

*Test produktionsbedingter
Laufzeitfehler in
hochintegrierten, digitalen Schaltungen*

Dem Fachbereich für Physik und Elektrotechnik
der Universität Bremen

zur Erlangung des akademischen Grades eines
DOKTOR-INGENIEURS (Dr.-Ing.)
vorgelegte Dissertation

von
Volker Hans-Walther Meyer
aus Bremen

Gutachter: Professor Dr.-Ing. Walter Anheier
Professor Dr. Rolf Drechsler

Eingereicht am: 4. November 2003
Promotionskolloquium: 17. Dezember 2003

Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für theoretische Elektrotechnik und Mikroelektronik an der Universität Bremen im Rahmen einer Kooperation mit Philips Semiconductors Hamburg.

Mein Dank gilt Herrn Professor Dr.-Ing. W. Anheier für die Möglichkeit der Durchführung und seine Unterstützung dieser Arbeit sowie die Übernahme der Erstgutachtertätigkeit.

Herrn Professor Dr.-Ing. R. Drechsler danke ich für sein Interesse an meiner Arbeit und die Übernahme der weiteren Gutachtertätigkeit.

Herrn Professor Dr.-Ing. habil. G. Thiele sowie Herrn Professor Dr.-Ing. habil. B. Lohmann danke ich für die Übernahme der Prüfertätigkeit.

Mein ganz besonderer Dank gilt Herrn Dr.-Ing. F. Pöhl für seine Einführung in das Fachgebiet, die zahlreichen Diskussionen und die fortwährende fachliche Unterstützung dieser Arbeit. Viele Erkenntnisse stützen sich auf die von Herrn Pöhl geleistete Arbeit und hätten ohne diese so nicht gewonnen werden können.

Die vorliegende Arbeit hat durch die Zusammenarbeit mit dem industriellen Projektpartner Philips Semiconductors eine sehr praxisrelevante Ausrichtung erreicht. Für die geleistete finanzielle Unterstützung und die zahlreichen fachlichen Diskussionen bin ich dem Design Technology Center der Firma Philips Semiconductors sehr dankbar. Hervorzuheben sind in diesem Zusammenhang der Leiter des DTC, Herr F. Hapke, sowie die Herren Dipl.-Ing. A. Sticht und Dipl.-Phys. J. Schlöffel.

Schließlich möchte ich mich bei meinen Freunden und Kollegen am Institut, ohne deren Hilfe eine solche Arbeit nicht zu verwirklichen wäre, für ihre Unterstützung und ihre Aufmunterung bedanken.

Inhaltsverzeichnis

1	Einleitung	1
1.1	Motivation der Arbeit	2
1.2	Gliederung der Arbeit	5
2	Grundlagen, Begriffe und Definitionen	7
2.1	Defekt und Fehler	7
2.2	Test und Diagnose	8
2.3	Allgemeine Schaltungsbeschreibung	9
2.4	Testvektor, Testmuster und Testwertsatz	11
2.5	Fehlermodelle	11
2.5.1	Statische Fehlermodelle	13
2.5.2	Dynamische Fehlermodelle, Verzögerungsfehler	15
2.5.3	Strombasierte Testmethoden	20
2.6	Signallaufzeiten	20
2.7	Testmustergenerierung und Fehlersimulation	22
2.8	Testklassen dynamischer Fehlermodelle	22
2.8.1	Robuster Test	24
2.8.2	Nicht robuster Test	25
2.8.3	Testklassen	25
2.9	Algorithmen zur Testmustergenerierung	26
2.9.1	D-Algorithmus	26
2.9.2	PODEM	27
2.9.3	FAN	27
2.9.4	Recursive learning	29
2.9.5	Decision diagrams	30

2.9.6	Genetische Verfahren	30
2.10	Applikationsmodi	30
2.11	Taktschemata	35
2.12	Zusammenfassung	37
3	Anforderungen an den Test aktueller Schaltungen	39
3.1	Defekte in aktuellen CMOS Schaltungen	39
3.2	Fehlermodelle für aktuelle CMOS Schaltungen	42
3.3	Experimentelle Ergebnisse	43
3.4	Zusammenfassung	47
4	Fehlersimulation für Verzögerungsfehler	49
4.1	Fehlersimulation für Übergangsfehler	49
4.2	Fehlersimulation für Gatterverzögerungsfehler	50
4.3	Fehlersimulation für Pfadverzögerungsfehler	51
5	Testmustergenerierung für Verzögerungsfehler	55
5.1	Ansätze zur Testmustergenerierung für Verzögerungsfehlermodelle	55
5.1.1	Logiken zur Testmustergenerierung für Verzögerungsfehler	56
5.1.2	Sequentieller Ansatz	56
5.2	Testmustergenerierung	57
5.2.1	Testmustergenerierung für das Übergangsfehlermodell	57
5.2.2	Testmustergenerierung für das Gatterverzögerungsfehlermodell	61
5.2.3	Testmustergenerierung für das Pfadverzögerungsfehlermodell	63
5.3	Testmustergenerierung für Pfadverzögerungsfehler unter Verwendung des sequentiellen Ansatzes	64
5.3.1	Fehlerlisten und Pfadbeschreibung	65
5.3.2	Algorithmus zur Testmustergenerierung für Pfadverzögerungsfehler	68
5.3.3	Pfadsensitivierung	68
5.3.4	Einstellen der noch einzustellenden Knoten	69

