

Test Planning for Low-Power Built-In Self Test

Von der Fakultät Informatik, Elektrotechnik und Informationstechnik
der Universität Stuttgart zur Erlangung der Würde eines Doktors der
Naturwissenschaften (Dr. rer. nat.) genehmigte Abhandlung

Vorgelegt von

Christian Gerhard Zöllin

aus Müllheim, Deutschland

Hauptberichter: Prof. Dr. rer. nat. habil. Hans-Joachim Wunderlich

Mitberichter: Prof. Dr. rer. nat. habil. Ilia Polian

Tag der mündlichen Prüfung: 16. Mai 2014

Institut für Technische Informatik der Universität Stuttgart

2014

Abstract

Power consumption has become the most important issue in the design of integrated circuits. The power consumption during manufacturing or in-system test of a circuit can significantly exceed the power consumption during functional operation. The excessive power can lead to false test fails or can result in the permanent degradation or destruction of the device under test. Both effects can significantly impact the cost of manufacturing integrated circuits.

This work targets power consumption during Built-In Self-Test (BIST). BIST is a Design-for-Test (DfT) technique that adds additional circuitry to a design such that it can be tested at-speed with very little external stimulus. Test planning is the process of computing configurations of the BIST-based tests that optimize the power consumption within the constraints of test time and fault coverage.

In this work, a test planning approach is presented that targets the Self-Test Using Multiple-input signature register and Parallel Shift-register sequence generator (STUMPS) DfT architecture. For this purpose, the STUMPS architecture is extended by clock gating in order to leverage the benefits of test planning. The clock of every chain of scan flip-flops can be independently disabled, reducing the switching activity of the flip-flops and their clock distribution to zero as well as reducing the switching activity of the down-stream logic. Further improvements are obtained by clustering the flip-flops of the circuit appropriately.

The test planning problem is mapped to a set covering problem. The constraints for the set covering are extracted from fault simulation and the circuit structure such that any valid cover will test every targeted fault at least once. Divide-and-conquer is employed to reduce the computational complexity of optimization against a power consumption metric. The approach can be combined with any fault model and in this work, stuck-at and transition faults are considered.

The approach effectively reduces the test power without increasing the test time or reducing the fault coverage. It has proven effective with academic benchmark circuits, several industrial benchmarks and the Synergistic Processing Element (SPE) of the Cell/B.E.TM Processor (Riley et al., 2005). Hardware experiments have been

conducted based on the manufacturing BIST of the Cell/B.E.TM Processor and shown the viability of the approach for industrial, high-volume, high-end designs.

In order to improve the fault coverage for delay faults, high-frequency circuits are sometimes tested with complex clock sequences that generate test with three or more at-speed cycles (rather than just two of traditional at-speed testing). In order to allow such complex clock sequences to be supported, the test planning presented here has been extended by a circuit graph based approach for determining equivalent combinational circuits for the sequential logic.

In addition, this work proposes a method based on dynamic frequency scaling of the shift clock that utilizes a given power envelope to it full extent. This way, the test time can be reduced significantly, in particular if high test coverage is targeted.

Acknowledgments

I would like to express my sincere gratitude to everyone who has contributed to my Doctoral studies and this thesis.

First and foremost, I would like to thank my supervisor Prof. Dr. habil. Hans-Joachim Wunderlich. He has supported me throughout my studies, challenged me with exciting scientific work and allowed me to venture into many new and exciting research areas. I will value forever the way in which he has greatly influenced my approach to scientific problems and provided the guidance towards solving them. I also very much appreciate the support and advice of Prof. Dr. habil. Ilia Polian both during this thesis and the joint work that we did on the DFG project REALTEST.

I am very thankful for the many supporters at IBM Germany Research & Development GmbH. The grant from the IBM Center of Advanced Studies has enabled the work that led to this thesis and has allowed for joint work that would not have been possible otherwise. In particular, I would like to thank Dr. Jens Leenstra, Dipl.-Ing. Nicolas Maeding and Dr. Peter Hans Roth, who have sparked my interest for high-end, high-frequency chip design and have supported me ever since. A special thanks also goes to Michael Kessler, Knut Schuenemann and Otto Torreiter who have shared their insight on processor test and who enabled the invaluable experimental studies in this work. Furthermore, I'm indebted to my manager at IBM Research, Dr. Ruud Haring, who has enabled and encouraged me to continue work on this thesis during my work on the BlueGene/Q supercomputer project.

I would like to thank my colleagues at the Institut fuer Technische Informatik of the University of Stuttgart. Besides the adventures that we shared, I'm thankful for the ideas and discussions that they contributed and joint research that we did. I am especially thankful to Michael Kochte with whom I had many fruitful discussions, which led to numerous publications. I am also very grateful to Melanie Elm with which I have exchanged many ideas and enjoyed mentoring many students.

I would like to thank my family for their patience and support. In particular, I would like to mention my parents Gerhard and Hannelore and my brother Alexander. And finally, I would like to thank my beloved girlfriend Ursula, who has blessed me with the persistence required for the completion of this thesis.

Weinstadt,

Christian Zoellin
February 2014

Contents

1	Kurzfassung in deutscher Sprache	1
1.1	Selbsttest mit parallelen Prüfpfaden	2
1.2	Berechnung eines optimierten Testplans	3
1.2.1	Optimierungsalgorithmus	4
1.3	Untersuchung	7
1.3.1	Benchmarks und industrielle Schaltungen	8
1.3.2	Ergebnisse	9
2	Introduction	13
2.1	Motivation and Goal of the Work	13
2.2	Outline of this Work	15
3	Fault Models	19
3.1	Stuck-At Faults	21
3.2	Bridging Faults	22
3.3	Delay Faults	23
3.3.1	Gate Delay Faults	23
3.3.2	Path Delay Faults	24
4	Concepts of Built-In Self-Test	25
4.1	Short Taxonomy of BIST	25
4.2	Scan-based Built-In Self-Test	27
4.3	Test Pattern Generators	29
4.3.1	Pseudo-Random Pattern Generation	29
4.3.2	Weighted-Random Pattern Testing	31
4.4	Test Response Evaluation	32

4.5	The STUMPS Design	34
5	Power During the Test of Integrated Circuits	37
5.1	Impact of Test Power Consumption	37
5.1.1	Test Environment	38
5.1.2	Test Cost	38
5.2	Power Consumption of the Circuit under Test	39
5.2.1	Power Metrics	40
5.2.2	Power Dissipation in Synchronous Circuits	41
5.3	Power Consumption in Scan-based Testing	42
6	Related Power-Aware BIST Techniques	45
6.1	Modified Test Pattern Generators	45
6.2	Low Power Scan Cells	46
6.2.1	Scan Clock Gating	47
6.2.2	Toggle Suppression	49
6.3	Scan Path Organization	51
6.3.1	Scan Path Segmentation	51
6.3.2	Extended Clock Schemes for Scan Segmentation	53
6.3.3	Scan Tree and Scan Forest	55
6.3.4	Inserting Logic into the Scan Path	58
7	Power-Aware BIST DFT with Scan-Chain Disable	61
7.1	Scan Chain Enable by Clock Gating	61
7.2	Estimation of Power Consumption	63
7.2.1	Related Models for Test Power Consumption	64
7.2.2	Linearized Model	64
7.2.3	Evaluation of Model Accuracy	65
7.3	Principle	66
7.4	The BIST Planning Problem	67
7.5	Scan Cell Clustering	69
7.5.1	Formal Problem Statement	70
7.5.2	Partitioning Algorithm	71
8	BIST Planning for Scan Clock Gating	75
8.1	Related Work	75
8.2	Optimization Algorithm	78
8.3	Fault Classification and Heuristics	79
8.3.1	Hard Faults	80

8.3.2	Difficult Faults	82
8.3.3	Other Faults	83
9	Extension to Transition Faults and Frequency Scaling	85
9.1	Transition Faults	85
9.1.1	Sequential Graphs for Shift and Capture Clocks	86
9.1.2	Graph Generation from a Clock Sequence	89
9.2	Dynamic Frequency Scaling	90
9.3	Tradeoffs	91
10	Evaluation	93
10.1	Benchmarks and Industrial Circuits	93
10.2	Wafer Test Experiments with the 45nm Cell/B.E. Processor	95
10.2.1	Experimental Setup	96
10.2.2	Test Generation Flow	98
10.2.3	Experiment: Accuracy of Power Estimates	99
10.2.4	Experiment: Power Reduction by Test Planning	100
10.2.5	Frequency Scaling	101
10.3	Canonical Experiments with Benchmark Circuits	104
10.3.1	Test Planning for Stuck-At Tests	104
10.3.2	Impact of Test Plan Granularity	106
10.3.3	Impact of Test Length	107
10.3.4	Impact of Scan Chain Insertion Approach	108
10.3.5	Test Planning for Transition Fault Tests	110
10.3.6	Launch-Off-Capture and Launch-Off-Shift Tests	111
10.3.7	Complex Test-Clock Sequences	112
11	Conclusion	115
11.1	Summary	115
11.2	Contributions	116
11.3	Possible Future Work	117
	List Of Figures	119
	List Of Tables	123
	List Of Abbreviations	125
	References	127

Related Publications by the Author	145
12.1 Book Chapters	145
12.2 Journals and Conference Proceedings	145
12.3 Workshop Contributions	148

Kurzfassung in deutscher Sprache

Die fortschreitende Miniaturisierung der Schaltungsstrukturen, die steigende Betriebsfrequenz und die wachsende Komplexität und Fläche führen zu einer ständigen Zunahme sowohl der dynamischen als auch der statischen Verlustleistung hochintegrierter Schaltungen. Zusätzlich ist die Schaltaktivität während des strukturorientierten Tests um nahezu eine Größenordnung erhöht (Zorian, 1993), so daß die dynamische Verlustleistung beträchtlich zunimmt und sowohl die Ausbeute als auch die Zuverlässigkeit beeinträchtigt (Girard et al., 2009).

Ohne geeignete Gegenmaßnahmen kann die maximale Verlustleistung, die während eines Zeitpunkts im Verlauf des Tests auftritt, deutlich über derjenigen liegen, die für den Systembetrieb spezifiziert ist. Die maximale momentane Verlustleistung tritt üblicherweise zu Beginn eines Taktzyklus auf. Durch Spannungsabfall am ohmschen Widerstand des Verteilungsnetzwerks entstehen Spannungseinbrüche, induktive Effekte oder Rauschen auf den Signalleitungen der dazu führt, daß auch defektfreie Schaltungen fehlerhafte Ausgaben liefern, aussortiert werden und somit die Ausbeute mindern. Die durchschnittliche Verlustleistung ist die während eines Tests oder eines Testabschnitts umgesetzte Energie dividiert durch die Dauer und kann ohne geeignete Gegenmaßnahmen beträchtlich über derjenigen im Systembetrieb liegen. Die erhöhte Stromdichte und auch die daraus folgende erhöhte Temperatur setzen die Schaltung einem Stress aus, der unkontrolliert die Lebenserwartung verringert.

Die auf Grund der Skalierung schnell wachsenden Leckströme beeinträchtigen den System- und Testbetrieb in gleicher Weise, wobei die Reduktion der statischen Verlustleistung nicht Gegenstand dieser Arbeit ist. Dagegen ist die überproportional wachsende dynamische Verlustleistung testspezifisch und muß in der industriellen

Fertigung entsprechend berücksichtigt werden. Zu den industriell üblichen Gegenmaßnahmen gehören die Reduktion der Testgeschwindigkeit, Partitionierung der Schaltung und besonderer Aufwand bei der Kühlung (Girard et al., 2009).

Es haben sich in der Vergangenheit zahlreiche Publikationen mit der Verminderung der Schaltaktivität während des prüfpfadbasierten Selbsttests beschäftigt. Das Schieben von Testmustern und -antworten erfordert den größten Teil des Energieaufwands nicht nur im Prüfpfad selbst sondern auch durch die in der kombinatorischen Logik hervorgerufene Schaltaktivität. Diese kann verringert werden, indem der funktionale Ausgang während des Schiebevorgangs konstant gehalten wird (Gerstendörfer and Wunderlich, 1999). Einige Ansätze unterdrücken im Selbsttest solche Muster, die nicht zur Testqualität beitragen und verringern so den Energieverbrauch (Gerstendörfer and Wunderlich, 1999; Girard et al., 1999). Andere Methoden modifizieren beispielsweise durch Umordnen oder Einfügen von Gattern den Prüfpfad, um den Test zu beschleunigen oder die Schaltaktivität zu reduzieren (Bonhomme et al., 2004; Dabholkar et al., 1998; Whetsel, 2000). Die Parallelität des Tests und damit die Verlustleistung können reduziert werden, indem die Schaltung partitioniert und ein entsprechender Testplan berechnet wird (Girard et al., 2000b). Weiterhin wurden verlustleistungsgerechte Mustergeneratoren (Wang and Gupta, 2006) oder die Verwendung von globalem Clock Gating (Girard et al., 2001; Sankaralingam and Touba, 2002) vorgeschlagen.

In dieser Arbeit wird eine Methode vorgestellt, welche es erlaubt einzelne bzw. mehrere Prüfpfade vorübergehend zu deaktivieren ohne die Fehlerabdeckung zu beeinträchtigen. Hierdurch wird die Verlustleistung signifikant reduziert und deren Verlauf kann an verschiedene Hüllkurven angepasst werden. In der vorliegenden Arbeit wird gezeigt, dass durch spezielle Entwurfswerkzeuge zur Testplanung, welche die Freiheitsgrade während der Optimierung berücksichtigen, die Verlustleistung deutlich reduziert werden kann.

1.1 Selbsttest mit parallelen Prüfpfaden

Für den Selbsttest mit mehreren Prüfpfaden wurde in (Bardell et al., 1987) die STUMPS-Konfiguration vorgestellt, welche inzwischen die am weitesten verbreitete Struktur für den Logik-Selbsttest ist. Hierbei werden mehrere parallele Prüfpfade von einem Generator für Pseudo-Zufallsmuster (Pseudo-Random Pattern Generator, PRPG) mit Testmustern versorgt. Der PRPG besteht aus einem linear rückgekoppelten Schieberegister (Linear-Feedback Shift Register, LFSR), einem XOR-Netzwerk zur Phasenverschiebung und einer Logik zur Gewichtung der Häufigkeit von Einsen

und Nullen im Testmuster. Die Schaltungsantworten werden durch ein Signaturregister mit mehreren Eingängen (Multiple-Input Signature Register, MISR) ausgewertet.

Die hier vorgestellte Erweiterung dieser Selbstteststruktur gestattet es, den Schieberegister für einzelne Prüfpfade zu deaktivieren (Abb. 1.1). Das Taktsignal wird für einzelne Prüfpfade gänzlich unterdrückt wodurch die durch den Schieberegister verursachte Schaltaktivität blockiert und die Verlustleistung im Taktbaum verringert wird. Eine Variante dieser Architektur wird zum Beispiel im Cell/B.E.TM Prozessor benutzt (Pham et al., 2006).

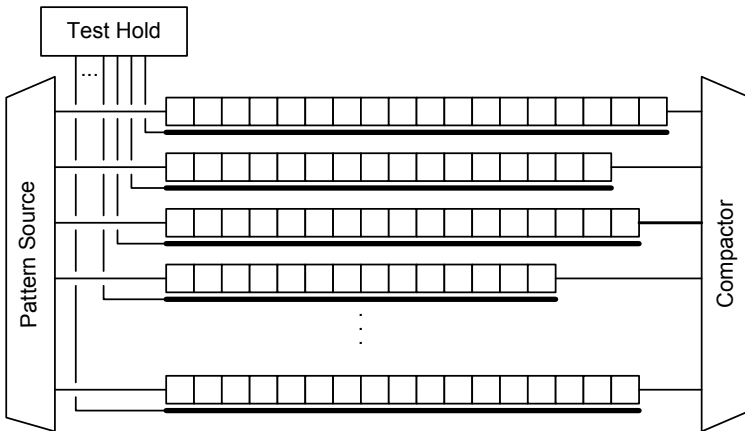


Fig. 1.1.1. DfT mit parallelen Prüfpfaden und Taktabschaltung pro Prüfpfad

1.2 Berechnung eines optimierten Testplans

Ziel der Berechnung eines optimierten Testplans ist das Erkennen einer gegebenen Fehlermenge mit minimierter Verlustleistung. Zu diesem Zweck wird für jeden Startwert des LFSRs eine möglichst große Menge von Prüfketten bestimmt, die abgeschaltet werden können, ohne die Fehlererfassung zu beeinträchtigen.

Ein Testblock b ist ein Tupel (s, sc) bestehend aus einem Startwert des Testmuster-generators (und der zugehörigen Testmenge fester Größe), sowie einer Menge von aktivierten Prüfpfaden $sc \subset SC$ im folgenden Konfiguration genannt. Durch Fehler-simulation unter Einbeziehung der Prüfpfadkonfiguration kann zu jedem Testblock die Fehlermenge F_b ermittelt werden, die durch diesen Block erkannt wird. Gesucht

ist eine Menge von Blöcken B , welche die Gesamtfehlermenge F mit minimalem Energieaufwand detektieren kann.

Ein Fehler f kann durch unterschiedliche Konfigurationen entdeckt werden. Man beachte, dass zu einem Startwert mehrere Blöcke existieren können, die sich jeweils in der Konfiguration unterscheiden und deshalb unterschiedliche Fehlermengen überdecken. Ein Block (s, sc) wird als minimal bezüglich des Fehlers f bezeichnet, wenn f nicht mehr durch den Block erkannt werden würde, falls ein beliebiger Prüfpfad aus sc entfernt wird.

Die Kosten einer Überdeckung sind eine Schätzung ihres Energieverbrauchs, indem die Zahl der Ketten, die pro Startwert aktiviert werden sollen, aufsummiert wird. Sei S_B die Menge der Startwerte aus B , dann kann zu jedem Startwert $s \in S_B$ die zugehörige Menge von Blöcken B_s angegeben werden. Die Kosten einer Menge von Blöcken kann nun abgeschätzt werden zu

$$\text{cost}(B) = \sum_{s \in S_B} \left| \bigcup_{(s, sc) \in B_s} sc \right|.$$

Falls die Prüfpfade nicht, wie allgemein üblich, von gleicher Länge sind, kann hier auch mit der Zahl der Flipflops gewichtet werden.

Für realistische Schaltungsgrößen ist die Bestimmung eines globalen Optimums nicht sinnvoll möglich, da die Zahl aller möglichen Startwerte des Testmustergenerators sehr groß ($> 2^{32}$) ist und die benötigten Fehlersimulationen sehr rechenintensiv sind. Im Folgenden zeigen wir eine Heuristik, welche das Überdeckungsproblem löst, indem für eine gegebene Menge S von Startwerten Blöcke berechnet werden, welche die zugehörige Fehlermenge annähernd optimal überdecken.

Beispiel

Zum Test der Schaltung in Abbildung 1.2 gehört die Menge der Prüfpfade $SC = \{sc_1, sc_2, sc_3\}$, die Menge der Fehler $F = \{f_1, f_2, f_3, f_4, f_5\}$ und die Menge der Startwerte $S = \{s_1, s_2, s_3\}$.

1.2.1 Optimierungsalgorithmus

Jeder Block $b = (s, sc)$ bestimmt eine Menge von Fehlern F_b , die bei der Abarbeitung dieses Blocks erkannt werden. Falls in F_b Fehler enthalten sind, die von keinem anderen Block erkannt werden, gehört b auf jeden Fall zur optimalen Lösung und wird essentieller Block genannt. Wir erhalten $B_0 := \{b \mid b \text{ essentiell}\}$ als Zwischenlösung und müssen nur noch die Fehler aus $F_0 = F \setminus \bigcup_{b \in B_0} F_b$ überdecken.

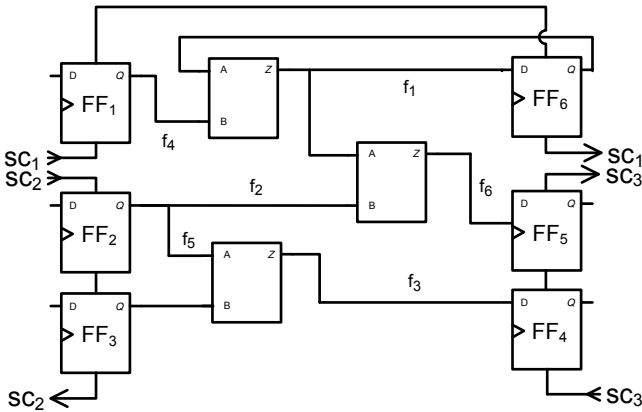


Fig. 1.2. Beispiel: Es sei $S = \{s_1, s_2, s_3\}$, $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$.

Auch bei den verbleibenden Fehlern wird die Komplexität durch einen „divide and conquer“ Ansatz reduziert. Zu diesem Zweck wird die Menge der Fehler in drei verschiedene Klassen unterteilt:

1. Harte Fehler sind solche, die nur durch einen Startwert aus S erkannt werden können. Die entsprechenden Startwerte nennen wir essentiell.
2. Schwer erkennbare Fehler können nur durch eine Zahl von Startwerten erkannt werden, die unter einer vom Anwender vorgegebenen Konstante lim liegt.
3. Die übrigen Fehler sind leicht erkennbar.

Diese drei Klassen werden mit unterschiedlichen Verfahren und Heuristiken behandelt.

Harte Fehler

Für jeden harten Fehler $f \in F_0$ mit zugehörigen Startwert s_f bestimmt die Funktion $c(f, s_f) = \{c \mid f \text{ wird von Block } (s_f, c) \text{ erkannt}\}$ die Scanketten und damit die Menge der minimalen Blöcke $B_f = \{(s_f, c) \mid c \in c(f, s_f)\}$.

Für harte Fehler wird zur Ermittlung der Blöcke eine Implementierung der Funktion $c(f, s)$ verwendet, die ausnutzt, dass zur Erkennung eines Fehlers bereits ein einzelnes Ausgangsflipflop ausreicht, in dem der Fehler zu einer Änderung der Schaltungsantwort führt. Damit wird für jedes erkennende Ausgangsflipflop eine Konfiguration ermittelt, die ausreichend ist, um den Fehler zu erkennen, und die nahe an einer minimalen Konfiguration liegt.

Für die Funktion $c(f, s)$ werden also aus der Fehlersimulation des Startwerts s_f diejenigen Flipflops ermittelt, welche f beobachten können. Zu jedem Flipflop werden anschliessend die Flipflops des zugehörigen Eingangskegels hinzugenommen und auf die Menge der benötigten Prüfpfade geschlossen.

Für jeden Block $b \in \bigcup_{f \text{ hart}} B_f$ wird wieder die Menge F_b aller Fehler bestimmt, die in b erkannt werden. Da die Zahl dieser Blöcke gering ist, kann mit einem Branch-and-Bound Verfahren eine Teilmenge $B_1 \subset \bigcup_{f \text{ hart}} B_f$ gefunden werden (Coudert, 1996), so daß $\bigcup_{b \in B_1} F_b$ alle harten Fehler aus F_0 überdeckt und $\text{cost}(B_0 \cup B_1)$ minimal ist. Die verbleibende Fehlermenge ist $F_1 = F_0 \setminus \bigcup_{b \in B_1} F_b$.

Beispiel

In der Beispielschaltung aus Abb. 8.2 werde f_4 nur durch s_1 erkannt und sei ein harter Fehler (wir nehmen $B_0 = \emptyset$ an). Bei der Fehlersimulation werde ermittelt, dass f_4 durch die Flipflops FF_5 sowie FF_6 beobachtet wird, wenn s_1 unter Aktivierung aller Prüfpfade angewendet wird. Daraus folgt gemäß der Approximation $c(f_4, s_1)$, dass f_4 durch Aktivierung einer der Flipflop Kombinationen $\{FF_1, FF_2, FF_5, FF_6\}$ oder $\{FF_1, FF_6\}$ bzw. der zugehörigen Prüfpfade $\{sc_1, sc_2, sc_3\}$ oder $\{sc_1\}$ erkannt wird. Die Menge der Blöcke für f_4 ergibt sich zu:

$$B_{f_4} = \{(s_1, \{sc_1, sc_2, sc_3\}), (s_1, \{sc_1\})\}$$

Die optimierte Überdeckung ist $B_1 = \{(s_1, \{sc_1\})\}$ mit Kosten $\text{cost}(B_1) = 1$. Im Beispiel wird durch Fehlersimulation ermittelt, dass B_1 bereits auch den Fehler f_1 erkennt.

Schwer erkennbare Fehler

Die nächste Klasse enthält schwer erkennbare Fehler, welche durch mehrere Startwerte erkannt werden, deren Zahl aber unter einem gesetzten Limit lim liegt. Um den damit einhergehenden Komplexitätszuwachs zu kompensieren, wird pro Startwert nur genau eine erkennende Konfiguration berücksichtigt. Dazu wird die Funktion $c_{min}(f, s_f) \in c(f, s_f)$ verwendet, welche nur auf die Konfiguration abbildet, welche mit minimalen Grenzkosten umgesetzt werden kann.

Wiederum ist $B_f = \{(s_f, c_{min}(f, s_f)) \mid s_f \text{ Startwert für } f\}$ die zugehörige Blockmenge, und eine Teilmenge $B_2 \subset \bigcup_{f \in F_{lim}} B_f$ wird erzeugt, so dass $\bigcup_{b \in B_2} F_b$ alle schwer erkennbaren Fehler aus F_1 überdeckt und $\text{cost}(B_0 \cup B_1 \cup B_2)$ minimal ist. Die verbleibende Fehlermenge $F_2 = F_1 \setminus \bigcup_{b \in B_2} F_b$ kann nur noch leicht erkennbare Fehler umfassen und wird daher sehr klein oder sogar leer sein.

Beispiel

Zu Abbildung 8.2 ergibt sich für $lim = 2$ die Menge $F_{lim} = \{f_2, f_3\}$. f_2 wird bei den Startwerten $\{s_1, s_2\}$ beobachtet in FF_6 , wodurch sich ergibt $c_{min}(f_2, s_f) = \{sc_1, sc_2, sc_3\}$. Für f_3 ergibt sich bei den Startwerten $\{s_2, s_3\}$, $c_{min}(f_3, s_f) = \{sc_2, sc_3\}$. Dazu gehören die folgenden Mengen von Blöcken:

$$B_{f_2} = \{(s_1, \{sc_1, sc_2, sc_3\}), (s_2, \{sc_1, sc_2, sc_3\})\},$$

$$B_{f_3} = \{(s_2, \{sc_2, sc_3\}), (s_3, \{sc_2, sc_3\})\}$$

Für Abb. 8.2 wird hierbei die optimale Teilmenge $B_2 = \{(s_2, \{sc_1, sc_2, sc_3\})\}$ mit Kosten $cost(B_2 \cup B_1 \cup B_0) = 4$ gefunden. Im Beispiel aus Abb. 8.2 wird bereits der letzte Fehler f_5 erkannt und der Algorithmus endet hier für diesen Fall. Die ermittelte Blockmenge beträgt $B = B_2 \cup B_1 \cup B_0 = \{(s_1, \{sc_1\}), (s_2, \{sc_1, sc_2, sc_3\})\}$ mit Kosten von $cost(B) = 4$ und detektiert alle Fehler.

Restliche Fehler

Um die restlichen Fehler zu erfassen, wird Schritt 2 nochmals mit $lim = \infty$ wiederholt, wodurch Blöcke für alle verbliebenen Fehler erzeugt werden. Da viele dieser Fehler durch eine besonders große Menge von Startwerten erkannt werden, ist der Suchraum bei der Optimierung sehr gross. Für solche Fehler wird nur eine Untermenge von Startwerten betrachtet. Zusätzlich wird die Optimierung durch das Branch-and-Bound Verfahren abgebrochen, sobald die Zeit bis zum nächsten (besseren) Zwischenergebnis ein festgelegtes Zeitlimit überschreitet. Dies beeinträchtigt das Ergebnis nur unwesentlich, beschränkt aber die Worst Case Zeitkomplexität auf eine wählbare obere Schranke.

In den Verfahren für die Klassen der harten und schwer erkennbaren Fehler kann die Zahl der Blöcke im schlimmsten Fall quadratisch mit der Zahl der Prüfpfade wachsen. Die Komplexität für die Klasse der restlichen Fehler ist konstant (durch Abbruch nach Zeitüberschreitung). Die Experimente im folgenden Kapitel bestätigen, dass das Verfahren in der Praxis noch besser skaliert.

1.3 Untersuchung

Das beschriebene Verfahren wurde in Java als Teil einer hauseigenen Entwurf-automatisierungslösung implementiert und auf eine Reihe von Benchmark-Schaltungen angewendet. Alle Resultate wurden mit Hilfe eines Quad-Opteron mit 2,4

GHz und 32 GB Hauptspeicher generiert, wobei die hohe Parallelisierbarkeit des Problems ausgenutzt wurde, um rechenintensive Aufgaben wie z.B. die Fehlersimulation zu beschleunigen.

1.3.1 Benchmarks und industrielle Schaltungen

Zur Evaluierung des vorgestellten Verfahrens wurden Schaltungsmodelle aus den folgenden Quellen verwendet:

- International Symposium on Circuits and Systems (ISCAS89)
- International Test Conference (ITC99)
- Industrielle Schaltungen von NXP
- Prozessorkern des Cell-Prozessors

Die Schaltungen aus ISCAS89 (s38417 sowie s38584) und ITC99 (b17, b18 und b19) sind die größten Schaltungen aus der jeweiligen Sammlung von sogenannten Benchmark-Schaltungen. Sie besitzen keine Selbsttestausstattung und wurden für diesen Beitrag um die benötigte Testarchitektur erweitert. Die Flipflops der Schaltungen wurden hierzu basierend auf der Schaltungstopologie in 32 parallelen Prüfpfaden angeordnet.

Die von NXP zur Verfügung gestellten Schaltungen (p286k, p330k, p388k, p418k und p951k) enthalten bereits einen prüfgerechten Entwurf mit parallelen Prüfpfaden und sind um ein vielfaches größer als die Schaltungen aus ISCAS89 und ITC99. Zudem repräsentieren sie die typischen Eigenschaften industrieller Schaltungen, nämlich kürzere Pfade sowie kleinere Ausgangskegel, bedingt durch die stärkere Optimierung auf hohe Taktfrequenzen bei geringer Fläche.

Als Beispiel für die Anwendung des vorgestellten Verfahrens auf eine Schaltung mit partiellen Prüfpfaden wurde eine Implementierung des Cell-Prozessors verwendet, welche aus etwa 250 Millionen Transistoren auf 115mm^2 Chipfläche in 45nm Technologie besteht (Takahashi et al., 2008). Die übergeordnete Selbsttestarchitektur des Cell Prozessors besteht aus 15 Selbsttestdomänen mit jeweils eigener STUMPS Instanz (Riley et al., 2005). Da 70% der Chipfläche von den 8 Synergistic Processing Elements (SPE) eingenommen werden und diese eine recht heterogene Struktur besitzen, werden sie im folgenden als repräsentativer Teil des Cell Prozessors betrachtet.

Die für den Test der SPE relevanten Kenndaten sind:

- 1.8 Millionen Logikgatter, 7 Millionen Transistoren in der Logik, 14 Millionen Transistoren in Speicherfeldern
- 150.000 Flipflops

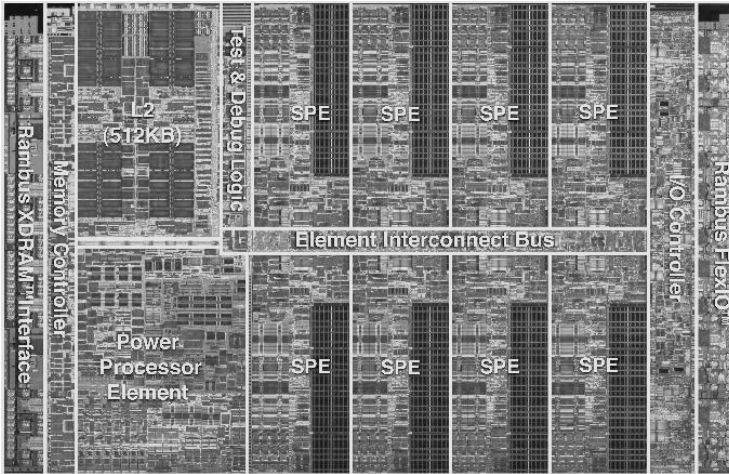


Fig. 1.3. Chip Foto des Cell-Prozessors

- 82.500 Flipflops sind angeordnet in 32 STUMPS Kanäle
- Speicher-Felder sind nicht Teil des Logikselbsttests und werden über einen gesonderten Selbsttest abgedeckt

Tabelle 1.1 zeigt eine Übersicht der Schaltungseigenschaften aller untersuchter Schaltungen. Die erste Spalte (Schaltung) gibt den Schaltungsnamen an, die zweite Spalte (Gatter) die Zahl der Logikgatter welche durch die in der dritten Spalte (Ketten) angegebene Zahl von Prüfpfaden getestet werden. Die Prüfpfade enthalten die in der Spalte FFs angegebene Zahl von Flipflops. Die Zahl der Haftfehler ist in der Spalte #Haftfehler angegeben.

1.3.2 Ergebnisse

Der folgende Abschnitt zeigt Ergebnisse für verschiedene Kombinationen aus Startwerten und erzeugten Mustern pro Startwert. Dabei beträgt die Anzahl der zufällig gewählten Startwerte 200 und pro Startwert generiert der PRPG eine feste Anzahl von 1024 Mustern. Daraus ergibt sich eine Gesamtanzahl der angelegten Testmuster von 204.800. Für Schritt 2 des Algorithmus wurde der Parameter *limit* auf einen Erfahrungswert von 3 gesetzt, für den sich eine gute Verteilung zwischen Problemgröße und erreichter Überdeckung ergibt.

Die detaillierten Ergebnisse für die ausgewählten Schaltungen finden sich in Tabelle 1.2.

Table 1.1. Parameter der für die Evaluierung eingesetzten Schaltungen

Schaltung	# Gatter	# Ketten	# FFs	# Haftfehler
s38417	24079	32	1770	32320
s38584	22092	32	1742	38358
b17	37446	32	1549	81330
b18	130949	32	3378	277976
b19	263547	32	6693	560696
p286k	332726	55	17713	648044
p330k	312666	64	17226	547808
p388k	433331	50	24065	856678
p418k	382633	64	29205	688808
p951k	816072	82	104624	1590490
SPE	1352044	32	40027	1065190

Der erste Spaltenblock zeigt neben dem Schaltungsnamen die Zahl der Haftfehler.

Die weiteren Spaltenblöcke geben jeweils die Ergebnisse für die drei Schritte des Optimierungsverfahrens an. $|F_0|$ ist die Zahl der harten Fehler, $|F_1|$ die Zahl der schwer erkennbaren Fehler und $|F_2|$ die Zahl der restlichen Fehler. Die Spalte *cost()* gibt jeweils die Bewertung des Zwischenergebnisses mit der Kostenfunktion an. Die Reduktion der Verlustleistung in Prozent nach Abschluss der Optimierung wird in der letzten Spalte *Red.* angegeben.

Table 1.2. Ergebnisse für Testpläne mit 200 Startwerten je 1024 Muster

Circuit	$ F_{det} $	Harte Fehler		FSIM	Harte+Schwere. F.	FSIM	Alle Fehler	
	Erkannt	$ F_0 $	cost()	$ F_1 $	cost()	$ F_2 $	cost()	Red. [%]
s38417	31661	500	1484	30836	1873	31431	2165	67.43
s38584	36389	26	90	13506	243	34922	350	94.42
b17	70547	1947	2691	70230	3152	70519	3246	49.29
b18	239753	6291	4164	239451	4446	239952	4498	33.36
b19	479689	13802	4915	479220	5112	480253	5128	22.69
p286k	610094	6226	9901	603509	9990	609269	10024	10.43
p330k	491199	3036	6214	436877	6733	490705	6799	41.94
p388k	839174	3239	3732	790766	4129	835695	4233	58.93
p418k	639750	8851	4895	577037	5372	638527	5470	45.37
p951k	1544131	7995	7640	1421168	7983	1542416	8069	46.72
SPE	904183	1805	1150	817125	1342	900114	1451	75.33

Besonders bei den Schaltungen b18, b19 und p286k ergibt sich eine drastische Verbesserung. Der Pseudozufallstest verlangt in der Regel mit steigender Schalungsgröße auch eine Zunahme der Testlänge. Bleibt die Testlänge konstant, wächst die Zahl der harten Fehler und beeinträchtigt die Effizienz des Verfahrens. Denn in der Folge werden für diese Schaltungen die Kosten stark von den harten Fehlern dominiert und die zusätzlichen Freiheitsgrade in den beiden späteren Schritten können nicht ausgenutzt werden. Zum Beispiel werden bei p951k im ersten Schritt bei $200 \cdot 512$ Mustern bereits 83% der Kosten zur Erkennung der harten Fehler benötigt, bei p286k gar 99%. Die erzielte Minderung der Verlustleistung ist also von den Parametern „Anzahl der Startwerte“ und „Muster pro Startwert“ abhängig.

Darum wird in Abbildung 1.4 für die Benchmarkschaltungen aus ISCAS und ITC die Anzahl der Startwerte variiert, wobei die Gesamtzahl der Muster konstant bei 204,800 gehalten wird. Es zeigt sich, dass durch eine feinere Granularität die Reduzierung der Verlustleistung erheblich verbessert werden kann. Dies ist darauf zurückzuführen, dass für eine höhere Zahl von Startwerten die Zahl von Mustern und damit die Zahl von Fehlern, die durch jeden Startwert erkannt werden, entsprechend abnimmt.

