it 5/2006

DFG-Projekt RealTest – Test und Zuverlässigkeit nanoelektronischer Systeme

DFG-Project – Test and Reliability of Nano-Electronic Systems

Bernd Becker, Ilia Polian, Universität Freiburg,
Sybille Hellebrand, Universität Paderborn,
Bernd Straube, Fraunhofer IIS-EAS Dresden,
Hans-Joachim Wunderlich, Universität Stuttgart

Zusammenfassung Entwurf, Verifikation und Test zuverlässiger nanoelektronischer Systeme erfordern grundlegend neue Methoden und Ansätze. Ein robuster Entwurf wird unabdingbar, um Fertigungsfehler, Parameterschwankungen, zeitabhängige Materialveränderungen und vorübergehende Störungen in gewissem Umfang zu tolerieren. Gleichzeitig verlieren gerade dadurch viele traditionelle Testverfahren ihre Aussagekraft. Im Rahmen des Projekts RealTest werden einheitliche Entwurfs- und Teststrategien entwickelt, die sowohl einen robusten Entwurf als auch eine darauf abgestimmte Qualitätssicherung unterstützen. ►►► **Summary** The increasing number of fabrication defects, spatial and temporal variability of parameters, as well as the growing impact of soft errors in nanoelectronic systems require a paradigm shift in design, verification and test. A robust design is mandatory to ensure dependable systems and acceptable yields. The quest for design robustness, however, invalidates many traditional approaches for testing and implies enormous challenges. Within the framework of the RealTest project unified design and test strategies are developed to support a robust design and a coordinated quality assurance after the production and during the lifetime of a system.

KEYWORDS B.7 [Integrated Circuits] Nanoelektronik, Entwurf, Test, Zuverlässigkeit, Fehlertoleranz/Nano-electronics, Design, Test, Dependability, Fault Tolerance

1 Hohe Qualität trotz unzuverlässiger Komponenten

Die Qualitätssicherung für nanoelektronische Schaltungen und Systeme steht im Mittelpunkt des DFG-Projekts RealTest (Test and Reliability of Nano-Electronic Systems), das von den Universitäten Freiburg (Prof. Dr. Bernd Becker, Dr. Ilia Polian), Paderborn (Prof. Dr. Sybille Hellebrand) und Stuttgart (Prof. Dr. Hans-Joachim Wunderlich) sowie dem Fraunhofer IIS-EAS Dresden (Prof. Dr. Bernd Straube) gemeinsam durchgeführt und von den Firmen Infineon Technologies München und Philips Semiconductors GmbH Hamburg unterstützt wird. Im Fokus des Projekts stehen integrierte Systeme, die aus Transistoren im Nanometerbereich aufgebaut sind. Wie auch schon bei traditioneller Mikroelektronik muss zur Qualitätssicherung die Korrektheit des Entwurfs durch Verifikation bzw. Validierung nachgewiesen werden. Daneben sind auch bei vollständig korrektem Entwurf Defekte in der Hardware nicht auszuschließen. Um fehlerhafte Chips zu erkennen und auszusortieren, muss jeder einzelne Chip nach der Produktion und wiederholt während des gesamten Lebenszyklus getestet werden.

Die fortschreitende Technologieentwicklung führt zu immer kleineren Strukturen und bietet damit viele neue Möglichkeiten, wie etwa die Integration ganzer Rechnercluster auf einem einzigen Chip. Gleichzeitig werden die Herstellungspro-

zesse jedoch empfindlicher gegenüber Störungen. Bei extrem kleinen Strukturen spielen bereits quantenphysikalische Effekte eine Rolle, und die einzelnen Herstellungsschritte erfordern eine viel größere Genauigkeit.

Bei nanoelektronischen Chips ist deshalb mit zahlreichen Defekten und auch mit großen Schwankungen der Schaltungsparameter, wie zum Beispiel der Schwellspannung der Transistoren, zu rechnen. Dabei müssen sowohl Abhängigkeiten der Parameter von der Position auf dem Chip als auch Materialveränderungen im Verlauf der Zeit stärker als bisher berücksichtigt werden. Schätzungen der „International Technology Roadmap for Semiconductors" gehen davon aus, dass bei fortschreitender Technologieentwicklung bis zum Jahr 2019 Strukturgrößen von 7 nm erreicht werden, aber nur noch zwischen 10% und 20% der produzierten Chips defektfrei sind [4]. Alle defekten Systeme wie bisher durch Tests einfach auszusortieren, würde keine wirtschaftliche Produktion mehr erlauben. Hinzu kommt außerdem noch eine erhöhte Anfälligkeit der Systeme gegenüber äußeren Störeinflüssen während des Betriebs.

Bei weiterer Skalierung wird der Test einen prinzipiellen Paradigmenwechsel erfahren. Es muss davon ausgegangen werden, dass vermehrt Maßnahmen wie Fehlertoleranz, Redundanz, Reparatur und auch Rekonfiguration eingesetzt werden, um wirtschaftliche Ausbeuten zu erzielen und die zunehmende Zahl von Fehlern während des Betriebs zu kompensieren. Solch ein „robuster" Entwurf bringt völlig neue Herausforderungen beim Test mit sich. Interne Komponenten sind kaum noch steuer- und beobachtbar, und viele der bisher verwendeten Überwachungsgrößen, wie z. B. Schalt- und Ruhestrommessungen, sind nicht mehr als Fehlerindikatoren geeignet. Darüber hinaus sind übliche Qualitätsmaße wie „Fehlerüberdeckung" nicht mehr aussagekräftig, da durch den robusten Entwurf ja auch ein bestimmtes Maß an Fehlern toleriert werden soll. Entsprechend können Verfahren für den Produktions- und Wartungstest („Offline Test") nicht mehr allein auf strukturorientierten Fehlermodellen aufbauen. Um die Fehlertoleranzmaßnahmen gegenüber Fehlern während des Betriebs zu unterstützen, sind effiziente Methoden für den „Online Test" notwendig. Dies umfasst sowohl permanente Fehler, die im System bleiben, als auch transiente Fehler („Soft Errors"), die nur kurzfristig auf das System einwirken.