6	Pfadverzögerungsfehlerbasierter Test	73
6.1	Testqualität	73
6.1.1	Nicht robuste Tests	75
6.1.2	Erhöhung der Qualität nicht robuster Tests	78
6.1.3	Nicht robuste Tests hoher Qualität	82
6.1.4	Zusammenfassung	84
6.2	Testmusterkompaktierung	86
6.3	Vergleiche mit Arbeiten aus der Literatur und Abschätzung der Leistungsfähigkeit	91
7	Anwendungen verzögerungsfehlerbasierter Testmuster	93
7.1	Produktionstest	93
7.2	Analyse	97
7.3	Speed binning	98
8	Fehlereffekte in Nanotechnologien	101
8.1	Elektromigrationserscheinungen	101
8.2	Signalkopplungen	102
8.2.1	Überschwinger	103
8.2.2	Unterschwinger	103
8.2.3	Temporäre Signalschwankungen	104
8.2.4	Signalverzögerungen	105
8.3	Einflüsse der Versorgungsspannung	106
9	Zusammenfassung	109

Liste der verwendeten Symbole

Symbol	Bedeutung
k	beliebiger Knoten innerhalb der Schaltung
k_{Φ}	Fehlerort (fehlerbehafteter Schaltungsknoten)
Φ	Fehler
δ	Größe des Fehlers
Δ_{Φ}	Fehlereffekt
P^k	kritischer Pfad
s	Reserve auf dem Pfad
V_I	Initialisierungsvektor
V_P	Propagationsvektor
T	Testmuster
Θ	Testwertsatz
$iv(x)$	Ausgangswert der Flanke x
$fv(x)$	Endwert der Flanke x
t_I	Taktzeitpunkt zu Beginn des Initialisierungszyklus
t_P	Taktzeitpunkt zu Beginn des Propagationszyklus
t_{CLK}	Taktzeitpunkt am Ende des Propagationszyklus



1 Einleitung

Seit der Realisierung der ersten integrierten Schaltung im Jahre 1959 durch Jack Kilby¹ entwickelt sich die Herstellung integrierter Schaltkreise mit atemberaubender Geschwindigkeit. Die durch technologische Entwicklungen beständig wachsende Zahl an integrierbaren Bauelementen je Schaltung verschafften den integrierten Schaltkreisen eine wachsende Funktionalität und eröffneten ihnen ein stetig wachsendes Einsatzgebiet. Die Vorteile, bei geringerem Gewicht, Platzbedarf und Leistungsaufnahme die gleiche Funktionalität zu realisieren, verschafften ihnen zunächst den Durchbruch in der Unterhaltungselektronik, wo sie in Radios oder Fernsehern die Röhren ersetzten. Weiter sinkende Abmaße bei gleichzeitig sinkendem Energiebedarf verhalfen ihnen zum Einsatz in der Medizintechnik (Beispiel Herzschrittmacher). Steigende Rechenleistung war der Grundstein für den Erfolg der Prozessortechnik der letzten 20 Jahre, und auch die heutzutage überall präsente Mobiltelefonie wäre ohne die Rechenkapazität und den minimalen Leistungsbedarf moderner integrierter Schaltungen nicht möglich.

Bei der Entwicklung der Mikroelektronik werden in den meisten Fällen die Fortschritte in der Fertigung hervorgehoben und nur selten erwähnt, dass der Produktionsprozeß einer integrierter Schaltung selbst nach langer Fertigungszeit und Optimierung keine Ausbeute² von 100% erreicht. Der Grund hierfür liegt in der Vielzahl der Einflüsse, die auf den Herstellungsprozess wirken und die Erschaffung vollkommen identischer Produkte verhindern. Als Beispiele können Verunreinigungen des Basismaterials ebenso angeführt werden wie Materialrückstände aus vorhergehenden Prozeßschritten oder Schwankungen in der Zusammensetzung verwendeter Materialien. Die Erkennung der fehlerhaften Schaltungen in der Menge aller gefertigten Schaltungen ist Aufgabe des Produktionstests. Sein Einfluss auf den Erfolg der mikroelektronischen Schaltungen wird zumeist unterschätzt, doch spätestens die Verwendung integrierter Schaltungen in sicherheitskritischen Anwendungen wie Kraftwerkssteuerungen oder Bremskreisregelungen in Kraftfahrzeugen verdeutlicht, welchen Anteil der Produktionstest am Erfolg oder Misserfolg einer produzierten Schaltung hat.