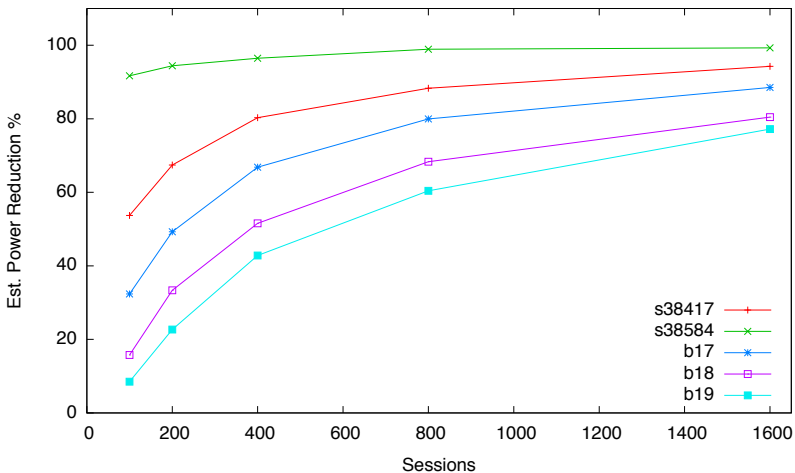


Fig. 1.4. Abhängigkeit von der Anzahl der Startwerte für ISCAS und ITC

Abbildung 1.5, zeigt eine Detaildarstellung eines einzelnen Testplans. Die X-Achse steht für die sequentielle Reihenfolge der Startwerte und die Y-Achse die Zahl der zu aktivierenden Flipflops. Es zeigt sich, dass der vorliegende Algorithmus

mus eine sehr gleichmässige Verteilung der Verlustleistung erreicht. Falls notwendig kann dieser Testplan durch Umsortieren der Startwerte und Konfiguration oder durch Anpassung der Schiebefrequenz weiter optimiert werden. Dadurch können auch spezielle Hüllkurven erreicht werden, zum Beispiel um die anfängliche Wärmekapazität der Schaltung zu nutzen.

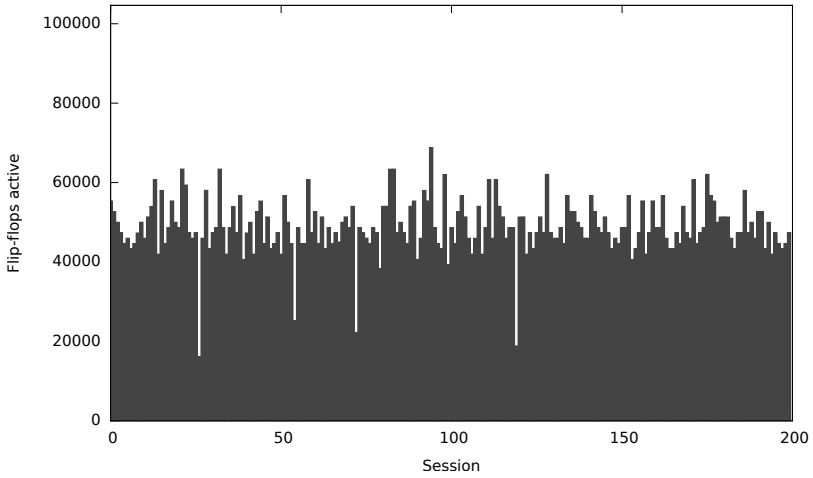


Fig. 1.5. Testplandetails für Schaltung p951k

Introduction

2.1 Motivation and Goal of the Work

During the manufacturing of integrated circuits, defects can be introduced by a large number of mechanisms that are difficult or impossible to avoid. Since each manufacturing step incurs additional cost, it is essential for the economic viability of the process to test and reject the defective parts as early in the manufacturing process as possible. Common steps for test in the manufacturing flow are tests after manufacturing of the wafers (wafer test), after packaging of a chip (module test), after integration with other components on a printed circuit board (board test) and in the final system (system or in-the-field test).

The size of circuits that can be integrated on a chip has increased steadily as outlined by Moore's Law and at the time of this writing can exceed one billion transistors. At the same time, the number of pins on the chips has not increased significantly and the increase in circuit complexity has resulted in difficulty to provide sufficient bandwidth for the test stimuli. Consequently, low-pin-count test techniques have been an important part of testing for several process generations. Furthermore, the designs often operate at frequencies of several Gigahertz and at-speed testing (testing for defects that impact circuit delay) has become a standard technique in manufacturing test.

Built-in Self-Test (BIST) is a Design-for-Test (DfT) technique that adds additional circuitry to a design such that it can be tested with very little external stimulus. In the most common technique, the BIST generates random patterns on-chip which requires little area overhead. BIST provides both low-pin-count and at-speed testing and it can be reused at wafer, module, board and system test. Consequently, all recent high-end chip designs include BIST structures to some extent.

Power consumption has become the most important issue in the design of integrated circuits (Bosio et al., 2012; Nebel and Mermet, 1997). Two types of power consumption are distinguished: The maximum instantaneous power consumption is called the peak power and relates to the requirement of on-chip decoupling capac-

itance to avoid voltage droop (which would slow down the circuits). The average power consumption relates to the excess heat that must be dissipated through cooling equipment as well as the sizing of power supplies including the batteries of mobile devices.

Both types of power consumption are taken into account in all steps of the design flow, starting with the architectural design down to the physical design. Techniques such as clock gating are used to reduce the switching activity during the functional operation. On the other hand, during the test (Girard et al., 2009) many of these techniques are disabled in order to improve observability and exercise additional nodes of the design in order to reduce test time. Furthermore, structural test can execute non-functional state transitions which exceed the carefully architected functional power specification.

Consequently, these test patterns result in power consumption that can significantly exceed the power consumption during functional operation (Zorian, 1993). The excessive peak power can lead to false test fails, degrading yield. The excessive average power consumption can result in the permanent degradation or destruction of the device under test. The most common-place industrial ad-hoc methods to deal with the elevated power consumption during test are:

- Oversizing power supply, package and cooling.
- Reduction of the clock frequency during the test.
- Partitioning of the chip and reducing the parallelism during the test.

All of these methods increase the test cost, either through the additional equipment or material, or through the increase in test time.

More advanced techniques that are gradually taken into account in state-of-the-art commercial design automation software include:

- Segmentation of the scan path and staggered clocking (Bonhomme et al., 2001; Whetsel, 2000)
- Blocking the propagation of switching activity by gating or partitioning (Gers-tendörfer and Wunderlich, 2000; Hertwig and Wunderlich, 1998)
- Power-aware pattern generation (Remersaro et al., 2006, 2007; Wang and Gupta, 1997; Wen et al., 2007b; Zhang et al., 1999)

The low-power test planning method presented in this work is orthogonal to these techniques and may be combined with any of them. (Further non-commercial techniques for power reduction and their relationship to this work are discussed in later chapters.)

Test planning is the process of computing configurations of the BIST-based tests that optimize the power consumption within the constraints of test time and fault coverage. In this work, a test planning approach is presented that targets the Self-Test Using Multiple-input signature register and Parallel Shift-register sequence generator (STUMPS) DfT architecture (Bardell et al., 1987) that is extended by scan clock gating. The approach effectively reduces the test power without increasing the test time or reducing the fault coverage.

The test planning is mapped to a set covering problem. The constraints for the set covering are extracted from fault simulation. Divide-and-conquer is used to reduce the computational complexity. The approach can be combined with any fault model. For the experiments, stuck-at and transition faults are considered.

The approach is evaluated with academic benchmark circuits, several industrial benchmarks and the Synergistic Processing Element (SPE) of the Cell/B.E.TM Processor (Riley et al., 2005). Hardware experiments have been conducted based on the manufacturing BIST of the Cell/B.E.TM Processor.

2.2 Outline of this Work

Chapter 3 (→ ‘Fault Models’ starting on page 19) describes the most common logical faults models. Understanding the choice and implication of the fault model is important in this work, since the method does all optimization with respect to the fault model that is selected. Section 3.1 introduces the stuck-at fault model. Section 3.2 briefly introduces the bridging fault model. In section 3.3, the transition fault and path delay fault models are described. In this work, only the stuck-at and transition faults models are used, but the methods applies to the other two fault models and they are mentioned to show their relation to the fault models targeted here.

The basic BIST techniques are presented in chapter 4 (→ ‘Concepts of Built-In Self-Test’ starting on page 25). Section 4.1 introduces a classification of BIST techniques. Section 4.2 introduces scan design for testability, which is the basis of practically all industrial BIST architectures. Test pattern generation (TPG) is a key part of any BIST technique, pseudo-random TPG approaches are discussed in section 4.3. Furthermore, test responses must be evaluated with test response evaluators, which are described in section 4.4. The most common logic BIST approach is the STUMPS architecture introduced in section 4.5.

Chapter 5 (→ ‘Power During the Test of Integrated Circuits’ starting on page 37) deals with the impact of elevated power consumption on manufacturing test and its sources in the circuit under test. Section 5.1 discusses how excessive test power must

be accounted for in the test environment and how it impacts test cost. In section 5.2, the common power metrics are reviewed as well as the sources of power consumption in synchronous circuits. The power consumption in scan-based circuits is considered in additional detail in section 5.3.

State-of-the-art power-aware BIST techniques are presented in chapter 6 (→ ‘Related Power-Aware BIST Techniques’ starting on page 45). Section 6.1 introduces TPGs that target reducing switching activity. A significant amount of the switching activity originates in the shifting of patterns in scan-based BIST and in the scan cells themselves. Section 6.2 presents important techniques to reduce the data as well as clock activity in scan cells through gating techniques. The scan path organization techniques described in section 6.3 are an important factor in reducing the shift power and in assisting power reduction techniques such as the one presented in this work.

A new design-for-test architecture for power-aware BIST is presented in chapter 7 (→ ‘Power-Aware BIST DfT with Scan-Chain Disable’ starting on page 61). This architecture is based on clock gating techniques to control the scan function which are presented in section 7.1. Section 7.2 deals with the power consumption characteristic of the proposed architecture and presents an approximation method that can be efficiently computed and is employed as a cost function for the optimization algorithms presented later-on. Section 7.3 presents the basic idea of test planning to take advantage of the proposed hardware structure and section 7.4 presents the formal problem definition and maps the problem to a set covering problem. An optimization approach that can be used to improve the hardware design is shown in section 7.5.

Chapter 8 (→ ‘BIST Planning for Scan Clock Gating’ starting on page 75) presents an algorithm to solve the test planning problem that is able to deal with industrial circuits. Related work from deterministic test generation and test scheduling is reviewed in section 8.1. A divide-and-conquer approach is used to reduce the computational complexity of the set-covering based optimization algorithm which is introduced in section 8.2. The heuristics used during the optimization are described in additional detail in section 8.3.

The basic algorithm presented in chapter 8 can be extended in several ways, which are outlined in chapter 9 (→ ‘Extension to Transition Faults and Frequency Scaling’ starting on page 85). Section 9.1 presents how the algorithm is modified that delay fault models can be targeted besides the regular stuck-at fault model. Here, a generalized method is introduced that allows to consider arbitrary clock sequences. In many applications, the goal is not to achieve the minimum test power possible,

but to match a power envelope that is close to the functional power consumption. Section 9.2 demonstrates how test planning together with dynamic frequency scaling can be employed to match power envelopes. Section 9.3 discusses the parameters that influence the effectiveness of the proposed approach and how these parameters can be optimized.

Extensive evaluations of the proposed method are presented in chapter 10 (→ ‘Evaluation’ starting on page 93). The benchmark and industrial circuits examined here are presented in section 10.1. Section 10.2 presents a series of experiments that were conducted in an industrial wafer-test scenario at IBM for a 45nm implementation of the Cell/B.E.TM processor architecture. The report on this experiment also includes details on how the proposed approach fits into a standard-tools based industrial design flow. And finally, the experiments for the Cell/B.E.TM processor are augmented by test time optimizations that are possible due to optimization of the power consumption beyond the specified power envelope. Canonical experiments on benchmark circuits are conducted in section 10.3 to demonstrate the applicability of the presented approach to a wide range of circuits and to thoroughly evaluate the trade-offs. Stuck-at tests are evaluated for the entire range of benchmarks in section 10.3.1 and the impact of parameters such as test granularity, test length and scan chain configuration is considered in the subsequent sections. Section 10.3.5 deals with the application of the proposed method to transition faults. Besides the two most common test-clock schemes, more advanced schemes are considered as well.

The work is concluded by a summary in chapter 11 (→ ‘Conclusion’ starting on page 115).

Fault Models

This chapter describes the abstraction of physical defects to faults at the logic level. This abstraction reduces the large space of physical defects that have to be described at the 3D layout and electrical level, to perturbations of the behavior of the circuit at the gate level. It is an essential part in developing tractable algorithms for the quantification of the quality of tests and for automated test generation and optimization.

First, the classes of defects and associated fault models is discussed. Then, the most common fault models are introduced in further detail with a special focus on the models employed in this work.

Flaws at the physical level are caused by various mechanisms, such as:

- Missing or surplus material
- Device effects: Parasitic transistors or degradation of transistors
- Material impurities
- Variation in atomic quanta
- Parametric variations, e.g. of the temperature or the power supply
- External influences (e.g. irradiation with high-energy particles)

Faults describe the impact of the physical defects at the logic level. For example, excess material can cause a short of a signal wire with a power supply line such as ground. With sufficient conductance this will cause the corresponding logic signal to behave permanently stuck low. Obviously, various defects can result in the same logic behavior and the abstraction provided by the fault models presented in this chapter covers a significant portion of actual defects in logic circuits. Through the fault model, every entity of a circuit model at the logic level is associated with one or more possible faults.

An important measure for the quality of a test is the fault coverage. The fault coverage is defined as the percentage of detected faults with respect to the total set of faults under a given fault model.

The fault models representing the aforementioned defects can be classified according to several criteria. These properties of defects and the associated fault mod-

els are important in comparing fault models with each other and in implementing the associated behavior of the fault at the logic level.

The frequency of occurrence of a fault plays an important role in how to diagnose and deal with a fault. The occurrence is grouped into three categories: Permanent faults, intermittent faults and transient faults. The offline test (e.g. during manufacturing) focuses mostly on permanent and intermittent faults, whereas transient faults are of growing concern for online-test methods (Hellebrand et al., 2007). In this work, the fault models are limited to permanent fault models and their extensions to intermittent faults. The proposed method is easily extended to transient faults by applying methods such as the one in (Amgalan et al., 2008).

Fault models are furthermore distinguished between functional faults and structural faults. Functional faults are modeled as perturbing the function at the behavioral level and are implemented at purely behavioral abstraction levels such as the register transfer level or the transaction level (Kochte et al., 2010b). Structural faults on the other hand retain some of the interconnect structure of the implementation and are modeled at the gate level. The structural BIST presented here relies on a structural fault model.

Especially in high-frequency designs, faults are also distinguished according to their sequential behavior. Delay faults are faults which impact the timing of the circuit (i.e. the propagation delays) and are categorized according to which entities are affected (gate, wiring, via, entire path). Other faults require a sequence of circuit states (e.g. in partial scan, Chapter 4). In both cases, at least two patterns must be applied to the circuit to activate these fault and/or propagate their consequences. On the other hand, static faults do not exhibit this sequential property and can be targeted with single-pattern tests.

Of particular importance to the tractability of a fault model is the assumption regarding the multiplicity of faults. The single-fault assumption is very wide-spread in dealing with fault models and it has been shown that tests targeting single-faults in most cases also detect the multiple-fault cases (Hughes and McCluskey, 1984). However, in cases where circuits are specifically designed for robustness certain circuit properties must be checked with a multiple-fault model (Hellebrand and Hunger, 2009). But this analysis is only feasible for small modules. Industrial circuits are typically analysed with single-fault models only.

The remainder of this chapter introduces the stuck-at fault model and the bridging fault model as two examples of static fault models. This is followed by two delay fault models, the transition fault model and the path delay model.

3.1 Stuck-At Faults

The stuck-at fault model is a structural static fault model that assumes faults are permanent during the application of a test (Galey et al., 1961). A defect that causes an interconnection to be permanently tied to a fixed voltage, representing the logic value $x \in \{0, 1\}$ is modeled as a stuck-at- x signal line at the gate level. A signal line at the gate level may be an input of a gate or the output of a gate. The most common variation of this fault model is the Single Stuck-at (SSA) fault model. Physical defects that naturally fall into this category are shorts to power and ground, but many other defects cause stuck-at behavior. Figure 3.1 shows an AND gate with the input signal line stuck at the logic value zero.

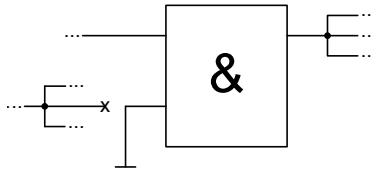


Fig. 3.1. A Stuck-at-0 Fault

The stuck-at fault model has a number of advantages that have facilitated its wide-spread use. The stuck-at fault model is not only simple to understand, it is also relatively straight-forward to implement in automated tools, requiring no access to other information sources such as layout or process technology specifications. The number of faults grows linearly in circuit size and the identification of fault-equivalence classes allows to further reduce the number of faults to about 50% (Bushnell and Agrawal, 2000). Furthermore, fault simulation and test generation algorithms can be improved with numerous algorithmic optimizations related to the properties of stuck-at faults (Kochte et al., 2010a; Schulz, 1988).

The stuck-at fault model was found to cover a wide range of physical defects and tests generated for the stuck-at fault model detect many faults from other fault models as well (Patel, 1998; Timoc et al., 1983). Furthermore, more advanced fault models may be constructed and analyzed using simple extensions retaining the aforementioned advantages (Holst and Wunderlich, 2009; Wunderlich and Holst, 2010).

A circuit node that has an undetectable stuck-at fault is said to be redundant, since the circuit can always be simplified by removing at least one gate or gate input. The generation of tests for stuck-at faults is in the set of NP-complete problems (Garey and Johnson, 1979; Ibarra and Sahni, 1975). The complete classification of

untestable faults of a circuit is a common benchmark for test generation methods (Drechsler et al., 2008).

3.2 Bridging Faults

In contrast to stuck-at faults, which focus on defects causing shorts to the power interconnect, bridging faults try to model shorts between the data interconnect lines (Bushnell and Agrawal, 2000; Renovell et al., 1994). Such defects can cause a wide range of effects on the logical behavior of a circuit. Figure 3.2 shows some of the most common types of bridging faults.

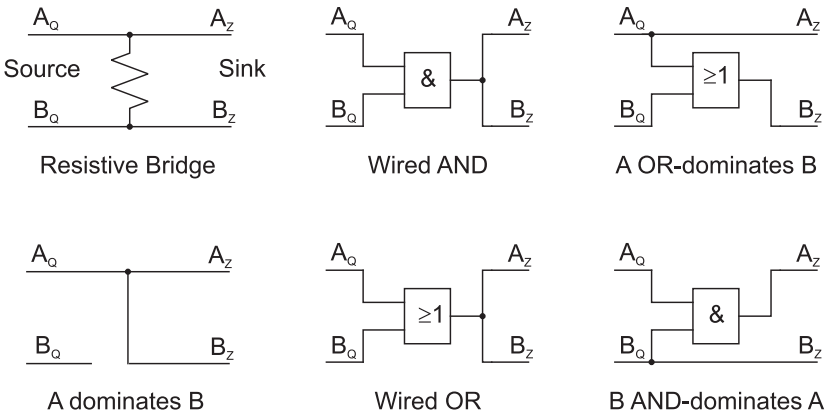


Fig. 3.2. Types of Bridging Faults

The most general of these models is the resistive bridging model (Renovell et al., 1995) that tries to model the specific resistance of the defect. The resistive bridging fault model is a dynamic fault model since most of the faults impact the delay of the circuit only. Since the resistance is a continuous variable, the methods for efficient simulation and analysis of faults are significantly more complex and computationally extensive than techniques that only take into account the logic level (Engelke et al., 2009).

Static bridging faults are modeled either using a simple AND or OR gate or they use an aggressor/victim model, where one signal line dominates another line in some way. Since this fault model is similar to the stuck-at fault model with the addition of the aggressor condition (Wunderlich and Holst, 2010) some of the methods and optimizations done for stuck-at faults can be applied.

One disadvantage of the bridging fault models lies in the number of faults to be considered. For example, for a circuit with N gate input or output signal lines, the number of possible aggressor/victim pairs is $N^2 - N$ for each of the aggressor/victim fault models in Figure 3.2. Linear complexity can only be achieved if technology and layout information is taken into account to determine the set of likely aggressor/victim pairs.

3.3 Delay Faults

Testing for defects that impact the timing of a circuit has become a standard option in all industrial test methodologies. Several fault models have been proposed that try to cover different mechanisms leading to a circuit exceeding its timing specification. Gate delay fault models (Barzilai and Rosen, 1983; Iyengar et al., 1990; Levendel and Menon, 1986; Pramanick and Reddy, 1988; Waicukauski et al., 1986) assume that the defect is contained and affects the timing of a single gate or gate interconnect. Path delay fault models (Lin and Reddy, 1987; Smith, 1985) assume that defect mechanism can also affect multiple gates. The segment delay model (Heragu et al., 1996) tries to strike a trade-off between generality of the path delay fault model and the computational efficiency of the gate delay fault model.

This section gives an introduction to the gate delay fault model and the path delay fault model. To test for delay faults, a transition must be created at a gate and propagated along at least one path from the gate to an output or sequential element. To create the transition at least two patterns must be applied to the circuit. This creates a sequential test generation problem. Section 9.1 (→ ‘Transition Faults’ starting on page 85) will discuss this problem in more detail.

3.3.1 Gate Delay Faults

Many defects such as those due to material impurities or process contamination often lead to locally contained impact affecting just a single transistor or part of an interconnect (e.g. a via). If such a defect affects the timing of the circuit, it is likely that it affects just one single gate on a given path.

The gate delay fault model distinguishes slow-to-rise and slow-to-fall faults, since most defects that affect one transistor only affect one of the two possible transitions. Furthermore, the gate delay fault model introduces a continuous variable, the amount of delay introduced. Only gate delay faults that exceed the timing slack available on a given propagation path will cause a faulty response. Much of the difficulty of the gate delay fault model goes into determining the size of this variable as part

of fault simulation (Iyengar et al., 1990) and deriving a representative fault coverage (Pramanick and Reddy, 1997).

The transition fault model removes this complexity by only considering gate delay faults that are larger than the cycle time. This is based on the observation that many delay faults indeed are gross delay defects (Geilert et al., 1990) (one study found gross defects up to 15 clock cycles long (Bula et al., 1990)). The resulting fault model only considers two faults per signal line, the gross slow-to-fall and slow-to-rise faults, and hence the number of faults behaves and grows similar to the stuck-at fault models. Furthermore, test generators and fault simulators for stuck-at faults can be adapted to support the transition fault model and some of the same algorithmic optimizations and evaluations apply.

Robust and non-robust tests for delay faults are distinguished. A robust test guarantees that a fault cannot be masked by other delay faults and variations in the circuit. This is achieved by requiring all of the off-path signals along the transition path to be at static values and only the on-path signal have a transition. Not all paths may have robust tests. The concept of robust tests is most often applied with the path delay faults presented in the next section (Lin and Reddy, 1987; Park and Mercer, 1987). But it can also be extended to gate-delay faults (Pramanick and Reddy, 1988).

In this work, the evaluations for delay tests are done for the transition fault model and does not distinguish between robust and non-robust tests.

3.3.2 Path Delay Faults

Defect mechanisms can also affect the timing behavior of larger portions of a circuit. Especially in recent process technology nodes, gate delay is also subject to statistical variation (Srivastava et al., 2005), which means that every gate may be subject to increased delay due to this. The path delay model considers all of the gates and interconnections along a path (Lin and Reddy, 1987; Smith, 1985).

The disadvantage of the path delay model is that in the worst case the number of paths is exponential in the number of signal lines in the circuit. Pruning techniques based on static timing can significantly reduce the number of path delay faults to be considered (Liou et al., 2002). But power-aware and statistical design increase the number of potentially critical paths that must be considered.

Concepts of Built-In Self-Test

Built-In Self-Test (BIST) is a class of frequently applied Design-for-Test techniques that allow a circuit to be tested with very little external stimuli (Bardell et al., 1987; Wunderlich, 1998). Often only one signal for the BIST mode and one oscillator for the Phase-Locked Loop (PLL) are required.

BIST is used in a large number of test applications, such as:

- For at-speed testing for defects impacting the delay of the circuit. Since all actual test stimuli are generated on-chip, they are easily applied at the functional frequency of the design.
- To reduce the need for expensive Automated Test Equipment (ATE), since fewer external stimuli have to be provided and only at low frequency.
- To reduce the test application time, since a large number of tests can be applied at high frequency.
- To protect third-party intellectual-property (IP) since only a minimal test specification must be provided.
- To allow for testing during the life-cycle of the product, since BIST can be applied in the field and short test times allow it to be executed frequently or even concurrently.

This chapter will give an overview of the types of BIST usually encountered. Furthermore, it will describe the most common-place class of BIST in additional detail.

4.1 Short Taxonomy of BIST

A plethora of BIST techniques have been developed. Their properties allow them to be classified into the groups shown in Figure 4.1. Offline BIST targets the application of self tests while the circuit is in a special mode and environment, such as in a special

mode for manufacturing tests applied by an ATE. If the BIST is executed by a functional design unit such as a micro-processor it is called functional BIST. In structural BIST, special hardware is added to execute the self-test. On the other hand, Online BIST applies self-tests while the entire circuit is in the functional mode and in the environment of the final product. The BIST can either be executed concurrently with the main-objective that the circuit performs. Or it can be performed non-concurrently with the circuits main-functionality temporarily suspended, in which case some of the offline techniques can be reused.

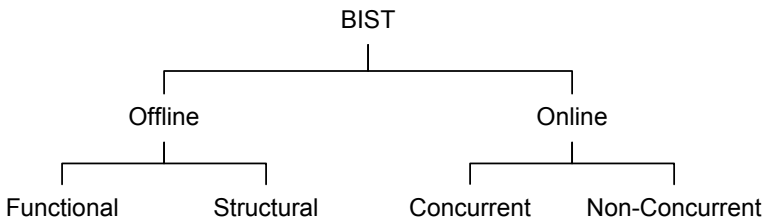


Fig. 4.1. BIST Classification

Figure 4.2 depicts the general structure of a Circuit Under Test (CUT) that has been extended by hardware that performs a structural BIST. A Test Pattern Generator (TPG) applies test patterns to the inputs of the CUT and a Test Response Evaluator (TRE) measures the circuit responses. A BIST controller coordinates the TPG and TRE and ensures that the CUT is cycled through the appropriate modes of test execution.

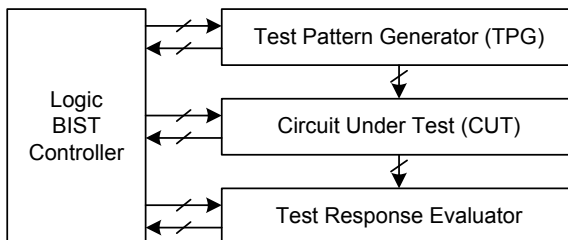


Fig. 4.2. Basic Principle of Structural BIST

Two types of structural BIST are distinguished. In test-per-clock structural BIST a new set of faults in the CUT is tested in every clock cycle. This technique is

best-suited for purely combinational designs or simple pipelines. Test-per-clock techniques for general sequential circuit integrated the function of the TPG and the TRE into the flip-flops of the circuit, for example by Built-In Logic Block Observers (BILBO) (Könemann et al., 1979) or the circular self-test path (Krasniewski and Pilarski, 1987).

Test-per-scan structural BIST (abbreviated: scan-based BIST) takes advantage of special sequential cells in the circuit that can operate in a special test mode to form a shift register through all cells. This way, each sequential element becomes easily controllable and observable. Testing a set of faults then consists of three steps, shifting in a stimulus, capturing the circuit response in functional mode and shifting out the response. If multiple tests are applied, the shift steps of two tests can be overlapped. Test-per-scan structural BIST is the most widely used BIST class for random logic circuits and is supported by all commercial DfT tools and methodologies.

The novel techniques outlined in this work target the structural test-per-scan offline BIST. But the basic ideas can be applied to other BIST classes as well. The next sections introduce the common-place components of test-per-scan BIST such as scan design, TPGs and TREs in additional detail.

4.2 Scan-based Built-In Self-Test

This section introduces the scan-path as the central DfT structure in test-per-scan BIST. It also presents the most common TPG and TRE structures employed in test-per-scan BIST. Finally it presents the Self-Test using MISR and Parallel shift register Sequence Generator (STUMPS) architecture that is used in most industrial test-per-scan BIST implementations.

Scan design is the most important technique in design for testability. It significantly increases the testability of sequential circuits and enables the automation of structural testing with a fault-model. Figure 4.3 shows the general principle of scan design. Each sequential element of the circuit is replaced by a scan cell. When the signal *scan enable* is set to '1', the circuit is operated in a way that all of the memory elements form a single shift register, the scan path. Multiple, parallel scan shift registers are called scan chains. An arbitrary circuit state can thus be shifted in through the scan path, even one that is not ordinarily reachable in functional operation.

The two most common types of scan cells are the mux-D cell (or mux-scan cell) (Williams and Angell, 1973) and the level-sensitive scan-design (LSSD) cell (Eichelberger and Williams, 1977) (Figure 4.4). The mux-D cell is a regular edge-triggered

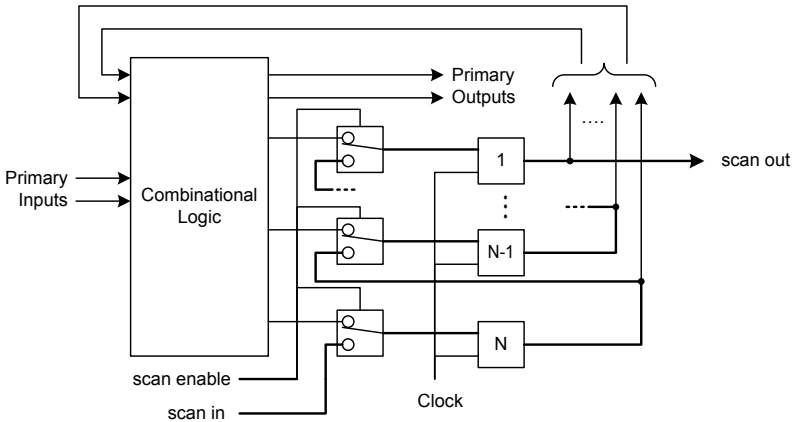


Fig. 4.3. Principle of the Scan Path

memory cell with a multiplexer that selects between the data input and the scan input. Level-sensitive scan-design (LSSD) is a scan method for latch-based designs.

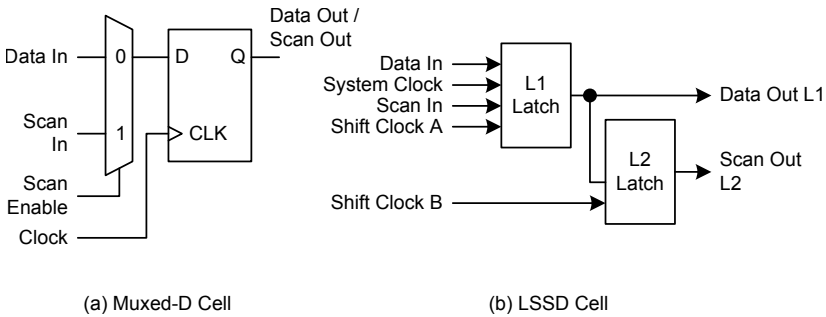


Fig. 4.4. The Two Most Common Scan Cell Designs

To reduce the overhead of scan design, only a subset of the sequential elements may be replaced by scan cells (Trischler, 1980). If certain criteria are met in the selection of sequential elements to be replaced, full testability of the CUT can still be achieved (Gupta and Breuer, 1989; Kunzmann and Wunderlich, 1990).

4.3 Test Pattern Generators

TPGs for structural BIST are commonly classified into 3 categories that are pseudo-random generators (Bardell et al., 1987), exhaustive or pseudo-exhaustive pattern generation (Gupta et al., 1996; Hellebrand and Wunderlich, 1989; McCluskey and Bozorgui-Nesbat, 1981; Wang and McCluskey, 1986) and deterministic generation of pre-computed patterns stored on-chip (Agarwal and Cerny, 1981; Koenemann, 1991). Pseudo-random techniques may be augmented by weighted pattern generation (Bardell et al., 1987; Wunderlich, 1987). Approaches that fall into multiple classes at once are called mixed-mode schemes (Hakmi et al., 2007; Wunderlich and Kiefer, 1996).

Deterministic pattern generation can have undesirable area requirements to store or represent the pre-computed patterns and exhaustive or pseudo-exhaustive techniques have prohibitively long test times on large designs. Hence, most BIST applications resort to pseudo-random test generation and its variations. This section gives an overview of the pseudo-random TPGs and their combination with weighted pattern generation.

4.3.1 Pseudo-Random Pattern Generation

Pseudo-Random Pattern Generators (PRPGs) are attractive TPGs for their low hardware overhead with reasonable test times on random-testable circuits. PRPGs can be implemented as Linear Feedback Shift Registers (LFSRs), one-dimensional Linear Hybrid Cellular Automata (LHCAs) or as accumulator-based designs. LHCAs are a collection of locally communicating memory cells whose interaction is governed by a simple set of rules (Hortensius et al., 1989; Serra et al., 1990). LFSRs are simple Moore machines that implement shift registers with an input function that is a linear function of the current state.

LFSRs and other linear feedback ring generators are the most wide-spread and best-understood TPGs. The rest of this subsection focuses on this type of TPG and its implications on scan-based BIST DfT.

The canonical form of the LFSR is the Standard LFSR (SLFSR). The SLFSR with length k implements a feedback function $x_{k-1} = \sum_{i=0}^{k-1} h_i x_i$ where the sum is defined as the modulo-2 addition, h_0, \dots, h_{k-1} is a set of coefficients and x_0, \dots, x_{k-1} are the state variables of the sequential elements in the shift register. All other x_i take the value of their predecessor $x_i = x_{i+1}$. Figure 4.5 shows the generalized SLFSR implementing this feedback function. Note that the coefficients h_i simply translate into the presence of a connection and an XOR gate. The outputs of the sequential

elements x_i are also called taps of the LFSR. If the SLFSR is initialized to a state that is not all-0 or all-1, it will produce a periodic sequence of states with associated output (Golomb, 1981).

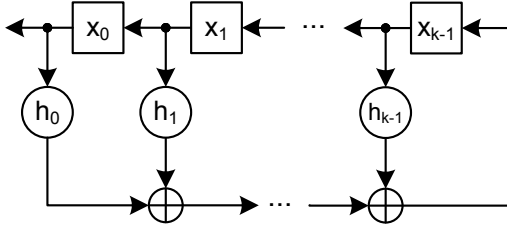


Fig. 4.5. Standard LFSR

The feedback function of the SLFSR can be associated with a polynomial over the finite field $GF(2)$:

$$h(X) = X^k + \sum_{i=0}^{k-1} h_i X^i$$

This polynomial is called the characteristic polynomial or feedback polynomial of the LFSR.

The period of the SLFSR of length k is the smallest positive integer P such that $h(X)$ divides $1 + x^P$. The period cannot be greater than $2^k - 1$ and LFSRs with this period are said to have a maximum-length sequence. The associated polynomial is called a primitive polynomial of degree k over $GF(2)$. A primitive polynomial is irreducible, i.e. it cannot be expressed as a product of two or more non-trivial factors.

The output sequence of a SLFSR with length k and primitive feedback polynomial has the following properties (Golomb, 1981):

- It consists of 2^{k-1} ones and $2^{k-1} - 1$ zeros.
- There is one consecutive sub-sequence of k ones and one consecutive sub-sequence of $k - 1$ zeros.
- There is an autocorrelation between the possible sequences of the SLFSR. Two sequences of the LFSR circularly shifted from each other will be identical in $2^{k-1} - 1$ positions and will differ in 2^{k-1} positions.

If multiple scan chains are fed from several taps of the same SLFSR, each sequence will be shifted by just a few cycles from each other. Even if the LFSR is very long, linear dependency between the sub-sequences will impact the quality of the generated test patterns. This problem can be alleviated by inserting a phase shifter

(Figure 4.6) that creates sub-sequences with larger distance from each other (Bardell, 1990; Rajski and Tyszer, 1998). The phase shifter selects a linear combination of several taps of the LFSR which results in a sequence that is cyclically shifted from the original. Ideally, the phase is selected for a high absolute distance from the original sequence (Rajski and Tyszer, 1998).

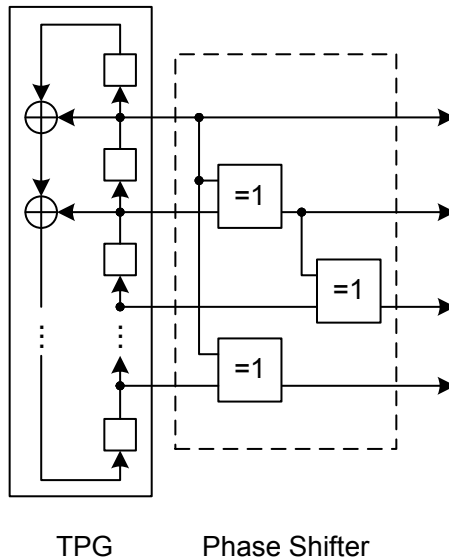


Fig. 4.6. LFSR-based TPG with Phase Shifter

Besides the SLFSR, LFSRs may also be constructed with the XOR-gates between the sequential elements. The so-called Modular LFSR is depicted in Figure 4.7. MLFSRs are faster than SLFSRs since they avoid the deep XOR-tree and the sequences from individual taps have larger distance.

4.3.2 Weighted-Random Pattern Testing

While pseudo-random pattern generators can generate a large test set with simple hardware, certain stimuli may not occur at all even if all subsequences are applied exhaustively (e.g. because of restrictive LFSR length or linear dependence as mentioned in the previous section). Often only a limited test set is applied due to test time

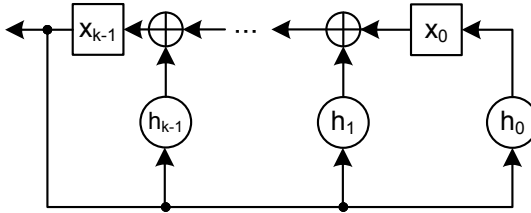


Fig. 4.7. Modular LFSR

constraints, in which case just the likelihood of certain specific stimuli may be too low (Savir and Bardell, 1983; Seth et al., 1990; Wunderlich, 1985).