Im Rahmen des Projekts RealTest werden einheitliche Entwurfs- und Teststrategien entwickelt, die sowohl einen robusten Entwurf als auch eine darauf abgestimmte Qualitätssicherung nach der Produktion und während des Betriebs unterstützen. Die Forschungsschwerpunkte liegen dabei in den Bereichen

- Fehlermodellierung und Fehleranalyse (Dresden),
- Speicher- und Zustandsüberwachung für komplexe Systeme (Stuttgart),
- Test fehlertoleranter nanoelektronischer Systeme (Paderborn), und
- Modellierung, Verifikation und Test akzeptablen Verhaltens (Freiburg).

Die Forschungsaktivitäten sind eng miteinander verzahnt. Soll zum Beispiel ein System robust gegenüber Störungen während des Betriebs entworfen werden, wird als Ausgangspunkt eine Analyse der zu erwartenden Defekt- und Störmechanismen benötigt, die insbesondere auch statistische Schwankungen der Parameter berücksichtigen kann und entsprechend das resultierende Verhalten statistisch charakterisiert. Für Speicherelemente wie Flipflops und Latches, die in freier Logik einen immer größeren Anteil einnehmen und besonders störanfällig sind, gibt es bisher kaum kostengünstige Ansätze des zuverlässigen Entwurfs. Im Rahmen des Projekts werden deshalb effiziente Techniken zur Überwachung und zur Kompensation von Fehlern entwickelt. Die Entwurfsstrategie steckt zusammen mit den Daten aus der Fehleranalyse die Randbedingungen für den Test ab. Hier muss nicht nur geprüft werden, ob das System das gewünschte Verhalten liefert, sondern auch in welchem Maß dazu bereits Fehlertoleranz eingesetzt wird und wie robust das Systemverhalten noch ist. Trotz entsprechender Entwurfs- und Testverfahren wird es nicht immer möglich sein, mit vertretbaren Kosten alle Fehler während des Betriebs abzufangen. Hier ist es entscheidend, das gewünschte Systemverhalten möglichst genau und anwendungsspezifisch zu modellieren. Fehler, die nicht kritisch sind, müssen dann nicht mehr berücksichtigt werden.

Herausforderungen und Lösungsansätze, die sich daraus in den einzelnen Forschungsschwerpunkten ergeben, werden in den folgenden Abschnitten ausführlicher beschrieben.

2 Fehleranalyse

Defekte, Störungen und Parametervariationen in nanoelektronischen Systemen werden durch die bekannten Modelle nicht mehr oder nur unzureichend beschrieben. Die extrem kleinen Strukturen führen dazu, dass schon Defekte mit sehr geringen Abmessungen das Systemverhalten beeinflussen. Schwankungen der Versorgungsspannungen und anderer Parameter, wie Kanallänge, Kanalbreite, Dicke des Gate-Oxids, Schwell- und Substratspannung, Diffusionswiderstand, Via- und Kontaktwiderstände, Gleichförmigkeit der Metallverbindungen, wirken sich wesentlich stärker als bisher aus. Neben diesen meistens permanenten Defekten spielen die durch Strahlungsteilchen hervorgerufenen, kurzzeitig mit unterschiedlicher Energie wirkenden, transienten Fehler eine immer stärkere Rolle [2].

Um diese vielfältigen und komplexen physikalischen Zusammen-

hänge beim Entwurf und Test robuster Systeme adäquat berücksichtigen zu können, sind sie möglichst realistisch auf Logikebene zu charakterisieren. So müssen zum Beispiel die notwendigen Eigenschaften von Testmustern oder Testfolgen durch logische und statistische Bedingungen beschrieben werden. Außerdem werden insbesondere für transiente Fehler Informationen darüber benötigt, mit welcher zeitlichen und räumlichen Verteilung zu rechnen ist.

Ausgangspunkt zur Lösung dieser Aufgabe sind Verfahren zur induktiven Fehleranalyse, welche die Auswirkungen von Defekten zunächst auf elektrischer Ebene modellieren [5]. Dort werden dann mit Simulationen und Analysen Testfolgen bestimmt und validiert. Die bisher üblichen Verfahren zur induktiven Fehleranalyse berücksichtigen allerdings die räumliche und zeitliche Variabilität von Parametern noch nicht und unterstützen auch keine statistische Charakterisierung von transienten Fehlern. Im Rahmen des Projekts werden deshalb entsprechende Erweiterungen entwickelt.