Der Erfolg einer gefertigten Schaltung lässt sich nüchtern an dem Gewinn messen, der durch sie erzielt wird. Wirtschaftlich betrachtet haben darauf eine Vielzahl von Faktoren Einfluss. Die Kosten einer produzierten Schaltung setzen sich einerseits aus den eingesetzten Ressourcen, andererseits aus der Anzahl der fehlerfreien (verwertbaren) Schaltungen zusammen. Die eingesetzten Ressourcen umfassen dabei neben den eingesetzten Materialien auch die Kosten für die

¹für die er 2000 den Nobelpreis der Physik verliehen bekam

²Anteil der fehlerfreien Schaltungen an der Menge aller gefertigten Schaltungen, auch als Yield bezeichnet

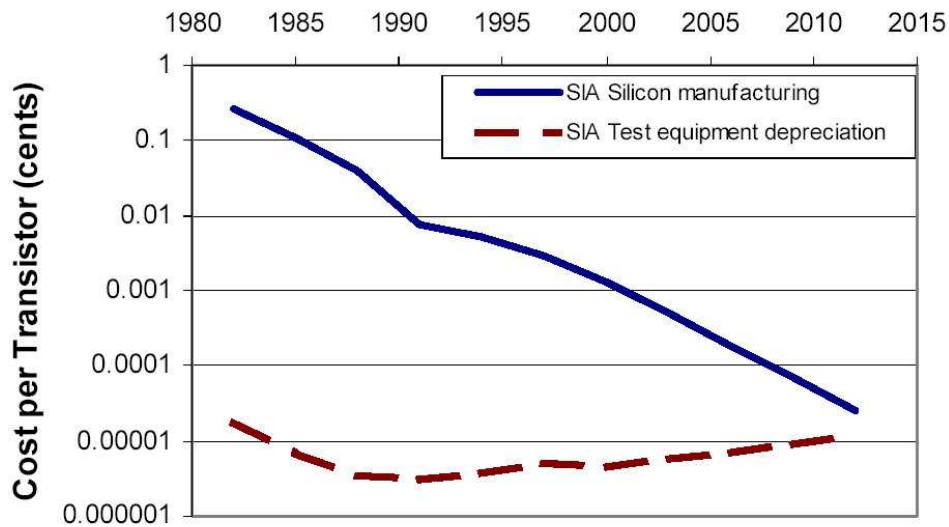


Abbildung 1: Kosten für die Produktion und den Test eines Transistors einer integrierten Schaltung [90]

verwandten Produktionsstätten (Anschaffungs- und Betriebskosten) sowie die Kosten für den Test. Die ITRS-Roadmap [90] der Halbleiterindustrie, die auf den Daten der Vergangenheit basierend die Trends der kommenden Jahre voraussagen versucht, zeigt deutlich, dass die Kosten für die Fertigung eines Transistors durch die zunehmende Integrationsdichte tendenziell sinken, während die Kosten für den Test eines Transistors tendenziell konstant bleiben (Abbildung 1). Insgesamt bedeutet dies, dass der Anteil der Testkosten an den Gesamtkosten einer gefertigten Schaltung zunimmt. Die Gründe für diese Entwicklung liegen hauptsächlich in der steigenden Integrationsdichte sowie den durch die technologischen Fortschritte bedingten veränderten Fehlermechanismen, wodurch insgesamt die Menge möglicher Fehlerorte und Fehlereffekte, deren Abwesenheit der Produktionstest sicherstellen muss, anwächst.

1.1 Motivation der Arbeit

In der Halbleiterindustrie ist der Bedarf an Testmustern für Verzögerungsfehler als Teil des Produktionstests, gerade mit Blick auf den Anstieg der Taktfrequenzen bei den Prozessoren in den vergangenen Jahren (Abbildung 2, siehe auch Tabelle 10 im Anhang), hochaktuell. Die bislang gemachten Erfahrungen sowie die Prognosen der Halbleiterindustrie sehen für die kommenden Jahre einen weiteren Anstieg der Integrationsdichte und Taktfrequenzen voraus. Aus diesen

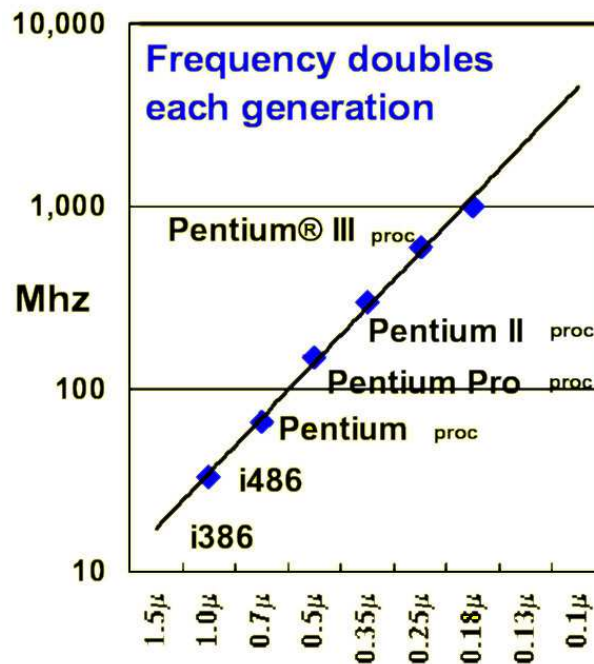


Abbildung 2: Entwicklung der Taktfrequenz am Beispiel der Intel Prozessorfamilie [107]