In this case, weighting circuitry can be added to the pseudo-random TPG that biases the probability of zeros and ones. This simple circuitry usually consists of a set of AND and OR gates to achieve the desired effect (Bardell et al., 1987). The weights are usually made programmable, since several weights may be required to achieve acceptable test coverage (Wunderlich, 1987). Very often, different probabilities are required for each LFSR tap (Wunderlich, 1988) and programmability is present for each tap individually.

4.4 Test Response Evaluation

Similarly to the input side, the test response evaluation requires that all expected circuit responses must be stored on-chip. The number of observables especially in scan-testing can be very high and it is not viable to store all of the responses. Usually, the responses are compacted significantly to very few bits and subsequently only the so-called signature must be compared (Frohwerk, 1977). Naturally, the information loss involved may lead to masking and aliasing of erroneous circuit responses.

The literature distinguishes two types of compaction (Saluja and Karpovsky, 1983; Sinanoglu and Orailoglu, 2001): space compaction and time compaction. In space compaction, a combinational network computes check-bits for each circuit response, for example using a linear code or some other type of error detecting or correcting code. In time compaction, a finite state machine is constructed that changes its state based on the circuit responses and thus computes a signature across all circuit responses in a given time. Due to the significantly higher compaction ratio of time compaction, scan-based BIST almost always uses time compaction or a combination of space and time compaction.

The types of time compaction can be based on LFSRs implementing a type of cyclic redundancy check (Bushnell and Agrawal, 2000). Other approaches may employ simple accumulators or count the number of ones or transitions in the circuit response (Hayes, 1976). In BIST applications, LFSR-based time compaction is usually preferred due to its high compaction rate with low aliasing. This section introduces the aspects of LFSR-based time-compression that are relevant to BIST.

For LFSR-based time compaction, the LFSR is used as a circuit to divide a given polynomial (representing the circuit response) by the characteristic polynomial and storing the remainder of this division. The input is thus represented as the data polynomial $D = d_k X^n + \dots + d_1 X + d_0$ where $d_k \dots d_0$ are the consecutive data bits at the LFSR input. The signature of all responses is then obtained as the remainder polynomial of this division. Figure 4.8 shows an MLFSR used as a time compactor for the stream of bits provided at d . $s_{k-1} \dots s_0$ store the state or signature bits of the LFSR and represent the remainder polynomial of the polynomial division.

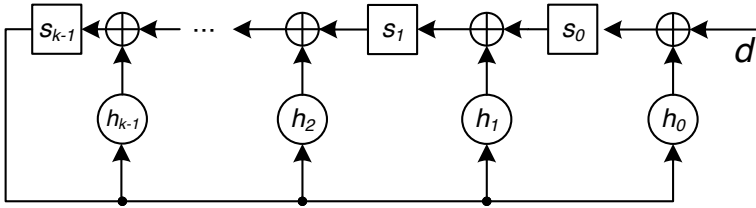


Fig. 4.8. Compactor Based on MLFSR

The MLFSR-based compactor in Figure 4.8 has only a single input. It is often desirable to share one LFSR among several circuit outputs or scan chains. LFSRs used for signature computation are easily extended to accept multiple inputs. Figure 4.9 shows an MLFSR that accepts the inputs $d_{k-1} \dots d_0$ in a single clock cycle. This compactor is called a Multiple-Input Signature Register (MISR).

As mentioned before, due to the unavoidable loss of information involved in compaction, faulty circuit responses may be masked or several faulty circuit responses may alias each other. The quality of a compaction method is thus measured in the likelihood that a masking or aliasing event occurs (Gupta and Pradhan, 1988; Williams et al., 1988).

If it is assumed that errors at the inputs $d_{k-1} \dots d_0$ of a MISR of length k occur independent of each other, then the probability of aliasing is 2^{-k} and this probability is independent of the initial state of the MISR (Williams et al., 1988). However, when

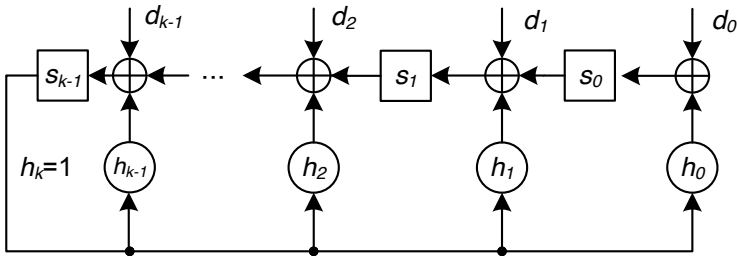


Fig. 4.9. Multiple-Input Signature Register (MISR)

this assumption is not true more accurate analysis can improve the results (Gupta and Pradhan, 1988; Stroele and Wunderlich, 1991).

Furthermore, it has been shown that the LFSR signature in many cases contains sufficient information to diagnose specific defects based on the signature alone (Elm and Wunderlich, 2010).

4.5 The STUMPS Design

The Self-Test using MISR and Parallel shift register Sequence Generator (STUMPS) DfT architecture (Figure 4.10) first presented in (Bardell and McAnney, 1982) uses the components for structural scan-based BIST outlined in the previous sections. It employs several taps of an LFSR-based TPG to shift test patterns into several scan chains in parallel and uses a MISR as the TRE.

The main advantage of the STUMPS architecture is that almost all industrial circuits employ scan design for testing the random logic and adding support for STUMPS-based BIST require very little hardware overhead and very simple hardware structures. Most commercial design automation tools support the automated insertion of the required structures. Hence, STUMPS is a popular BIST technique in industrial designs (Hetherington et al., 1999; Riley et al., 2005). The techniques for power reduction presented in this work focus on extensions of the STUMPS DfT architecture.

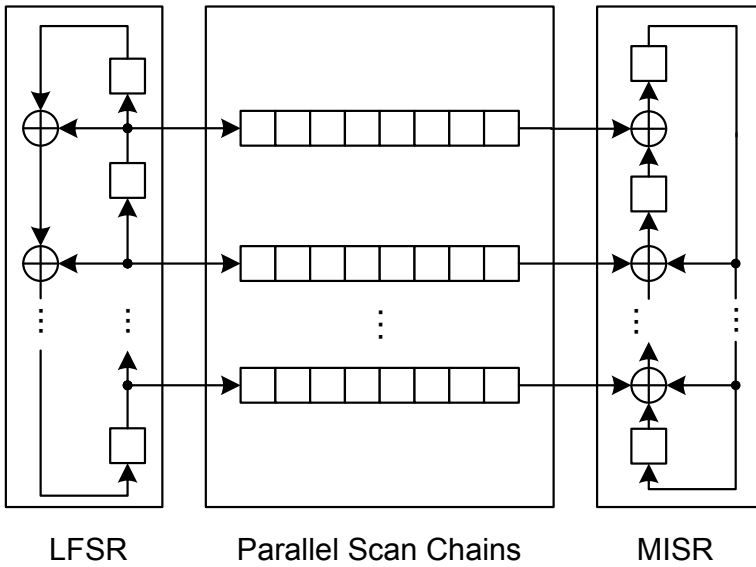


Fig. 4.10. Self-Test using MISR and Parallel shift register Sequence Generator (STUMPS)

Power During the Test of Integrated Circuits

It has been shown (Girard et al., 2009; Zorian, 1993), that the power consumed during test (test power) can be significantly higher than the power during normal mode (i.e. functional mode). Both peak power and average power are affected. There are several reasons for this:

1. Most of the circuit states during functional mode are confined to reachable states and show significant correlation. During scan-based test the circuit can be exercised in non-functional states that often show very little correlation, especially when testing with pseudo-random patterns.
2. It is common to test all sub-circuits or units of a chip in parallel to reduce the necessary test time at the cost of excessive power dissipation.
3. Sub-circuits that are only used to improve the test of the chip are idle during functional operation. Often, this overhead of test circuitry can be quite large.

This chapter will give an overview of the power consumption in integrated circuits with a focus on power consumption during scan-based testing. The first section will outline the impact of excessive power consumption during test. Then, the general power dissipation mechanisms in CMOS circuit are described as well as power metrics that are used to evaluate the power consumption during design, test and functional operation. Finally, the case of power consumption in scan-based testing is considered with additional detail.

5.1 Impact of Test Power Consumption

The impact of excessive power consumption on the test of a circuit can be subdivided into two effects: provisioning of the test environment and the increase in test cost. These two effects are related and must be considered together (Girard et al., 2009).

5.1.1 Test Environment

A circuit's timing behavior is highly dependent on the voltage and temperature of the test environment. These parameters must be well controlled during a test, otherwise even defect-free circuits can show errors that lead to discarding a chip. This yield loss can have considerable financial impact and must be avoided.

Providing a stable voltage to the circuit under test is a challenging task. During the test, the instantaneous power consumption of the circuit under test can be highly fluctuating, causing noise and voltage droop which severely degrade the timing behavior (Lin, 2012). This effect can occur at several hierarchy levels. At the transistor level, power is delivered through a grid that was designed to sustain functional operating current. At the module or chip level, long connectors from the power supply to the CUT result in high inductance and contacts are not soldered during the test and are subject to increased and varying resistivity.

The temperature fluctuates at a much higher time scale mostly due to changes in average power consumption. At the module test cooling can be challenging and at the wafer test power may be partially dissipated through the fragile test probes. High temperature during the test can increase the resistivity of the on-chip interconnect and degrade the timing behavior. Moreover, excessive temperature can degrade the chip itself and even lead to feedback effect called thermal runaway. This can lead to the destruction of the circuit under test and can require expensive repair of the test fixture or of test probes.

5.1.2 Test Cost

As outlined above, components that provide the test environment such as power supplies, tester probes and cooling must be appropriately sized, involving significant cost (Ishida et al., 2012). The cost of test equipment must be amortized across the entire production time and contributes significantly to the overall cost of manufacturing a product.

At the chip level, power grids may have to be adapted to test power supply requirements, transistor density in hot-spots may have to be decreased and chip packaging may have to be adapted to accommodate high current and temperature. This has impact on the cost of the product even before taking into account test cost.

Sometimes, an ad-hoc solution to test power consumption can be to decrease the shift frequency of scan-based tests or to test cores sequentially instead of testing several in parallel. While this solution can alleviate some of the effects, it always results in a significant increase in the test time. The test time in turn affects tester through-

put and is an even more significant cost driver than oversizing of tester components. Hence, the method presented in this work avoids increasing the test time.

As mentioned above, defect-free chips may be classified as faulty due to power supply noise or temperature impacting the timing behavior. The yield reduction will have significant impact on manufacturing cost. At worst, operating the circuit outside the specified limits can affect long-term reliability of the shipped product, leading to expensive field returns.

5.2 Power Consumption of the Circuit under Test

The power consumption or dissipation of CMOS circuits can be split into static and dynamic parts (Rabaey et al., 1996).

$$P = P_{stat} + P_{dyn} \quad (5.1)$$

Dynamic power is consumed by the charging (and discharging) of gate capacitors whenever a circuit node incurs a switching event (i.e. a logic transition $0 \rightarrow 1$ or $1 \rightarrow 0$). The static power dissipation is caused by the leakage current as well as other currents that constantly draw from the external power supply.

Major contributors to static power consumption are transistor effects such as the reverse-bias pn junction leakage, the subthreshold conduction current and the pattern-dependent oxide tunneling current (Agarwal et al., 2004). Recent advances in circuit technology brought a substantial increase in leakage current, which is currently tackled by techniques such as power gating or sleep transistors (Kim et al., 2004). In the following, the static power consumption is not taken into account.

The dynamic power is dissipated mainly due to charging and discharging of capacitance, but also through short-circuit currents. Figure 5.1 shows a simple CMOS inverter with a load capacitance C_L . When the input voltage of the inverter V_{in} switches from Gnd to V_{DD} , the NMOS transistor is turned on and the PMOS transistor is turned off. Subsequently, the load capacitance C_L is discharged through a resistive path through the NMOS transistor.

This process dissipates a certain amount of energy in the NMOS transistor (or PMOS transistor for the opposite transition). The energy dissipated in the transistor is:

$$E_{diss} = \frac{1}{2} C_L V_{DD}^2$$

If the transistor is turned on and off with frequency f , then the average power dissipation is:

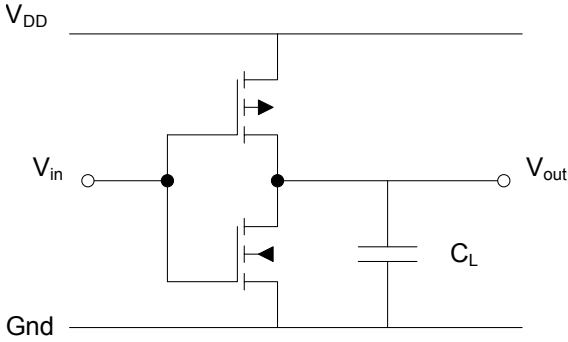


Fig. 5.1. CMOS Inverter With Load Capacitance C_L

$$P_{diss} = C_L V_{DD}^2 f$$

Since the switching of the transistors is not discrete but gradual, there is a small time frame in which both transistors are conducting and create a path from V_{DD} to Gnd . The duration of this short-circuit current depends on the threshold voltages of the transistors and the slew of the input transition.

5.2.1 Power Metrics

Power consumption is usually characterized statistically. This section introduces the commonly used measures.

The *total energy* is the energy consumed by the circuit in a given time-frame and is computed as:

$$E = \int_{t_0}^{t_1} P(t) dt$$

where t_0 and t_1 are mission times, for example the start and end of a test session. This measure is essential for battery operated circuits, where the run time is limited by the energy stored in the battery.

The *average power* is important to determine heating effects and long-term aging effects such as electromigration, which are accelerated by increased current and temperature. It is computed as $P_{avg} = E/(t_1 - t_0)$.

The *maximum instantaneous power* is the power at the time instant of the maximum amount of charge flux. The maximum instantaneous power can often be several orders of magnitude higher than the average power, due to the exponential discharge and low resistances to on-chip capacitance.

In contrast to this, the *peak single-cycle power* (sometimes simply referred to as *peak power*) is the maximum power consumption with a time window of one clock cycle. This measure is attributed to the fact that the instant with the highest power consumption is often decoupled by local capacitance and related to the clock signal switch. Transitions in the combinational logic that follow the clock transition must be supplied also from external decoupling capacitance and the peak power over the entire clock cycle determines the current supplied. In the remainder of this work, whenever the term peak power is used, it refers specifically to the peak single-cycle power.

The *peak n -cycle power* is the maximum of the power averaged over n clock cycles (Hsiao et al., 2000). The peak n -cycle power is used to determine local thermal stress as well as the required external cooling capability. Most of the common literature does not distinguish between average power and peak n -cycle power and often includes n -cycle averages under the term average power. When the term average power is used in this work, it refers to the average of the power consumption over a time frame long enough to include thermal effects.

The rate of change measured by di/dt , describes sudden changes in the power consumption. After a di/dt event, the parasitic inductance L of the circuit causes a decrease of the power supply voltage V_{DD} by Ldi/dt . Similar to the peak power, this measure determines the design of required decoupling capacitance. However, it is also used to model resonance effects in the power distribution network.

5.2.2 Power Dissipation in Synchronous Circuits

In synchronous sequential circuits, the memory elements update their state at a well-defined point in time. After the state of the memory elements has changed, the associated switching events propagate through the combinational logic gates. The gates at the end of the longest circuit paths are the last gates to receive switching events. At the same time, the longest paths usually determine the clock cycle.

Fig. 5.2 outlines the typical waveform of the current $I(t)$ during the clock cycles of a synchronous sequential circuit. In the example, the circuit contains edge triggered memory elements (e.g., flip-flops), so the highest current during the clock cycle is typically encountered at the rising clock edge. Subsequently, the switching events propagate through the combinational logic and the current decreases. The clock network itself also includes significant capacitance and both clock edges contribute to the dynamic power consumption as well.

As mentioned in the previous section, the peak single-cycle power is the maximum power consumed during a single clock-cycle, i.e. when the circuit makes a

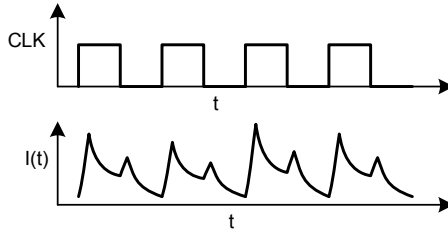


Fig. 5.2. Typical Power Consumption During a Clock Cycle

transition from a state s_1 to a state s_2 . Iterative (unrolled) representations of a sequential circuit are a common method to visualize the sequential behavior. Fig. 5.3 shows a sequential circuit, which makes a transition from state s_1 with input vector v_1 to state s_2 with input vector v_2 .

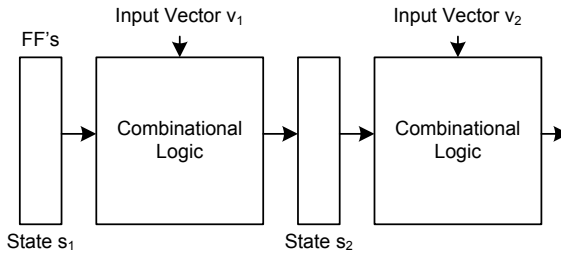


Fig. 5.3. Sequential Model to Analyze Peak Single-cycle Power

5.3 Power Consumption in Scan-based Testing

This section discusses the power consumption in circuit designs that implement one or more scan paths. The phases of a scan test and their implications on power are described, so that the techniques described in the rest of this work can be evaluated. The scan test consists of shifting, launch and capture cycles and many techniques only reduce the power consumption for a subset of these three phases.

For scan-based tests, power is divided into shift power and capture power (Girard et al., 2009). Shift power is the power consumed while using the scan path to bring a circuit from state s_a (e.g. a circuit response) to s_b (e.g. a new test pattern) (Fig. 5.4).

Power may also be distinguished according to the structure where it is consumed. A large part of the test power is consumed in the clock tree and the scan cells. In high frequency designs, clocking and scan cells can consume as much power as the combinational logic. Usually, only a small part of the power is consumed in the control logic (such as BIST controllers), the pattern generators and the signature registers. A detailed analysis of the contributors to test power may be found in (Gerstendörfer and Wunderlich, 2000).

Related Power-Aware BIST Techniques

The focus of this chapter is on BIST techniques for circuits that implement scan design to improve testability. This applies to all current VLSI designs. The first part of the chapter deals with test pattern generators which produce patterns that result in reduced peak and average test power. This is followed by a section discussing improvements of the design of the scan cells. Here, unnecessary switching activity is avoided by preventing the scan cells to switch during scan. This is achieved by gating either the functional output of a scan cell during shifting or by clock gating of the scan cell. Through careful test planning, clock gating can be employed to reduce test power without impacting fault coverage.

The third section of the chapter deals with the scan paths in the circuit. Here, the segmentation of the scan path reduces the test power without increasing test time. Other techniques insert tree-like muxing and logic gates into the scan network that can be used to reduce test power consumption.

6.1 Modified Test Pattern Generators

Low-power test pattern generators for BIST are based on reducing the number of transitions in the test patterns in order to reduce shift power. This is a straight-forward extension of regular weighted random pattern testing as introduced in section 4.3.2 (→ ‘Weighted-Random Pattern Testing’ starting on page 31). However, here the target is a trade-off between fault coverage and power consumption. Substantial power reductions can be achieved for the combinational logic (Zhang et al., 1999). But since a large amount of power is consumed in the clock distribution, reducing the number of transitions in the shifted pattern is often insufficient.

Based on the idea of weighted random pattern testing, techniques have been proposed that use split LFSRs (Sato et al., 2012; Wang and Gupta, 1997). Toggle flip-flops (Wang and Gupta, 2006) and hold flip-flops (Rajski et al., 2012) allow to control the number of transitions in a scan pattern, rather than just biasing the distribution

of 0s and 1s. A similar approach is to limit the number of transitions to a single scan chain per shift vector using gray code counters (Girard et al., 2000a). However, even with advanced techniques (Rajski et al., 2012) the test time (to reach equivalent fault coverage) is increased by an order of magnitude compared to standard pseudo-random test, unless test planning is applied (Solecki et al., 2012).

In deterministic BIST, it is possible to stop the TPG when no care bits should be injected into the scan pattern (Mrugalski et al., 2007). However, also with this power consumption in the clock tree is substantial and clock gating-based approaches are preferred if possible (Czysyz et al., 2009).

6.2 Low Power Scan Cells

Scan based on the muxed-D cell requires only a single clock signal to be routed to the scan cell and any type of flip-flop may be used as its basis. Hence, muxed-D can take advantage of a number of low-power flip-flop designs such as double-edge triggered flip-flops (Chung et al., 2002).

For LSSD, the shift operation is exercised using two separate clock signals A and B. These clock signals are driven by two non-overlapping clock waveforms, which provides increased robustness against variation and shift-power related IR-drop events. Figure 6.1 shows an LSSD scan cell implemented with transmission gates. The transmission gate design has very low overall power consumption (Stojanovic and Oklobdzija, 1999).

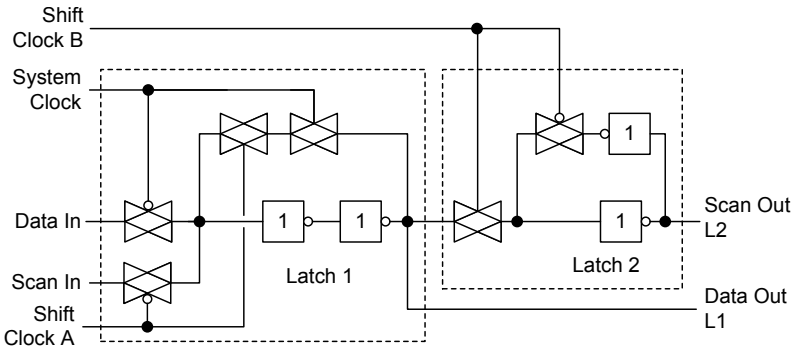


Fig. 6.1. LSSD Scan Cell Using Transmission Gate Latches

For designs such as Figure 6.1, a significant portion of the power is consumed in the clock buffers driving the transmission gates. Hence, clock gating at the local clock buffers is an important technique to further reduce power.

6.2.1 Scan Clock Gating

Clock gating is an important technique for power reduction during functional mode. Clock gating reduces the switching activity by two means: first by preventing memory elements from creating switching events in the combinational logic. Second by preventing the clock transitions in the leaves of the clock tree. A common application of clock gating during scan testing is to deactivate the scan clock during the application of useless patterns (Figure 6.2). Useless patterns do not detect additional faults that are not already detected by other patterns.

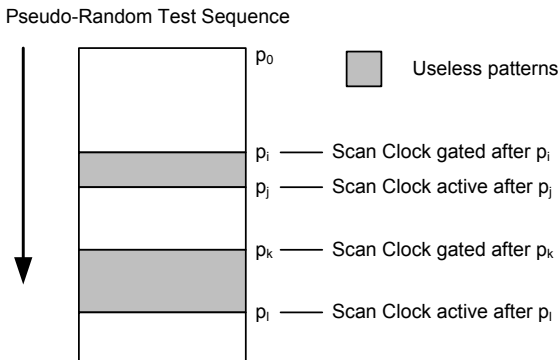


Fig. 6.2. Scan Clock Gating of Useless Patterns

During scan-based BIST, a sequence of pseudo-random test patterns is applied to the circuit. Fault simulation is used to determine the useless patterns and resimulating the patterns in reverse and permuted order may uncover additional useless patterns.

The pattern suppression of (Gerstendörfer and Wunderlich, 1999) employs a simple controller that deactivates the scan clock during useless patterns. (Girard et al., 1999) present a similar technique of suppressing useless patterns in non-scan circuits.

Figure 6.3 shows the DfT architecture for clock gating during the test. The circuit has the common self-test DfT of a scan path with a TPG and a TRE. The test controller generates the scan clocks and contains a pattern counter. A simple decoder generates the clock gating signal from the pattern count.

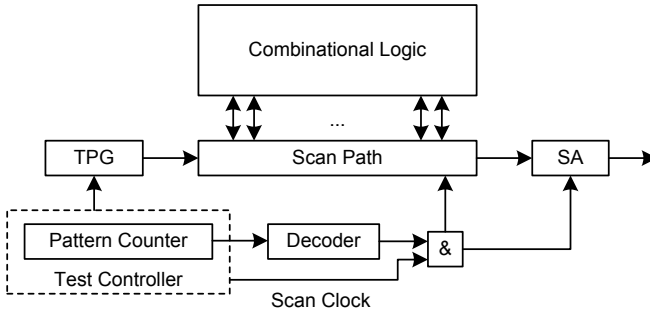


Fig. 6.3. Design for Test with Scan Clock Gating During Useless Patterns

Using the information obtained from fault simulation, a simple table is constructed. For example, the result of the fault simulation may look as follows:

Table 6.1. Fault Simulation Result for a Test Set with 16 Patterns

index	binary	# faults	index	binary	# faults
0	0000	17	8	1000	2
1	0001	9	9	1001	0
2	0010	4	10	1010	0
3	0011	0	11	1011	1
4	0100	5	12	1100	0
5	0101	2	13	1101	0
6	0110	3	14	1110	0
7	0111	0	15	1111	0

The first three patterns detect new faults and the pattern with index 3 does not. Shifting is suspended during patterns 3, 7, 9 and 10, and enabled during patterns 0, 1, 2, 4, 5, 6, 8, 11 and 12. The clock is enabled for pattern 12 to shift out the circuit response of pattern 11. The test controller stops the BIST after pattern 12. Now, the resulting Boolean function of the decoder is:

```

{0000, 0001, 0010, 0100, 0101,
 0110, 1000, 1011, 1100}           on-set

{0011, 0111, 1001, 1010}           off-set

{1101, 1110, 1111}                 dc-set

```

Fig. 6.4. Boolean Function of the Decoder

This function is minimized and synthesized using a standard tool flow. Figure 6.5 shows the decoder for the example.

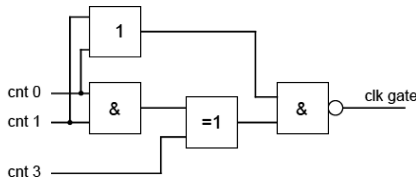


Fig. 6.5. Decoder for Pattern Suppression for the Example

For larger circuits, the overhead for the decoder is just a few percent as reported in (Gerstendörfer and Wunderlich, 1999). It has been shown that pattern suppression reduces the average power consumption by around 10%. However, the reduction may be significantly higher if the test length is very high or if the circuit has been designed for random testability. For pattern suppression, the scan clocks can be gated at the root of the clock tree.

6.2.2 Toggle Suppression

During shifting, the functional outputs of the scan cells continue to drive the combinational logic. Hence, the combinational logic is subject to high switching activity during the scanning of a new pattern. Except for the launch cycle in launch-off-shift transition tests, the shifting does not contribute to the result of the test.

Hence a very effective method to reduce the shift power is to gate the functional output of the scan cell during shifting. The gating can be achieved by just inserting an AND or OR gate after the functional output of the scan cell. However, in this case the entire delay of the AND or OR gate will impact the circuit delay. Instead, it is more desirable to integrate the gating functionality into the scan cell itself.

Figure 6.6 shows a muxed-D scan cell based on a master-slave flip-flop. The NAND gate employed to gate the functional output incurs only a very small delay overhead, since the NAND-input can be driven by the QN node of the slave latch. (Hertwig and Wunderlich, 1998) have reported that toggle suppression reduces average shift power by almost 80% on average. But the switching activity during the capture cycle is not reduced and overall peak power consumption is almost unaffected.

In order to use toggle suppression with launch-off-shift transition tests, the control signal for the output gating has to be separated from the scan enable signal. How-

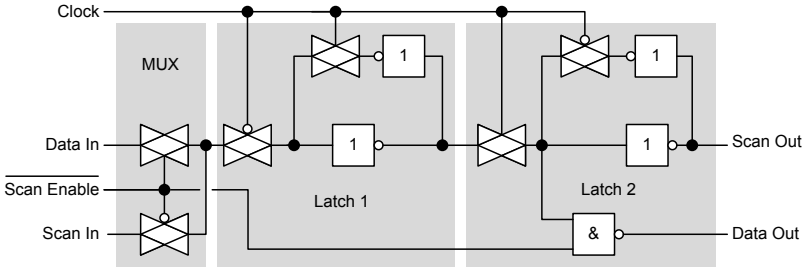


Fig. 6.6. Master-slave Muxed-D Cell with Toggle Suppression

ever, this increases the wiring overhead of the scheme significantly. If the separate control signal is combined with a latch shadowing the slave latch of a master-slave scan cell (Lin et al., 2012), the launch cycle toggling can be selectively suppressed per scan cell.

The techniques above reduce the peak power during shifting since all of the scan cells are forced to a specific value and the application of the test pattern to the combinational logic may incur switching in up to 50% of the scan cells. To provide additional control over the peak power for launch and capture cycles, the functional output of the scan cell can be gated using a memory element. The memory element then stores the circuit response of the preceding pattern and by appropriately ordering the test patterns, the peak power can be reduced. For example, (Zhang and Roy, 2000) have proposed the structure in Figure 6.7, which uses an asynchronous feedback loop across a multiplexer to implement a simple latch. Similar to the NAND-based approach, the impact on the circuit delay can be reduced by integrating the gating functionality into the scan cell. (Parimi and Sun, 2004) use a master-slave edge triggered scan cell and duplicate the slave latch.

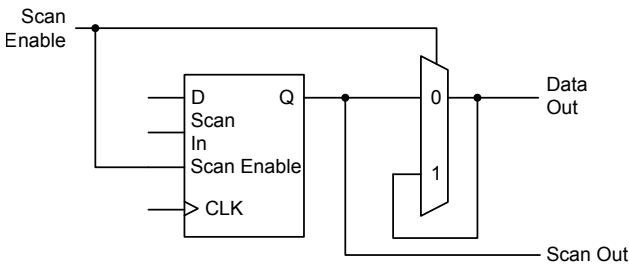


Fig. 6.7. Toggle Suppression Implemented with Multiplexer

It may be sufficient to apply toggle suppression to a subset of the scan cells. (ElShoukry et al., 2007) use a simple heuristic to select scan cells to be gated. The cost function is based on a scan cell's contribution to the power consumption and takes into account available timing slack. It was shown that adding toggle suppression to just 50% of the scan cells achieves almost 80% of the power reduction compared to adding toggle suppression to all of the scan cells.

6.3 Scan Path Organization

This section discusses how the scan path can be organized in a way that the shifting process uses less power and that it assists other techniques for power reduction. Figure 6.8 shows the general flow of scan insertion into a design. Commercial tools support all of these steps and the techniques discussed in this section may be used to extend or replace some of the steps in Figure 6.8.

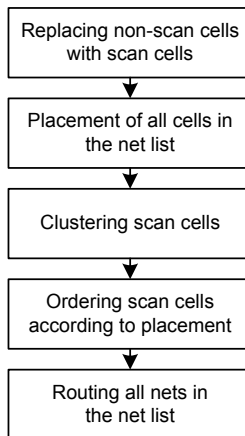


Fig. 6.8. General Scan Insertion Flow

6.3.1 Scan Path Segmentation

A common method to reduce the excess switching activity during shifting is to split the scan path into several segments. Shifting is then done one segment after the other. The segments not currently active are not clocked and do not contribute to shift power. The technique reduces both peak and average power.

Figure 6.9 shows the structure proposed by (Whetsel, 2000). Here, a scan path of length t is split into 3 segments of length $t/3$.

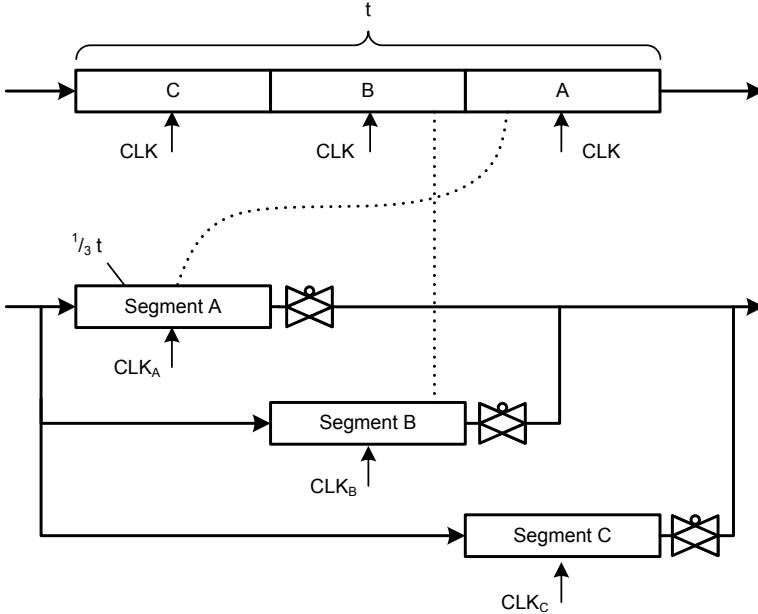


Fig. 6.9. Scan Path Segmentation

The activation of the segments is controlled using the scan clocks. Because the shift input is multiplexed using the clocks, only a multiplexer for the shift outputs is required. However, either the shift clocks for each segment have to be routed individually or scan clock gating is employed as described in section 6.2.1 (\rightarrow ‘Scan Clock Gating’ starting on page 47). Figure 6.10 shows the clock sequence for the example above.

For launch-off-shift transition faults, in the clock sequence of Figure 6.10 only the shift of the segment lastly activated launches a transition to be captured. In this case, it is possible to apply an additional launch shift cycle to all segments just before the capture cycle.

If the segmentation is done this way, the test time remains the same. For two segments, shift power is reduced by approximately 50%, for three segments the reduction is about 66%. (Whetsel, 2000) has reported that two or three segments have

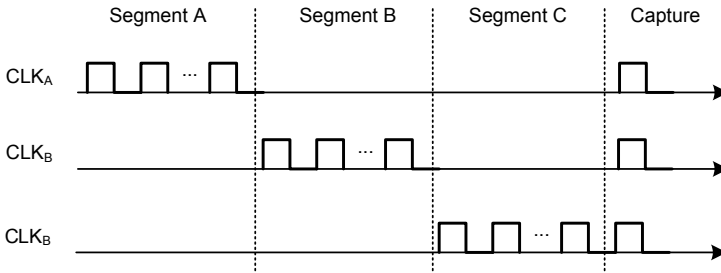


Fig. 6.10. Clock Sequence for the Scan Segmentation in Figure 6.9

the best ratio of power reduction versus implementation overhead. The technique reduces both the peak power during shifting as well as the overall test energy. Since test time is kept, average power is reduced as well. However, the power consumption during the capture cycle is not reduced by the clock sequence above.

If the DfT architecture consists of multiple scan chains anyway, like in the STUMPS architecture, the technique can also be applied using just the scan clock gating from Figure 7.1 from section 6.2.1 (→ ‘Scan Clock Gating’ starting on page 47). In this case, the test time is increased compared to scanning all chains in parallel.

6.3.2 Extended Clock Schemes for Scan Segmentation

The clock sequence in Figure 6.10 has two remaining drawbacks: first, the clock frequency used to shift the individual scan segment is not reduced and may be subject to local IR-drop effects. Second, the power of the capture cycle is not reduced, which is a significant issue especially in transition tests.

To solve the first problem, instead of shifting individual segments at full clock frequency the segments can be shifted in an alternating fashion. This technique is also often called staggered clocking or skewed clocking in low-power design. Figure 6.11 shows the clock sequence for the three scan segments A, B and C of Figure 6.9 as proposed by (Bonhomme et al., 2001). Now, each individual clock has a lower frequency. This increases the robustness of the pattern shifting against IR-drop events. (Girard et al., 2001) show how staggered clocking may be applied to the pattern generator as well.

The peak power of the launch and capture is not reduced with the previous clock sequence. The staggered clocking above can be done for the launch and capture clocks as well (Figure 6.12).

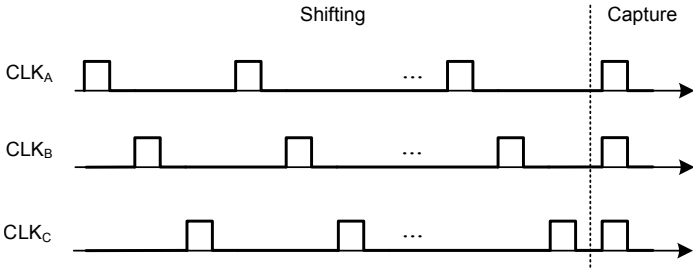


Fig. 6.11. Staggered Clock Sequence for Shift Peak Power Reduction

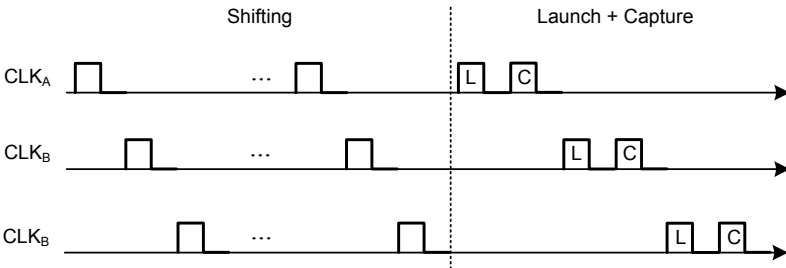


Fig. 6.12. Clock Sequence for Launch-capture Peak Power Reduction

In this case, only transition faults that are launched and captured by the same segment can be tested. Figure 6.13 shows an example where all of the flip-flops are contained in a single segment, which launches the transition, sensitizes the propagation path and observes the fault. In this case, the fault can be tested by only executing the launch and capture cycle for segment A. However, capturing segment A before segment B may change the justification for segment B and special care is required during test generation.

Most of the fault-coverage can be retained by using additional combinations of launch and capture clocks. The combinations of segments to be activated to detect a certain set of faults can be determined by mapping to and solving the set covering problem discussed in chapter 8 (→ ‘BIST Planning for Scan Clock Gating’ starting on page 75). The set of faults that can be tested using just the launch and capture clocks of a single or a few segments can be increased by clustering the scan cells into scan segments appropriately. (Rosinger et al., 2004) report that appropriate planning and clustering allow to reduce the peak power during capture by about 30-50%.