Bei der Bestimmung von Testfolgen durch elektrische Simulationen muss stärker als bisher berücksichtigt werden, dass die elektrischen Eingangsfolgen kontinuierliche Umschaltvorgänge zwischen zwei Spannungspegeln sind, die den logischen Werten „0“ und „1“ entsprechen. Werden ideale Umschaltvorgänge angenommen, können die Ergebnisse verfälscht werden und zu falschen Schlussfolgerungen über das Fehlerverhalten führen. Bei einer zu analysierenden eingebetteten digitalen Komponente muss statt dessen beachtet werden, dass die jeweiligen Übergänge stark verschliffene Verläufe haben und dass die entsprechenden Spannungspegel nicht immer erreicht werden.

Um eine Umsetzung der elektrischen Eingangssignale in Testfolgen auf Logikebene zu ermöglichen, wird das elektrische Netzwerk nur mit Spannungsverläufen stimuliert, die Bitfolgen auf Logikebene entsprechen. Bei der Analyse eines elektrischen Fehlers können dadurch auch folgende Probleme aufgedeckt werden:

- Der Fehler führt zu einer Ausgangsspannung, die von der Sollspannung abweicht, bewirkt aber keinen „eindeutigen“ Fehler auf Logikebene.
- Der Fehler hat zwar einen veränderten Versorgungsstrom zur Folge, bewirkt aber keine signifikante Spannungsabweichung.
- Der Fehler wird nur mit sehr langen Eingangsfolgen erkannt und kann damit nur schwer von digitalen Testverfahren behandelt werden.
- Der Fehler lässt sich nicht als Fehler auf Logikebene beschreiben.

In diesen Fällen dienen die Analyseergebnisse als Grundlage für einen prüfgerechten Entwurf und für Entwurfs- und Prozessverbesserungen zur Defektvermeidung.

Wenn eine Schaltung robust gegenüber transienten Fehlern entworfen werden soll, müssen für den Produktionstest auch die Wechselwirkungen zwischen permanenten und transienten Fehlern charakterisiert werden. Außerdem muss analysiert werden, ob es besonders „kritische“ transiente Fehler gibt, die mit hoher Wahrscheinlichkeit auftreten und für die während des Betriebs nur wenige Testmuster bzw. Testfolgen vorkommen (vgl. Abschnitt 4).

3 Speicher- und Zustandsüberwachung

Es ist ein bislang ungebrochener Trend, dass der Anteil von Flipflops in freier Logik stetig zunimmt. Diese Entwicklung folgt unter anderem aus dem massiven Pipelining oder dem Anwachsen der Registersätze, die beispielsweise zur Unterstützung von Spekulation, Multithreading und Befehlsscheduling notwendig sind. Auch Fehlertoleranztechniken erhöhen die Zahl der Speicherelemente in freier Logik, und bereits heute sind Schaltungen mit Millionen von Flipflops anzutreffen [8]. Diese Beobachtungen treffen nicht nur für Datenpfade zu sondern auch für kontrolldominierte Module, bei denen immer mehr Regularität und Geschwindigkeit im Vordergrund stehen.

Die Flipflops einer Schaltung sind in besonderem Maße gegenüber Einwirkungen der Umgebung anfällig und erfordern Schutzmechanismen wie sie heute schon bei regulären Speicherfeldern üblich sind [2]. Einige der gegenwärtig industriell eingesetzten Verfahren sind hier Reparatur und Rekonfiguration, Fehlererkennung und Fehlerkorrektur durch Kodierung, periodisches Auffrischen der Daten („Scrubbing“) gegen Fehlerakkumulation sowie eingebaute Selbsttestverfahren mit Redundanzanalyse und Selbstreparatur.

Besonders kritisch wirkt sich auch die Tatsache aus, dass zur Reduktion der Verlustleistung die Zahl der schaltenden Flipflops so gering wie möglich gehalten wird („Clock-Gating“). Dies hat zur Folge, dass eine beträchtliche Anzahl von Flipflops ihren Wert über einen längeren Zeitraum speichern muss. Damit sind die Speicherelemente ähnlich wie ein reguläres, dynamisches Speicherfeld über längere Zeit externen Einflüssen ausgesetzt und transiente Fehler können sich akkumulieren. Eine periodische Auffrischung der Speicherinformation ist hier genauso notwendig wie bereits heute in regulären Speicherfeldern [3].

Durch die steigenden Soft Error Raten (SER) für kombinatorische Bauelemente und die ständigen Verkürzungen der Logiktiefe werden außerdem vermehrt Fehler aus der Kombinatorik in die Speicherelemente propagiert [19]. Diese Effekte müssen ebenfalls durch eine geeignete Überwachung der Speicherelemente und entsprechende Fehlertoleranzverfahren kompensiert werden. Zusätzlich bietet sich auch die Möglichkeit, kombinatorische Ele-

mente und Latches gegen transiente Fehler zu härten [7].

Die zunehmende Zahl der Speicherelemente und die erforderliche Zusatzausstattung für eine erhöhte Zuverlässigkeit erschweren zugleich den Produktionstest, der bereits heute ein dominierender Kostenfaktor ist. Für freie Logik sind Teststrategien mit Prüfpfad am weitesten verbreitet. Hier werden die Testdaten seriell in die Schaltung geschoben und ausgelesen. Zur Verkürzung der Testzeit verwendet man meist mehrere Prüfpfade parallel, erzeugt die Muster im Selbsttest auf dem Chip oder führt komprimierte Testinformation von außen zu, die auf dem Chip dekomprimiert wird. Entsprechend wird die Testantwort komprimiert nach außen geführt. Bild 1 verdeutlicht dieses Prinzip des eingebetteten Tests.