Gründen ist für die Hersteller der Testwerkzeuge (wie Testmustergeneratoren, Fehlersimulatoren oder Diagnoseprogrammen) der Bereich der Testmustergenerierung für dynamische Fehlereffekte von großer Bedeutung. Den Entwurfsablauf einer digitalen Schaltung im Top-Down Design sowie die einzelnen Punkte, an denen der Test berücksichtigt werden sollte, zeigt Abbildung 3. In enger Zusammenarbeit mit der Philips Semiconductors GmbH entstanden dabei Testmustergeneratoren für Verzögerungsfehlereffekte. Der hervorgehobene Testmustergenerator für Pfadverzögerungsfehler bildet den Schwerpunkt dieser Arbeit.

Durch seine Ausrichtung auf die Anwendung im Produktionstest geht der Testmustergenerator im Unterschied zu den meisten in der Literatur vorhandenen Beiträge von einem Standard-Scan Design aus. Die daraus folgenden Abhängigkeiten werden erläutert und auf ihre Einsatzmöglichkeit im Produktionstest hin untersucht. Von besonderer Bedeutung sind dabei die Anforderungen, die der Test an das zu verwendende Equipment stellt, sowie die zu erwartende Messgenauigkeit.

Im Unterschied zum überwiegenden Teil der in der Literatur verfolgten Ansätze wird die Testmustergenerierung für Verzögerungsfehler in zwei Teilprobleme aufgespalten, wie sie von der Testmusterberechnung in sequentiellen Schaltungen bekannt sind. Als Folge kann der Kern eines