(Yoshida and Watari, 2002) use even more fine-grained clock staggering by manipulating the duty cycles of each scan clock. This allows to more closely interleave

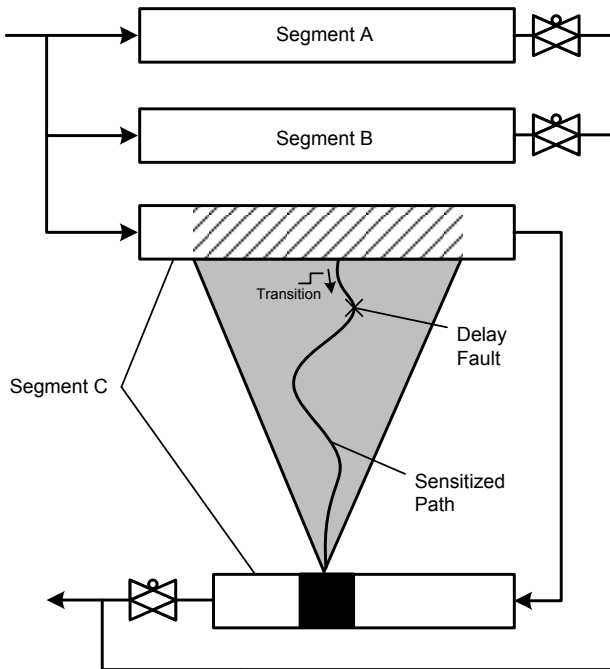


Fig. 6.13. Input Cone that is Contained in a Single Scan Segment

the shifting of several segment and can increase the possible shift frequency. However, the modification of the clock duty cycle requires a significantly higher design and verification effort.

6.3.3 Scan Tree and Scan Forest

The scan tree is a generalization of the scan path. In a scan tree, a scan cell's output may be connected to several other scan cells as seen in Figure 6.14. Scan forests are the extension of this concept to parallel scan chains.

Scan cells connected to the same fan-out will receive the same test stimuli. To avoid any impact on fault coverage, special care must be taken. For example, for stuck-at faults it is sufficient to ensure that all the scan cells in the input cone can be controlled independently (c.f. Figure 7.7). Two scan cells are called compatible if they are not part of any common input cone. For the scan tree, scan cells should be

ordered by their compatibility. (Chen and Gupta, 1995) use a graph-based approach to find (pseudo-)primary inputs that may receive the same test vectors.

Scan trees are often used to reduce test time and test data volume. The Illinois scan architecture by (Hamzaoglu and Patel, 1999) is a special case of the scan tree in which only the scan-in has a fan-out larger than one. (Hellebrand et al., 2000) combine the scan tree with a test pattern decompression technique to improve the compression efficiency.

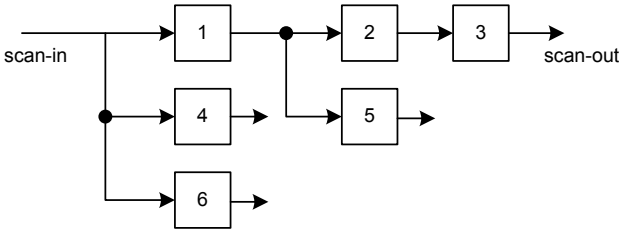


Fig. 6.14. Example of a Scan Tree

The scan tree may also be combined with a regular scan path (Figure 6.15). This mode of operation is often called "serial mode" and is used to provide conventional scan access to the circuit for debugging as well as for cases, where some of the scan cells are incompatible.

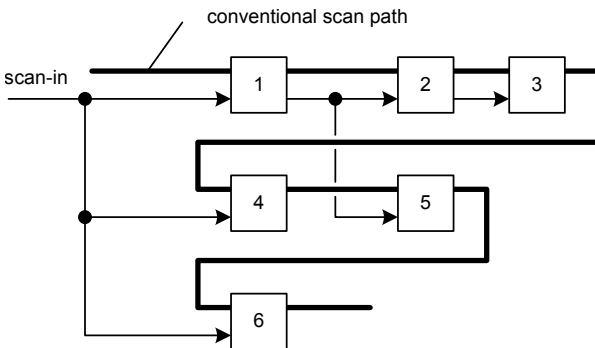


Fig. 6.15. Scan Tree Combined with Conventional Scan Path

The principle may also be applied to the fan-in of scan cells. For example in the double-tree design of Figure 6.16, the scan cells 8, 9 and 10 are computed as the

XOR of the two predecessors. Alternatively, additional control signals are used to select a predecessor with a multiplexer as suggested by (Bhattacharya et al., 2003).

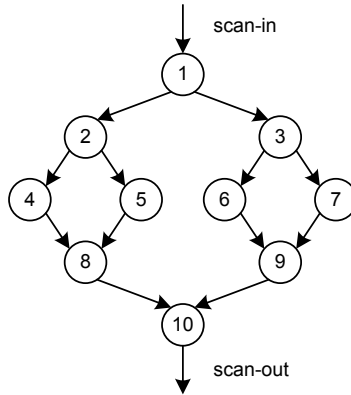


Fig. 6.16. Double Scan-tree

The scan segmentation of section 6.3.1 (→ ‘Scan Path Segmentation’ starting on page 51) is a special case of the double scan-tree with multiplexers in Figure 6.16. In fact, power reduction for the more general structure works in a similar way. Here, scan clock gating is used to reconfigure the double tree according to the care-bits in a test pattern. The scan gating is implemented such that any complete path through the double tree can be activated at a time. If a test pattern has care bits in scan cells 1, 5 and 8 it is sufficient to scan just the cells in the path 1 – 2 – 5 – 8 – 10 (Figure 6.17).

Select = 00 → Path-0: 1→2→4→8→10
 Select = 01 → Path-1: 1→2→5→8→10
 Select = 10 → Path-2: 1→3→6→9→10
 Select = 11 → Path-3: 1→2→7→8→10

Fig. 6.17. Scan Path Configurations for Figure 6.16

In most test sets, care bits are rather sparse and often only a few paths have to be scanned for a complete pattern. When constructing the scan tree of Figure 6.16, the scan cells that are most likely to contain a care bit should be closer to the root of the tree. The problem of clustering and ordering scan cells in this way can be mapped to the algorithms presented in sections 7.5 (→ ‘Scan Cell Clustering’ starting on

page 69) and to the scan cell ordering shown in (Bonhomme et al., 2003). (Xiang et al., 2007) have presented such a technique for constructing forests of scan trees.

For the double scan tree with clock gating, (Bhattacharya et al., 2003) report a reduction in shift power consumption of up to 90%. Similar to the scan segmentation in section 6.3.1, special attention is required for the peak power consumption during launch and capture cycles of transition tests. Also, the routing overhead must be taken into account when constructing the scan tree.

6.3.4 Inserting Logic into the Scan Path

Combinational logic can be inserted to apply certain patterns with lower shift power consumption. In most ATPG test sets, many patterns have similar assignments to the pseudo-primary inputs because of common path sensitization criteria between faults. The scan-in value of a scan cell can be inverted from other scan cells by inserting an inverter on the scan path before and after the scan cell. A scan cell that is inverted this way, is said to have inverse polarity with respect to the scan-in pin at the beginning of the scan path.

If the test set for a circuit is known, the probability of the assignment in each scan cell can be computed. This prediction is subsequently used to select the optimal polarity of the scan cells. This reduces the number of transitions during shifting, but not during the capture cycle. Figure 6.18 shows a single test cube that is filled using the repeat fill method. The test pattern has two transitions and the associated test response has one transition. By using the inverted output of the second scan cell in the example, the number of transitions in the final pattern and response is reduced to just one.

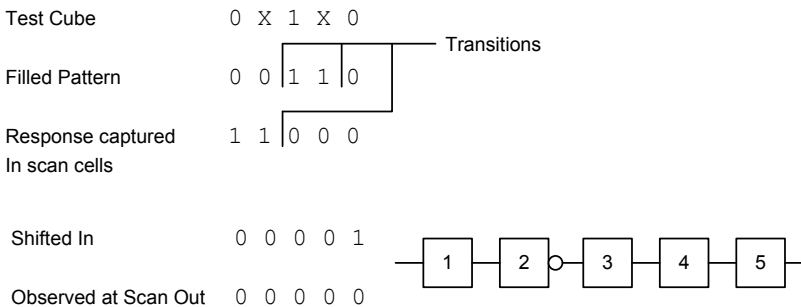


Fig. 6.18. Example of Scan Segment Inversion

However, often it is highly undesirable to have DfT structures that rely on a specific test set, since even a slight change in the test generation process may change the test set. Instead, more general measures from testability analysis can be employed, for example the methods COP by (Brglez et al., 1984) or PROTEST by (Wunderlich, 1985).

Correlation between the assignments to certain pseudo-primary inputs can be exploited to improve the prediction of the scan cell value and further reduce the number of transitions in the pattern. (Sinanoglu et al., 2002) embed a linear function into the scan path as depicted in Figure 6.19. Here issues of routing overhead and computational complexity mandate that the linear function is implemented over just a short segment of the scan path. The algorithm proposed by (Sinanoglu et al., 2002) works by the divide-and-conquer paradigm and uses a given test set as the input.

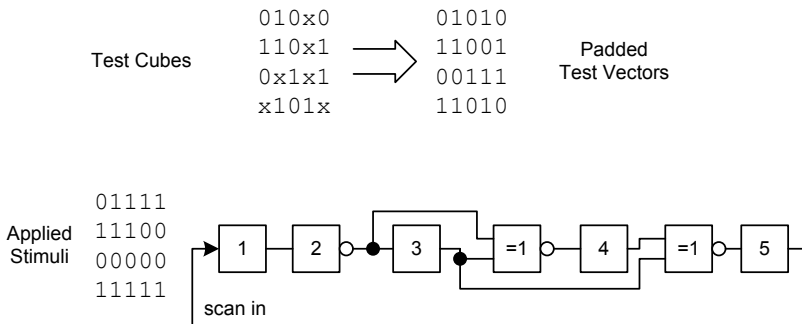


Fig. 6.19. Scan Segment Inversion by Embedding a Linear Function

This technique provides a 10-20% reduction of the shift power. But other than the selection of the scan cell polarity, the approach causes an area overhead of about 5%.

Inversion of certain scan segments can also improve the efficiency of pseudo-random BIST. Here, the goal is to increase the detection probability for test patterns with low weight (i.e. probability of a '1' \ll 50%). This is directly related to the estimation of the weights of the pseudo-random pattern test such as by (Wunderlich, 1987). To further increase the effectiveness of the test, (Lai et al., 2004) make the segment inversion configurable (Figure 6.20).

The BIST application is split into several sessions with different configurations. In this case, the reduction of the energy consumption of the entire test is achieved

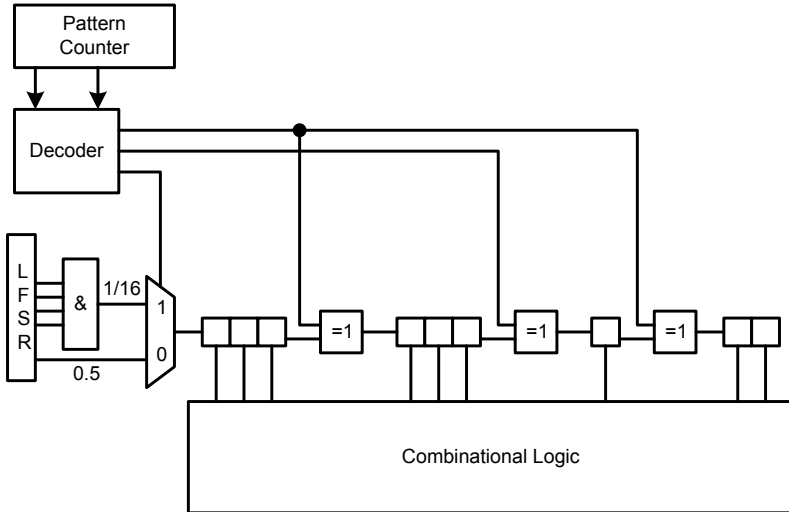


Fig. 6.20. Pseudo-random BIST with Scan Segment Inversion

by reducing the test time to achieve a certain fault coverage, similar to deterministic BIST.

Power-Aware BIST DfT with Scan-Chain Disable

The previous chapter has shown that clocking and organization of the scan path are two key factors in trying to reduce test power during BIST. The architecture presented here can be employed with industrial designs and its insertion and optimization can be fully automated.

A circuit designed this way has parallel scan chains that can be deactivated by clock gating, and the shifting of a scan chain may be avoided completely if the values controlled and observed by that scan chain do not contribute to the fault coverage of the test. In other words, turning off the clocks of the scan chain does not alter the fault coverage if just the chains with this property are deactivated. Since most faults can be detected in numerous ways, the search space for valid tests is very large. The problem of determining a set of BIST tests and configurations for such an architecture is called the BIST planning problem.

This chapter presents the DfT architecture in section 7.1. Section 7.2 qualitatively analyzes the power consumption of the architecture and introduces an approximation of the power consumption that can be used as a cost function when solving the BIST planning problem described in section 7.3. A formal description of the BIST planning problem is presented in section 7.4. A DfT synthesis approach for the automated implementation and optimization of the DfT architecture presented here is shown in section 7.5.

7.1 Scan Chain Enable by Clock Gating

To achieve improved granularity of the clock gating, the clocks may be gated closer to the memory cells (Wagner, 1988). However, the savings obtained using clock gating diminish if it is applied to individual cells. In functional design, clocks are usually gated at the register granularity (e.g. of 32 or 64 bits). During test, an acceptable granularity is to deactivate a scan chain, a group of scan chains or a test partition. Figure 7.1 shows a common-place design for test architectures that employ parallel

scan chains such as the STUMPS design (Self-Test Using MISR and Parallel SRSG). Here, the scan clock of every scan chain may be disabled individually by setting a Test Hold register.

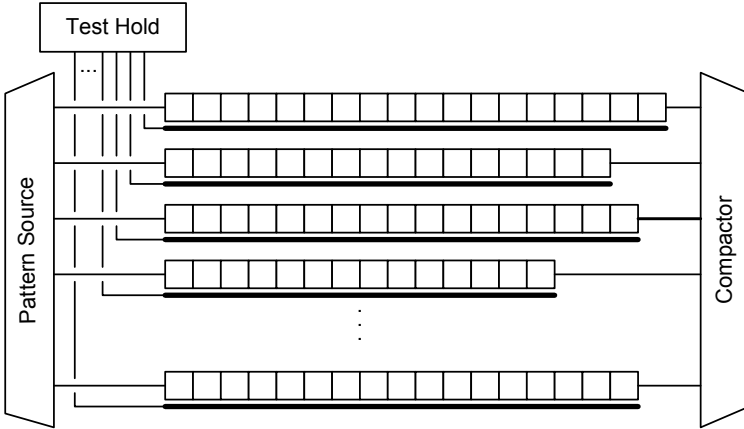


Fig. 7.1. DfT with Parallel Scan Chains and Clock Gating per Chain

In order to implement the scan clock gating, all of the clock gating functionality can be added to the local clock buffers. Figure 7.2 shows an example of a local clock buffer from (Pham et al., 2006) that allows for clock gating during functional mode and during scan as well. If signal *Test Mode* is set to '0', then the clock gating is controlled by *Activate* and the outputs *Load Clock B* and *System Clock* operate an associated LSSD cell in a master/slave mode. If *Test Mode* is set to '1', then the *Scan Clock A* and the *Load Clock B* operate the LSSD cell in scan mode. The signal *Test Hold* deactivates the clock during both scan and capture clocks of the test.

Such a clock buffer employs a dynamic logic gate or a latch for the clock gating. This allows to design the clock buffer in such a way that the clocks stay off during the complete clock cycle, even if one of the clock gating signals exhibits glitches. This way race conditions are avoided.

In the partial-scan design only a subset of all memory elements of a circuit can be scanned. In this case, it is highly beneficial to disable the clocks of the non-scan elements during shifting. This avoids the power consumption in the non-scan cells and the associated clock buffers. It also blocks the switching events of the combinational logic attached to the scan cells from propagating further through the circuit. Figure 7.3 outlines the principle.

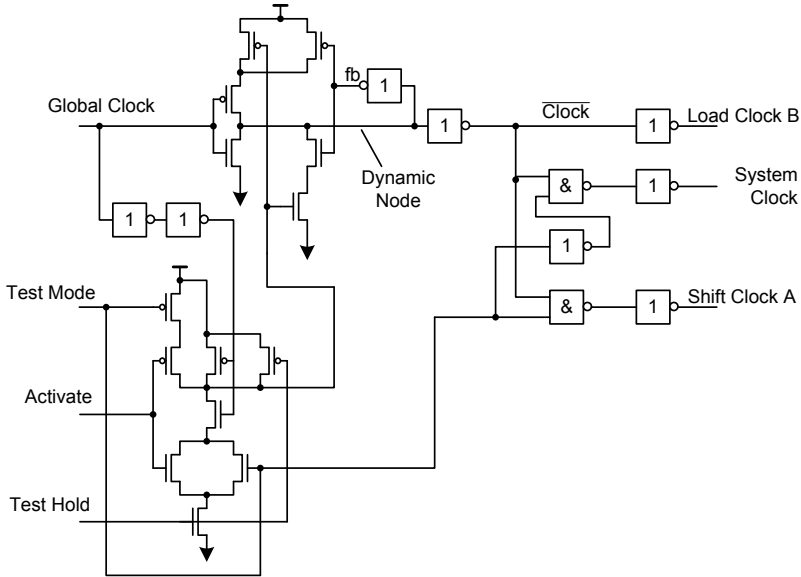


Fig. 7.2. Local Clock Buffer for Functional and Scan Clock Gating

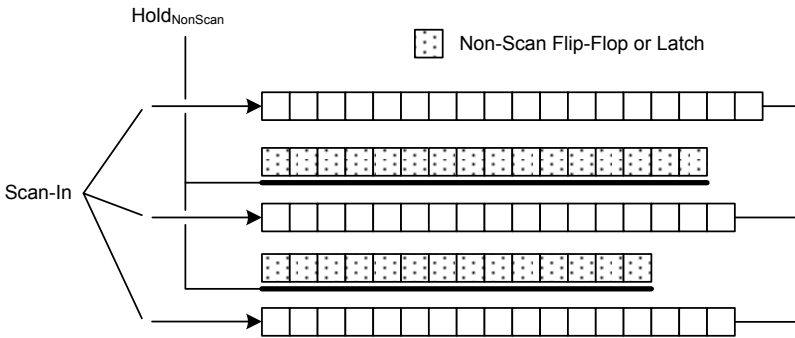


Fig. 7.3. Using Non-scan Cells to Block the Propagation of Switching Activity

7.2 Estimation of Power Consumption

For the optimization methods presented in the subsequent chapter, it is necessary to estimate the power consumption of an intermediate solution. The set covering based optimization method creates a significant amount of intermediate solutions, and therefore the estimation method must be very efficient. This section presents the linearized power estimation used in this work and evaluates its accuracy.

7.2.1 Related Models for Test Power Consumption

A fine grained estimation may be obtained by computing the weighted switching activity of each block during shift and capture. In the weighted switching metric, all switching events during a clock cycle are recorded/counted and are then weighted with the capacitance at the circuit node (respectively the electric charge that had to be moved). This metric is straight forward but computationally expensive. In the following, this method is used only for validation of the metric outlined in this section. Using acceleration hardware for scientific computation, the weighted switching activity can be computed orders of magnitude faster than with general purpose CPUs (Holst et al., 2012). Nevertheless, the run time of a single estimate is on the same order of magnitude as the set covering solving done in the test planning method presented here.

A more efficient estimate of shift power can be computed using the weighted transition metric, which considers transitions in the shift pattern (Sankaralingam et al., 2000). This metric is very useful, when generating, compacting or compression deterministic tests. However, for large circuits and pseudo-random test patterns, the additional costs do not provide significant gains of estimation accuracy since transitions are very frequent and randomly distributed.

To provide more accurate capture power checking, simulation is augmented by layout information (Kokrazy and Ravikumar, 2004). The overhead for the simulation is prohibitive for test generation or planning, and instead a power estimate can also be derived by looking at the number of detected non-target transition faults (Lee and Tehranipoor, 2008), where a higher number of detected faults implies higher switching activity. Since the effects of instantaneous power consumption on circuit delay also depends on the locally available decoupling capacitance, additional accuracy can be obtained by regional assessment of pattern safety (Devanathan et al., 2007).

7.2.2 Linearized Model

Up to 50% of the dynamic power is consumed in the sequential elements and the clock distribution to the sequential elements. Since this work focuses on power reduction based on scan clock gating, it seems that the number of clocked flip-flops is a suitable estimate. This essentially amounts to a linearization of the power consumption in the number of clocked flip-flops.

Hence, the estimated power consumption P^* of the circuit can be expressed as a function of the number of clocked flip-flops FF :

$$P^*(FF) = P_{base} + FF \cdot P_{flip-flop}$$

where P_{base} is the static power consumption of the circuit plus the power consumed in the branches of the clock distribution before the clock gates. $P_{flip-flop}$ is a fitting parameter that can be determined through simulation of a smaller number of test patterns.

7.2.3 Evaluation of Model Accuracy

In this section, the power model employed here is compared to a state-of-the-art power analysis tool that computes the weighted switching activity based on a simulation trace and takes the circuit timing into account.

For the evaluation of the accuracy of the linearized power estimate a DfT based on the DfT architecture in Figure 7.1 in section 6.2.1 (\rightarrow ‘Scan Clock Gating’ starting on page 47) was implemented for each circuit. This included full scan insertion according to the scan clustering presented in section 7.5 (\rightarrow ‘Scan Cell Clustering’ starting on page 69) with clock gating on a per scan chain basis.

All benchmark circuits were then synthesized with Synopsys Design Compiler targeting a 45nm cell library at a nominal process corner. For each circuit, the cycle time was manually determined and the maximum timing optimization effort was enabled. The timing information from the Static Timing Analysis (STA) in Design Compiler was exported via a Standard Delay Format (SDF) file.

The gate-level netlist and SDF file were then imported into Mentor Graphics ModelSim to allow timing accurate simulation including all glitching and clocking. A full Value Change Dump (VCD) trace was captured for 200 experiments per circuit. Each experiment simulated 10 pseudo-random patterns for a randomly selected configuration of the scan clocking. Before each experiment the sequential elements of the circuit are initialized to random values in order to avoid biasing due to all-0 or all-U initialization.

The gate-level netlist and VCD trace of each experiment were then analyzed using Synopsys PrimeTime PX. From each experiment, the average power consumption and the peak instantaneous power have been extracted for statistical evaluation. These results were then fitted using linear regression. The linear regression method employed here is ordinary least squares.

The root mean square error (RMSE) and mean absolute percentage error (MAPE) of the fitting were computed to evaluate the accuracy of the linearized power prediction.

$$RMSE = \sqrt{\frac{1}{N} \sum_{t=1}^N (s_t - s_t^*)^2}$$

$$MAPE = \frac{1}{N} \sum_{t=1}^N \left| \frac{s_t - s_t^*}{s_t} \right|$$

where $N = 200$ is the number of experiments s_t is the measured sample and s_t^* is the predicted value for the experiment.

Table 7.1. Estimation Errors Caused by Linearized Power Model

Circuit	Leakage + Base [mW]	Max. Avg. Power [mW]	Dynamic Power per FF [μ W]	RMSE [mW]	Error MAPE
p286k	3.157	8.680	0.296	0.138	1.91%
p330k	2.688	7.172	0.262	0.122	1.97%
p388k	4.264	14.50	0.429	0.285	2.44%
p418k	4.490	21.68	0.583	0.628	4.28%
p951k	10.21	86.02	0.830	0.912	1.82%

Figures 7.4 and 7.5 show two examples of the simulated power consumption over the predicted power consumption. Since the peak instantaneous power consumption has an entirely different magnitude from the average power consumption a different scale has been used in these graphs. In the graph the linear relationship between the prediction method and the simulation results is clearly visible. In the remainder of this work, this prediction method is used to estimate the power consumption of the circuit during a test plan. Further evaluations of this estimation method with actual hardware are reported in section 10.2 (\rightarrow ‘Wafer Test Experiments with the 45nm Cell/B.E. Processor’ starting on page 95).

7.3 Principle

Most faults detected by the complete test can be detected by several sessions and may often be observed in several scan cells. In the example of Figure 7.6, the fault f may be detected by a test session started using seed a and a test session started by seed b . In the case of seed a , the fault is detected in scan cell 19 and in the case of seed b the fault is detected in cells 19 and 20. Only one of these combinations is required.

To ensure that the path that detects the fault is completely sensitized, it is sufficient to activate all of the scan cells in the input cone together with the scan cell observing the fault effect. For example, to detect the fault in cell 19 of Figure 7.6, it is sufficient to activate the scan cells $\{4, 5, 6, 7, 8, 9, 10, 19\}$. Since in practice the clocks cannot be activated individually but only per scan chain, this is mapped to enabling the scan chains $\{sc_1, sc_2\}$.

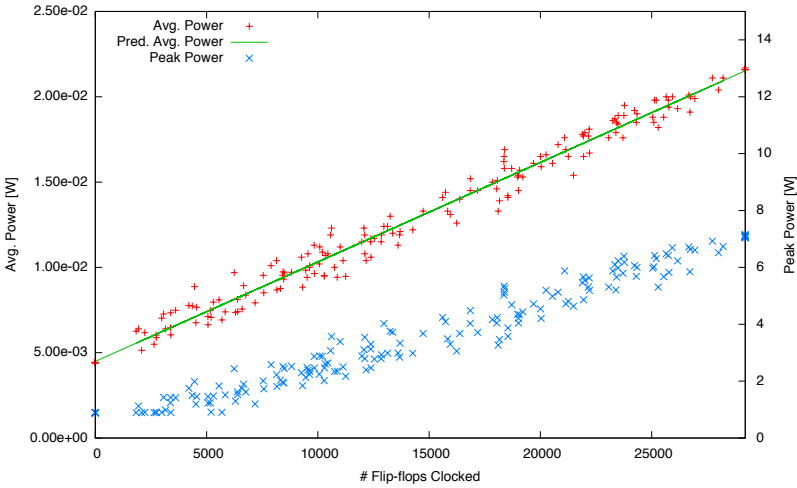


Fig. 7.4. Linearized Power Consumption vs. Simulation Results for Circuit p418k

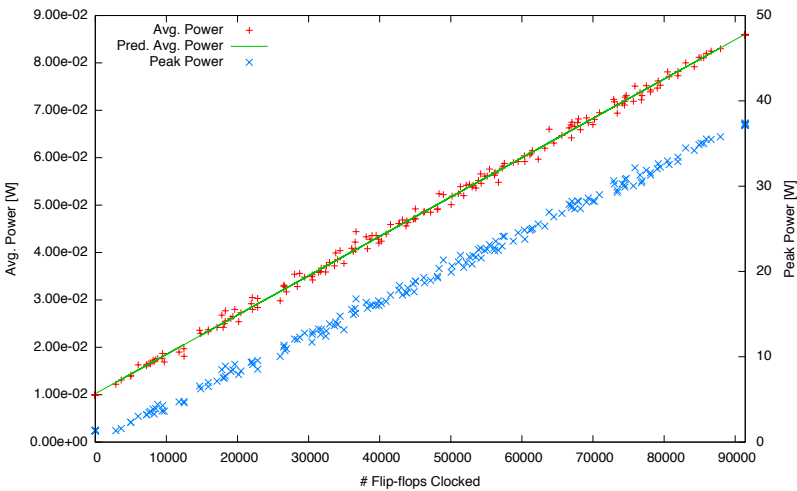


Fig. 7.5. Linearized Power Consumption vs. Simulation Results for Circuit p951k

7.4 The BIST Planning Problem

The goal of determining an optimized test plan is to construct a modified test for a given set of faults with minimized power dissipation. For every seed of the LFSR,

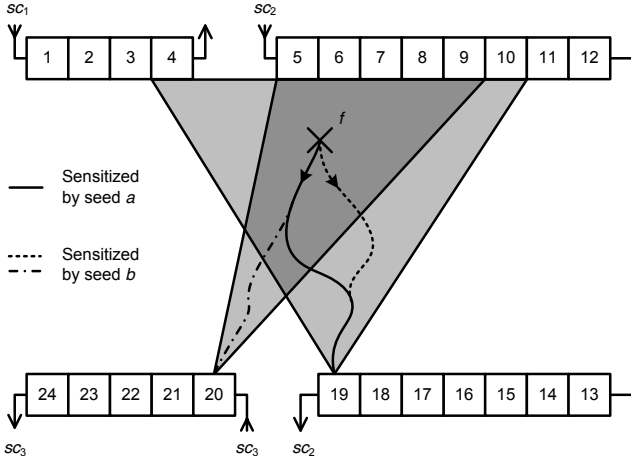


Fig. 7.6. Example of Detecting a Fault f and the Associated Input Cone

a set of scan chains is determined, which can be switched off without affecting the fault coverage.

A test session t is defined as a tuple (s, SC_t) consisting of seeds for the pattern generator and a set of activated scan chains $SC_t \subset SC$ called configuration. A seed corresponds to an associated test set of fixed size N . A fault set F_t can be determined for every test session t which contains all faults that can be detected by this block. We are looking for a set T of test sessions called the test plan, which detects the complete set of faults F with minimal power dissipation.

A fault f can be detected by different configurations, and hence there may be several sessions for one seed. They only differ in their configuration and may cover different sets of faults. Test sessions with the same seed may be merged $t_s = (s, \bigcup_{t \in T_s} SC_t)$ where T_s is a set of test sessions associated with a seed.

The session (s, SC_t) is called minimal regarding a fault f , if f would not be detected by the test session if any scan chain is removed from SC_t .

The cost of a covering is an estimation of its energy dissipation, which is determined by summing up for every seed the number of flip-flops in active chains as defined in section 7.2 (\rightarrow 'Estimation of Power Consumption' starting on page 63). Let S_T be the set of seeds from T . Then for every seed $s \in S_T$, the cost of a set of blocks can now be estimated as

$$\text{cost}(T) = \sum_{s \in S_T} \left| \bigcup_{(s, SC_t) \in T_s} SC_t \right|.$$

For industrial sized circuits, the determination of a global optimum is not feasible, as solving the set covering problem and the required fault simulations are computationally expensive.

7.5 Scan Cell Clustering

In many power reduction techniques, the effectiveness of the method is influenced by the organization of the scan chain. For example, in the scan path segmentation presented above, the shifting of a segment may be avoided completely if there is no fault observed in the segment and the segment contains no care bit (i.e. bits that are not don't care) of the pattern. In the example in Figure 7.7, a path is sensitized by a test pattern and the response is captured in scan cell 11. If all of these flip-flops are in the same scan segment like in Figure 6.13, only that segment has to be activated to test all of the faults along the path. In fact, similar relations hold for many other, more advanced test generation and test planning techniques.

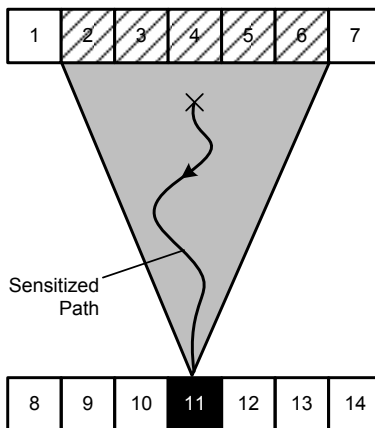


Fig. 7.7. Scan Cell Observing a Fault with Input Cone

The goal of scan clustering is to organize the scan cells of a circuit into k segments or parallel scan chains, where each segment contains at most t scan cells (Figure 7.8). The clustering tries to increase the likelihood that the scan cells with care-bits and the observing scan cells are in the same segment. Since it is undesirable to

synthesize DfT hardware based on a specific test set, the optimization is based on the circuit structure. In the DfT insertion process, the scan clustering is followed by layout-based scan cell ordering which tries to minimize the routing overhead of the scan design.

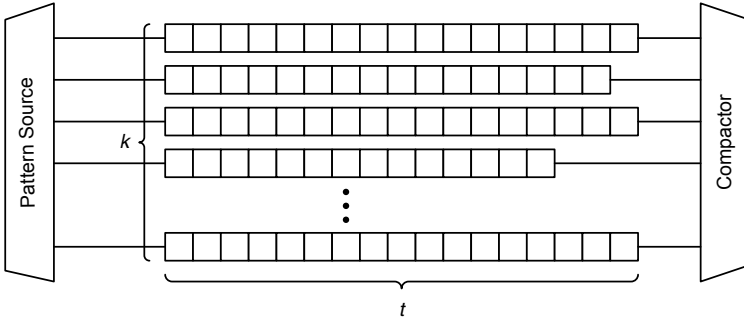


Fig. 7.8. Parameters k and t in Scan Chain Clustering

In the following two sections, the problem is first mapped to partitioning of a hypergraph and second an efficient heuristic to solve the partitioning problem is introduced.

7.5.1 Formal Problem Statement

The problem of clustering the scan cells is mapped to a graph partitioning problem. The technique described here uses a hypergraph representation of all the constraints. In this hypergraph $H = (V, E)$ the set of vertices $V = \{v_1, v_2, v_3, \dots, v_n\}$ represents the scan elements of the circuit. The set of hyperedges $E \subseteq \mathcal{P}(V)$ represents the set of cones derived from the circuit structure.

Using this representation, for every observing flip-flop o a hyperedge is derived which includes the vertex v_o corresponding to the flip-flop itself and all the vertices $v_{i1}, v_{i2}, \dots, v_{in}$ derived from its input cone. Since every flip-flop is a potential observer, the number of hyperedges $|E|$ equals the number of vertices $|V|$.

Figure 7.9 shows the hyperedge for the example of Figure 7.7. For the example the hyperedge for cell 11 is $\{2, 3, 4, 5, 6, 11\}$. It also contains hyperedges for a few more cones.

Now, the optimized clustering is the partitioning of the vertices of the hypergraph into k disjoint sets of up to t vertices such that the global edge cut (GEC) is minimized.

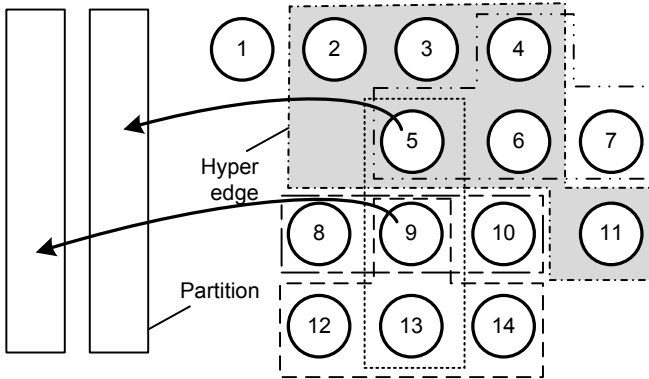


Fig. 7.9. Hypergraph and Hyperedge for Figure 7.7

$$GEC = \sum_{e \in E} (\#spanned_partitions(e) - 1)$$

The hypergraph partitioning problem is NP-complete. Up to now, substantial research effort has been dedicated to fast and reliable partitioning heuristics for graphs and hypergraphs aiming at different optimization criteria. Examples are spectral methods (Chan et al., 1993), min cut (Fiduccia and Mattheyses, 1982; Kernighan and Lin, 1970), coarsening (Karypis and Kumar, 1999) or variations of the Fiduccia Mattheyses algorithm (Fiduccia and Mattheyses, 1982). All of these general heuristics do not target large problem instances and are thus not applicable to the problem established here. For scan clustering a problem-specific heuristic such as the one used here can achieve favorable results with very low computation time (linear-time complexity) even for multi-million gate designs. This kind of clustering can improve the effectiveness of power reduction techniques by around 40% compared to regular scan insertion techniques.

7.5.2 Partitioning Algorithm

First a cost function is presented to evaluate partial solutions, the subsections deal with the three phases of the algorithm, and finally some complexity issues are discussed.

Evaluating configurations

A configuration Π of the partitioning process is a state, where s vertices were already assigned to partitions and $n - s$ vertices are left. A configuration is evaluated by a cost

function, which displays its quality and the difficulty to complete the configuration. This is achieved by assigning a label $L(e)$ to each hyperedge $e \in E$. The label of an edge is the number of partitions this edge is spanning in the current configuration. The cost function is defined as the sum of all these labels.

$$C(\Pi) = \sum_{e \in E} L(e)$$

Initial Configurations

If the hypergraph is not connected, each of its connected components can be processed independently in order to reduce complexity. All vertices in components smaller than t are simply assigned to a single arbitrary partition, as this can be done without cutting edges. Without loss of generality the hypergraph is considered as a connected component.

Figure 7.9 depicts some vertices, which induce a natural, initial partitioning of two vertices of the hypergraph as they are the centers of two different strongly connected areas of the hypergraph. If vertex 5 and vertex 11 are assigned to two different partitions, the number of cut hyperedges will still be minimal, and a partitioning algorithm to assign the remaining vertices can be guided.

The initial configuration tries to identify k vertices like those in figure 7.9 and assigns them to different, empty partitions. These preassigned vertices will then attract all vertices in their neighborhood to the same partition in the next steps.

The next vertices to be prepartitioned are determined by evaluating the connectivity of each vertex. The vertex with the most incident hyperedges is selected and put into the first partition. Afterwards, all incident hyperedges are marked and the search proceeds to find another vertex, with the most incident, yet unmarked, hyperedges. This repeats until a vertex has been assigned to all the k partitions.

Graph Partitioning

The main algorithm adds one vertex after the other to the configuration obtained so far. The cost function evaluates for each vertex, to which of the k partitions it should be added. After the best partition has been found, the vertex is assigned to this partition, and the labels and the cost function are updated accordingly.

Not only the choice of the initial partitioning is important, but also the order in which the vertices are processed. For this reason, a label $L(v)$ is defined for each vertex $v \in V$ that allows to guide the search.

$$L(v) = \max_{\{e|v \in e\}} L(e)$$

The label of a vertex is the maximum label of all the edges incident to this vertex. The next vertex to choose for partitioning is one with maximum label and maximum incidence.