Mit den Kompressionsmethoden begegnet man dem akuten Problem, dass die Bandbreite zwischen Chips und Testautomaten deutlich langsamer wächst als der Umfang der Testdaten [10; 17]. Der steigende Anteil von Flipflops und die beträchtliche Redundanz zur Steigerung der Zuverlässigkeit verschärfen dieses Testproblem noch beträchtlich.

Ziel des Projekts ist die Entwicklung einer einheitlichen Entwurfsmethodik für speichernde Elemente, welche die Probleme der Zuverlässigkeit und Fehlertoleranz, des Offline-Tests und des Online-Tests behandelt. Hierzu werden die einzelnen Prüfpfade nach Bild 2 in Segmente geeigneter Größe zerlegt und ihnen Redundanz zur Maskierung oder Reparatur permanenter Fehler hinzugefügt, sodass die entstehende Struktur immer noch tolerant gegenüber transienten Fehlern ist.

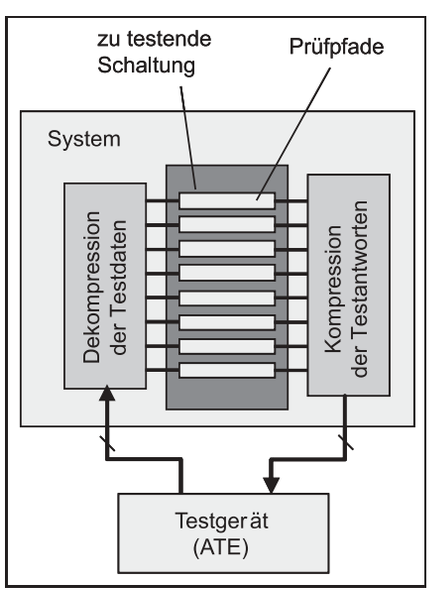

Bild 1 Eingebetteter Test.

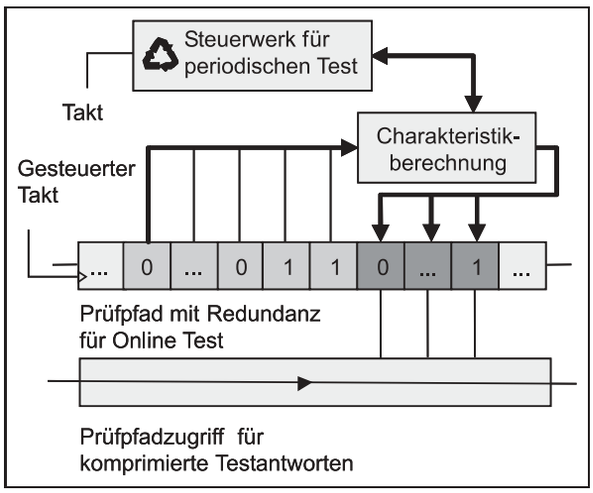

Bild 2 Online- und Offline Test für Prüfpfade.

Ein Prüfpfad kann als ein eindimensionaler Speicher interpretiert werden, und die entsprechenden Testverfahren für reguläre Speicherfelder, wie der periodische Test, der Online-Test und der transparente Test, lassen sich darauf anwenden. Wiederholtes Lesen und Rückschreiben würde jedoch den Zugriff auf die Flipflops einschränken und den Systembetrieb belasten. Stattdessen ist es sinnvoll, einen transparenten periodischen Selbsttest einzusetzen [3; 12]. Mit einer einfachen Logik lässt sich eine „Restcharakteristik" berechnen (Bild 2), die es erlaubt, den Inhalt eines Prüfpfades konsistent zu halten und kontinuierlich, periodisch zu überwachen.

Die zusätzliche Hardware, die für den Online-Test der Speicherelemente in die Schaltung integriert wurde, lässt sich zur Kompression der Testantworten verwenden. So muss nur die berechnete Charakteristik ausgewertet werden, von welcher dann auf falsche Schaltungsantworten geschlossen werden kann. Ein vollständiges Auslesen der teilweise redundanten Prüfpfadinformation ist bei dieser Lösung für den Offline Test nicht nötig, und die Testzeit wird ohne Zusatzaufwand dramatisch verkürzt.

Für die Eingangsdaten des Prüfpfads können ohne wesentliche Änderungen die derzeit bekannten Verfahren der Testdatenkompression eingesetzt werden.

4 Test fehlersicherer Schaltungen

Beim Test robuster Systeme muss insbesondere auch bewertet werden, in welchem Umfang bereits Fehlertoleranzmaßnahmen genutzt wurden und wie robust das System noch ist.

Klassische Fehlertoleranztechniken, wie etwa die n-fach modulare Redundanz, sind mit sehr hohen Kosten verbunden und werden meist nur für extrem sicherheitskritische Anwendungen eingesetzt [16]. Für das große Spektrum anderer Anwendungen sind kostengünstigere Überwachungsstrategien von besonderem Interesse. Die speziellen Aspekte und Herausforderungen beim Test solcher „selbstprüfender" Schaltungen („self-checking circuits") werden im Folgenden exemplarisch für „stark fehlersichere" Schaltungen aufgezeigt.

4.1 Stark fehlersichere Schaltungen

Um eine Boolesche Funktion als selbstprüfende Schaltung zu implementieren, kodiert man die Ein- und Ausgaben als Elemente von fehlererkennenden Codes $I \subset \{0,1\}^n$ und $O \subset \{0,1\}^m$ und überprüft die Ausgaben mithilfe eines entsprechenden Checkers (vgl. Bild 3) [13; 14; 18].