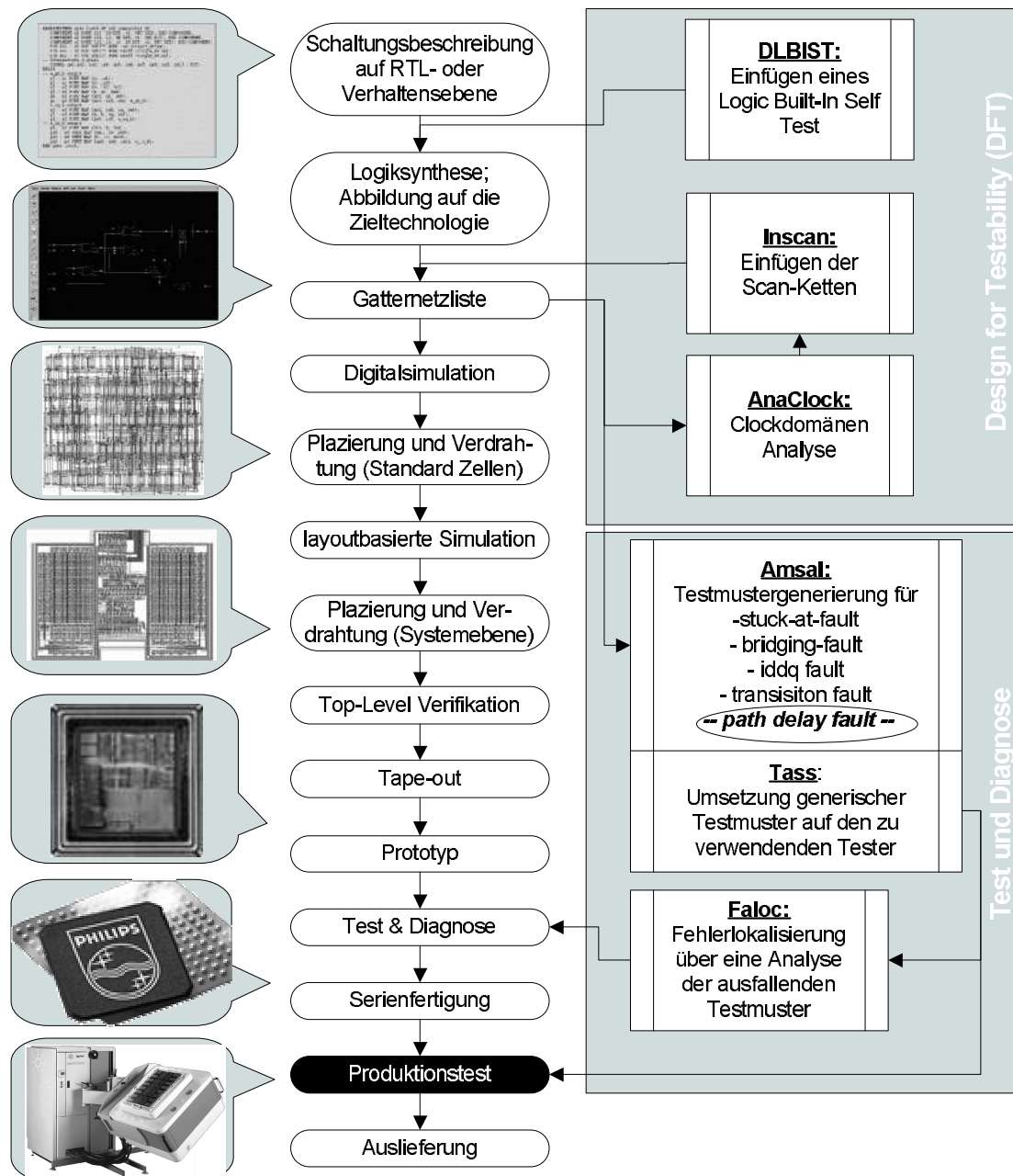


Abbildung 3: Darstellung der testbezogenen Anteile im Top-Down Entwurfsablauf sowie die vom Projektpartner Philips Semiconductors verfügbaren Tools für den einzelnen Testaufgaben

Testmustergenerators für Haftfehler für die Berechnung verwendet werden. Dies führt zu einer Minimierung des Integrations- und Wartungsaufwandes beim Industriepartner Philips Semiconductors. Darüber hinaus gewährt der Rückgriff auf bestehende Codefragmente die Verwendung bereits getesteter Implementierungen, was insbesondere die Zuverlässigkeit des Prototypen erhöht; der Generator erreicht damit schneller seine Einsatzreife.

1.2 Gliederung der Arbeit

Das Umfeld des Produktionstests von Schaltungen aktueller Größe und Fertigungstechnologie bildet den Rahmen, in dem die vorliegende Arbeit entstanden ist (Abbildung 3). Aus diesem Anlass wird nach einer Darstellung der Grundlagen (Kapitel 2) zunächst die Frage geklärt, in welchen Technologien aktuelle Schaltungen gefertigt werden. Mit Blick auf die Testmustergenerierung wird in Kapitel 3 erarbeitet, welche typischen Fehlereffekte in aktuellen Schaltungen aufgrund der verwendeten Technologie und Technologiefortschritte auftreten (Abschnitt 3.1) um entscheiden zu können, welches Fehlermodell dem Produktionstest zugrunde gelegt werden muss. Die Aussagen stützen sich dabei auf praktische Untersuchungen aus der Literatur (Abschnitt 3.3).

Basierend auf den Ergebnissen wird in Kapitel 4 kurz die Fehlersimulation für dynamische Fehlermodelle erläutert. Der im Rahmen dieser Arbeit entstandene und dort beschriebene Pfadverzögerungsfehlersimulator ist von integraler Bedeutung für die Bewertung der Testmuster, wie sie in Kapitel 6 dargestellt ist. Kapitel 5 beschäftigt sich mit der Testmustergenerierung für Verzögerungsfehlermodelle, wobei der in Abschnitt 5.3 vorgestellte Testmustergenerator für Pfadverzögerungsfehler den Kern dieser Arbeit bildet.

Kapitel 6 beschreibt die im Rahmen der Arbeit durchgeführten Untersuchungen, die sich mit der Qualität der erhaltenen Testmuster befassen. Es zeigt sich, dass eine Testmustergenerierung, die auf den in der Literatur als hinreichend für einen nicht robusten Test angegebenen Bedingungen basiert, zweifelhafte Ergebnisse liefert, weswegen in Kapitel 6 Modifikationen erarbeitet werden. Die Fehlersimulationen der daraus erhaltenen Testmuster belegen die erreichten Verbesserungen.

Kapitel 7 beschreibt mögliche Anwendungsgebiete der in dem Projekt entstandenen Testmustergeneratoren und fasst die in diesem Zusammenhang publizierten Arbeiten aus dem Projekt zusammen. Abschließend beleuchtet Kapitel 8 Probleme in der Schaltungsfertigung, die mit weiter voranschreitender Miniaturisierung bei steigenden Integrationsdichten auf den Produktionstest zukommen. Eine Bewertung der Effekte sowie erste Vorschläge, wie die Testmustergenerierung für den Test dieser Fehlereffekte realisiert werden könnte, bilden Abschluß der Arbeit.

2 Grundlagen, Begriffe und Definitionen

Vor dem Test einer integrierten Schaltung müssen die möglichen Fehlermechanismen, die in der Schaltung auftreten können, analysiert werden (→3.1). Wird ein fehlerfreies Schaltungsdesign vorausgesetzt und Entwurfsfehler damit ausgeschlossen, muss der Grund für eine fehlerhafte Schaltungsfunktion in Defekten liegen, die durch den Produktionsprozeß eingebracht wurden. Folglich kann auch von einem defektorientierten Test gesprochen werden.

2.1 Defekt und Fehler

Der Begriff *Defekt* beschreibt dabei eine physikalische Abnormalität, die zu einer ungewollten Veränderung der (elektrischen) Eigenschaften einer Schaltung führt. Einen ungewollt während des Produktionsprozesses eingebrachten Partikel als mögliche Defektursache zeigt Abbildung 4. Defekte lassen sich im allgemeinen nicht direkt erkennen, da sie zum einen eine möglicherweise geringe geometrische Ausdehnung haben, zum anderen nach mehreren Prozessschritten unter den nachfolgenden Schichten vergraben liegen können. Um Defekte in mikroelektronischen Schaltungen nachzuweisen, wird ihr Einfluss auf die elektrischen Eigenschaften der Schaltung ausgenutzt. Ein *Fehlermodell* (→2.5) beschreibt die Abbildung eines Defektes über seine elektrischen Auswirkungen auf einen Fehler. Ein *Fehler* ist damit die durch einen Defekt verursachte Veränderung der Schaltungsfunktion.