As the label cannot exceed k , one can find the next critical vertex fast by storing the vertices in buckets according to their label. The evaluation of the incidence can be done, when the graph is built. Again the vertices are put into buckets, and the next vertex is found in constant time.

Post-partitioning

The process described above touches each vertex only once, and some vertices may be partitioned based on incomplete information. In a post-processing step some of the vertices are removed again considering their cost impact. These vertices are re-evaluated and the partitioning decision may be revised.

The contribution of a vertex to the overall cost can be estimated by

$$C(v) = \sum_{\{e|v \in e\}} L(e)$$

The vertices are ranked by their cost $C(v)$, and according to this ranking, the first 10% of vertices are removed. In order to gain more freedom in repartitioning vertices, also the 10% vertices ranked lowest are removed. These vertices are partitioned as before a second time. But here, the order in which the vertices are selected is based on the incidence of a vertex, starting with the vertices with highest incidence.

The post-partitioning step can be applied to any configuration, which may be the original scan design or which may be constructed by the steps described above. In cases where the original scan design has already been optimized and should not be changed completely, this postprocessing alone may provide further gain.

Complexity Considerations

During each of the three steps — initial configuration, graph partitioning, post processing — each of the n vertices is only touched once. However, the number of operations during a step depends on the cardinality of a hyperedge or on the number of hyperedges the vertex is a member of.

The cardinality of a hyperedge corresponds to the size of an input cone, and the number of hyperedges incident to one vertex corresponds to the size of its output cone. In general, the size of an input or output cone is independent of the circuit

size n and is limited by a constant, say c_1 . However, often a small set of signals is connected (fans out) to a very large number of flip-flops. Examples are buffer trees or asynchronous signals. Again, the number of signals of this kind is limited in a real circuit for timing and regularity reasons to say c_2 . Together, this yields a complexity estimation of $c_1 \cdot n + c_2 \cdot n$, which indicates a linear complexity with circuit dependent constants c_1, c_2 .

BIST Planning for Scan Clock Gating

As defined in the previous chapter, BIST planning is the process of assigning configurations of the scan clock gating for each session of a test such that a certain set of faults is detected. For example, in a BIST based on the STUMPS design, the BIST may consist of several sessions where each session is started by a seed of the linear feedback shift register (LFSR). For every seed, test planning computes a configuration of the scan chains such that fault coverage is not impaired.

This chapter introduces the proposed search algorithm for BIST planning. In the first section of the chapter, an overview of the related work is presented. Section 8.2 presents an algorithm that is suitable to solve the BIST planning problem for industrial circuit. To reduce the computational complexity, section 8.3 presents a divide-and-conquer approach and heuristics which are applied in each iteration of the divide-and-conquer method.

8.1 Related Work

Test planning can be conducted at several steps in the creation of a test program. The existing techniques in literature focus on test planning for two steps: The system or chip level scheduling of core tests and the test generation step. The method presented here belongs to the test generation approaches. The two steps are orthogonal and hence techniques for core test scheduling can be combined with test generation methods. Substantial improvements in power consumption can be achieved on both levels. While the method presented here is targeting the test generation step for BIST, a short overview of the scheduling approaches will be given as well.

When first documenting the issue of power dissipation in (Zorian, 1993), Zorian proposed to schedule the parallel execution of tests according to the power envelope. Scheduling approaches have been developed based on resource graphs (Chou et al., 1997) and bin-packing (Xia et al., 2003). Making core tests preemptable and resumable, significantly reduces the computational complexity of the problem (Iyengar

and Chakrabarty, 2001). More accurate modeling of heating effects (Rosinger et al., 2005) and peak power (Samii et al., 2006) have been shown to be important, since test can exhibit vastly different behavior.

Often, the problem of test scheduling is also subject to constraints in the test access mechanisms such as the number of pins connected to the scan chains of a core (Sehgal et al., 2008). This requires the test scheduling to add additional constraints (Larsson and Peng, 2001; Pouget et al., 2005). Often, the scheduling and design optimization steps are done together to find optimal solutions to the scheduling problem (Huang et al., 2002). A large focus of more recent publications is on this design optimization using test scheduling predictions (Larsson and Peng, 2006).

The method presented here is a test planning method for the step of test generation for BIST. Hence, it can be employed together with any of the above methods and these methods will be more effective if core tests have been generated for power efficiency with methods such as the one presented here.

Test planning in the test generation step focuses mostly on deterministic tests from automated test pattern generation (ATPG). In all of these cases, the target architecture is a scan-design with multiple parallel scan-chains that can be clock gated individually. The initial work done in (Sankaralingam and Toubia, 2002) has shown that test planning for deterministic tests is challenging. Additional constraints for the automated test generation have to be added and the effectiveness of the approach depends largely on the ability to cluster the flip-flops. To obtain acceptable results, the test generation and clustering steps are overlapped. This makes the hardware design dependent on the test set, which is often undesirable. In (Sankaralingam and Toubia, 2002), only results for the smallest circuit evaluated here have been presented.

In (Czysz et al., 2009), the author present a method for deterministic test generation that can be applied to industrial circuits and does not require special consideration of the scan chain clustering besides the common methods described in section 7.5 (\rightarrow ‘Scan Cell Clustering’ starting on page 69). It targets a clock gated DfT architecture with parallel scan chains. The method takes advantage of the fact that in very large circuits, power constraints can be imposed during the dynamic test compaction step. The dynamic test compaction targets several faults with a single scan pattern. Whenever a new target fault is added the power constraint is checked before admitting the fault, but otherwise scan chains are enabled as necessary. This approach adds a very simple constrained algorithm to the test generation. Similarly, simple constraints have been recently added to commercial test generation tools (Chakravadhanula et al., 2009), with the concept of the test plan used to guide the test generation (Uzzaman et al., 2009).

General methods for power-aware ATPG are orthogonal to test planning. Such methods focus on rejecting unsafe patterns that exceed test power limits, constraining / guiding the ATPG algorithm (mostly to reduce capture power) and filling unspecified bits (mostly for shift power reduction). Safety checking of patterns estimates the power consumptions of a test and compares it to the specification in which the power supply network and transistors are supposed to achieve the desired timing delay properties. The result of the check is either used to reject an unsafe pattern or continue the search for a pattern for a target fault that is safe. The most common techniques are those outlined in section 7.2 (→ ‘Estimation of Power Consumption’ starting on page 63).

Guidance of the ATPG is an important approach to reduce test generation time, by guiding the search for a pattern along the most observable/controllable paths. These algorithmic techniques can also be used to guide the search along paths with lower transition probability and hence capture power (Wang and Gupta, 1994). However, the guidance is just probabilistic and a resulting pattern is still able to exceed the desirable capture power. For stricter power limitation, the gate-level model used for ATPG can be augmented by logic gates which intentionally limit the search space (Ravi et al., 2007). Alternatively, by introducing an entirely new type of constraints (and hence conflicts) into the pattern generation algorithm, both optimization targets and hard limits can be covered with the same algorithm (Wen et al., 2006). Besides reducing power to within the circuit’s specification, it can occasionally be required to increase capture power consumption on critical paths closer to the functional specification in order to provide the best coverage for small delay defects (Czutro et al., 2012).

During ATPG, a large number of bits are left unspecified (don’t care bits), which can be selected to arbitrary values without affecting coverage of the target faults. Usually, such bits are filled with random values in order to detect as many non-target faults as possible. However, the random fill will lead to very high shift power like during LBIST. In order to reduce shift power during shift-in, don’t-care bits can instead be filled with fixed values (Nicolici et al., 2000), with repeating the previous value providing the best reduction (Sankaralingam and Touba, 2002). Besides the shift-in power, more advanced fill methods also consider the power consumption during shift-out (Song et al., 2008). The impact of the fill on capture power can be taken into account either through signal probabilities (Remersaro et al., 2006) or by using ATPG-like justification (Wen et al., 2007a). To avoid the excessive performance overhead of the justification, its search space can be significantly constrained (Miyase et al., 2013).

However, filling the unspecified bits as outlined above may lead to a larger test set with many more patterns compared to random fill of the unspecified bits. Because of this, a more effective approach can be to first fill all of the unspecified bits with random values in order to obtain a small test set and then reverting specified bits to unspecified bits that are not needed after all. This technique is called bits-stripping (Kochte et al., 2008; Miyase and Kajihara, 2004) and the finally unspecified bits can then be filled according to the rules above (Kochte et al., 2011; Miyase and Kajihara, 2004) providing a favorable trade-off of test time and power consumption. If clock gating is present in the design under test, the better results can be achieved by specifically targeting the clock gating signals (Miyase et al., 2010).

Since the method presented here is targeting pseudo-random BIST, it cannot influence which faults are detected by which patterns. While a simple algorithm can obtain acceptable results (Zoellin et al., 2006), considerable improvements are obtained by using the more advanced optimization methods outlined in this work.

8.2 Optimization Algorithm

Figure 8.1 shows the general flow of the algorithm. In the first step, fault simulation is used to determine for every test session $t = (s, SC_t)$ the set of faults F_t detected while executing the session. A fault isolation table is generated to capture the outputs at which faults are observed.

If F_t contains faults that cannot be detected by any other test session, then t is part of the optimal solution and is called essential session. We get $T_0 := \{t \mid t \text{ essential}\}$ as an intermediate result and only have to cover the faults in $F_0 = F \setminus \bigcup_{t \in T_0} F_t$.

For the remaining faults F_0 , the complexity is further reduced by a „divide and conquer“ approach. For this purpose, the set of faults is partitioned into three classes:

1. Hard faults are faults that can only be detected by one seed out of S . The corresponding seeds are called essential.
2. Difficult faults can only be detected by a number of seeds, which is below a user defined constant λ .
3. All other faults are easy to detect.

Those three classes are tackled with different methods and heuristics as outlined in the next section. To describe these methods, the circuit in Figure 8.2 will serve as an example. In the example, we assume $S = \{s_1, s_2, s_3\}$, $F = \{f_1, f_2, f_3, f_4, f_5, f_6\}$.

The heuristics outlined below create constraints for the associated set covering problem. The set covering is then solved using the jSAT software package with a

special constraint and bound to reflect the power according to the estimation method outlined in section 7.2 (→ ‘Estimation of Power Consumption’ starting on page 63).

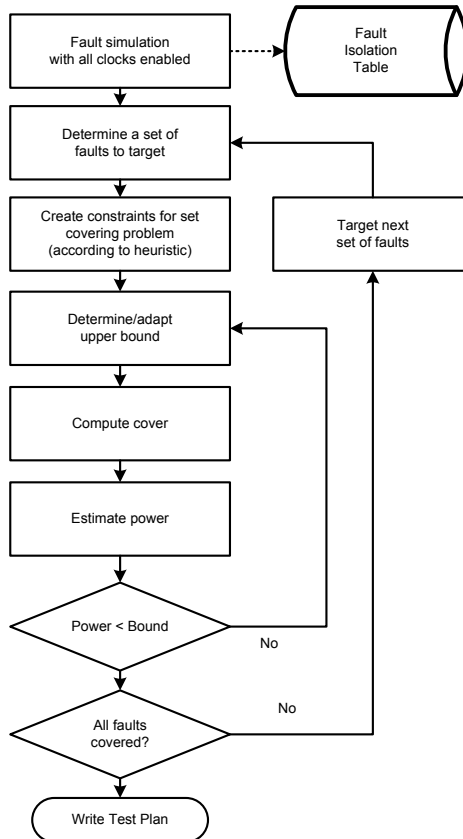


Fig. 8.1. Flow Chart of Test Planning Algorithm

8.3 Fault Classification and Heuristics

This section introduces the divide-and-conquer approach, which starts by analyzing fault detectability and then divides faults into different classes. A set covering problem is then constructed for each class iteratively. The three classes are "Hard Faults", "Difficult Faults" and "Other Faults".

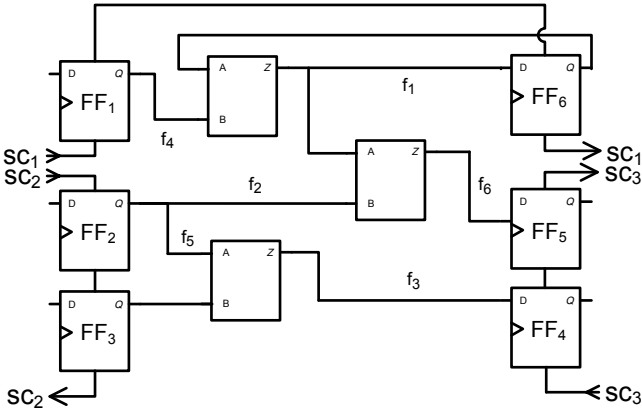


Fig. 8.2. Circuit for Test Planning Example

8.3.1 Hard Faults

For hard faults, algorithm 1 computes the set of test sessions with the lowest cost (i.e. power consumption).

Algorithm 1 Hard Faults

```

for all  $f \in F_0$  do
     $s_f \leftarrow$  seed detecting  $f$ 
     $c \leftarrow \{SC_t \mid f \text{ is detected by session } (s_f, SC_t)\}$ 
     $T \leftarrow T \cup \{(s_f, SC_t) \mid SC_t \in c\}$ 
end for
for all  $s \in S$  do
    for all  $t = (s, SC) \in T$  do
         $F_t \leftarrow$  faults detected by  $t$ 
        Add constraint  $SC_t \rightarrow F_t$  to set covering problem
    end for
    minimize  $T_1 \subset T$  for  $cost(T_1)$  subject to  $F_0 = \bigcup_{t \in T_1} F_t$ 
end for

```

For every hard fault $f \in F_0$ with corresponding seed s_f , the function $c(f, s_f) = \{SC_t \mid f \text{ is detected by session } (s_f, SC_t)\}$ defines the sets of scan chains and consequently the set of minimal sessions $T_f = \{(s_f, SC_t) \mid SC_t \in c(f, s_f)\}$.

Since a single output flip-flop is sufficient for detecting a fault, each set in $c(f, s_f)$ contains just one output scan chain and a minimal number of input chains. For all the patterns generated by a seed s_f , the output information is found in the fault isolation

table. Afterwards, for every flip-flop the flip-flops from the corresponding input cone are collected and the set of required scan chains is determined.

Instead of generating a fault isolation table, it is also possible to use the „Support Region“ (Hamzaoglu and Patel, 1998) of a fault f . It contains all flip-flops of the output cone of f , as well as all flip-flops that are part of the input cones of these output flip-flops. This approximation is independent of the generated test set and does not require the fault isolation table. However, it significantly overestimates the number of scan chains in the configurations created by the mapping function c .

For every test session $t \in \cup_{f \in \text{hard}} T_f$, the set F_t of all the faults, which are detected by t is determined. Because the number of faults and configurations is small, a branch-and-bound method (such as (Coudert, 1996), or based on an optimization package like the jSAT program used here) can be used to find a subset $T_1 \subset \cup_{f \in \text{hard}} T_f$, such that $\cup_{t \in T_1} F_t$ covers all the hard faults from F_0 and $\text{cost}(T_0 \cup T_1)$ is minimal. The remaining set of faults is $F_1 = F_0 \setminus \cup_{t \in T_1} F_t$.

For an example see Figure 8.3. For the example, let f_4 be a hard fault detected by seed s_1 either in FF_5 or in FF_6 . $c(f_4, s_1) = \{\{sc_1\}, \{sc_1, sc_2, sc_3\}\}$. $T_{f_4} = \{(s_1, \{sc_1\}), (s_1, \{sc_1, sc_2, sc_3\})\}$. Solution of the covering problem $T_1 = \{(s_1, \{sc_1\})\}$ has $\text{cost}(T_1 \cup T_0) = 1$. f_4 is detected in fault simulation of $T_1 \cup T_0$, which now detects these faults with $\text{cost}(T_1 \cup T_0) = 1$.

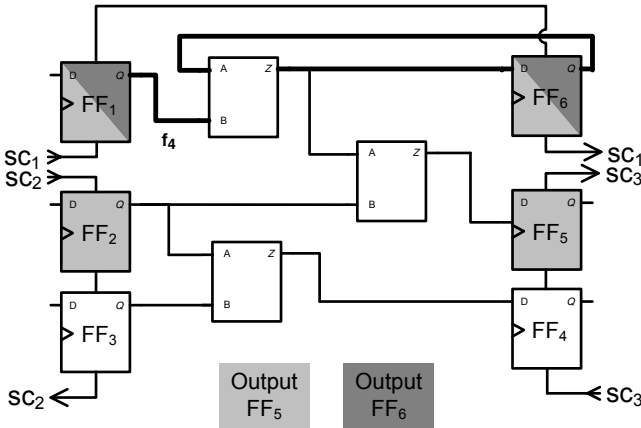


Fig. 8.3. Test Planning Example Considering a Hard Fault

8.3.2 Difficult Faults

Algorithm 2 determines the set of test sessions with the lowest cost (i.e. power consumption) for faults that are detected by more than one seed.

Algorithm 2 Difficult Faults

```

 $F_\lambda \leftarrow \{f \mid |S_f| < \lambda\}$ 
for all  $f \in F_\lambda$  do
   $s_f \leftarrow$  seed detecting  $f$ 
   $c_{min} \leftarrow \min(\{SC_t \mid f \text{ is detected by session } (s_f, SC_t)\})$ 
   $T \leftarrow T \cup \{(s_f, SC_t) \mid SC_t \in c_{min}\}$ 
end for
for all  $t = (s, SC) \in T$  do
   $F_t \leftarrow$  faults detected by  $t$ 
  Add constraint  $SC_t \rightarrow F_t$  to set covering problem
end for
minimize  $T_2 \subset T$  for  $cost(T_0 \cup T_1 \cup T_2)$  subject to  $F_\lambda = \bigcup_{t \in T_2} F_t$ 

```

The fault class dealt with in this step contains difficult faults F_λ that can be detected by a number of seeds $|S_f|$, where $|S_f|$ is below a fixed limit λ . For these faults, the constraint set does not decompose into subsets for each seed since faults can be detected by more than one seed. This significantly increases the time searching for a solution to the set covering. To restrict the complexity, only one detecting configuration is considered per seed. Only the configuration $c_{min}(f, s_f) \in c(f, s_f)$ is selected that has minimal cost.

In this case, $T_f = \{(s_f, c_{min}(f, s_f)) \mid s_f \text{ seed for } f\}$ is the corresponding set of test sessions for fault f and a subset $T_2 \subset \bigcup_{f \in F_\lambda} T_f$ is generated so that $\bigcup_{t \in T_2} F_t$ covers all difficult faults F_λ from F_1 and $cost(T_0 \cup T_1 \cup T_2)$ is minimal. The remaining set of faults $F_2 = F_1 \setminus \bigcup_{t \in T_2} F_t$ only contains easy to detect faults and will be very small or even empty, since they may be already detected in the previous steps.

For the example in Figure 8.4, let λ be 2. $F_\lambda = \{f_2, f_3\}$. $S_{f_2} = \{s_2, s_3\}$. $S_{f_3} = \{s_1, s_3\}$. $c_{wc}(f_2) = \{sc_1, sc_2, sc_3\}$, $T_{f_2} = \{(s_2, \{sc_1, sc_2, sc_3\}), (s_3, \{sc_1, sc_2, sc_3\})\}$. $c_{wc}(f_3) = \{sc_2, sc_3\}$, $T_{f_3} = \{(s_1, \{sc_2, sc_3\}), (s_3, \{sc_2, sc_3\})\}$. Optimal solution $T_2 = \{(s_3, \{sc_1, sc_2, sc_3\}), (s_3, \{sc_2, sc_3\})\}$ has $cost(T_2 \cup T_1 \cup T_0) = 4$. f_5, f_6 are detected in fault simulation of $T_2 \cup T_1 \cup T_0$, which now detects all faults with $cost(T_2 \cup T_1 \cup T_0) = 4$.

Even after this optimization, the space for the branch-and-bound optimization is very large. However, a very good intermediate solution is found after relatively short runtime. Hence, the optimization with the branch-and-bound method is aborted as

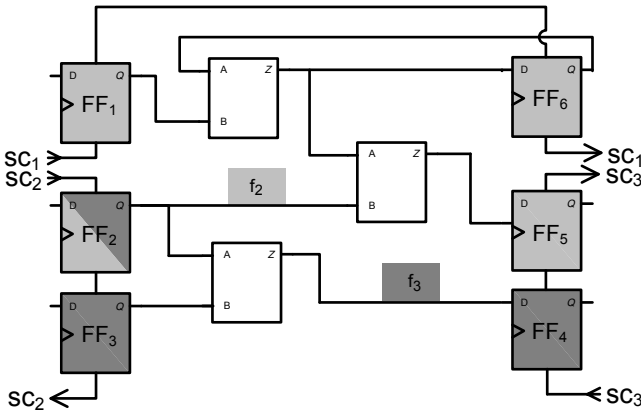


Fig. 8.4. Test Planning Example Considering A Difficult Fault

soon as the time for finding the next (improved) intermediate solution exceeds a certain time limit. This affects the result only marginally.

8.3.3 Other Faults

To cover all the remaining faults step 2 is re-executed with $\lambda = \infty$, so that blocks for all remaining faults are generated. However, some faults in this class can be detected by a very large number of seeds. Some faults may even be detected by every seed in S . This severely degrades the quality of the early intermediate solutions found before the abort of the branch-and-bound algorithm.

A simple heuristic is added to alleviate this problem that takes advantage of the fact that these faults can be detected by so many seeds. During constraint generation, only a subset of all possible test sessions is allowed for faults that can be detected by too many seeds. For each possible test session $t_f \in T_f$ the cost $cost(T_2 \cup T_1 \cup T_0 \cup \{t_f\})$ is evaluated and this cost is used to rank all possible test sessions for the fault. Only the test sessions that are ranked in the top 10 are used for the constraint generation.

This reduces the solution space by several orders of magnitude and restores the quality of the intermediate solutions found until an abort. Especially for the larger circuits evaluated in chapter 10 (\rightarrow ‘Evaluation’ starting on page 93) this heuristic results in a very efficient set cover. For these circuits, the cost for detecting the last set of faults is just 2-5% of the final cost.

Extension to Transition Faults and Frequency Scaling

The preceding chapter has presented the general core algorithm for BIST planning. This chapter presents applications and extensions of the core algorithm that are intended for dealing with industrial designs.

The first section in this chapter shows how the sequential behavior when testing for delay faults can be modeled in the circuit graph such that BIST test planning can be applied. Often, industrial designs support a variety of clock sequences that must be accounted for. The method presented here introduces a general clock sequence modeling and applies to a wide range of algorithms that use the circuit graph and hence it is also used for the fault simulation.

Another extension required when testing industrial circuits is to control the power envelope during the test and consider test time improvements. In a manufacturing test environment, the goal is often not the lowest test power and energy consumption but to remain within the limits of the power supply and cooling capacity. The second section shows how the presented approach can be combined with dynamic frequency scaling to optimize the test time within the given power consumption envelope.

Numerous parameters factor into the hardware architecture and the BIST application that influence the effectiveness of the presented method. Parameters are for example the number of scan chains and their organization, or the number of patterns and test sessions. The third section of this chapter discusses these tradeoffs.

9.1 Transition Faults

Tests that target the transition fault model introduced in section 3.3.1 (→ ‘Gate Delay Faults’ starting on page 23) are a standard technique to validate the timing behavior of a circuit under test (CUT). If the transition test is executed at-speed, it can detect small structural delay defects in the order of the timing slack.

In scan-based transition test, mainly three techniques are used to justify and launch transitions: launch-on-capture (LOC) (Savir and Patil, 1994), launch-on-shift

(LOS) (Savir, 1992) and scan-hold (Dasgupta et al., 1981). LOC uses two system clock cycles to launch and capture a transition. LOS uses the last clock pulse of the scan shift to launch the transition, directly followed by a system clock for capture. Scan-hold allows to launch arbitrary transitions. Since scan-hold requires special scan cells with additional overhead, it is not considered here.

While LOS often provides higher fault coverage than LOC, there are also faults untestable by LOS that are testable by LOC. Hence, it is often beneficial to apply both types of tests to a circuit (Ahmed and Tehranipoor, 2005; Madge et al., 2003; Wang et al., 2004). Furthermore, fault coverage of LOC can be increased by using additional functional clock cycles (Abraham et al., 2006; Zhang et al., 2006). Finally, both schemes may be combined to form Launch-on-Capture-Shift (LOCS) and Launch-on-Shift-Capture (LOSC) (Park and McCluskey, 2008).

This section shows how to deal with arbitrary clock sequences in test planning. First, a procedure is presented to derive graph data structures that reflect a specified clock sequence. While the technique remains true to the idea of 'unrolling' a circuit (McCluskey, 1958), we formalize the implications of shift and capture cycles in arbitrary clock sequences. The graph types employed in the test plan generation are circuit graphs for fault simulation (Antreich and Schulz, 1987) and S-graphs of the flip-flops (Kunzmann and Wunderlich, 1990; Wunderlich, 1989). However, the presented technique is easily adapted to any graph representation of the circuit (e.g. for scan chain synthesis, ATPG or diagnosis) known for balanced circuits (Gupta and Breuer, 1989; Narayanan et al., 1992).

9.1.1 Sequential Graphs for Shift and Capture Clocks

Below, we will formalize the transformation of a sequential circuit with scan to a combinational circuit graph. The generated graph corresponds to the combinational function if the sequential circuit were subject to a given clock sequence. For both capture and shift clock, we show how to compute a set of edges that connects two isomorphic copies of the circuit graph. The final graph is created by concatenating several isomorphic copies of the original graph.

For many design automation problems, circuits are modeled as directed graphs. In fault simulation and ATPG, the circuit is represented by a directed graph, where the vertices correspond to primary inputs, primary outputs and the outputs of the gates. In the synthesis of partial scan paths, S-graphs are used that are directed graphs where vertices correspond to the primary inputs, primary outputs and the flip-flops of the circuit.

Let $G(V, E)$ be a graph representation of the combinational function of the circuit, where V contains at least the primary and pseudo-primary inputs and outputs. The set of edges $E \subset V \times V$ is problem specific and may represent the data flow or a similar relationship between vertices. V includes the set of inputs $I \subset V$ and outputs $O \subset V$. For sake of simplicity we consider only full-scan circuits, but the presented formalism is easily extended to partial scan circuits.

The information about the structure of the scan chains and the input/output relation of scan flip-flops is not included in G . Scan flip-flops $FF \subset O \times I$ are edges between pseudo-primary outputs and pseudo-primary inputs. The scan chain organization $SC \subset \mathcal{P}(I)$ is a partitioning of the flip-flops in the circuit. For each scan chain $SC_i \subset I$ in SC the scan chain order $sc_i \in SC_i^*$ is given as a unique sequence $sc_i = (ppi_1, ppi_2, \dots)$. Figure 9.1 shows a circuit graph for a small example.

Figure 9.1 depict the circuit graph for a simple circuit. The sets of inputs and outputs to this circuit are $I = \{pi_1, ppi_1, ppi_2, ppi_3\}$ and $O = \{ppo_1, ppo_2, ppo_3, po_1\}$. The edges that represent the flip-flops and the scan in si and scan out so are not depicted here. They are $FF = \{(ppo_1, ppi_1), (ppo_2, ppi_2), (ppo_3, ppi_3)\}$. The single scan chain of the circuit is $sc_1 = (ppi_1, ppi_2, ppi_3)$.

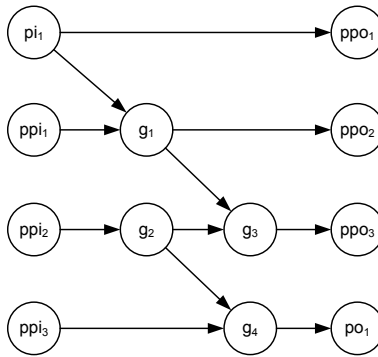


Fig. 9.1. A Circuit Graph

Let $G_t(V_t, E_t)$ be an isomorphic copy of G and let I_t, O_t be the sets of inputs and outputs in G_t . We say two vertices $v_{t_1} \in V_{t_1}$ and $v_{t_2} \in V_{t_2}$ with $t_1 \neq t_2$ are structurally equivalent (i.e. map to the same circuit node) if they are derived from the same node in G .

The circuit state and output after a clock can now be described as the concatenation of two isomorphic copies G_t and G_{t+1} of G .

Graph Concatenation for Capture Clock

A capture clock causes the data at the pseudo-primary outputs of G_t to appear at the pseudo-primary inputs of G_{t+1} . The data flow for this case is described by the edges represented in the set of flip-flops FF .

Hence two graphs G_t and G_{t+1} may be concatenated using the following set of edges $Cap_{t,t+1}$:

$$Cap_{t,t+1} \subset O_t \times I_{t+1}$$

$$Cap_{t,t+1} = \{ (o_t, i_{t+1}) \in O_t \times I_{t+1} \mid \exists (o_f, i_f) \in FF : o_t, o_f \text{ and } i_{t+1}, i_f \text{ are struct. equiv.} \}$$

Figure 9.2 shows the set of edges $Cap_{t,t+1}$ for the example circuit above.

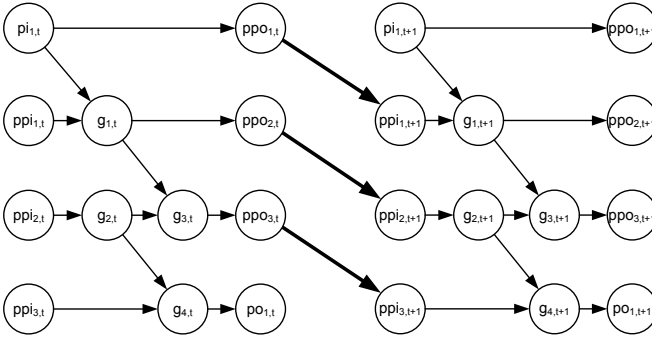


Fig. 9.2. The Graph Concatenation for a Capture Clock

Graph Concatenation for Shift Clocks

If a shift clock is applied instead of a capture clock, the inputs of one circuit graph are mapped to inputs of the other circuit graph. The edges $Shf_{t,t+1}$ concatenating the graphs G_t and G_{t+1} are derived from the scan chains of the circuit:

$$Shf_{t,t+1} \subset I_t \times I_{t+1}$$

$$Shf_{t,t+1} = \{ (i_1, i_2) \in I_t \times I_{t+1} \mid i_1 \in ff_k \wedge i_2 \in ff_{k+1} \}$$

where ff_k, ff_{k+1} are successive flip-flops in a scan chain $SC_j \in SC$ of the scan chain organization of the circuit.

Figure 9.3 shows $Shf_{t,t+1}$ for the example. Depending on the purpose of the graph, scan-in and scan-out nodes can be added for the shift clock cycle.

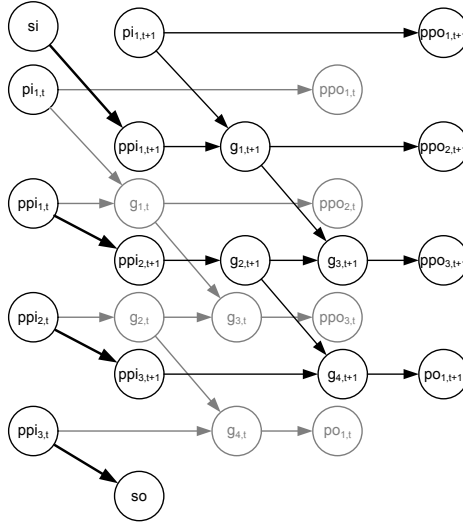


Fig. 9.3. The Graph Concatenation for a Shift Clock

9.1.2 Graph Generation from a Clock Sequence

A clock sequence is described by any sequence $c \in C^*$ over the alphabet $C = \{CAP, SHIFT\}$.

For the final graph $G_c = (V_c, E_c)$, $|c|$ isomorphic copies of the graph G are created one for each clock in the sequence. The vertices of the final graph are then:

$$V_c = \bigcup_{t=1}^{|c|} V_t$$

The edges of the final graph are the edges of each copy plus the edges for the concatenation of the graphs:

$$E_c = \bigcup_{t=1}^{|c|} \begin{cases} E_t \cup Cap_{t-1,t} & \text{if } \sigma = CAP \\ E_t \cup Shf_{t-1,t} & \text{if } \sigma = SHIFT \end{cases}$$

The presented formalization can be easily implemented using almost any graph representation. For example, the test planning algorithm presented in the next chapter uses two graph representations. Parallel pattern single fault propagation (PPSFP) is used for the transition fault simulation and the circuit is represented by a regular circuit graph of the gates. An S-graph of the sequential gates is used to determine the predecessors of a flip-flop that detects a fault. Both graphs can share the same algorithm to deal with arbitrary clock sequences. The set of software interfaces required is rather small and is easily factored into the implementation.

9.2 Dynamic Frequency Scaling

In a manufacturing test environment, very often the goal is not to minimize the power consumption, but to provide the shortest test time within a given power envelope. Using the method presented here to its full extent will result in power consumption that is significantly lower than the specified envelope for a significant number of test sessions.

Since the shift portion of the scan-based logic BIST is the largest contributor to the overall test time, it is highly desirable to select the fastest scan frequency with which a BIST session of the generated test plan remains within the given envelope. Scaling the scan frequency changes the dynamic power consumed during shifting. Often the scan clock during BIST is just divided from the PLL frequency, hence there is still a certain amount of control logic that remains at speed. In addition, frequency scaling by itself does not change the static power consumption. In contrast to functional operation, frequency scaling during BIST cannot be combined with voltage scaling to improve static power consumption, since the launch and capture cycles are still operated at their original frequency. (Please note that often timing of the shift path is only closed to a frequency f_{scan} that is significantly lower than the functional frequency and will limit the maximum scaling. However, actual shift frequencies are often significantly lower than the timed frequency due to the power supply noise concerns, leaving ample headroom for the frequency scaling proposed here.)

Let P_{min} be the average power consumption with all scan clocks stopped and P_{max} the avg. power consumption with scan frequency at the highest possible value that satisfies f_{scan} . Then the difference P_{range} between these two operating points is nearly all dynamic power consumption and is the control range for the frequency scaling.

Let's assume that frequency scaling is possible through division of the PLL frequency with an integer d , hence the divider at the timing limit of the scan path is

$$d_{scan} = \left\lfloor \frac{f_{PLL}}{f_{scan}} \right\rfloor.$$

Since P_{range} is just dynamic power that scales linearly with the frequency, the scaled power consumption is as follows:

$$P_{scaled} = \frac{d_{scan}}{d} P_{range}$$

Similarly, given a test session that has estimated power P_{est} at max. shift frequency (e.g. using the power estimation approach outlined in section 7.2), the power consumption $P_{session}$ of the test session with frequency scaling is

$$P_{session} = P_{min} + \frac{d_{scan}}{d} (P_{est} - P_{min}).$$

For a limited power envelope P_{limit} , only a subset of frequencies are permitted:

$$P_{limit} \geq P_{min} + \frac{d_{scan}}{d} (P_{est} - P_{min})$$

The smallest integer divider within P_{limit} that does not exceed the timed scan frequency f_{scan} is:

$$d_{min} = \max \left(\left\lfloor d_{scan} \frac{P_{est} - P_{min}}{P_{limit} - P_{min}} \right\rfloor, d_{scan} \right)$$

9.3 Tradeoffs

The method presented here allows for a number of tradeoffs during the optimization process, that can either be selected a priori or during the optimization process. The variables are as follows:

- **The granularity of the test session** is the number of patterns applied during a test session. Since a configuration of the scan chains must be constant during an entire test session, having more sessions with fewer patterns (that in consequence also detect fewer essential/difficult faults) allows to disable chains more often.
- **The overall number of patterns** applied during the BIST is the sum of all of the patterns applied during all of the test sessions. If more patterns are applied overall, more faults can be detected more than once or just a few times. This reduces the number of essential and difficult faults, significantly increasing the degree of freedom during the optimization step. The additional test time required for applying the patterns can be offset to a certain degree by the frequency scaling described in the previous section.

- **The clustering of the flip-flops** into scan-chains has significant impact on the test planning. Ideally, a fault's support is contained entirely within a single scan chain such that only that scan chain must be enabled to detect the fault. However, standard methods for DfT insertion often do not take the circuit topology into account and are focused just on reducing the layout overhead of the scan wires (if at all optimizing for physical design).
- **The number of scan chains** is a factor for both test time as well as effectiveness of the method presented here. Having more scan chains that are shorter reduces test time, since the longest chain determines the number of shift cycles per pattern. Since an entire chain must be enabled even if only a few flip-flops are part of the fault support, short chains also reduce the number of don't-care flip-flops that are enabled unnecessarily. Clock-gating introduces overhead to the design and is most effective at the level of the clock buffers, which drive around 32 flip-flops or at even coarser granularity. Hence there are limits to shortening scan chain length for clock gating, shorter scan chains can also result in additional wiring overhead. Overall, this parameter is highly design-specific and is not considered in significant detail beyond accepting the parameters chosen by the design supplier.
- **PRPG weighting** plays a role in achieving high fault coverage pseudo-random BIST. Test patterns with weight other than 1:1 are useful to detect random resistant faults that often exist in OR/AND-reduce networks (very wide AND/OR). These patterns generally target very few remaining faults not detected by the 1:1 pattern, hence they will result in very good clock gating efficiency. However, they do also generally consume less power (in the data path) and in many cases test point insertion is preferred over running many different weights. The influence of weighting on test power has been evaluated in detail in (Zhang et al., 1999) and is not considered here.

The experiments in chapter 10 try to explore the first three of these tradeoffs. But in general, these tradeoffs are subject to the design-specific aspects of test resource, test time and physical design impact and only general observations can be made in this work.

Evaluation

This section presents an evaluation of the methods proposed in this work. The first set of experiments was conducted with the Cell Processor in an industrial manufacturing test setting and measurements were done with wafers of the Cell processor. The focus of these experiments is the implementation of the presented method in this scenario as well as creating a reference for the results found in the simulation-based experiments.