In der Praxis werden dafür häufig Paritätscodes oder ungeordnete Codes (z. B. Berger Codes) eingesetzt. Die Eigenschaften der Codes und der interne Aufbau der Schaltung bestimmen, welche Fehler erkannt werden können. „Selbstprüfend" bezieht sich deshalb immer auf eine bestimmte Fehlermenge Φ. Wird die Antwort auf eine Eingabe $i \in I$ beim Auftreten eines Fehlers $\varphi \in \Phi$ mit $f(i, \varphi)$ bezeichnet, lassen sich folgende drei Fälle unterscheiden:

- $f(i, \varphi) = f(i)$: der Fehler wird maskiert.
- $f(i, \varphi) \neq f(i)$, $f(i, \varphi) \notin O$: der Fehler wird vom Checker erkannt.
- $f(i, \varphi) \neq f(i)$, $f(i, \varphi) \in O$: der Fehler führt zu einem falschen Ergebnis, wird aber vom Checker nicht erkannt.

Ein „fehlersicherer" Entwurf versucht, den dritten Fall auszuschließen. Hier wird sichergestellt, dass für alle Fehler $\varphi \in \Phi$ und alle Eingaben $i \in I$ stets $f(i, \varphi) \notin O$ oder $f(i, \varphi) = f(i)$ gilt. Damit ist aber nicht unbedingt garantiert, dass f auch „selbsttestend" ist, d. h. dass es für jeden Fehler ein Testmuster mit $f(i, \varphi) \notin O$ gibt. Wenn im Lauf der Zeit mehrere Fehler $\varphi_1, \ldots, \varphi_k$ nacheinander auftreten und die ersten Fehler maskiert werden, kann der Mehrfachfehler $\langle \varphi_1, \ldots, \varphi_k \rangle$ zu einem falschen Ergebnis innerhalb des Codes führen. Es ist also nicht sicher, dass jeder Fehler erkannt wird, wenn er zum ersten Mal ein falsches Ergebnis liefert („Totally Self-Checking Goal"). Dieses Ziel wird erst mit „stark fehlersicheren" Schaltungen erreicht [20].

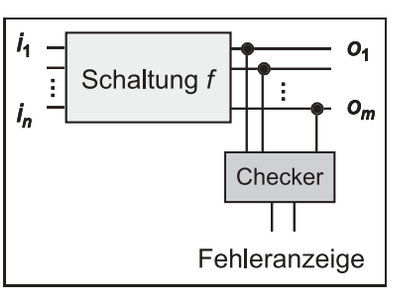

Bild 3 Selbstprüfende Schaltung („self-checking circuit").

Man nennt eine Schaltung f „stark fehlersicher" („strongly fault secure"), wenn für alle Fehler $\varphi \in \Phi$ gilt: Entweder

(i) f ist selbsttestend und fehlersicher bezüglich $\varphi \in \Phi$ oder

(ii) f ist fehlersicher bezüglich $\varphi \in \Phi$, und falls ein weiterer Fehler $\psi \in \Phi$ auftritt, dann gilt für den Mehrfachfehler $\langle \varphi, \psi \rangle$ entweder (i) oder (ii).

Entwurfsregeln für starke Fehlersicherheit bezüglich einfacher Fehlermodelle finden sich bereits in [20], weitergehende Ansätze sind in [9] zusammengefasst.

4.2 Herausforderungen und Erweiterungen

Nicht immer lassen sich für die zu erwartenden Fehler einfache Entwurfsregeln angeben. In solchen Fällen muss die Fehlersicherheit explizit verifiziert werden, bzw. es muss analysiert werden, mit welchen Einschränkungen zu rechnen ist. Weiterhin stimmt die Menge Φ der transienten Fehler in der Regel nicht mit dem Fehlermodell Ψ für Herstellungsfehler überein. Es müssen also insbesondere auch Wechselwirkungen zwischen beiden Fehlermodellen berücksichtigt werden.

Sowohl für den Nachweis der Fehlersicherheit als auch für die Testvorbereitung können prinzipiell Verfahren zur algorithmischen Testmustererzeugung („automatic test pattern generation", ATPG) eingesetzt werden. Dabei sind jedoch spezielle Anforderungen und Randbedingungen zu berücksichtigen.

Um die starke Fehlersicherheit bezüglich Φ zu verifizieren, muss für jeden einzelnen Fehler $\varphi \in \Phi$ geprüft werden, ob es ein Testmuster $i \in I$ mit $f(i, \varphi) \notin O$ gibt. Falls nicht, muss das Verhalten bei Fehlerakkumulierung durch Testerzeugung für entsprechende Mehrfachfehler analysiert werden. Insgesamt wird eine Analysestrategie benötigt, die mit möglichst wenigen Schritten eine vollständige Bewertung der Schaltung erlaubt. Außerdem muss die Zugehörigkeit der Testmuster zum Eingabecode durch „Constraints" geeignet modelliert werden.

Für den Produktionstest müssen Testmuster für die Fehler aus Ψ erzeugt werden. Für Fehler aus dem Durchschnitt $\Phi \cap \Psi$ können die Verifikationsmuster verwendet werden. Wenn für einen Fehler ψ aus der Restmenge $\Psi \setminus \Phi$ kein Testmuster aus dem Eingabecode existiert, wirkt sich der Fehler während des Betriebs nicht aus, die Schaltung kann jedoch nur eingesetzt werden, wenn sie weiterhin stark fehlersicher ist. Dazu müssen Mehrfachfehler, die ψ und weitere Fehler aus Φ umfassen, analysiert werden. Wenn die Schaltung mit ψ nicht mehr stark fehlersicher ist, muss sichergestellt werden, dass ψ im Produktionstest mit einem Testmuster aus $\{0,1\}^n \setminus I$ erkannt wird. Dafür sind eventuell zusätzliche Maßnahmen für den prüfgerechten Entwurf („design for testability", DFT) notwendig.