Fehlermodelle basieren auf Annahmen über die Auswirkung möglicher Defekte auf die Schaltungsfunktion. Aus den dem jeweiligen Fehlermodelle zugrunde liegenden Annahmen folgt, welche Defekte das Fehlermodell abdeckt bzw. welche Defekte damit nicht modelliert werden können. Die modellierten Fehler sind -im Gegensatz zu den sie verursachenden Defekten- mit Hilfe geeigneter elektrischer Signale (Testmuster) nachweisbar. Kann ein Fehler innerhalb einer Schaltung detektiert werden, muss diese folglich einen Defekt aufweisen. Die Rücktransformation eines Fehlers auf den ursächlichen Defekt ist jedoch nicht eindeutig, da verschiedene Defekte die gleiche elektrische Auswirkung haben können und sich in diesem Fall im gleichen Fehler manifestieren (Abbildung 5).

Im Folgenden wird bei der Verwendung des Begriffs Test der *strukturelle* Test einer Schaltung vorausgesetzt. Strukturelle Testverfahren basieren auf einer Fehlermodellierung in einer strukturellen Beschreibung der Schaltung. Ein *funktionaler* Test einer Schaltung ist ebenfalls denkbar und findet in der Industrie trotz wachsender Schaltungsgrößen weiterhin Anwendung [85][127]. Der Funktionaltest erfordert grundsätzlich ein exaktes Wissen über die Schaltungsfunktion, um

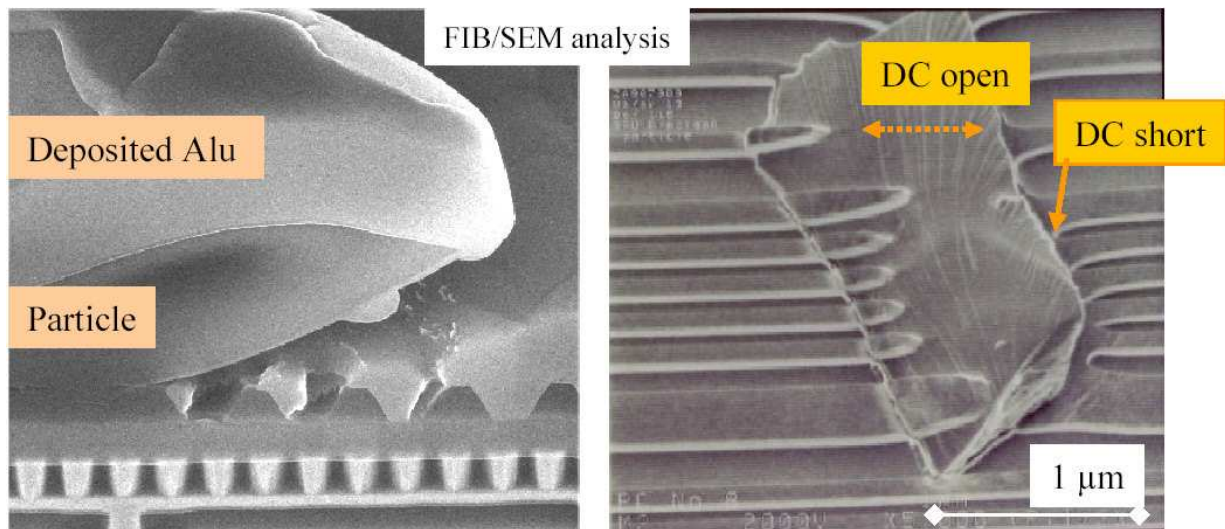


Abbildung 4: Defekte (z.B. Partikel) bewirken Fehler in einer Schaltung [132]

sinnvolle Testmuster zu erzeugen, eine automatische Testmuster-generierung ist nicht möglich [14]. Funktionale Testmuster verwenden für ihre Erstellung kein Fehlermodell, eine auf strukturelle Fehlermodelle bezogene Fehlerabdeckung kann jedoch über eine entsprechende Fehler-simulation bestimmt werden. Ein vollständiger funktionaler Test ist für Schaltungen heutiger Größe nicht möglich, da es zu viele mögliche Zustände gibt, welche die Schaltung einnehmen kann³.