In the second set of experiments, several simulations are conducted with benchmark circuits as well as industrial circuits that are described in the first section of this chapter. The simulation-based experiments are performed to evaluate the overall performance of the presented approach and to establish the influence of parameters such as the number of BIST sessions. Section 10.3 of this chapter deals with BIST targeting stuck-at faults and also on transition fault tests with typical clock sequences.

The method outlined in this work was implemented in Java as part of an electronic design automation tool developed at the Institut für Technische Informatik.

10.1 Benchmarks and Industrial Circuits

For evaluation of the presented method, circuit models from the following sources were used (Table 10.1):

- International Symposium on Circuits and Systems (ISCAS89) (Brglez et al., 1989)
- International Test Conference (ITC99) (Davidson, 1999)
- Industrial circuits from NXP
- SPE Processor core of the IBM/Sony/Toshiba Cell processor (Riley et al., 2005)

Table 10.1 lists the test characteristics of the circuits. The first column gives the circuit name, where circuits from ISCAS89 start with s^* , those from ITC99 with b^* , NXP circuits are p^* and the Cell SPE is listed as SPE. The second column gives the

number of two-input gate equivalents in the simulation model. The third and fourth columns give the number of scan chains in the circuit and the number of scan flip-flops in the scan chains. The fifth column gives the number of collapsed single stuck-at (SSA) faults in the LBIST testable logic of the circuit. The last column reports the number of collapsed transition faults in the circuit. Not all of the collapsing for the stuck-at faults applies to transition faults, so the number of collapsed transition faults is generally more than twice the number for stuck-at faults. The number of transition faults is not listed for the SPE, since the circuit model was not available at the time the transition fault experiments were conducted.

Table 10.1. Parameters of Circuits used for Evaluation

Circuit	# Gates	# Chains	# FFs	# SSA Faults	# Trans. Flts.
s38417	24079	32	1770	32320	65364
s38584	22092	32	1742	38358	52018
b17	37446	32	1549	81330	143346
b18	130949	32	3378	277976	487136
b19	263547	32	6693	560696	981866
p286k	332726	55	17713	648044	1117520
p330k	312666	64	17226	547808	947798
p388k	433331	50	24065	856678	1476348
p418k	382633	64	29205	688808	1173036
p951k	816072	82	104624	1590490	2634564
SPE	1352044	32	40027	1065190	-

The circuits from ISCAS89 (s38417 and s38584) and ITC99 (b17, b18 and b19) are the largest circuits from each collection of benchmark circuits. They do not contain any kind of design for test (DfT) and were extended with the required BIST architecture. The flip-flops of each circuit were arranged into 32 parallel scan chains.

The circuits provided by NXP (p286k, p330k, p388k, p418k and p951k) already contained DfT with parallel scan chains and are several times larger than the circuits from ISCAS89 and ITC99. Moreover, they represent the typical properties of industrial circuits, namely shorter paths and smaller output cones as a consequence of the stronger optimization for high clock rates and low area.

The current implementation of the Cell processor (Figure 10.1) serves an example for the application of the presented method to a circuit with partial scan. It consists of about 250 million transistors and its top-level self-test architecture is partitioned into 15 self-test domains (so called BIST-satellites), each with its own STUMPS instance (Riley et al., 2005). The DfT in each self-test domain follows

the architecture outlined in Figure 7.1 in Chapter 6. The Synergistic Processing Elements (SPE) was chosen as a representative BIST-satellite, as 70% of the chip area is covered by the 8 identical SPEs.

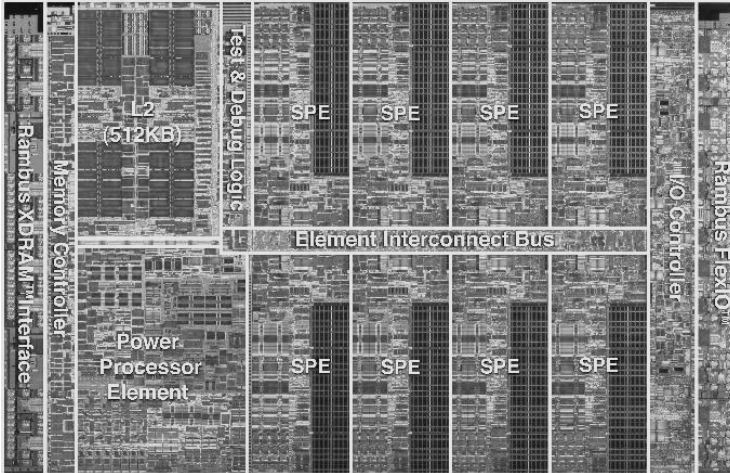


Fig. 10.1. Die Photo of the Cell-Processor

The characteristics relevant for the test of the SPE are:

- 1.8 million logic gates, 7 million transistors in the logic, 14 million transistors in memory arrays
- 150,000 flip-flops
- 82,500 flip-flops arranged in 32 STUMPS channels
- Memory arrays are not part of the logic BIST and are covered by a separate self-test structure and procedure

10.2 Wafer Test Experiments with the 45nm Cell/B.E. Processor

This section presents power measurements obtained for 45nm designs of the Cell Processor (Takahashi et al., 2008) on Automated Test Equipment at IBM Research & Development Boeblingen, Germany. The experiments consist of experiments verifying the general validity of the assumptions made in this work as well as actual measurements of a test plan generated using the method presented here. Furthermore, some simple variations of the method presented here are evaluated that may be desirable in wafer and module test.

10.2.1 Experimental Setup

The experiments discussed in this section were conducted in a wafer-test scenario based on the actual manufacturing tests for the Cell Processor presented in (Riley et al., 2005). The ATE employed for these experiments is a V93000™ tester (Figure 10.2) with special power supplies that facilitate product characterization. All results are limited to the SPE cores of the Cell chip, all other BIST domains are disabled during the experiments.

The existing test flow executes 200,000 patterns in the logic BIST run. For the evaluations conducted here, the existing BIST sessions are subdivided into 100 sessions overall to allow for the granularity required in the test planning algorithms presented here.



Fig. 10.2. Advantest V93000™ Test Equipment

Figure 10.3 shows a rough outline of the power measurements and the sampling point for measuring. The existing test flow initializes the entire chip for a logic BIST session, consisting of setting up the PLLs and scanning all of the flip-flops on the chip to create a well-defined state. This initialization sequence takes roughly 1.5ms. Then, the actual BIST session is executed. The run time of this session depends mostly on the scan frequency employed, for example at 400MHz frequency, the BIST session is roughly 3ms. After the BIST session is complete, the entire chip is scanned again to determine the MISR signatures and characterization data.

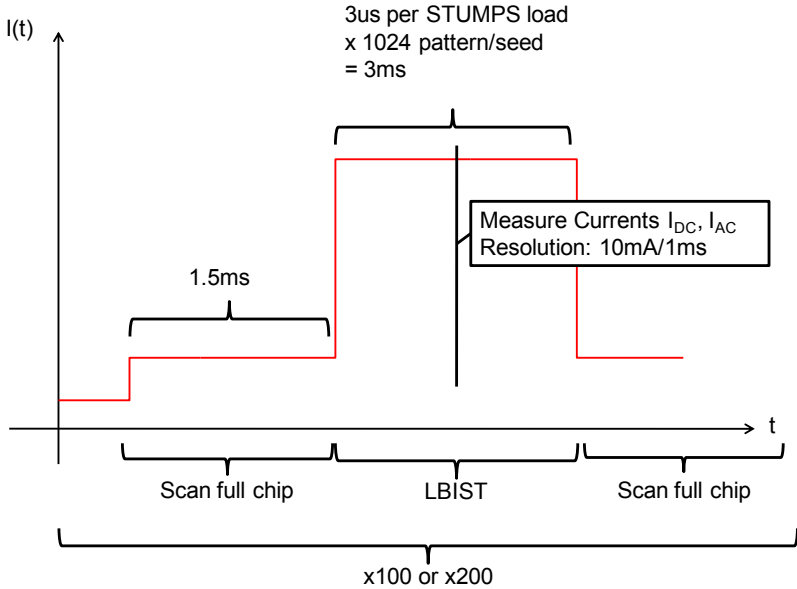


Fig. 10.3. Experiment Characteristics and Measuring Time

For each experiment, the external clock oscillator and PLL are operated at functional parameters. The voltage and scan frequency depend on the experiment conditions. Please note that the runtimes for setup and result evaluation can be avoided in a production environment, where only the seed and scan enables have to be changed and just the MISR signatures can be read for a go/no-go decision. Hence, the lead and trail times are disregarded in the following observations.

The measurement of power consumption focuses on the dynamic power consumption. Due to the large amount of infrastructure running at-speed, the dynamic power consumption is not zero even if no actual scan-chain is being clocked.

The results reported in the following focus mostly on the impact on power consumption during the test plan. Each BIST session has been repeated several times and for 10 chip sites on the same wafer. Only chips passing the original manufacturing flow have been selected. Due to this relatively small sample, the experimental results are not representative of a production run and no evaluation regarding the defect coverage of the test plans has been attempted.

10.2.2 Test Generation Flow

The test planning approach presented here poses several challenges regarding the integration into an existing test methodology and tool flow. This section describes some of the DfT details required to implement the clock control mechanisms that test planning takes advantage of, as well as the steps required to introduce the test planning into the industrial flow.

Some of the challenges are representative of the tool level available at the time. The concept of the test plan has been recently added to state-of-the-art EDA tools (Uzzaman et al., 2009) and in general, the support for the DfT structure to enable the fine-grained support for scan clocking that the Cell Processor provides has been improved.

The Cell Processor allows to not only select scan clocking on a chain-by-chain basis, but also supports arbitrary wave-form generation for scan and functional clocks. Figure 10.4 shows the generalized structure of this DfT. The wave-form generation required that all clock-related control signals are propagated at least at the clock frequency. To close the timing requirement imposed by this design, all of the control signals are staged by special flip-flops. 11 stages are introduced from the chip-level control to the BIST engine of a BIST domain (e.g. an SPE core). Another 7 stages for pipelining are added inside a BIST domain to fan-out the local signals.

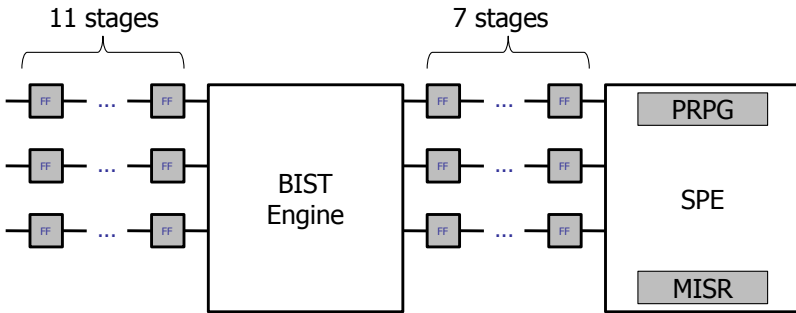


Fig. 10.4. Clock and BIST Control Pipelining in the Circuit

Figure 10.5 shows the flow of steps to run a complete testplan on the ATE. The EDA tool for test generation and fault simulation at IBM is Cadence EncounterTest™. Tool limitations (at the time) were encountered both with respect to the particular design, as well as the way in which test planning uses the design. A major tool limitation in dealing with the specific DfT setup outlined above was imposed

by the lack of support of the staged control signal architecture. While this was not a functional impediment, it significantly reduced productivity, for example in the automated decoding of the correspondence between ATE vectors and scan flip-flop values. Furthermore, the existing test flow required computationally expensive operations when creating BIST sessions. The original test flow has only very few sessions of high pattern count limiting the impact of this issue. But the test planning increases the number of individual test sessions, exposing this particular issue.

Due to these limitations, the scan enable vectors themselves are introduced after the computationally expensive checks. Consequently, the scan configuration imposed by a session was not subject to the verification techniques provided by Encounter Test. A limited number of configurations was subjected to the full flow with all checks to ensure that no invalid assumptions were made in this approach.

After translation of the pattern for the Verigy ATE, it was found that the existing test methodology at the time did not take the large number of BIST sessions into account that is required for test planning. Additional, manual steps would have been required including editing ATE microcode. To overcome this, additional automation was added to patch the ATE microcode for each BIST session of each experiment automatically.

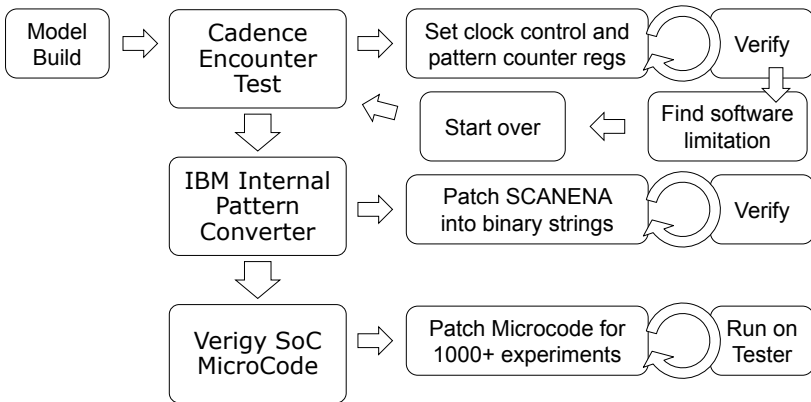


Fig. 10.5. Workflow for Test Data Generation and Test Execution

10.2.3 Experiment: Accuracy of Power Estimates

The first experiment presented here tries to evaluate the validity of the assumptions regarding the predictability of the test power consumed in a given scan configura-

tion. During the set covering optimization, a very efficient test power estimation is required. The associated linear model has been presented in section 7.2 (→ ‘Estimation of Power Consumption’ starting on page 63) together with a simulation-based verification.

For the verification, 200 experiments have been executed with random scan configurations, where each experiment was repeated ten times for the ten chips sites mentioned before. In each experiment, the dynamic power consumption is measured and the linear power model in section 7.2 is fitted to this scenario. Figure 10.6 shows the error between the predicted and the measured results. It is observed that the prediction error is just 3% on average and 10% in the worst case.

Furthermore, each execution on the same chip site showed negligible variation and variation across chip sites was limited to the variation experienced by the original manufacturing test suite.

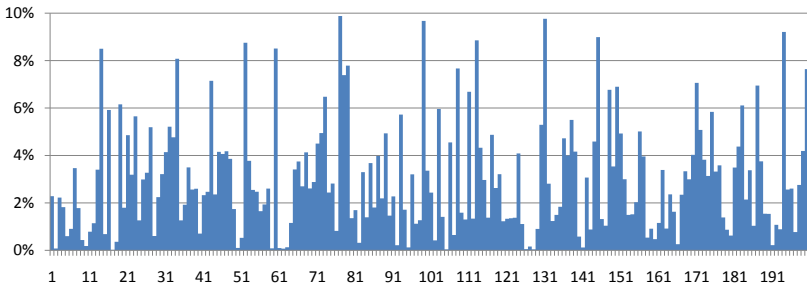


Fig. 10.6. Prediction Error Across 200 Experiments

10.2.4 Experiment: Power Reduction by Test Planning

In the next experiment, an actual test plan was computed using the method outlined in this work for 200 BIST sessions consisting of 1024 patterns each. Each session of the test plan was repeated ten times for each of the ten chip sites mentioned before. Figure 10.7 shows the measured power consumption for this test plan.

The result matches the results encountered in the purely model-based results generated in this chapter. This is supported by the prediction error results in the previous subsection. In order to provide a more systematic view on the impact of test planning, another experiment was conducted with a canonical test plan that does not take into account test coverage requirements like the test planning presented in this work.

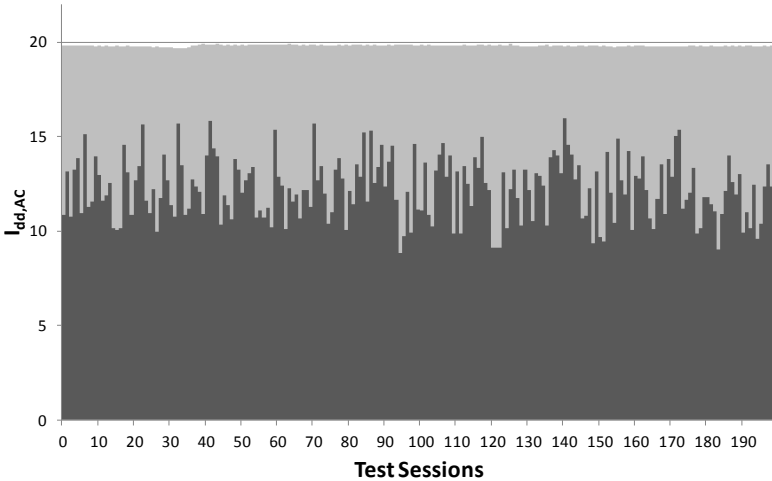


Fig. 10.7. Measured Power Reduction with the Cell Processor (Lower Bars are After Power Reduction)

For this experiment, ten BIST sessions were generated for scan configurations selected with $\{4, 8, 12, 16, 20, 24, 28, 32\}$ scan chains chosen randomly out of the 32 available. The result of this experiment is reported in Figure 10.8.

10.2.5 Frequency Scaling

This subsection evaluates the ability to scale the scan frequency of the DfT in the Cell Processor chips tested here. Based on this experiment, two additional test plans are evaluated that take into account the scan frequency according to section 9.2 in order to further average the test power and to use as much of the test power envelope as possible.

Figure 10.9 shows the repetition of the canonical experiment employed in the previous subsection for the scan frequencies $\{400, 533, 800, 1067, 1600\}$ Megahertz ($d_{scan} = 2$ according to section 9.2. Please note that the x-Axis is discrete and not linear.

The frequency scaling works by modulating the pulse width of the scan clock enable signal. This means that the base-line dynamic power consumption of the DfT infrastructure remains unaltered. This is clearly visible in the case of no enabled scan chains. Dynamic power consumption still varies with the scan frequency due to the

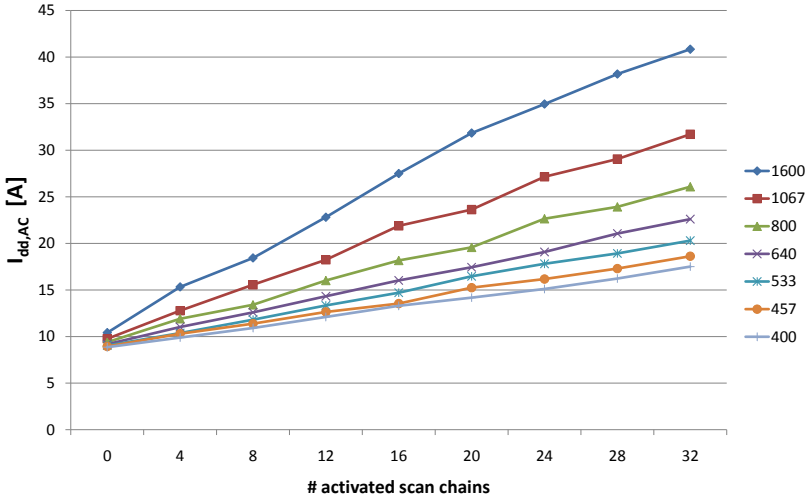


Fig. 10.8. Relation between Current and Number of Enabled Scan Chains

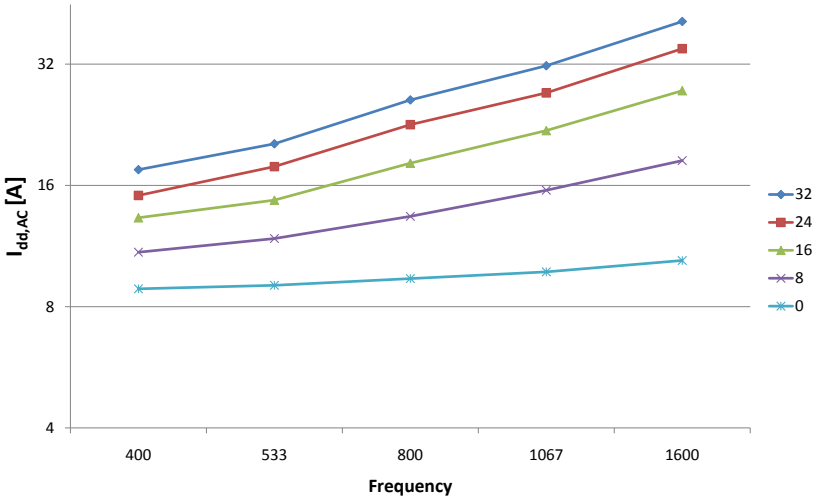


Fig. 10.9. Frequency Scaling with 0, 8, 16, 24 and 32 Scan Chains Enabled

toggleing of the control signals, staging flip-flops, etc.. When frequency scaling is employed, this base-line consumption must be taken into account.

Moreover, it was observed that at very low voltages higher frequencies could be achieved if few scan chains were activated compared with clocking all scan chains. While this observation has not been investigated quantitatively, it implies that power supply noise and instantaneous power consumption are reduced by scan chain disabling as well. This observation is supported by the simulation results in section 7.2.

Figures 10.10 and 10.11 show two enhanced test plans that take advantage of frequency scaling and take into account the impact of the baseline dynamic power consumption. The power (current) envelope was chosen as 20A, which is realistic but does not necessarily reflect actual production parameters. The basis test plan minimizes test power as presented in the previous chapter. Based on the scan configurations and seed selected in this test plan, the scan frequency is then selected to either average out the test power consumption (Figure 10.10) or to increase the scan frequency so that the test power envelope is used up (Figure 10.11).

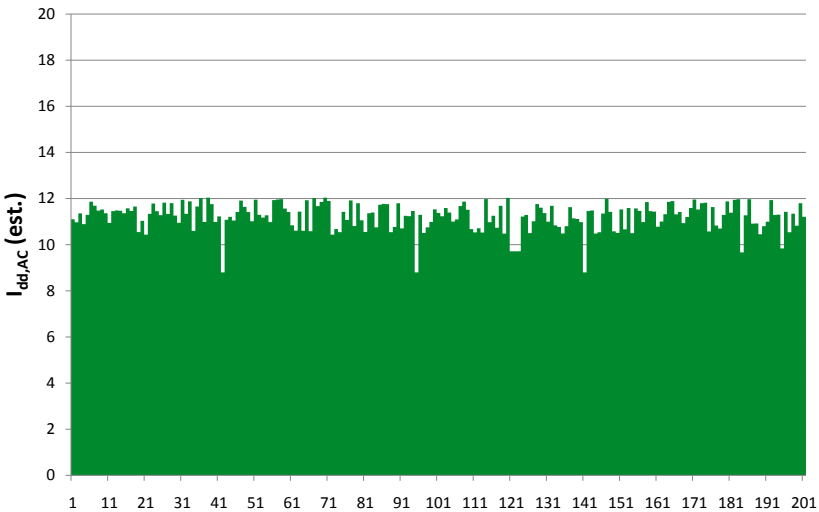


Fig. 10.10. Power Averaging through Frequency Scaling

When using up the given test power envelope, the increase in scan frequency leads to an overall reduction of 75% of the BIST-related test time when compared to the BIST at the default scan frequency. As mentioned before, the experimental methodology used here does not evaluate defect coverage and the impact of process variation is only limited to a single wafer.

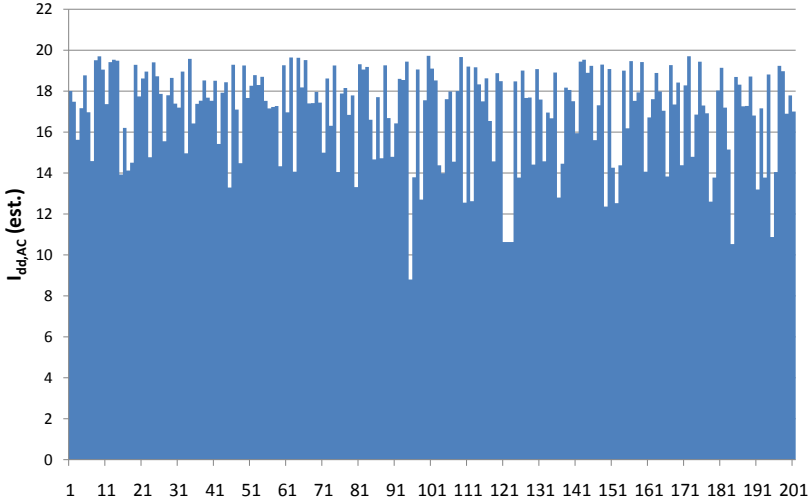


Fig. 10.11. Current When Using Frequency Scaling to Use All of a Given Power Envelope

Instead of using existing test power envelopes that have been determined according to conventional assumptions regarding BIST test power, it may be beneficial to use test power envelopes based on functional power profiles or the ability to deliver power through wafer probes.

10.3 Canonical Experiments with Benchmark Circuits

In the following, a set of simulation experiments is presented, which use the circuits presented in section 10.1 and assume parameters suitable for comparison and analysis of the method and its parts.

10.3.1 Test Planning for Stuck-At Tests

This sub-section presents results for a typical self-test scenario with 200k pseudo-random patterns. The number of randomly chosen seeds is 200, and per seed a fixed number of 1024 patterns is generated by the PRPG. For step 2 of the algorithm, the parameter *lim* was set to a value of 3. The first group of columns of Table 10.2 show the circuit name as well as the number of stuck-at faults detected by the chosen seeds.

The second to fourth column groups show results for the three steps of the presented optimization algorithm. For hard faults, the second group of columns lists their absolute quantity as well as the cost of the generated sessions $cost(T_1 \cup T_0)$.

Table 10.2. Evaluation of Test Plans with 200 Seeds, 1024 Patterns

Circuit	$ F_{det} $	Hard Faults		FSIM	Hard+Diff. F.	FSIM	All Faults	
	Detected	$ F_{hard} $	cost()	$ F_1 $	cost()	$ F_2 $	cost()	PR [%]
s38417	31661	500	1484	30836	1873	31431	2165	67.43
s38584	36389	26	90	13506	243	34922	350	94.42
b17	70547	1947	2691	70230	3152	70519	3246	49.29
b18	239753	6291	4164	239451	4446	239952	4498	33.36
b19	479689	13802	4915	479220	5112	480253	5128	22.69
p286k	610094	6226	9901	603509	9990	609269	10024	10.43
p330k	491199	3036	6214	436877	6733	490705	6799	41.94
p388k	839174	3239	3732	790766	4129	835695	4233	58.93
p418k	639750	8851	4895	577037	5372	638527	5470	45.37
p951k	1544131	7995	7640	1421168	7983	1542416	8069	46.72
SPE	904183	1805	1150	817125	1342	900114	1451	75.33

The column $FSIM$ provides information on the number of faults $|F_1|$ detected by those sessions. For the difficult faults in the next column group, the result of the cost function $cost(B_2 \cup B_1 \cup B_0)$ after optimization is followed by result $|F_2|$ of the fault simulation of the intermediate result. Finally, the last column group lists the cost needed for the final result and the achieved power reduction (PR) using the estimation method outlined in section 7.2 (\rightarrow ‘Estimation of Power Consumption’ starting on page 63).

For circuits with a large number of hard faults, the degrees of freedom in the last two optimization steps cannot be fully exploited. For example, when considering p951k already 95% of the cost is required in order to detect the hard faults (for p286k this is even 99%). Additional degrees of freedom can be created by changing the parameters „number of seeds“ and „patterns per seed“. This is evaluated in section 10.3.2 (\rightarrow ‘Impact of Test Plan Granularity’ starting on page 106).

The runtimes of the approach are dominated by the fault simulation of the pseudo-random patterns.

In Figure 10.12, the details of a single test plan are shown. It plots the number of activated flip-flops per seed over the test sessions. It is obvious that the test plan computed by the presented algorithm is not affected by a bias and the power consumption is distributed relatively evenly. The test plan can be further improved by reordering seeds (together with their corresponding configurations) or by changing other parameters that influence power such as the shift frequency. This allows to further adapt the power dissipation to a given envelope. Examples of this are shown in section 10.2.5 (\rightarrow ‘Frequency Scaling’ starting on page 101).

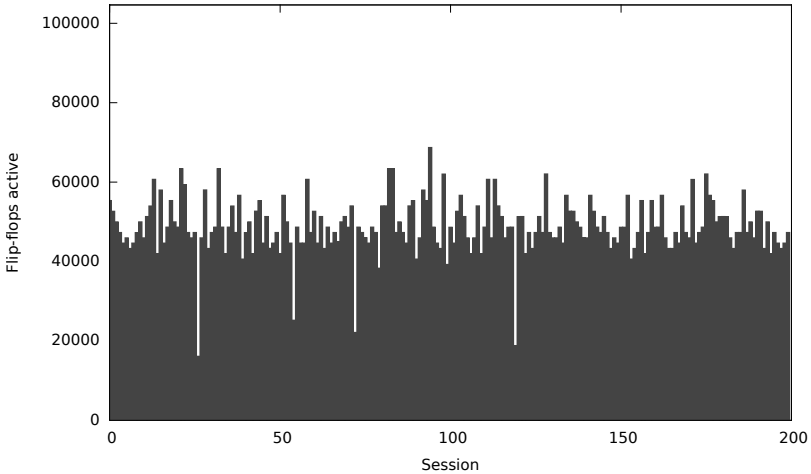


Fig. 10.12. Enabled Flip-flops for each Test Session in the Test Plan for Circuit p951k

10.3.2 Impact of Test Plan Granularity

One important parameter that influences the effectiveness of the presented approach is the number of test sessions. In the previous section, the number of sessions was set to a realistic value given the common industrial manufacturing flows. However, with careful design of hardware and software, higher numbers of sessions may be implemented as well. This subsection will demonstrate the behavior of the proposed approach if the number of test sessions is varied.

Figure 10.13 gives estimates of the power reduction when the number of seeds is varied for the benchmarks from ISCAS and ITC. In this experiment the overall pattern count is kept constant at 204,000.

In the previous sections, it was shown that the hard to test faults impose the highest fraction of power dissipation on the test plan. By varying just the number of test sessions, the overall test set and hence the number of hard faults remains the same. But in most cases just a single pattern in the session associated with the hard fault does indeed detect the fault, and by splitting one session into several the power consumed for hard faults can be reduced significantly.

On the other hand, reducing power consumption below 25-30% of the usual test power envelope will in many cases reduce the power consumption well below the functional in-the-field power consumption. Often, this reduced power supply noise is not desirable in terms of defect coverage, since circuit delay depends on the local

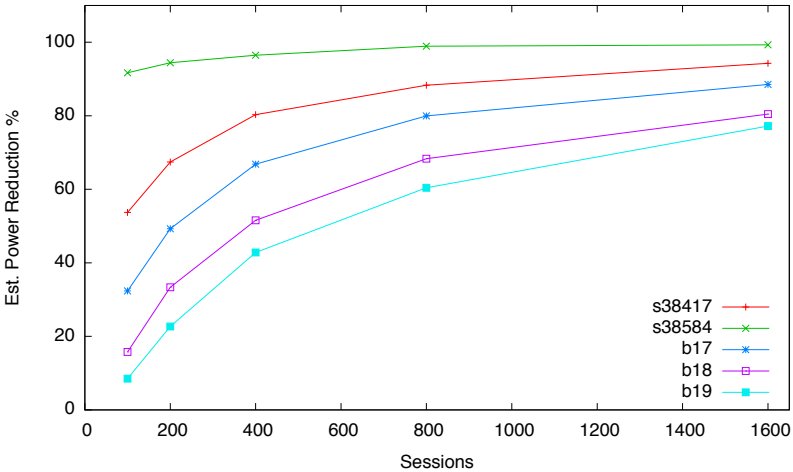


Fig. 10.13. Dependency on the Number of Sessions for 200k Patterns

supply voltage (and hence the detectability of small delay defects). However, the higher granularity can then be used with reseeding methods based on deterministic patterns such as (Koenemann, 1991). This allows to specifically target hard-to-test faults compared to a regular pseudo-random BIST.

10.3.3 Impact of Test Length

Besides the number of test sessions, the overall test-length is an important parameter. First, it determines the test time, which in turn is an important factor in the monetary cost of test. Second, it determines the occurrence of hard-to-test faults, which influences the effectiveness of the test planning approach.

To evaluate the influence of the test length, Table 10.3 presents results for two test lengths of 100k and 200k patterns which are organized into 200 sessions of 512 or 1024 patterns. The results for the circuits reference in column one are organized into three groups. The first group lists the results for 100k patterns and gives the number of faults $|F_{det}|$ detected by the BIST, the number of hard faults $|F_{hard}|$ observed and the power reduction PR achieved by the test planning. The second group of columns gives the same results for the case of 200k patterns. The last column compares the average power consumption with 200k patterns with the power consumption while applying 100k patterns. Please note that the test energy required for applying 200k patterns is twice as high (in the worst case) by principle.

Table 10.3. Comparison of Two Test Lengths with Session Count 200

Circuit	512 pats. / session			1024 pats. / session			$\frac{P_{avg,200k}}{P_{avg,100k}}$
	$ F_{det} $	$ F_{hard} $	PR [%]	$ F_{det} $	$ F_{hard} $	PR [%]	
s38417	31189	569	69.08	31661	500	67.43	1.05
s38584	36365	39	89.45	36389	26	94.42	0.53
b17	67419	2965	56.27	70547	1947	49.29	1.16
b18	234599	9839	32.54	239753	6291	33.36	0.99
b19	468538	21212	24.32	479689	13802	22.69	1.02
p286k	604906	7353	7.98	610094	6226	10.43	0.97
p330k	489007	4066	39.66	491199	3036	41.94	0.96
p388k	836816	4055	55.13	839174	3239	58.93	0.92
p418k	633090	10018	43.43	639750	8851	45.37	0.97
p951k	1539400	9522	43.45	1544131	7995	46.72	0.94
SPE	902929	2392	71.15	904183	1805	75.33	0.85

Increasing the overall test length reduces power consumption in most cases, while significantly increasing the number of detected faults. The additional patterns applied to the circuit detect a large number of the faults that are classified as hard at just 100k, such that these faults are now detected more than once. This additional freedom is effectively used by the test planning algorithm to reduce power consumption. Three academic benchmark circuits show an increase in power consumption, which is caused by the large number of faults that are detected in addition to the 100k pattern test. None of industrial circuits show this behavior. Often these circuits are designed to achieve high random-test coverage with reasonable test lengths (Williams and Angell, 1973), like the SPE. In addition, the design for high frequency and layout constraints result in higher random testability.

10.3.4 Impact of Scan Chain Insertion Approach

Besides the BIST execution parameters such as test length and number of sessions, design properties such as the number of scan chains or the clustering used to group flip-flops into scan chains also have significant impact on the effectiveness of the test planning. The number of scan chains influences the granularity of the test planing similar to the granularity evaluated in subsection 10.3.2 (\rightarrow ‘Impact of Test Plan Granularity’ starting on page 106). This subsection focuses on the impact of the clustering with which flip-flops are grouped into scan chains.

Commercial EDA tools for scan insertion mostly use topological or hierarchical algorithms to determine the clustering and will then order scan chains according

to layout and wiring requirements. Instead of using an unconstrained topological or hierarchical approach for the clustering, additional constraints can be introduced that benefit test planning without constraining the ordering step of scan insertion (section 7.5 (\rightarrow ‘Scan Cell Clustering’ starting on page 69)). This way, the optimized scan clustering does not significantly impact the wiring overhead.

In very rare cases, the scan insertion is already done manually as part of the high-level design. For example, in the case of the Cell processor, the scan paths are timed at-speed to facility debugging and initialization (Riley et al., 2005). The at-speed timing and manual hierarchical scan insertion result in a scan chain clustering that is very contained and reduces the amount of combinational paths between multiple scan chains.

Table 10.4 compares three different clustering approaches. For each circuit, the original scan clustering is used as the baseline, which is based on unconstrained topological insertion in all cases but the SPE. The SPE uses a full-custom design insertion as outlined above. This configuration is first compared with a random clustering of the scan chains to show the worst case behavior of a clustering approach. The third organization was created with the power-aware approach presented in section 7.5. All three scan organizations are created with the same number of scan chains as outlined in table 10.1. For the evaluation, a test plan was generated with each clustering for 200 test sessions of 1024 patterns each.

Table 10.4. Comparison of Three Scan Chain Insertion Algorithms

Circuit	$ F_{det} $			% Power rel. original	
	Random	Original	Partition	Random	Partition
s38417	31556	31589	31663	212.89	37.72
s38584	36385	35989	36389	395.21	60.22
b17	70535	70300	70238	181.09	75.78
b18	239594	240385	239599	153.06	84.56
b19	479901	479834	479037	134.88	91.14
p286k	609609	610070	610072	109.77	59.09
p330k	491079	491528	491464	169.64	48.64
p388k	839075	839352	839004	237.39	59.29
p418k	639787	639786	640146	182.05	72.85
p951k	1545320	1544906	1544512	185.68	48.87
SPE	904534	904544	904434	378.82	102.55
SPE*	904534	904544	904303	378.82	93.56

The table reports the number of detected faults $|F_{det}|$ for each clustering, since the location of a flip-flop within the STUMPS channels determines the patterns applied through it. However, this is a random process and the impact is very small. The last

two columns of table 10.4 compare the power consumption of the random and power-aware clustering with the original scan clustering of the circuit. For all circuits with the exception of the SPE, the power consumption is significantly reduced by the power-aware scan clustering. The power-aware scan clustering requires to activate fewer scan chains to detect a given hard fault. Two results are given for the SPE. The results labeled *SPE* discards the original hand-crafted scan chain organization and uses the full algorithm outlined in section 7.5. This clustering almost achieves the power reduction of the hand-crafted solution. The result labeled *SPE** uses the hand-crafted cluster as the basis for the iterative post-optimization from the algorithm and is able to improve the power reduction over the original scan insertion.

10.3.5 Test Planning for Transition Fault Tests

Transition faults require tests that apply at least two patterns to the CUT. In scan-based circuits, these patterns are created through shift or capture clocks applied at speed as described in section 9.1 (→ ‘Transition Faults’ starting on page 85).