Idealerweise wird in einer stark fehlersicheren Schaltung ein Fehler erkannt bevor der nächste Fehler auftritt. Gutartiges Verhalten bei Fehlerakkumulierung wird deshalb nur für redundante Fehler sichergestellt. Wenn die Fehlerhäufigkeit zunimmt oder Testmuster für manche Fehler nur sehr selten vorkommen, können sich auch irredundante Fehler akkumulieren. Starke Fehlersicherheit im bisherigen Sinn genügt dann nicht mehr, um die Zuverlässigkeit der Schaltung zu gewährleisten, und realistische Fehlerannahmen müssen besser als bisher berücksichtigt werden. Die zu erwartende Fehlerhäufig-

keit bestimmt beispielsweise, welche Zeitspanne zwischen zwei Fehlern zu erwarten ist. Abhängig davon ist bei Fehlern, deren Erkennungswahrscheinlichkeit unter einer gewissen Schranke liegt, mit Akkumulierung zu rechnen. Eine Analyse dieser besonders kritischen Fehler liefert Anhaltspunkte für speziell angepasste Teststrategien, sinnvolle Entwurfsmodifikationen und verbesserte Entwurfsregeln.

5 Toleranzintervall für Soft Errors

Traditionell werden beim Test integrierter Systeme alle Schaltungen aussortiert, die nicht genau die vorher berechneten Sollausgaben liefern. Für manche Anwendungen ist eine solche strenge Übereinstimmung mit den Referenzwerten jedoch gar nicht notwendig. Bei einem DVD-Player kann z. B. eine Ungenauigkeit in der Ausgabe toleriert werden, solange sie vom Betrachter nicht wahrgenommen wird. Wenn wirtschaftliche Ausbeuten erreicht und die Kosten für Fehlertoleranzmaßnahmen minimiert werden sollen, sind „kleine“ Abweichungen der Testergebnisse von den Sollergebnissen zuzulassen („error tolerance“) [1; 6]. Dazu muss die akzeptable Bandbreite für Schwankungen in der Ausgabe, in der Geschwindigkeit oder anderen Parametern akkurat und anwendungsspezifisch modelliert werden.

5.1 Akzeptables Verhalten während des Betriebs

Während des Betriebs beeinflussen Soft Errors das Verhalten entscheidend, da sie den Systemzustand verändern und damit auch über längere Zeit hinweg zu falschen Ergebnissen führen können. Um akzeptables Verhalten während des Betriebs zu spezifizieren, ist es daher wichtig, das Systemverhalten im Zeitverlauf zu charakterisieren. Kehrt etwa ein System innerhalb einer vorgegebenen Zahl k von Zyklen nach dem Auftreten des Fehlers zum Referenzverhalten zurück, so kann der Fehlereffekt vernachlässigbar sein.

Um diese Idee zu präzisieren, wird das Referenzverhalten einer Schaltung durch Eingaben I, Ausgaben O, Zustände S, eine Übergangsfunktion δ: $I \times S \to S$ und eine Ausgabefunktion λ: $I \times S \to O$ beschrieben. Wenn von einem Startzustand $s \in S$ aus eine Eingabefolge $i \in I^m$ mehrere Zustandsübergänge nacheinander produziert, wird dies durch $\delta^m(i, s) = \delta(i_m, \delta(\ldots\delta(i_1, s)\ldots))$ angegeben. Entsprechend ist $\lambda^m(i, s)$ durch $\lambda(i_m, \delta(\ldots\delta(i_1, s)\ldots))$ definiert. Es wird angenommen, dass sich Fehler nur für einen Taktzyklus auswirken und die Zeitspanne bis zum Auftreten des nächsten Fehlers größer ist als die Länge der betrachteten Eingabefolgen. Wie in Abschnitt 4 wird das Schaltungsverhalten bei einem Fehler $\varphi \in \Phi$ durch $\delta^m(i, s, \varphi)$ bzw. $\lambda^m(i, s, \varphi)$ bezeichnet.

Außerdem sei eine Metrik d: $O \times O \to \mathbb{R}$ gegeben, mit der sich Abweichungen der Ausgaben von den Referenzwerten quantifizieren lassen. Berechnet das System etwa Datenwerte auf einem Ausgangsbus, so könnte d die maximale Differenz zwischen den Sollwerten und den Ausgaben der fehlerbehafteten Schaltung sein („Threshold Testing“ [6]). Nun lässt sich akzeptables Verhalten wie folgt beschreiben.

Das Verhalten der Schaltung mit einem Fehler $\varphi \in \Phi$ ist akzeptabel bezüglich eines Toleranzintervalls der Länge k und einer Schranke τ für die Abweichungsmetrik d, wenn für alle Eingabefolgen $(i_1, \ldots, i_k, i_{k+1}) \in I^{k+1}$, alle Startzustände $s \in S$ und alle m, $1 \le m \le k$, gilt:

- $d(\lambda^m(i_1, \ldots, i_m, s), \lambda^m(i_1, \ldots, i_m, s, \varphi)) \le \tau$,
- $\lambda^{k+1}(i_1, \ldots, i_k, i_{k+1}, s) = \lambda^{k+1}(i_1, \ldots, i_k, i_{k+1}, s, \varphi)$ und
- $\delta^{k+1}(i_1, \ldots, i_k, i_{k+1}, s) = \delta^{k+1}(i_1, \ldots, i_k, i_{k+1}, s, \varphi)$.