2.2 Test und Diagnose

Generell hat der *Test* einer Schaltung zum Ziel, deren fehlerfreie Funktion zu gewährleisten, weshalb der Test einer Schaltung abgebrochen werden kann, sobald ein Fehler detektiert wurde. Da der Test Teil der Produktion ist und auf jede produzierte Schaltung angewendet werden muss, sind die Anforderungen an einen optimalen Test, die maximal mögliche Fehlerabdeckung bei minimaler Testdauer und minimaler Größe des Testwertsatzes zu erreichen. Die aus dem Test erhaltenen Informationen können lediglich zur Bestimmung der korrekten Funktion der Schaltung dienen. Die zwangsläufig erwünschte hohe Fehlerabdeckung jedes einzelnen Testmusters geht damit zu Lasten der Aussagekraft, welcher der modellierten Fehler bei einer fehlerhaften

³Ein vollständiger Test weist für jeden testbaren der modellierten Fehler dessen Existenz oder Abstinenz in der Schaltung nach. Ein vollständiger funktionaler Test muss den Prüfung in jeden funktionsbedingt möglichen Zustand bringen um dessen korrekte Funktion zu gewährleisten.

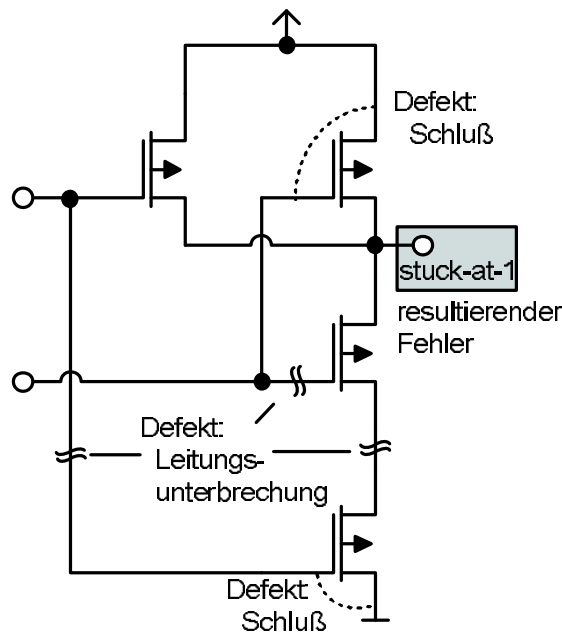


Abbildung 5: Verschiedene, einzeln vorliegende Defekte können sich in dem gleichen Fehler manifestieren

Testantwort in der Schaltung vorliegt.

Die *Diagnose* soll, im Unterschied zum Test, die Bestimmung des Fehlerortes innerhalb der Schaltung mit dem Ziel ermöglichen, den dem Fehler zugrunde liegenden Defekt aufzuspüren. Die Anforderungen an die zu verwendenden Testmuster sind -im Gegensatz zum Test- eine möglichst eindeutige Zuordnung zwischen einem modellierten Fehler und einem Testmuster, um für ein ausfallendes Testmuster über eine möglichst geringe Anzahl an von ihm abgedeckten Fehlern die Zahl der in Frage kommenden Fehlerorte gering zu halten. Durch die Verwendung mehrerer Testmuster, die eine unterschiedliche Kombination an Fehlern abdecken, kann die Menge der möglicherweise in der Schaltung vorliegenden Fehler über eine Schnittmengenbestimmung der von den ausfallenden Vektoren detektierbaren Fehler verringert werden.

2.3 Allgemeine Schaltungsbeschreibung

Bei der zu prüfenden Schaltung handelt es sich im allgemeinen um eine *sequentielle Schaltung*, wie sie Abbildung 6 zeigt. Für den Sonderfall, dass die zu testende Schaltung keine Speicherelemente verwendet und damit als rein *kombinatorische Schaltung* vorliegt, ist der kombinatorische

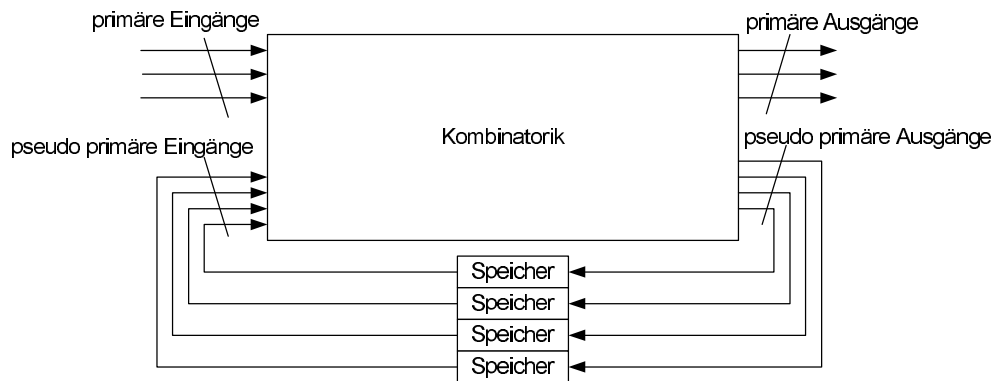


Abbildung 6: Allgemeines Modell einer sequentiellen Schaltung

Schaltungsteil vollständig über die primären Eingänge⁴ erreichbar.