The difficulty of test planning for transition fault tests is two-fold: First, the transition tests require special consideration to model the sequential behavior of the circuit. Second, the multi-cycle tests can lead to much more extensive fault propagation, involving many more flip-flops to sensitize the multi-cycle propagation path. The evaluation in this section employs the circuit modeling presented in section 9.1 allow to apply the existing test planning algorithm to transition faults. In order to quantify the impact of the sensitization and propagation issues mentioned above, various clock schemes are compared and evaluated. The first subsection in this section is devoted to the simple two-cycle launch-capture tests LOS and LOC. The section subsection evaluates test planning for the more advanced LOS and LOSC multi-cycle clock schemes.

All the test plans are generated for a BIST setup with 200 seeds, and 1024 patterns are generated from each seed. Transition faults are two pattern tests and they have lower detectability than stuck-at faults. Hence, it may be acceptable or desirable to apply test sequences even longer than 200k patterns. It has been shown in section 10.3.3 that longer tests improve the results obtained with test planning. In these experiments, the power-aware scan organizations from section 7.5 are used to improve power reduction.

The clock sequences evaluated here are:

- A single capture cycle (LOC)
- A single shift cycle (LOS)
- A capture cycle followed by a shift cycle (LOCS)

- A shift cycle followed by a capture cycle (LOSC)

10.3.6 Launch-Off-Capture and Launch-Off-Shift Tests

This subsection will investigate the two clock sequences LOC and LOS. LOS has the interesting property that no additional scan chains have to be activated to enable this transition test, since it only requires a shift cycle for the launch and hence no paths between scan chains are added. On the other hand, LOS tests are known to detect significantly more transition faults than LOC tests.

Table 10.5 reports the results for LOC and LOS. $|F_{det}|$ is the number of faults detectable by the seeds of the tests and targeted by the test plan. $|F_{hard}|$ is the number of (hard) faults, detected by just a single seed from the overall set of seeds. PR is the estimated power reduction in percent of regular execution of the test without turning off any scan chains. Again, the power is estimated by the method from section 7.2.

Table 10.5. Power Reduction for LOC and LOS Transition Clock Schemes

Circuit Name	LOC			LOS		
	$ F_{det} $	$ F_{hard} $	PR [%]	$ F_{det} $	$ F_{hard} $	PR [%]
s38584	47527	502	84.67	58645	182	90.94
s38417	47869	1283	83.89	49209	1179	84.07
b17	89814	4099	43.89	113476	7017	43.12
b18	259294	23026	30.9	383652	19414	22.7
b19	518771	40038	22.15	768408	40908	15.35
p286k	802947	41401	20.96	1020417	16170	29.59
p330k	753738	16568	43.47	823477	8580	61.43
p388k	1256203	17920	41.19	1416672	7153	61.52
p418k	866561	26480	39.64	1035798	20019	43.9
p951k	2280840	19812	62.58	2418250	16207	63.55

As expected, LOS detects significantly more faults than LOC. With LOC, the random patterns are launched through the logic network and this introduces significant correlation between the two patterns. With LOS, the shift cycle causes correlation between consecutive flip-flops in the scan chains, but this is less severe compared to LOC.

In most cases, the highest reduction of the test power is achieved when using the LOS clock scheme. For LOC, the set of flip-flops that has to be activated to detect a target fault is relatively large. In addition to each flip-flop observing the consequences of a fault, it includes all of the flip-flops in its input cone and in turn

all the flip-flops in the inptheut cones of these flip-flops. These flip-flops span many more scan chains than the small set that is sufficient for LOS.

10.3.7 Complex Test-Clock Sequences

Next, more advanced clock sequences are evaluated, which consist of two launch cycles combining shift and capture cycles. Since both LOCS and LOSC contain a capture cycle, they are subject to the increase in the number of chains that are required to detect a fault that is observed with the LOC scheme.

Table 10.6 reports the results for LOCS and LOSC test clock sequences. Again, $|F_{det}|$ is the number of faults detectable by the seeds of the tests and targeted by the test plan. $|F_{hard}|$ is the number of (hard) faults. PR is the estimated power reduction.

Table 10.6. Power Reduction for LOCS and LOSC Complex Transition Clock Sequences

Circuit Name	LOCS			LOSC		
	$ F_{det} $	$ F_{hard} $	PR [%]	$ F_{det} $	$ F_{hard} $	PR [%]
s38584	57353	314	88.4	61174	288	91.08
s38417	48128	1089	86.53	50511	947	88.53
b17	110467	7874	46.81	117766	4732	41.64
b18	374471	20725	25.6	389137	14725	22.41
b19	755697	42435	17.41	778310	30475	15.32
p286k	980883	19999	22.66	1010510	18945	26.95
p330k	786042	19875	48.16	830790	8725	60.99
p388k	1399901	14714	46.66	1416907	9627	55.29
p418k	944916	17666	44.85	1045834	21316	43.73
p951k	2409177	18034	63.43	2449348	15186	66.33

LOCS uses a capture clock cycle followed by a shift clock cycle. The shift cycle is able to randomize much of the correlation caused the combinational logic. Consequently, LOCS is very close to LOS in terms of fault coverage. Moreover, LOSC has some interesting properties: First, the leading shift cycle activates a large number of transition faults as expected from LOS. Second, the capture cycle effectively propagates the circuit responses and at the same time it activates additional transition faults. Hence, LOSC has the highest fault coverage of all four clock schemes for all the circuits except *p286k*. In contrast, the responses of the leading capture cycle in the LOCS scheme can only be used for justification since transition faults are not propagated by the subsequent shift cycle.

LOCS and LOSC suffer from the rather large input cones due to the capture cycle that LOC does. However they exhibit some rather interesting properties: Many faults are detected by many more seeds compared to LOC and an indication of this is the significantly reduced number of hard faults for LOCS and LOSC. This effect is even more pronounced for LOSC, since faults are activated in both cycles of the clock scheme. The additional degree of freedom is effectively used by the test planning and the power reduction achieved with LOSC is comparable to that of LOS and even exceeds LOS for *s38584*, *s38417* and *p951k*.

If the best clock scheme is selected for each of the circuits, the power reduction obtained here is in the same range as the power reduction obtained for stuck-at-faults in section 10.3.1.

Conclusion

11.1 Summary

This work presents a technique for power-aware logic built-in self-test (LBIST) that yields substantial power reductions and scales to industrial-size designs. The implemented scheme is based on the wide-spread STUMPS architecture and can be combined with any of the common-place at-speed test techniques. For this, the STUMPS architecture is extended by clock gating, which is often already present in power-aware designs. The test planning algorithm proposed in this work optimizes the test to take advantage of this clock gating and reduces the dynamic power consumption by a substantial amount.

The state-of-the-art techniques for power-aware LBIST have been reviewed in depth in this work. Many techniques can be combined with the technique proposed here and others are superseded by it. This technique shows, for the first time, how clock gating can be effectively used during LBIST such that the total energy and power consumption is reduced without impacting fault coverage.

The algorithm presented here has been specifically designed such that it can handle industrial-size circuits. The run-times and memory requirements do not increase prohibitively with circuit size. The algorithmic optimizations are chosen such that the test planning is particularly robust regarding the circuit's structure and the at-speed clocking chosen for the test as exhibited by the experimental results. The results are even better for the most challenging high-performance designs such as the Cell/B.E. Processor™.

Besides experiments for the most common structures and clock sequences, this work also takes into account the more recent at-speed test-clock sequences such as Launch-Off-Shift-Capture and Launch-Off-Capture-Shift. The experiments conducted here show that these techniques are very effective in detecting additional transition faults and work in conjunction with the method presented here. The resulting test plans for these two clock schemes use similar or sometimes less power than the current state-of-the-art at-speed clock schemes.

11.2 Contributions

The scientific contributions of the work presented here are:

- **Power-aware synthesis of a clock-gated STUMPS architecture, Chapter 7:** The traditional STUMPS architecture has been extended by clock gating and the clustering of the flip-flops into scan chains is automatically selected with an efficient heuristic such that the efficiency of test planning and other methods is improved.
- **Scalable Power-Aware Test Plan Generation, Chapter 8:** The work presented here provides an innovative approach to test planning for LBIST. It scales well to industrial-size circuits and fits very well into an industrial design automation tool-flow. The approach results in a significant power-reduction compared to regular STUMPS-based LBIST and is compatible to other power-reduction techniques such as toggle suppression or staggered clocking.
- **Extension of the Approach to Arbitrary Clock Schemes, Sections 9.1 and 10.3.5:** At-speed testing for transition faults is an important aspect of BIST. High test coverage is essential for high-performance designs that have a significant percentage of critical paths. Complex test-clock schemes can increase the fault coverage of the LBIST substantially. In this work, arbitrary test-clock schemes are modeled by a novel graph-generation approach that maps the sequential behavior to a combinational equivalent. Even with these more complex test-clock schemes, the test planning approach provides a similar or better power reduction compared to the stuck-at tests or simple transition test schemes. The graph modeling can be applied to test EDA algorithms besides BIST planning.
- **Hardware Experiments with a State-of-the-Art High-Performance Design, Section 10.2:** In order to evaluate the applicability of the approach to a high-performance design and judge the implementation with respect to an industrial standard-tools design flow, experiments have been conducted with a 45nm design of the Cell/B.E.TM processor architecture. These experiments were carried out in a wafer-test scenario and showed substantial improvements in power consumption. Furthermore, the ability to reduce test-time was successfully demonstrated. For this, the scan frequency was increased for the test sessions that did not use the specified power-envelope.
- **Simulation Experiments with Industrial Benchmark Circuits, Chapter 10.3:** An extensive set of large academic and industrial benchmark circuits has been evaluated in order to assess the presented method. The experiments explore various parameters of the algorithm, such as test length or scan configuration, in

order to assess its robustness when being applied to other designs or different test requirements. The power estimation methodology used here has been verified with gate-level cycle-accurate simulations.

11.3 Possible Future Work

The DfT architecture presented here can be made fully embedded by providing on-chip stores for seeds and configurations. For example by reusing existing on-chip memories during test. The algorithms presented do not have to be altered for such an architecture. The approach could then be combined with advanced signature diagnosis techniques like (Elm and Wunderlich, 2010) to form a completely embedded, low-power, diagnosable architecture.

List of Figures

1.1	DfT mit parallelen Prüfpfaden und Taktabschaltung pro Prüfpfad . . .	3
1.2	Beispielschaltung	5
1.3	Chip Foto des Cell-Prozessors	9
1.4	Abhängigkeit von der Anzahl der Startwerte für ISCAS und ITC	11
1.5	Testplandetails für Schaltung p951k	12
3.1	A Stuck-at-0 Fault	21
3.2	Types of Bridging Faults	22
4.1	BIST Classification	26
4.2	Basic Principle of Structural BIST	26
4.3	Principle of the Scan Path	28
4.4	The Two Most Common Scan Cell Designs	28
4.5	Standard LFSR	30
4.6	LFSR-based TPG with Phase Shifter	31
4.7	Modular LFSR	32
4.8	Compactor Based on MLFSR	33
4.9	Multiple-Input Signature Register (MISR)	34
4.10	Self-Test using MISR and Parallel shift register Sequence Generator (STUMPS)	35
5.1	CMOS Inverter With Load Capacitance C_L	40
5.2	Typical Power Consumption During a Clock Cycle	42
5.3	Sequential Model to Analyze Peak Single-cycle Power	42
5.4	Circuit States and Transitions During Shifting	43
5.5	Launch-Off-Shift (LOS) and Launch-Off-Capture (LOC)	43

6.1	LSSD Scan Cell Using Transmission Gate Latches	46
6.2	Scan Clock Gating of Useless Patterns	47
6.3	Design for Test with Scan Clock Gating During Useless Patterns	48
6.4	Boolean Function of the Decoder	48
6.5	Decoder for Pattern Suppression for the Example	49
6.6	Master-slave Muxed-D Cell with Toggle Suppression	50
6.7	Toggle Suppression Implemented with Multiplexer	50
6.8	General Scan Insertion Flow	51
6.9	Scan Path Segmentation	52
6.10	Clock Sequence for the Scan Segmentation in Figure 6.9	53
6.11	Staggered Clock Sequence for Shift Peak Power Reduction	54
6.12	Clock Sequence for Launch-capture Peak Power Reduction	54
6.13	Input Cone that is Contained in a Single Scan Segment	55
6.14	Example of a Scan Tree	56
6.15	Scan Tree Combined with Conventional Scan Path	56
6.16	Double Scan-tree	57
6.17	Scan Path Configurations for Figure 6.16	57
6.18	Example of Scan Segment Inversion	58
6.19	Scan Segment Inversion by Embedding a Linear Function	59
6.20	Pseudo-random BIST with Scan Segment Inversion	60
7.1	DfT with Parallel Scan Chains and Clock Gating per Chain	62
7.2	Local Clock Buffer for Functional and Scan Clock Gating	63
7.3	Using Non-scan Cells to Block the Propagation of Switching Activity	63
7.4	Linearized Power Consumption vs. Simulation Results for Circuit p418k	67
7.5	Linearized Power Consumption vs. Simulation Results for Circuit p951k	67
7.6	Example of Detecting a Fault f and the Associated Input Cone	68
7.7	Scan Cell Observing a Fault with Input Cone	69
7.8	Parameters k and t in Scan Chain Clustering	70
7.9	Hypergraph and Hyperedge for Figure 7.7	71
8.1	Flow Chart of Test Planning Algorithm	79
8.2	Circuit for Test Planning Example	80
8.3	Test Planning Example Considering a Hard Fault	81
8.4	Test Planning Example Considering A Difficult Fault	83

9.1	A Circuit Graph	87
9.2	The Graph Concatenation for a Capture Clock	88
9.3	The Graph Concatenation for a Shift Clock	89
10.1	Die Photo of the Cell-Processor.	95
10.2	Advantest V93000™ Test Equipment	96
10.3	Experiment Characteristics and Measuring Time	97
10.4	Clock and BIST Control Pipelining in the Circuit	98
10.5	Workflow for Test Data Generation and Test Execution	99
10.6	Prediction Error Across 200 Experiments	100
10.7	Measured Power Reduction with the Cell Processor (Lower Bars are After Power Reduction)	101
10.8	Relation between Current and Number of Enabled Scan Chains	102
10.9	Frequency Scaling with 0, 8, 16, 24 and 32 Scan Chains Enabled.	102
10.10	Power Averaging through Frequency Scaling	103
10.11	Current When Using Frequency Scaling to Use All of a Given Power Envelope	104
10.12	Enabled Flip-flops for each Test Session in the Test Plan for Circuit p951k	106
10.13	Dependency on the Number of Sessions for 200k Patterns	107

List of Tables

1.1	Parameter der für die Evaluierung eingesetzten Schaltungen	10
1.2	Ergebnisse für Testpläne mit 200 Startwerten je 1024 Muster	10
6.1	Fault Simulation Result for a Test Set with 16 Patterns	48
7.1	Estimation Errors Caused by Linearized Power Model	66
10.1	Parameters of Circuits used for Evaluation	94
10.2	Evaluation of Test Plans with 200 Seeds, 1024 Patterns	105
10.3	Comparison of Two Test Lengths with Session Count 200	108
10.4	Comparison of Three Scan Chain Insertion Algorithms	109
10.5	Power Reduction for LOC and LOS Transition Clock Schemes	111
10.6	Power Reduction for LOCS and LOSC Complex Transition Clock Sequences	112

List of Abbreviations

ATE	Automated Test Equipment
ATPG	Automated Test Pattern Generation
BILBO	Built-In Logic Block Observers
BIST	Built-In Self-Test
CMOS	Complementary Metal-Oxide Semiconductor
CUT	Circuit Under Test
DFT	Design For Test
EDA	Electronic Design Automation
FF	Flip-Flop
FSIM	Fault Simulator
IP	Intellectual Property
LFSR	Linear Feedback Shift Register
LHCA	Linear Hybrid Cellular Automata
LOC	Launch On Capture
LOCS	Launch On Capture-Shift
LOS	Launch On Shift
LOSC	Launch On Shift-Capture
LSSD	Level-Sensitive Scan-Design
MAPE	Mean Absolute Percentage Error
MISR	Multiple-Input Signature Register
MLFSR	Modular Linear Feedback Shift Register
PLL	Phase-Locked Loop
PPSFP	Parallel Pattern Single Fault Propagation
PRPG	Pseudo-Random Pattern Generator
RMSE	Root Mean Square Error

SC	Scan Chain
SDF	Standard Delay Format
SLFSR	Standard Linear Feedback Shift Register
SPE	Synergistic Processing Element
SSA	Single Stuck-At (Fault)
STA	Static Timing Analysis
STUMPS	Multiple-input signature register and Parallel Shift-register sequence generator
TPG	Test Pattern Generator
TRE	Test Response Evaluator
VCD	Value Change Dump
VLSI	Very-Large Scale Integration
XOR	Exclusive-Or

References

- Jais Abraham, Uday Goel, and Arun Kumar. Multi-cycle sensitizable transition delay faults. In *24th IEEE VLSI Test Symposium (VTS), 30 April - 4 May 2006, Berkeley, California, USA*, pages 306–313, 2006.
URL <http://doi.ieeeecomputersociety.org/10.1109/VTS.2006.49>.
- Amit Agarwal, Chris H. Kim, Saibal Mukhopadhyay, and Kaushik Roy. Leakage in nano-scale technologies: mechanisms, impact and design considerations. In *41st ACM/IEEE Design Automation Conference (DAC), San Diego, CA, USA, June 7-11, 2004*, pages 6–11, 2004.
URL <http://doi.acm.org/10.1145/996566.996571>.
- V. K. Agarwal and E. Cerny. Store and generate built-in-testing approach. In *International Symposium on Fault-Tolerant Computing (FTCS), June 24-26, 1981, Portland, Maine*, pages 35–40, 1981.
- Nisar Ahmed and Mohammad Tehranipoor. Improving transition delay fault coverage using hybrid scan-based technique. In *20th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT), 3-5 October 2005, Monterey, CA, USA*, pages 187–198, 2005.
URL <http://doi.ieeeecomputersociety.org/10.1109/DFTVS.2005.69>.
- Uranmandakh Amgalan, Christian Hachmann, Sybille Hellebrand, and Hans-Joachim Wunderlich. Signature rollback - a technique for testing robust circuits. In *26th IEEE VLSI Test Symposium (VTS), April 27 - May 1, 2008, San Diego, California, USA*, pages 125–130, 2008.
URL <http://doi.ieeeecomputersociety.org/10.1109/VTS.2008.34>.
- Kurt Antreich and Michael H. Schulz. Accelerated fault simulation and fault grading in combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 6(5):704–712, 1987.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=28443&arnumber=1270316&count=19&index=1.
- Paul H. Bardell. Design considerations for parallel pseudorandom pattern generators. *Journal of Electronic Testing - Theory and Applications (JETTA)*, 1(1):73–87,

1990.
URL <http://dx.doi.org/10.1007/BF00134016>.
- Paul H. Bardell and William H. McAnney. Self-testing of multi-chip logic modules. In *IEEE International Test Conference (ITC), Philadelphia, PA, Nov. 15-18, 1982*, pages 200–204, 1982.
- Paul H. Bardell, William H. McAnney, and Jacob Savir. *Built-In Test for VLSI: Pseudorandom Techniques*. John Wiley & Sons, 1987.
- Zeev Barzilai and Barry K. Rosen. Comparison of AC self-testing procedures. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, October 1983*, pages 89–94, 1983.
- Bhargab B. Bhattacharya, Sharad C. Seth, and Sheng Zhang. Double-tree scan: A novel low-power scan-path architecture. In *IEEE International Test Conference (ITC), 28 September - 3 October 2003, Charlotte, NC, USA*, pages 470–479, 2003.
URL <http://csdl.computer.org/comp/proceedings/itc/2003/2063/00/20630470abs.htm>.
- Yannick Bonhomme, Patrick Girard, Loïs Guiller, Christian Landrault, and Serge Pravossoudovitch. A gated clock scheme for low power scan testing of logic ICs or embedded cores. In *10th Asian Test Symposium (ATS), 19-21 November 2001, Kyoto, Japan*, pages 253–258, 2001.
URL <http://csdl.computer.org/comp/proceedings/ats/2001/1378/00/13780253abs.htm>.
- Yannick Bonhomme, Patrick Girard, Loïs Guiller, Christian Landrault, and Serge Pravossoudovitch. Efficient scan chain design for power minimization during scan testing under routing constraint. In *IEEE International Test Conference (ITC), 28 September - 3 October 2003, Charlotte, NC, USA*, pages 488–493, 2003.
URL <http://csdl.computer.org/comp/proceedings/itc/2003/2063/00/20630488abs.htm>.
- Yannick Bonhomme, Patrick Girard, Loïs Guiller, Christian Landrault, Serge Pravossoudovitch, and Arnaud Virazel. Design of routing-constrained low power scan chains. In *Design, Automation and Test in Europe Conference and Exposition (DATE), 16-20 February 2004, Paris, France*, pages 62–67, 2004.
URL <http://csdl.computer.org/comp/proceedings/date/2004/2085/01/208510062abs.htm>.
- Alberto Bosio, Luigi Dilillo, Patrick Girard, Aida Todri, and Arnaud Virazel. Why and how controlling power consumption during test: A survey. In *21st IEEE Asian Test Symposium (ATS), Niigata, Japan, November 19-22, 2012*, pages 221–226, 2012.
URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2012.30>.
- Franc Brglez, Philip Pownall, and Robert Hum. Applications of testability analysis: From ATPG to critical delay path tracing. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, October 1984*, pages 705–712, 1984.

- Franc Brglez, David Bryan, and Krzysztof Kozminski. Combinational profiles of sequential benchmark circuits. In *IEEE International Symposium on Circuits and Systems (ISCAS), May 8 - May 11, Portland, OR, USA*, volume 3, pages 1929–1934, 1989.
- Orest Bula, John Moser, Jeanne Trinko, Mark Weissman, and Frank Woytowich. Gross delay defect evaluation for a CMOS logic design system product. *IBM Journal of Research and Development*, 34(2-3):325–338, 1990.
- Michael L. Bushnell and Vishwani D. Agrawal. *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Springer Netherlands, 2000. ISBN 0792379918.
- K. Chakravadhanula, V. Chickermane, B. Keller, P. Gallagher, and P. Narang. Capture power reduction using clock gating aware test generation. In *IEEE International Test Conference (ITC), Austin, TX, USA, November 1-6, 2009*, pages 1–9, 2009.
- Pak K. Chan, M.D.F. Schlag, and J.Y. Zien. Spectral k-way ratio-cut partitioning and clustering. In *30th Conference on Design Automation (DAC), Dallas, Texas, USA, 14-18 June, 1993*.
- Chih-Ang Chen and Sandeep K. Gupta. A methodology to design efficient BIST test pattern generators. In *IEEE International Test Conference (ITC), Washington, DC, USA, October 21-25, 1995*, pages 814–823, 1995.
- Richard M. Chou, Kewal K. Saluja, and Vishwani D. Agrawal. Scheduling tests for VLSI systems under power constraints. *IEEE Transactions on VLSI Systems (TVLSI)*, 5(2):175–185, 1997.
URL <http://doi.ieeeecomputersociety.org/10.1109/92.585217>.
- Wai Chung, T. Lo, and M. Sachdev. A comparative analysis of low-power low-voltage dual-edge-triggered flip-flops. *IEEE Transactions on VLSI Systems (TVLSI)*, 10(6):913–918, 2002.
URL <http://doi.ieeeecomputersociety.org/10.1109/TVLSI.2002.808429>.
- Olivier Coudert. On solving covering problems. In *33rd ACM/IEEE Design Automation Conference (DAC), Las Vegas, Nevada, USA, June 3-7, 1996*, pages 197–202, 1996.
- Alexander Czutro, Matthias Sauer, Iliia Polian, and Bernd Becker. Multi-conditional SAT-ATPG for power-droop testing. In *17th IEEE European Test Symposium (ETS), May 28th - June 1st 2012, Annecy, France*, pages 1–6, 2012.
URL <http://dx.doi.org/10.1109/ETS.2012.6233026>.
- Dariusz Czysz, Mark Kassab, Xijiang Lin, Grzegorz Mrugalski, Janusz Rajski, and Jerzy Tyszer. Low-power scan operation in test compression environment. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 28(11):1742–1755, 2009.
URL <http://dx.doi.org/10.1109/TCAD.2009.2030445>.

- Vinay Dabholkar, Sreejit Chakravarty, Irith Pomeranz, and Sudhakar M. Reddy. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 17(12):1325–1333, 1998.
URL <http://doi.ieeecomputersociety.org/10.1109/43.736572>.
- Sumit Dasgupta, Ronald G. Walther, Thomas W. Williams, and Edward B. Eichelberger. An enhancement to LSSD and some applications of LSSD in reliability, availability, and serviceability. In *11th Symposium on Fault-Tolerant Computing (FTCS), June 24-26, 1981, Portland, Maine*, pages 32–34, 1981.
- Scott Davidson. ITC'99 benchmark circuits - preliminary results. In *IEEE International Test Conference (ITC), Atlantic City, NJ, USA, 27-30 September 1999*, page 1125, 1999.
- V. R. Devanathan, C. P. Ravikumar, and V. Kamakoti. A stochastic pattern generation and optimization framework for variation-tolerant, power-safe scan test. In *IEEE International Test Conference (ITC), Santa Clara, California, USA, October 21-26, 2007*, pages 1–10, 2007.
URL <http://dx.doi.org/10.1109/TEST.2007.4437596>.
- Rolf Drechsler, Stephan Eggersgluss, Goerschwin Fey, Andreas Glowatz, Friedrich Hapke, Juergen Schloeffel, and Daniel Tille. On acceleration of SAT-based ATPG for industrial designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 27(7):1329–1333, 2008. ISSN 0278-0070.
- Edward B. Eichelberger and Thomas W. Williams. A logic design structure for LSI testability. In *14th ACM/IEEE Design Automation Conference (DAC), New Orleans, Louisiana, USA, June 20-22, 1977*, pages 462–468. IEEE Press, 1977.
- Melanie Elm and Hans-Joachim Wunderlich. BISD: Scan-based built-in self-diagnosis. In *Design, Automation and Test in Europe (DATE), Dresden, Germany, March 8-12, 2010*, pages 1243–1248, 2010.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?arnumber=5456997.
- Mohammed ElShoukry, Mohammad Tehranipoor, and C. P. Ravikumar. A critical-path-aware partial gating approach for test power reduction. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 12(2), 2007.
URL <http://doi.acm.org/10.1145/1230800.1230809>.
- Piet Engelke, Bernd Becker, Michel Renovell, Jürgen Schlöffel, Bettina Braitling, and Ilia Polian. SUPERB: Simulator utilizing parallel evaluation of resistive bridges. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 14(4), 2009.
URL <http://doi.acm.org/10.1145/1562514.1596831>.
- C.M. Fiduccia and R.M. Mattheyses. A linear-time heuristic for improving network partitions. In *19th Conference on Design Automation (DAC), Las Vegas, Nevada, USA, June 14-16, 1982*, pages 175–181, 1982.

- Robert A. Frohwerk. Signature analysis: A new digital field service method. *Hewlett-Packard Journal*, 28(9):2–8, 1977.
- J. Michael Galey, Ruth E. Norby, and J. Paul Roth. Techniques for the diagnosis of switching circuit failures. In *2nd Symposium on Switching Circuit Theory and Logical Design, 17-20 October, 1961, Detroit, Michigan, USA*, pages 152–160, 1961.
- Michael R. Garey and David S. Johnson. *Computers and intractability*. Freeman San Francisco, 1979.
- Manfred Geilert, Jürgen Alt, and Michael Zimmermann. On the efficiency of the transition fault model for delay faults. In *IEEE/ACM International Conference on Computer-Aided Design (ICCAD), November 11-15, 1990, Santa Clara, CA, USA*, pages 272–275, 1990.
- Stefan Gerstendörfer and Hans-Joachim Wunderlich. Minimized power consumption for scan-based BIST. In *IEEE International Test Conference (ITC), Atlantic City, NJ, USA, Sep. 28-30, 1999*, 1999.
- Stefan Gerstendörfer and Hans-Joachim Wunderlich. Minimized power consumption for scan-based BIST. *Journal of Electronic Testing - Theory and Applications (JETTA)*, 16(3):203–212, 2000.
- Patrick Girard, Loïs Guiller, Christian Landrault, and Serge Pravossoudovitch. A test vector inhibiting technique for low energy BIST design. In *17th IEEE VLSI Test Symposium (VTS), 25-30 April 1999, San Diego, CA, USA*, pages 407–412, 1999.
URL <http://csdl.computer.org/comp/proceedings/vts/1999/0146/00/01460407abs.htm>.
- Patrick Girard, Loïs Guiller, Christian Landrault, and Serge Pravossoudovitch. An adjacency-based test pattern generator for low power BIST design. In *9th Asian Test Symposium (ATS), 4-6 December 2000, Taipei, Taiwan*, page 459, 2000a.
- Patrick Girard, Christian Landrault, Loïs Guiller, and Serge Pravossoudovitch. Low power BIST design by hypergraph partitioning: methodology and architectures. In *IEEE International Test Conference (ITC), Atlantic City, NJ, USA, October 2000*, pages 652–661, 2000b.
- Patrick Girard, Loïs Guiller, Christian Landrault, Serge Pravossoudovitch, and Hans-Joachim Wunderlich. A modified clock scheme for a low power BIST test pattern generator. In *19th IEEE VLSI Test Symposium (VTS), 29 April - 3 May 2001, Marina Del Rey, CA, USA*, pages 306–311, 2001.
URL <http://csdl.computer.org/comp/proceedings/vts/2001/1122/00/11220306abs.htm>.
- Patrick Girard, Nicola Nicolici, and Xiaoqing Wen. *Power-Aware Testing and Test Strategies for Low Power Devices*. Springer Verlag, 2009. ISBN 1441909273.
- Solomon W. Golomb. *Shift Register Sequences*. Aegean Park Press Laguna Hills, CA, USA, 1981. ISBN 0894120484.

- Rajesh Gupta and Melvin A. Breuer. BALLAST: a methodology for partial scan design. In *19th International Symposium on Fault-Tolerant Computing (FTCS)*, 21-23 June, 1989, Chicago, IL, USA, pages 118–125, 1989.
- Sandeep K. Gupta and Dhiraj K. Pradhan. A new framework for designing and analyzing BIST techniques: Computation of exact aliasing probability. In *IEEE International Test Conference (ITC)*, Washington, D.C., USA, September 1988, pages 329–342, 1988.
- Sanjay Gupta, Janusz Rajski, and Jerzy Tyszer. Arithmetic additive generators of pseudo-exhaustive test patterns. *IEEE Transactions on Computers (TC)*, 45(8): 939–949, 1996.
- Abdul-Wahid Hakmi, Hans-Joachim Wunderlich, Christian G. Zoellin, Andreas Glowatz, Friedrich Hapke, Juergen Schloeffel, and Laurent Souef. Programmable deterministic built-in self-test. In *IEEE International Test Conference, Santa Clara, CA, USA, 2007*.
- Ilker Hamzaoglu and Janak H. Patel. New techniques for deterministic test pattern generation. In *16th IEEE VLSI Test Symposium (VTS)*, 28 April - 1 May 1998, Princeton, NJ, USA, pages 446–452, 1998.
- Ilker Hamzaoglu and Janak H. Patel. Reducing test application time for full scan embedded cores. In *International Symposium on Fault-Tolerant Computing (FTCS)*, June 15-18, 1999, Madison, Wisconsin, USA, pages 260–267, 1999.
URL <http://computer.org/proceedings/ftcs/0213/02130267abs.htm>.
- John P. Hayes. Transition count testing of combinational logic circuits. *IEEE Transactions on Computers (TC)*, 25(6):613–620, 1976.
- Sybille Hellebrand and Marc Hunger. Are robust circuits really robust? In *24th IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT)*, 7-9 October 2009, Chicago, Illinois, USA, pages 77–77, 2009.
URL <http://doi.ieeecomputersociety.org/10.1109/DFT.2009.28>.
- Sybille Hellebrand and Hans-Joachim Wunderlich. The pseudo-exhaustive test of sequential circuits. In *IEEE International Test Conference (ITC)*, Washington, D.C., USA, August 1989, pages 19–27, 1989.
- Sybille Hellebrand, Hans-Joachim Wunderlich, and Huaguo Liang. A mixed mode BIST scheme based on reseeding of folding counters. In *IEEE International Test Conference (ITC)*, Atlantic City, NJ, USA, October 2000, pages 778–784, 2000.
- Sybille Hellebrand, Christian G. Zoellin, Hans-Joachim Wunderlich, Stefan Ludwig, Torsten Coym, and Bernd Straube. A refined electrical model for particle strikes and its impact on SEU prediction. In *22nd IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT)*, 26-28 September 2007, Rome, Italy, pages 50–58, 2007.
URL <http://doi.ieeecomputersociety.org/10.1109/DFT.2007.43>.
- Keerthi Heragu, Janak H. Patel, and Vishwani D. Agrawal. Segment delay faults: a new fault model. In *14th IEEE VLSI Test Symposium (VTS)*, April 28 - May 1,

- 1996, Princeton, NJ, USA, pages 32–41, 1996.
 URL <http://csdl.computer.org/comp/proceedings/vts/1996/7304/00/73040032abs.htm>.
- Andre Hertwig and Hans-Joachim Wunderlich. Low power serial built-in self-test. In *IEEE European Test Workshop (ETW), Sitges, Barcelona, Spain, May 27 - 29, 1998*, pages 49–53, 1998.
- Graham Hetherington, Tony Fryars, Nagesh Tamarapalli, Mark Kassab, Abu S. M. Hassan, and Janusz Rajski. Logic BIST for large industrial designs: real issues and case studies. In *IEEE International Test Conference (ITC), Atlantic City, NJ, USA, 27-30 September 1999*, pages 358–367, 1999.
- Stefan Holst and Hans-Joachim Wunderlich. Adaptive debug and diagnosis without fault dictionaries. *Journal of Electronic Testing - Theory and Applications (JETTA)*, 25(4-5):259–268, 2009.
 URL <http://dx.doi.org/10.1007/s10836-009-5109-3>.
- Stefan Holst, Eric Schneider, and Hans-Joachim Wunderlich. Scan test power simulation on GPGPUs. In *21st IEEE Asian Test Symposium (ATS), Niigata, Japan, November 19-22, 2012*, pages 155–160, 2012.
 URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2012.23>.
- Peter D. Hortensius, Robert D. McLeod, and Howard C. Card. Parallel random number generation for VLSI systems using cellular automata. *IEEE Transactions on Computers (TC)*, 38(10):1466–1473, 1989.
- Michael S. Hsiao, Elizabeth M. Rudnick, and Janak H. Patel. Peak power estimation of VLSI circuits: new peak power measures. *IEEE Transactions on VLSI Systems (TVLSI)*, 8(4):435–439, 2000.
 URL <http://doi.ieeecomputersociety.org/10.1109/92.863624>.
- Yu Huang, Sudhakar M. Reddy, Wu-Tung Cheng, Paul Reuter, Nilanjan Mukherjee, Chien-Chung Tsai, Omer Samman, and Yahya Zaidan. Optimal core wrapper width selection and SOC test scheduling based on 3-D bin packing algorithm. In *IEEE International Test Conference (ITC), Baltimore, MD, USA, October 7-10, 2002*, pages 74–82, 2002.
 URL <http://csdl.computer.org/comp/proceedings/itc/2002/7543/00/75430074abs.htm>.
- Joseph L. A. Hughes and Edward J. McCluskey. An analysis of the multiple fault detection capabilities of single stuck-at fault test sets. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, October 1984*, pages 52–58, 1984.
- Oscar H. Ibarra and Sartaj Sahni. Polynomially complete fault detection problems. *IEEE Transactions on Computers (TC)*, 24(3):242–249, 1975.
- Masahiro Ishida, Toru Nakura, Toshiyuki Kikkawa, Takashi Kusaka, Satoshi Komatsu, and Kunihiro Asada. Power integrity control of ATE for emulating power supply fluctuations on customer environment. In *IEEE International Test Conference (ITC), Anaheim, CA, USA, November 5-8, 2012*, pages 1–10, 2012.
 URL <http://doi.ieeecomputersociety.org/10.1109/TEST.2012.>

6401553.