Bild 4 veranschaulicht dies für drei Beispiele. Das Referenzverhalten ist als dicke Linie eingezeichnet, das Verhalten der Schaltung bei Soft Errors ist durch gestrichelte Linien dargestellt.

Die vertikale Achse zeigt die Abweichung bezüglich der Metrik d. Wenn der „Schlauch“ mit dem Radius τ verlassen wird, ist die Differenz zu groß (inakzeptables Verhalten 1). Auch wenn die Ausgaben innerhalb des Schlauchs bleiben, ist es nicht akzeptabel, wenn sie zu lange von den Referenzwerten abweichen (inakzeptables Verhalten 2). Nur wenn die Schaltungsausgaben innerhalb des Schlauchs bleiben und die Schaltung nach k Zyklen wieder zum Referenzverhalten zurückkehrt, ist das Verhalten akzeptabel.

5.2 Anwendungen und erste Ergebnisse

Das Konzept des Toleranzintervalls liefert die Grundlage für einen kostengünstigen robusten Entwurf. Anstatt aufwendige Fehlertoleranzarchitekturen zu implementieren oder alle Schaltungsteile zu härten, wird

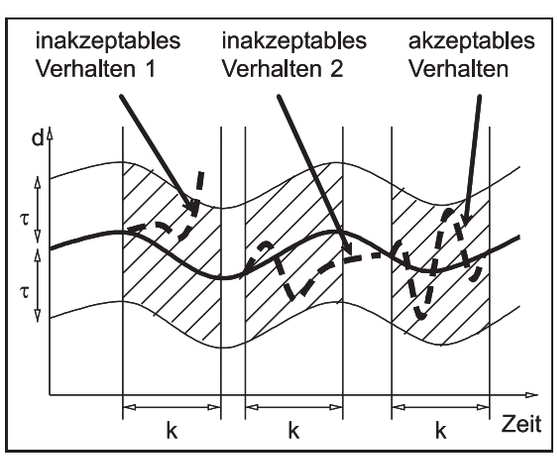

Bild 4 Inakzeptables und akzeptables Verhalten bei Soft Errors.

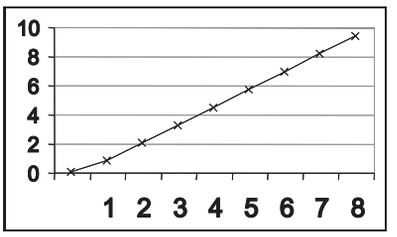

Bild 5 Anteil der unkritischen Fehler (in %) in Abhängigkeit von der Länge k des Toleranzintervalls.

zunächst analysiert, welche Fehler zu inakzeptablem Verhalten führen, also kritisch für einen sicheren Betrieb sind. Es genügt dann die von den kritischen Fehlern betroffenen Schaltungsteile zu härten („Selective Hardening“) [11].

In einer ersten Fallstudie wurde dazu ein Block zur Bewegungserkennung für ein MPEG System untersucht [15]. Die Metrik d wurde ähnlich wie in [6] definiert. Bild 5 zeigt, dass die Zahl der unkritischen Fehler deutlich zunimmt, wenn die Länge k des Toleranzintervalls erhöht wird.

6 Zusammenfassung

Die Entwicklung von der Mikro- zur Nanoelektronik bringt neben neuen Möglichkeiten auch zahlreiche Herausforderungen mit sich. Die starke Zunahme von Fabrikationsdefekten und Parameterschwankungen sowie die wachsende Anfälligkeit gegenüber transienten Fehlern erfordern einen robusten Entwurf, der mit darauf abgestimmten Maßnahmen zur Qualitätssicherung gekoppelt ist. Die Arbeiten im Projekt RealTest schaffen dafür die Grundlage mit speziell angepassten Verfahren zur Fehleranalyse, Strategien zur Überwachung während des Betriebs und zum Test fehlertoleranter Strukturen sowie einer anwendungsspezifischen Modellierung des akzeptablen Systemverhaltens.

Literatur

[1] M. Breuer, “Error-Tolerance and Related Test Issues”, Proc. Asian Test Symp. (ATS’04), Kenting, Tawain, Nov. 2004.

[2] P.E. Dodd and L.W. Massengill, “Basic mechanisms and modeling of single-event upset in digital microelectronics”, IEEE Trans. on Nuclear Science, 50 (3), pp. 583–602, June 2003.

[3] S. Hellebrand et al., “Efficient online and offline testing of embedded DRAMs”, IEEE Trans. on Computers, 51 (7), pp. 801–809, 2002.

[4] SIA, “Int. technology roadmap for semiconductors”, Technical report, Semiconductor Industry Association, 2003, available at: http://public.itrs.net/

[5] A. Jee and F.J. Ferguson, “Carafe: An Inductive Fault Analysis Tool for CMOS VLSI Circuits”, Proc. 11th IEEE VLSI Test Symp., pp. 92–98, 1993.