- Vijay S. Iyengar, Barry K. Rosen, and John A. Waicukauski. On computing the sizes of detected delay faults. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 9(3):299–312, 1990.
URL <http://doi.ieeecomputersociety.org/10.1109/43.46805>.
- Vikram Iyengar and Krishnendu Chakrabarty. Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip. In *19th IEEE VLSI Test Symposium (VTS), 29 April - 3 May 2001, Marina Del Rey, CA, USA*, pages 368–374, 2001.
URL <http://csdl.computer.org/comp/proceedings/vts/2001/1122/00/11220368abs.htm>.
- George Karypis and Vipin Kumar. Multilevel k-way hypergraph partitioning. In *36th Conference on Design Automation (DAC), New Orleans, LA, USA, June 21-25, 1999*, pages 343–348, 1999.
URL <http://doi.acm.org/10.1145/309847.309954>.
- B. W. Kernighan and S. Lin. An efficient heuristic procedure for partitioning graphs. *Bell System Technical Journal, February*, 49(2):291–307, 1970.
- Suhwan Kim, Stephen V. Kosonocky, Daniel R. Knebel, and Kevin Stawiasz. Experimental measurement of a novel power gating structure with intermediate power saving mode. In *International Symposium on Low Power Electronics and Design (ISLPED), August 9-11, 2004, Newport Beach, California, USA*, pages 20–25, 2004.
- Michael A. Kochte, Christian G. Zoellin, Michael E. Imhof, and Hans-Joachim Wunderlich. Test set stripping limiting the maximum number of specified bits. In *4th IEEE International Symposium on Electronic Design, Test and Applications (DELTA), Hong Kong*, pages 581–586, 2008.
URL <http://doi.ieeecomputersociety.org/10.1109/DELTA.2008.64>.
- Michael A. Kochte, Marcel Schaal, Hans-Joachim Wunderlich, and Christian G. Zoellin. Efficient fault simulation on many-core processors. In *47th ACM/IEEE Design Automation Conference (DAC), Anaheim, California, USA, July 13-18, 2010*, pages 380–385, 2010a.
URL <http://doi.acm.org/10.1145/1837274.1837369>.
- Michael A. Kochte, Christian G. Zoellin, Rafal Baranowski, Michael E. Imhof, Hans-Joachim Wunderlich, Nadereh Hatami, Stefano Di Carlo, and Paolo Prinetto. Efficient simulation of structural faults for the reliability evaluation at system-level. In *IEEE 19th Asian Test Symposium (ATS), Shanghai, China, December 1-4, 2010*, 2010b.
- Michael A. Kochte, Kohei Miyase, Xiaoqing Wen, Seiji Kajihara, Yuta Yamato, Kazunari Enokimoto, and Hans-Joachim Wunderlich. SAT-based capture-power reduction for at-speed broadcast-scan-based test compression architectures. In *International Symposium on Low Power Electronics and Design (ISLPED), 2011, Fukuoka, Japan, August 1-3, 2011*, pages 33–38, 2011.
URL <http://portal.acm.org/citation.cfm?id=2016812&CFID=>

- 34981777-{}&}CFTOKEN=25607807.
- Bernd Koenemann. LFSR-coded test patterns for scan designs. In *2nd European Test Conference*, pages 237–242, 1991.
- Aman Kokrady and C. P. Ravikumar. Fast, layout-aware validation of test-vectors for nanometer-related timing failures. In *17th International Conference on VLSI Design, 5-9 January 2004, Mumbai, India*, pages 597–, 2004.
URL <http://doi.ieeecomputersociety.org/10.1109/ICVD.2004.1260984>.
- B. Könemann, J. Mucha, and G. Zwiehoff. Built-in logic block observation technique. In *IEEE International Test Conference (ITC)*, pages 37–41, 1979.
- Andrzej Krasniewski and Slawomir Pilarski. Circular self-test path: A low-cost BIST technique. In *24th ACM/IEEE Design Automation Conference (DAC), Miami Beach, FL, June 28 - July 1, 1987*, pages 407–415, 1987.
URL <http://doi.acm.org/10.1145/37888.37949>.
- Arno Kunzmann and Hans-Joachim Wunderlich. An analytical approach to the partial scan problem. *Journal of Electronic Testing - Theory and Applications (JETTA)*, 1(2):163–174, 1990.
- Liyang Lai, Janak H. Patel, Thomas Rinderknecht, and Wu-Tung Cheng. Logic BIST with scan chain segmentation. In *IEEE International Test Conference (ITC), October 26-28, 2004, Charlotte, NC, USA*, pages 57–66, 2004.
URL <http://doi.ieeecomputersociety.org/10.1109/ITC.2004.115>.
- Erik Larsson and Zebo Peng. Test scheduling and scan-chain division under power constraint. In *10th Asian Test Symposium (ATS), 19-21 November 2001, Kyoto, Japan*, pages 259–264, 2001.
URL <http://csdl.computer.org/comp/proceedings/ats/2001/1378/00/13780259abs.htm>.
- Erik Larsson and Zebo Peng. Power-aware test planning in the early system-on-chip design exploration process. *IEEE Transactions on Computers*, 55(2):227–239, 2006.
URL <http://doi.ieeecomputersociety.org/10.1109/TC.2006.28>.
- Jeremy Lee and Mohammad Tehranipoor. LS-TDF: Low-switching transition delay fault pattern generation. In *26th IEEE VLSI Test Symposium (VTS), April 27 - May 1, 2008, San Diego, California, USA*, pages 227–232, 2008.
URL <http://doi.ieeecomputersociety.org/10.1109/VTS.2008.48>.
- Ytzhak Levendel and Premachandran R. Menon. Transition Faults in Combinational Circuits: Input Transition Test Generation and Fault Simulation. In *16th International Fault Tolerant Computing Symposium (FTCS)*, pages 278–283, 1986.
- Chin Jen Lin and Sudhakar M. Reddy. On delay fault testing in logic circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 6(5):694–703, 1987.
URL http://ieeexplore.ieee.org/xpls/abs_all.jsp?isnumber=28443&arnumber=1270315&count=19&index=0.

- Xijiang Lin. Power supply droop and its impacts on structural at-speed testing. In *21st IEEE Asian Test Symposium (ATS), Niigata, Japan, November 19-22, 2012*, pages 239–244, 2012.
URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2012.63>.
- Yi-Tsung Lin, Jiun-Lang Huang, and Xiaoqing Wen. A transition isolation scan cell design for low shift and capture power. In *21st IEEE Asian Test Symposium (ATS), Niigata, Japan, November 19-22, 2012*, pages 107–112, 2012.
URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2012.29>.
- Jing-Jia Liou, Angela Krstic, Li-C. Wang, and Kwang-Ting Cheng. False-path-aware statistical timing analysis and efficient path selection for delay testing and timing validation. In *39th ACM/IEEE Design Automation Conference (DAC), New Orleans, LA, USA, June 10-14, 2002*, pages 566–569, 2002.
URL <http://doi.acm.org/10.1145/513918.514061>.
- Robert Madge, Brady Benware, and W. Robert Daasch. Obtaining high defect coverage for frequency-dependent defects in complex ASICs. *IEEE Design & Test of Computers*, 20(5):46–53, 2003.
URL <http://csdl.computer.org/comp/mags/dt/2003/05/d5046abs.htm>.
- Edward J. McCluskey. Iterative combinational switching networks—general design considerations. *IRE Transactions on Electronic Computers*, 7:285–291, 1958.
- Edward J. McCluskey and Saied Bozorgui-Nesbat. Design for autonomous test. *IEEE Transactions on Computers (TC)*, 30(11):866–875, 1981.
- Kohei Miyase and Seiji Kajihara. XID: Don’t care identification of test patterns for combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 23(2):321–326, 2004.
URL <http://doi.ieeecomputersociety.org/10.1109/TCAD.2003.822103>.
- Kohei Miyase, Xiaoqing Wen, Hiroshi Furukawa, Yuta Yamato, Seiji Kajihara, Patrick Girard, Laung-Terng Wang, and Mohammad Tehranipoor. High launch switching activity reduction in at-speed scan testing using CTX: A clock-gating-based test relaxation and x-filling scheme. *IEICE Transactions*, 93-D(1):2–9, 2010.
URL http://search.ieice.org/bin/summary.php?id=e93-d_1_2.
- Kohei Miyase, Matthias Sauer, Bernd Becker, Xiaoqing Wen, and Seiji Kajihara. Search space reduction for low-power test generation. In *22nd Asian Test Symposium (ATS), Yilan County, Taiwan, November 18-21, 2013*, pages 171–176, 2013.
URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2013.40>.
- Grzegorz Mrugalski, Janusz Rajski, Dariusz Czysz, and Jerzy Tyszer. New test data decompressor for low power applications. In *44th ACM/IEEE Design Automation Conference (DAC), San Diego, CA, USA, June 4-8, 2007*, pages 539–544, 2007.
URL <http://doi.ieeecomputersociety.org/10.1109/DAC.2007>.

- 375223.
- Sridhar Narayanan, Charles A. Njinda, Rajesh Gupta, and Melvin A. Breuer. SIESTA: a multi-facet scan design system. In *European Conference on Design Automation, March 16-19, 1992, Brussels, Belgium*, pages 246–251. IEEE Computer Society Press Los Alamitos, CA, USA, 1992.
- Wolfgang Nebel and Jean P. Mermet. *Low power design in deep submicron electronics*. Springer, 1997. ISBN 079234569X.
- Nicola Nicolici, Bashir M Al-Hashimi, and Alan C Williams. Minimisation of power dissipation during test application in full-scan sequential circuits using primary input freezing. *IEE Proceedings-Computers and Digital Techniques*, 147(5): 313–322, 2000.
- Nitin Parimi and Xiaoling Sun. Toggle-masking for test-per-scan VLSI circuits. In *19th IEEE International Symposium on Defect and Fault-Tolerance in VLSI Systems (DFT), 10-13 October 2004, Cannes, France*, pages 332–338, 2004.
URL <http://csdl.computer.org/comp/proceedings/dft/2004/2241/00/22410332abs.htm>.
- Eun Sei Park and M. Ray Mercer. Robust and nonrobust tests for path delay faults in a combinational circuit. In *IEEE International Test Conference (ITC), Washington, D.C., USA, pages 1027–1034, 1987*.
- Intaik Park and Edward J. McCluskey. Launch-on-shift-capture transition tests. In *IEEE International Test Conference (ITC08), October 26 - 31, 2008, Santa Clara, CA*, page 35.3, 2008.
- Janak H. Patel. Stuck-at fault: a fault model for the next millennium. In *IEEE International Test Conference (ITC), Washington, DC, USA, October 18-22, 1998*, page 1166, 1998.
URL <http://www.computer.org/proceedings/itc/5093/50931166.pdf>.
- Dac C. Pham, Tony Aipperspach, David Boerstler, Mark Bolliger, Rajat Chaudhry, Dennis Cox, Paul Harvey, Paul M. Harvey, H. Peter Hofstee, Charles Johns, and Jim Kahle. Overview of the architecture, circuit design, and physical implementation of a first-generation Cell processor. *IEEE Journal of Solid-State Circuits (JSSC)*, 41(1):179–196, 2006.
- Julien Pouget, Erik Larsson, and Zebo Peng. Multiple-constraint driven system-on-chip test time optimization. *Journal of Electronic Testing - Theory and Applications (JETTA)*, 21(6):599–611, 2005.
URL <http://dx.doi.org/10.1007/s10836-005-2911-4>.
- Ankan K. Pramanick and Sudhakar M. Reddy. On the detection of delay faults. In *IEEE International Test Conference (ITC), Washington, D.C., USA, September 1988*, pages 845–856, 1988.
- Ankan K. Pramanick and Sudhakar M. Reddy. On the fault coverage of gate delay fault detecting tests. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 16(1):78–94, 1997.

- URL <http://doi.ieeecomputersociety.org/10.1109/43.559333>.
- Jan M. Rabaey, Anantha P. Chandrakasan, and Borivoje Nikolić. *Digital integrated circuits: a design perspective*. Prentice Hall, 1996. ISBN 0131786091.
- Janusz Rajski and Jerzy Tyszer. Design of phase shifters for BIST applications. In *16th IEEE VLSI Test Symposium (VTS), 28 April - 1 May 1998, Princeton, NJ, USA*, pages 218–224, 1998.
URL <http://csdl2.computer.org/dl/proceedings/vts/1998/8436/00/84360218.pdf>.
- Janusz Rajski, Jerzy Tyszer, Grzegorz Mrugalski, and Benoit Nadeau-Dostie. Test generator with preselected toggling for low power built-in self-test. In *30th IEEE VLSI Test Symposium (VTS), Maui, Hawaii, USA, 23-26 April 2012*, pages 1–6, 2012.
URL <http://dx.doi.org/10.1109/VTS.2012.6231071>.
- Srivaths Ravi, V. R. Devanathan, and Rubin A. Parekhji. Methodology for low power test pattern generation using activity threshold control logic. In *International Conference on Computer-Aided Design (ICCAD), November 5-8, 2007, San Jose, CA, USA*, pages 526–529, 2007.
URL <http://doi.acm.org/10.1145/1326073.1326181>.
- Santiago Remersaro, Xijiang Lin, Z. Zhang, Sudhakar M. Reddy, Irith Pomeranz, and Janusz Rajski. Preferred fill: A scalable method to reduce capture power for scan based designs. In *IEEE International Test Conference (ITC), Santa Clara, CA, USA, October 22-27, 2006*, pages 1–10. IEEE, 2006.
URL <http://dx.doi.org/10.1109/TEST.2006.297694>.
- Santiago Remersaro, Xijiang Lin, Sudhakar M. Reddy, Irith Pomeranz, and Janusz Rajski. Low shift and capture power scan tests. In *20th International Conference on VLSI Design, 6-10 January 2007, Bangalore, India*, pages 793–798, 2007.
URL <http://doi.ieeecomputersociety.org/10.1109/VLSID.2007.101>.
- Michel Renovell, P. Huc, and Yves Bertrand. CMOS bridging fault modeling. In *12th IEEE VLSI Test Symposium (VTS), April 25-28, 1994, Cherry Hill, New Jersey, USA*, pages 392–397, 1994. doi: 10.1109/VTEST.1994.292283.
- Michel Renovell, P. Huc, and Yves Bertrand. The concept of resistance interval: a new parametric model for realistic resistive bridging fault. In *13th IEEE VLSI Test Symposium (VTS), April 30 - May 3, 1995, Princeton, New Jersey, USA*, pages 184–189, 1995.
URL <http://csdl.computer.org/comp/proceedings/vts/1995/7000/00/70000184abs.htm>.
- M. Riley, L. Bushard, N. Chelstrom, N. Kiryu, and S. Ferguson. Testability features of the first-generation Cell processor. In *IEEE International Test Conference (ITC)*, pages 111–119, 2005.
- Paul M. Rosinger, Bashir M. Al-Hashimi, and Nicola Nicolici. Scan architecture with mutually exclusive scan segment activation for shift- and capture-power

- reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 23(7):1142–1153, 2004.
 URL <http://doi.ieeecomputersociety.org/10.1109/TCAD.2004.829797>.
- Paul M. Rosinger, Bashir M. Al-Hashimi, and Krishnendu Chakrabarty. Rapid generation of thermal-safe test schedules. In *Design, Automation and Test in Europe Conference and Exposition (DATE), 7-11 March 2005, Munich, Germany*, pages 840–845, 2005.
 URL <http://doi.ieeecomputersociety.org/10.1109/DATE.2005.252>.
- Kewal K. Saluja and Mark G. Karpovsky. Testing computer hardware through data compression in space and time. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, October 1983*, pages 83–88, 1983.
- Soheil Samii, Erik Larsson, Krishnendu Chakrabarty, and Zebo Peng. Cycle-accurate test power modeling and its application to SOC test scheduling. In *IEEE International Test Conference (ITC), Santa Clara, CA, USA, Oct. 24-26, 2006*, pages 1–10, 2006.
- Ranganathan Sankaralingam and Nur A. Touba. Reducing test power during test using programmable scan chain disable. In *1st IEEE International Workshop on Electronic Design, Test and Applications (DELTA), 29-31 January 2002, Christchurch, New Zealand*, pages 159–166, 2002.
 URL <http://csdl.computer.org/comp/proceedings/delta/2002/1453/00/14530159abs.htm>.
- Ranganathan Sankaralingam, Rama Rao Oruganti, and Nur A. Touba. Static compaction techniques to control scan vector power dissipation. In *18th IEEE VLSI Test Symposium (VTS), 30 April - 4 May 2000, Montreal, Canada*, pages 35–42, 2000.
 URL <http://csdl.computer.org/comp/proceedings/vts/2000/0613/00/06130035abs.htm>.
- Yasuo Sato, Senling Wang, Takaaki Kato, Kohei Miyase, and Seiji Kajihara. Low power BIST for scan-shift and capture power. In *21st IEEE Asian Test Symposium (ATS), Niigata, Japan, November 19-22, 2012*, pages 173–178, 2012.
 URL <http://doi.ieeecomputersociety.org/10.1109/ATS.2012.27>.
- Jacob Savir. Skewed-load transition test: Part 1, calculus. In *IEEE International Test Conference (ITC), Baltimore, Maryland, USA, September 20-24, 1992*, pages 705–713, 1992.
- Jacob Savir and Paul H. Bardell. On random pattern test length. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, October 1983*, pages 95–107, 1983.
- Jacob Savir and Srinivas Patil. Broad-side delay test. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 13(8): 1057–1064, 1994.
 URL <http://doi.ieeecomputersociety.org/10.1109/43.298042>.

- Michael H. Schulz. *Testmuster-generierung und Fehlersimulation in digitalen Schaltungen mit hoher Komplexität*. Springer, 1988. ISBN 3540500510.
- Anuja Sehgal, Sudarshan Bahukudumbi, and Krishnendu Chakrabarty. Power-aware soc test planning for effective utilization of port-scalable testers. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 13(3), 2008.
URL <http://doi.acm.org/10.1145/1367045.1367062>.
- Micaela Serra, Terry Slater, Jon C. Muzio, and D. Michael Miller. The analysis of one-dimensional linear cellular automata and their aliasing properties. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 9(7):767–778, 1990.
URL <http://doi.ieeecomputersociety.org/10.1109/43.55213>.
- Sharad C. Seth, Vishwani D. Agrawal, and Hassan Farhat. A statistical theory of digital circuit testability. *IEEE Transactions on Computers (TC)*, 39(4):582–586, 1990.
- Ozgur Sinanoglu and Alex Orailoglu. Space and time compaction schemes for embedded cores. In *IEEE International Test Conference (ITC), Baltimore, MD, USA, 30 October - 1 November 2001*, pages 521–529, 2001.
- Ozgur Sinanoglu, Ismet Bayraktaroglu, and Alex Orailoglu. Test power reduction through minimization of scan chain transitions. In *20th IEEE VLSI Test Symposium (VTS), 28 April - 2 May 2002, Monterey, CA, USA*, pages 166–172, 2002.
URL <http://csdl.computer.org/comp/proceedings/vts/2002/1570/00/15700166abs.htm>.
- Gordon L. Smith. Model for delay faults based upon paths. In *IEEE International Test Conference (ITC), Philadelphia, PA, USA, November 1985*, pages 342–351, 1985.
- Jedrzej Solecki, Jerzy Tyszer, Grzegorz Mrugalski, Nilanjan Mukherjee, and Janusz Rajski. Low power programmable PRPG with enhanced fault coverage gradient. In *IEEE International Test Conference (ITC), Anaheim, CA, USA, November 5-8, 2012*, pages 1–9, 2012.
URL <http://doi.ieeecomputersociety.org/10.1109/TEST.2012.6401559>.
- DongSup Song, Jin-Ho Ahn, Taejin Kim, and Sungho Kang. MTR-fill: A simulated annealing-based x-filling technique to reduce test power dissipation for scan-based designs. *IEICE Transactions*, 91-D(4):1197–1200, 2008.
URL <http://dx.doi.org/10.1093/ietisy/e91-d.4.1197>.
- Ashish Srivastava, Dennis Sylvester, and David Blaauw. *Statistical analysis and optimization for VLSI: Timing and power*. Springer Verlag, 2005. ISBN 0387257381.
- Vladimir Stojanovic and Vojin G. Oklobdzija. Comparative analysis of master-slave latches and flip-flops for high-performance and low-power systems. *IEEE Journal of Solid-State Circuits (JSSC)*, 34(4):536–548, 1999.
- Albrecht P. Stroele and Hans-Joachim Wunderlich. Signature analysis and test scheduling for self-testable circuits. In *International Symposium on Fault-*

- Tolerant Computing (FTCS)*, June 25-27, 1991, Montreal, Canada, pages 96–103, 1991.
- Osamu Takahashi, C Adams, D Ault, E Behnen, O Chiang, SR Cottier, P Coulman, J Culp, G Gervais, MS Gray, et al. Migration of cell broadband engine from 65nm SOI to 45nm SOI. In *IEEE International Solid-State Circuits Conference (ISSCC)*, San Francisco, CA, USA, February 3-7, 2008, pages 86–597. IEEE, 2008.
- C. Timoc, M. Buehler, T. Griswold, C. Pina, F. Stott, and L. Hess. Logical models of physical failures. In *IEEE International Test Conference (ITC)*, Philadelphia, PA, USA, October 1983, pages 546–553, 1983.
- Erwin Trischler. Incomplete scan path with an automatic test generation methodology. In *IEEE International Test Conference (ITC)*, pages 153–162, 1980.
- Anis Uzzaman, Brion L. Keller, Thomas J. Snethen, Kazuhiko Iwasaki, and Masayuki Arai. Automatic handling of programmable on-product clock generation (OPCG) circuitry for low power aware delay test. *Journal of Low Power Electronics - Theory and Applications (JETTA)*, 5(4):520–528, 2009.
URL <http://dx.doi.org/10.1166/jolpe.2009.1050>.
- Kenneth D. Wagner. Clock system design. *IEEE Design & Test of Computers*, 5(5): 9–27, 1988.
- John A. Waicukauski, Eric Lindbloom, Vijay S. Iyengar, and Barry K. Rosen. Transition fault simulation by parallel pattern single fault propagation. In *IEEE International Test Conference (ITC)*, Washington, D.C., USA, September 1986, pages 542–551, 1986.
- Laung-Terng Wang and Edward J. McCluskey. Circuits for pseudo-exhaustive test pattern generation. In *IEEE International Test Conference (ITC)*, Washington, D.C., USA, September 1986, pages 25–37, 1986.
- Seongmoon Wang and Sandeep K. Gupta. ATPG for heat dissipation minimization during test application. In *IEEE International Test Conference (ITC)*, Washington, DC, USA, October 2-6, 1994, pages 250–258, 1994.
URL <http://dx.doi.org/10.1109/TEST.1994.527956>.
- Seongmoon Wang and Sandeep K. Gupta. DS-LFSR: A new BIST TPG for low heat dissipation. In *IEEE International Test Conference (ITC)*, Washington, DC, USA, November 3-5, 1997, pages 848–857, 1997.
- Seongmoon Wang and Sandeep K. Gupta. LT-RTPG: a new test-per-scan BIST TPG for low switching activity. *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 25(8):1565–1574, 2006.
URL <http://doi.ieeecomputersociety.org/10.1109/TCAD.2005.855927>.
- Seongmoon Wang, Xiao Liu, and Srimat T. Chakradhar. Hybrid delay scan: A low hardware overhead scan-based delay test technique for high fault coverage and compact test sets. In *Design, Automation and Test in Europe Conference and Exposition (DATE)*, 16-20 February 2004, Paris, France, pages 1296–1301,

2004.
 URL <http://csdl.computer.org/comp/proceedings/date/2004/2085/02/208521296abs.htm>.
- Xiaoqing Wen, Seiji Kajihara, Kohei Miyase, Tatsuya Suzuki, Kewal K. Saluja, Laung-Terng Wang, Khader S. Abdel-Hafez, and Kozo Kinoshita. A new ATPG method for efficient capture power reduction during scan testing. In *24th IEEE VLSI Test Symposium (VTS), 30 April - 4 May 2006, Berkeley, California, USA*, pages 58–65, 2006.
 URL <http://doi.ieeecomputersociety.org/10.1109/VTS.2006.8>.
- Xiaoqing Wen, Kohei Miyase, Seiji Kajihara, Tatsuya Suzuki, Yuta Yamato, Patrick Girard, Yuji Ohsumi, and Laung-Terng Wang. A novel scheme to reduce power supply noise for high-quality at-speed scan testing. In *IEEE International Test Conference (ITC), Santa Clara, California, USA, October 21-26, 2007*, pages 1–10, 2007a.
 URL <http://dx.doi.org/10.1109/TEST.2007.4437632>.
- Xiaoqing Wen, Kohei Miyase, Tatsuya Suzuki, Seiji Kajihara, Yuji Ohsumi, and Kewal K. Saluja. Critical-path-aware x-filling for effective IR-drop reduction in at-speed scan testing. In *44th ACM/IEEE Design Automation Conference (DAC), San Diego, CA, USA, June 4-8, 2007*, pages 527–532, 2007b.
 URL <http://doi.ieeecomputersociety.org/10.1109/DAC.2007.375221>.
- Lee Whetsel. Adapting scan architectures for low power operation. In *IEEE International Test Conference (ITC), Atlantic City, NJ, USA, October 2000*, pages 863–872, 2000.
- Michael J.Y. Williams and James B. Angell. Enhancing testability of large-scale integrated circuits via test points and additional logic. *IEEE Transactions on Computers (TC)*, 22(1):46 – 60, 1973. doi: 10.1109/T-C.1973.223600.
- Tom W. Williams, Wilfried Daehn, Matthias Gruetzner, and Cordt W. Starke. Bounds and analysis of aliasing errors in linear feedback shift registers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems (TCAD)*, 7(1):75–83, 1988.
 URL <http://doi.ieeecomputersociety.org/10.1109/43.3132>.
- Hans-Joachim Wunderlich. PROTEST: a tool for probabilistic testability analysis. In *22nd ACM/IEEE Conference on Design Automation (DAC), Las Vegas, Nevada, USA*, pages 204–211, 1985.
 URL <http://doi.acm.org/10.1145/317825.317858>.
- Hans-Joachim Wunderlich. On computing optimized input probabilities for random tests. In *24th ACM/IEEE Design Automation Conference (DAC), Miami Beach, FL, June 28 - July 1, 1987*, pages 392–398, 1987.
 URL <http://doi.acm.org/10.1145/37888.37947>.
- Hans-Joachim Wunderlich. Multiple distributions for biased random test patterns. In *IEEE International Test Conference (ITC), Washington, D.C., USA, September 1988*, pages 236–244, 1988.

- Hans-Joachim Wunderlich. The design of random-testable sequential circuits. In *19th International Symposium on Fault-Tolerant Computing (FTCS), 21-23 June, 1989, Chicago, IL, USA*, pages 110–117, 1989.
- Hans-Joachim Wunderlich. BIST for systems-on-a-chip. *Integration*, 26(1-2):55–78, 1998.
URL [http://dx.doi.org/10.1016/S0167-9260\(98\)00021-2](http://dx.doi.org/10.1016/S0167-9260(98)00021-2).
- Hans-Joachim Wunderlich and Stefan Holst. Generalized fault modeling for logic diagnosis. *Models in Hardware Testing*, pages 133–155, 2010.
- Hans-Joachim Wunderlich and Gundolf Kiefer. Bit-flipping BIST. In *International Conference on Computer-Aided Design (ICCAD), November 10-14, 1996, San Jose, CA, USA*, pages 337–343, 1996.
URL <http://doi.acm.org/10.1145/244522.244827>.
- Yu Xia, Malgorzata Chrzanowska-Jeske, Benyi Wang, and Marcin Jeske. Using a distributed rectangle bin-packing approach for core-based soc test scheduling with power constraints. In *International Conference on Computer-Aided Design (ICCAD), November 9-13, 2003, San Jose, CA, USA*, pages 100–106, 2003.
URL <http://doi.acm.org/10.1145/996070.1009876>.
- Dong Xiang, Kaiwei Li, Jianguang Sun, and Hideo Fujiwara. Reconfigured scan forest for test application cost, test data volume, and test power reduction. *IEEE Transactions on Computers (TC)*, 56(4):557–562, 2007.
URL <http://dx.doi.org/10.1109/TC.2007.1002>.
- Takaki Yoshida and Masafumi Watari. MD-SCAN method for low power scan testing. In *11th Asian Test Symposium (ATS), 18-20 November 2002, Guam, USA*, pages 80–85, 2002.
URL <http://csdl.computer.org/comp/proceedings/ats/2002/1825/00/18250080abs.htm>.
- Xiaodong Zhang and Kaushik Roy. Power reduction in test-per-scan BIST. In *6th IEEE International On-Line Testing Workshop (IOLTW), 3-5 July 2000, Palma de Mallorca, Spain*, pages 133–138, 2000.
URL <http://csdl.computer.org/comp/proceedings/ioltw/2000/0646/00/06460133abs.htm>.
- Xiaodong Zhang, Kaushik Roy, and Sudipta Bhawmik. POWERTEST: A tool for energy conscious weighted random pattern testing. In *12th International Conference on VLSI Design, 10-13 January 1999, Goa, India*, pages 416–422, 1999.
- Zhuo Zhang, Sudhakar M. Reddy, Irith Pomeranz, Xijiang Lin, and Janusz Rajski. Scan tests with multiple fault activation cycles for delay faults. In *24th IEEE VLSI Test Symposium (VTS), 30 April - 4 May 2006, Berkeley, California, USA*, pages 343–348, 2006.
URL <http://doi.ieeecomputersociety.org/10.1109/VTS.2006.91>.
- Christian Zoellin, Hans-Joachim Wunderlich, Nicolas Maeding, and Jens Leenstra. BIST power reduction using scan-chain disable in the Cell processor. In *IEEE International Test Conference (ITC), Santa Clara, CA, USA, Oct. 24-26, 2006*,

pages 1–8, 2006.

Yervant Zorian. A distributed BIST control scheme for complex VLSI devices. In *11th IEEE VLSI Test Symposium (VTS), 6 April - 8 April, 1993, Atlantic City, NJ, USA*, pages 4–9, 1993.

Related Publications by the Author

12.1 Book Chapters

- **Power-Aware Design-for-Test**

H.-J. Wunderlich, C. G. Zoellin In: Power-Aware Testing and Test Strategies for Low Power Devices, P. Girard, N. Nicolici, X. Wen, Springer-Verlag Berlin Heidelberg, 2009, ISBN: 9781441909275

12.2 Journals and Conference Proceedings

1. **Efficient Multi-level Fault Simulation of HW/SW Systems for Structural Faults**

R. Baranowski, S. Di Carlo, N. Hatami, M. E. Imhof, M. Kochte, P. Prinetto, H.-J. Wunderlich, C. G. Zoellin, *SCIENCE CHINA Information Sciences*, Vol. 54(9), pp. 1784-1796

2. **Efficient Simulation of Structural Faults for the Reliability Evaluation at System-Level**

M. A. Kochte, C. G. Zoellin, R. Baranowski, M. E. Imhof, H.-J. Wunderlich, N. Hatami, S. Di Carlo, P. Prinetto, *IEEE 19th Asian Test Symposium (ATS)*, Shanghai, China, December 1-4, 2010

3. **System Reliability Evaluation Using Concurrent Multi-Level Simulation of Structural Faults**

M. A. Kochte, C. G. Zoellin, R. Baranowski, M. E. Imhof, H.-J. Wunderlich, N. Hatami, S. Di Carlo, P. Prinetto, *IEEE International Test Conference (ITC)*, Austin, TX, USA, October 31 - November 5, 2010

4. **Effiziente Simulation von strukturellen Fehlern für die Zuverlässigkeitsanalyse auf Systemebene**
M. A. Kochte, C. G. Zoellin, R. Baranowski, M. E. Imhof, H.-J. Wunderlich, N. Hatami, S. Di Carlo, P. Prinetto, 3. VDE Tagung "Zuverlässigkeit und Entwurf" (ZuE), Wildbad Kreuth, Germany, September 13-15, 2010
5. **Efficient Concurrent Self-Test with Partially Specified Patterns**
M. A. Kochte, C. G. Zoellin, H.-J. Wunderlich, *Journal of Electronic Testing: Theory and Applications (JETTA)*, vol 26(5), 2010, pp. 581-594
6. **Efficient Fault Simulation on Many-Core Processors**
M. A. Kochte, M. Schaal, H.-J. Wunderlich, C. G. Zoellin, 47th ACM/IEEE Design Automation Conference (DAC), Anaheim, CA, USA, June 13 - 18, 2010
7. **Low-Power Test Planning for Arbitrary At-Speed Delay-Test Clock Schemes**
C. G. Zoellin, H.-J. Wunderlich, *IEEE VLSI Test Conference (VTS)*, Santa Cruz, CA, USA, April 18 - 21, 2010
8. **Concurrent Self-Test with Partially Specified Patterns For Low Test Latency and Overhead**
M. A. Kochte, C. G. Zoellin, H.-J. Wunderlich, *IEEE European Test Symposium (ETS)*, Sevilla, Spain, May 25-29, 2009, pp. 53-58
9. **Test Exploration and Validation Using Transaction Level Models**
M. A. Kochte, C. G. Zoellin, M. E. Imhof, R. Salimi Khaligh, M. Radetzki, H.-J. Wunderlich, S. Di Carlo, P. Prinetto, *Design, Automation and Test in Europe (DATE)*, Nice, France, April 20-24, 2009
10. **Erkennung von transienten Fehlern in Schaltungen mit reduzierter Verlustleistung**
M. E. Imhof, H.-J. Wunderlich, C. G. Zoellin, 2. VDE GMM/ITG-Fachtagung Zuverlässigkeit und Entwurf (ZuE), Ingolstadt, Germany, 29.09. - 01.10.2008, pp. 107-114
11. **Integrating Scan Design and Soft Error Correction**
M. E. Imhof, H.-J. Wunderlich, C. G. Zoellin, 14th IEEE International On-Line Testing Symposium (IOLTS), Rhodes, Greece, July 7-9, 2008, pp. 59-64
12. **Scan Chain Clustering for Test Power Reduction**
M. Elm, M. E. Imhof, H.-J. Wunderlich, C. G. Zoellin, J. Leenstra, N. Maed-

ing, *45th ACM/IEEE Design Automation Conference (DAC), Anaheim, CA, USA, June 8-13, 2008, pp. 828-833*

13. **Selective Hardening in Early Design Steps**
C. G. Zoellin, H.-J. Wunderlich, I. Polian, B. Becker, *13th IEEE European Test Symposium (ETS), Lago Maggiore, Italy, May 25-29, 2008, pp. 185-190*
14. **Test Set Stripping Limiting the Maximum Number of Specified Bits**
M. A. Kochte, C. G. Zoellin, M. E. Imhof, H.-J. Wunderlich, *4th IEEE International Symposium on Electronic Design, Test & Applications (DELTA), Hong Kong, January 23-25, 2008, pp. 581-586, Best paper award*
15. **Testing and Monitoring Nanoscale Systems - Challenges and Strategies for Advanced Quality Assurance (Invited Paper)**
S. Hellebrand, C. G. Zoellin, H.-J. Wunderlich, S. Ludwig, T. Coym, B. Straube, *43rd International Conference on Microelectronics, Devices and Material with the Workshop on Electronic Testing (MIDEM), Bled, Slovenia, September 2007, pp. 3-10*
16. **A Refined Electrical Model for Drift Processes and its Impact on SEU Prediction**
S. Hellebrand, C.G. Zoellin, H.-J. Wunderlich, T. Coym, S. Ludwig, B. Straube, *22nd IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems (DFT), Rome, Italy, September 26-28, 2007, pp. 50-58*
17. **Programmable Deterministic Built-in Self-test**
A.-W. Hakmi, H.-J. Wunderlich, C.G. Zoellin, A. Glowatz, F. Hapke, J. Schloeffel, L. Souef, *IEEE International Test Conference (ITC), Santa Clara, CA, USA, October 23 - 25, 2007*
18. **Scan Test Planning for Power Reduction**
M. E. Imhof, C. G. Zoellin, H.-J. Wunderlich, *44th ACM/IEEE Design Automation Conference (DAC), San Diego, CA, USA, June 4-8, 2007, pp. 521-526*
19. **Verlustleistungsoptimierende Testplanung zur Steigerung von Zuverlässigkeit und Ausbeute**
M.E. Imhof, C.G. Zoellin, H.-J. Wunderlich, N. Maeding, J. Leenstra, VDE Tagung Zuverlässigkeit und Entwurf (ZuD), München, Deutschland, 26. - 28. März 2007, pp. 69-76

20. **BIST Power Reduction Using Scan-Chain Disable in the Cell Processor**
C. Zoellin, H.-J. Wunderlich, N. Maeding, J. Leenstra, *IEEE International Test Conference (ITC), Santa Clara, CA, USA, October 24 - 26, 2006*
21. **Blue Gene/L compute chip: Synthesis, timing, and physical design**
A. A. Bright, R. A. Haring, M. B. Dombrowa, M. Ohmacht, D. Hoenicke, S. Singh, J. A. Marcella, R. F. Lembach, S. M. Douskey, M. R. Ellavsky, C. G. Zoellin, and A. Gara, *IBM Journal of Research and Development, 2005, Vol. 49, No. 2/3, pp. 277-288*

12.3 Workshop Contributions

1. **Effiziente Fehlersimulation auf Many-Core-Architekturen**
M. A. Kochte, M. Schaal, H.-J. Wunderlich, C. G. Zoellin, *22nd ITG/GI/GMM Workshop "Testmethoden und Zuverlässigkeit von Schaltungen und Systemen" (TuZ), Paderborn, Germany, February 28 - March 2, 2010*
2. **Test Exploration und Validierung auf der Transaktionsebene**
M. A. Kochte, C. G. Zoellin, M. E. Imhof, R. Salimi Khaligh, M. Radetzki, Hans-Joachim Wunderlich, Stefano Di Carlo, Paolo Prinetto, *21st ITG/GI/GMM Workshop "Testmethoden und Zuverlässigkeit von Schaltungen und Systemen" (TuZ), Bremen, Germany, February 15-17, 2009*
3. **Integrating Scan Design and Soft Error Correction in Low-Power Applications**
M. E. Imhof, H.-J. Wunderlich, C. G. Zoellin, *1st Workshop on Low Power Design Impact on Test and Reliability (LPonTR), Verbania, Italy, May 25-29, 2008*
4. **Ein verfeinertes elektrisches Modell fuer Teilchentreffer und dessen Auswirkung auf die Bewertung der Schaltungsempfindlichkeit**
T. Coym, S. Hellebrand, S. Ludwig, B. Straube, H.-J. Wunderlich, C. Zoellin, *20th ITG/GI/GMM Workshop "Testmethoden und Zuverlässigkeit von Schaltungen und Systemen" (TuZ), Wien, Austria, February 24-26, 2008*
5. **Reduktion der Verlustleistung beim Selbsttest durch Verwendung testmenspez. Information**
M. E. Imhof, H.-J. Wunderlich, C. G. Zoellin, J. Leenstra, N. Maeding, *20th*

ITG/GI/GMM Workshop "Testmethoden und Zuverlaessigkeit von Schaltungen und Systemen" (TuZ), Wien, Austria, February 24-26, 2008

6. Programmable Deterministic Built-in Self-test

A.-W. Hakmi, H.-J. Wunderlich, C.G. Zoellin, A. Glowatz, J. Schloeffel, F. Hapke, *ITG/GI/GMM Workshop "Testmethoden und Zuverlaessigkeit von Schaltungen und Systemen" (TuZ), Erlangen, Germany, March 11-13, 2007*

7. BIST Power Reduction Using Scan-Chain Disable in the Cell Processor

C. Zoellin, H.-J. Wunderlich, N. Maeding, J. Leenstra, *ITG/GI/GMM Workshop "Testmethoden und Zuverlaessigkeit von Schaltungen und Systemen", Titisee, Germany, March 12-14, 2006*