[6] Z. Jiang and S.K. Gupta, “An ATPG for Threshold Testing: Obtaining Acceptable Yield in Future Processes”, Proc. IEEE Int. Test Conf. (ITC’02), Baltimore, MD, USA, pp. 824–833, Oct. 2002.

[7] Y. Komatsu et al., “A soft-error hardened latch scheme for Soc in a 90 nm technology and beyond”, Proc. IEEE Custom Integrated Circuits Conference (CICC’04), Orlando, FL, USA, pp. 329–332, Sep. 2004.

[8] R. Kuppuswamy et al., “Full hold-scan systems in microprocessors: Cost/benefit analysis”, Intel Technology Journal, 8 (1), pp. 63–72, Feb. 2004.

[9] P.K. Lala, “Self-Checking and Fault-Tolerant Digital Design”, Morgan Kaufmann Publishers, San Francisco, 2001.

[10] S. Mitra et al., “X-Tolerant Test Response Compaction”, IEEE Design & Test of Computers, 22 (6), pp. 566–574, 2005.

[11] K. Mohanram and N.A. Touba, “Cost-effective approach for reducing soft error failure rate in logic circuits”, Proc. IEEE Int. Test Conf. (ITC’03), Charlotte, NC, USA, pp. 893–901, Sept./Oct. 2003.

[12] M. Nicolaidis, “Theory of Transparent BIST for RAMs”, IEEE Trans. on Computers, 45 (10), pp. 1141–1156, 1996.

[13] M. Nicolaids and Y. Zorian, “On-Line Testing for VLSI – A Compendium of Approaches”, Journal of Electronic Testing: Theory and Applications (JETTA), 12 (1–2), pp. 7–20, February/April 1998.

[14] W.W. Peterson and E.J. Weldon, Jr., “Error-Correcting Codes”, 2nd Edition, MIT Press, Cambridge, MA, USA, 1972.

[15] I. Polian et al., “Period of Grace: A New Paradigm for Efficient Soft Error Hardening”, Handouts 18. ITG/GI/GMM Workshop “Testmethoden und Zuverlässigkeit von Schaltungen und Systemen”, Titisee, pp. 41–45, März 2006.

[16] D.K. Pradhan, “Fault Tolerant Computer System Design”, Prentice Hall, Upper Saddle River, NJ, USA, 1996.

[17] J. Rajski et al., “Finite memory test response compactors for embedded test applications”, IEEE Trans. on CAD, 24 (4), pp. 622–634, 2005.

[18] T.R.N. Rao and E. Fujiwara, “Error Control Coding for Computer Systems”, Prentice Hall, Englewood Cliffs, NJ, USA, 1989.

[19] P. Shivakumar et al., “Modeling the effect of technology trends on the soft error rate of combinational logic”, Proc. Int. Conf. on Dependable Systems and Networks (DSN’02), Bethesda, MD, USA, pp. 389–398, June 2002.

[20] J.E. Smith and G. Metze, “Strongly Fault Secure Logic Networks”, IEEE Trans. on Computers, c-27(6), pp. 491–499, June 1978.


1


2


3


4


5

1 Prof. Dr. Bernd Becker ist Inhaber des Lehrstuhls für Rechnerarchitektur an der

Albert-Ludwigs-Universität Freiburg. Seine Hauptarbeitsgebiete sind Entwurf, Verifikation und Test von Schaltungen und Systemen.
Adresse: Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Georges-Koehler-Allee, 79110 Freiburg,
E-Mail: becker@informatik.uni-freiburg.de

2 Prof. Dr. Sybille Hellebrand leitet die Arbeitsgruppe Datentechnik am Institut für Elektrotechnik und Informationstechnik der Universität Paderborn. Ihre Forschungsschwerpunkte liegen im Bereich Test und Diagnose von integrierten Schaltungen und Systemen.
Adresse: Arbeitsgruppe Datentechnik, EIM-E, Universität Paderborn, Warburger Str. 100, 33098 Paderborn,
E-Mail: sybille.hellebrand@date.upb.de

3 Dr. Ilia Polian hat 2003 an der Universität Freiburg promoviert (summa cum laude). Seitdem arbeitet er dort als Wissenschaftlicher Assistent an diversen Aspekten der Zuverlässigkeit von Mikro- und Nanoelektronik.
Adresse: Institut für Informatik, Albert-Ludwigs-Universität Freiburg, Georges-Koehler-Allee, 79110 Freiburg,
E-Mail: polian@informatik.uni-freiburg.de

4 Prof. Dr. Bernd Straube ist seit 1992 Gruppenleiter für Test und Verifikation in der Dresdner Außenstelle Entwurfsautomatisierung des Fraunhofer-Instituts für Integrierte Schaltungen. Seine Arbeitsgebiete sind Analog- und Mixed-Signal-Test sowie formale Verifikation.
Adresse: Fraunhofer IIS/EAS, Zeunerstr. 38, 01069 Dresden,
E-Mail: Bernd.Straube@eas.iis.fraunhofer.de

5 Prof. Dr. Hans-Joachim Wunderlich ist Direktor des Instituts für Technische Informatik (ITI) an der Universität Stuttgart. Er ist Autor und Co-Autor von vier Büchern und über 100 Publikationen auf den Gebieten Test, eingebauter Selbsttest, Zuverlässigkeit und Fehlertoleranz.
Adresse: Institut für Technische Informatik (ITI), Universität Stuttgart, Pfaffenwaldring 47, 70569 Stuttgart,
E-Mail: wu@informatik.uni-stuttgart.de