Der Großteil des Produktionstests zielt auf den kombinatorischen Schaltungsteil ab, womit die internen Zustandsleitungen eine besondere Bedeutung für den Test bekommen. Sind die *pseudo primären* Eingänge von außen nicht zugänglich, können die notwendigen Signale auf einem Teil der schaltungsinternen Knoten nicht direkt eingestellt werden. Um die Schaltung in den für den Test notwendigen Initialisierungszustand zu bringen, können deshalb mehrere Taktzyklen nötig sein [69]. Da die pseudo primären Ausgänge sequentieller Schaltungen zudem nicht von außen beobachtet werden können, sinkt die erzielbare Fehlerabdeckung⁵ des Tests bei gleichzeitig steigender Komplexität der Testmuster generierung und Testdurchführung. Um die Testbarkeit der Schaltung zu erhöhen, werden die Speicherelemente in der Regel durch Scan-Speicherelemente ersetzt⁶, die in einer sogenannten *Scan-Kette*⁷ [32] verschaltet werden. Als Folge wird ein externer, sequentieller Zugriff auf die internen Speicherelemente ermöglicht, die Schaltung erhält eine größere Anzahl an Testpunkten und ihre Testbarkeit steigt. Für die Durchführung eines Haftfehlers (→2.5.1) ist der sequentielle Zugriff ohne nennenswerten Einfluss, einzig die Testzeit steigt mit der Länge der verwendeten Scan-Kette, da der pseudo primäre Anteil des Testvektors vor dem Test eingetaktet und der pseudo primäre Anteil der Schaltungsantwort nach dem Test ausgetaktet werden muss. Für die dynamischen Fehlermodelle ergeben sich indes weitreichende Auswirkungen (→2.10).

⁴Unterschieden werden einerseits *primäre* Schaltungseingänge (-ausgänge), die extern beeinflussbar bzw. beobachtbar sind, sowie *pseudo primäre* Schaltungseingänge (-ausgänge), welche die internen Zustandsleitungen auf die (pseudo primären) Schaltungseingänge zurückkoppeln.

⁵Fehlerabdeckung: Menge der testbaren Fehler, bezogen auf die Gesamtfehlermenge

⁶Bei einem Scan-Speicherelement ist dem eigentlichen Speicherelement ein Multiplexer vorgeschaltet. Im Normal-Modus wird der Dateneingang, im Test-Modus der Testeingang an das Speicherelement geleitet.

⁷auch Standard-Prüfpfad genannt [139]

Die Verwendung der Scan-Ketten Technik hat sich industrieweit durchgesetzt, weshalb in den nachfolgenden Betrachtungen davon ausgegangen wird, dass die zu testende Schaltung als kombinatorische Schaltung vorliegt.

2.4 Testvektor, Testmuster und Testwertsatz

Die grundlegende Aufgabe eines *Testmusters* T besteht -unabhängig vom verwendeten Fehlermodell- darin, die Schaltung, dem angenommenen Fehler entsprechend, geeignet zu initialisieren und den möglichen Fehlereffekt an einen beobachtbaren Schaltungsausgang zu propagieren. Abhängig vom Fehlermodell ist die für den Test notwendige Zahl an *Testvektoren* V je Fehler unterschiedlich.

Für die Detektion statischer Fehler (\rightarrow 2.5.1) in *kombinatorischen* Schaltungen reicht die Anwendung eines Testvektors je Messung aus, da dieser in der Lage ist, die beiden vorgenannten Aufgaben zu übernehmen.

Für die Detektion statischer Fehler in *sequentiellen* Schaltungen müssen im allgemeinen mehrere Taktzyklen T_C aufgewendet werden, um die Schaltung zu initialisieren, da nicht alle Schaltungsknoten k direkt über die primären Eingänge eingestellt werden können (\rightarrow 2.3). Da für jeden Taktzyklus die Belegung der primären Eingänge neu erfolgen kann, werden im allgemeinen mehrere Testvektoren als Initialisierungssequenz Anwendung finden. Ein Testmuster für einen Fehler Φ besteht damit aus der Sequenz der Testvektoren, die für seine Detektion notwendig sind: $T = \langle V_1 \dots V_n \rangle$. Während diese Sequenz bei statischen Fehlermodellen zumeist nur einen Vektor enthält, kommen bei dynamischen Fehlermodellen (\rightarrow 2.5.2) mehrere Testvektoren zum Einsatz (üblicherweise werden die minimal notwendigen zwei Vektoren verwendet). Der letzte Testvektor hat dabei die Aufgabe, die für den Test notwendige Flanke am angenommenen Fehlerort zu erzeugen und zu einem Ausgang zu propagieren, weshalb er auch Propagationsvektor V_P genannt wird. Die vorhergehenden Vektoren dienen der Initialisierung der Schaltung und werden folglich auch Initialisierungsvektoren V_I genannt.

Die Menge der während eines Tests verwendeten Testmuster T wird im Folgenden als *Testwertsatz* Θ bezeichnet.

2.5 Fehlermodelle

Der Einfluss eines Defektes auf die Schaltungsfunktion kann mit seiner Lage innerhalb der Schaltung und seiner räumlichen Ausdehnung variieren. Ein Leitungsschluss zwischen einer Vorsor-

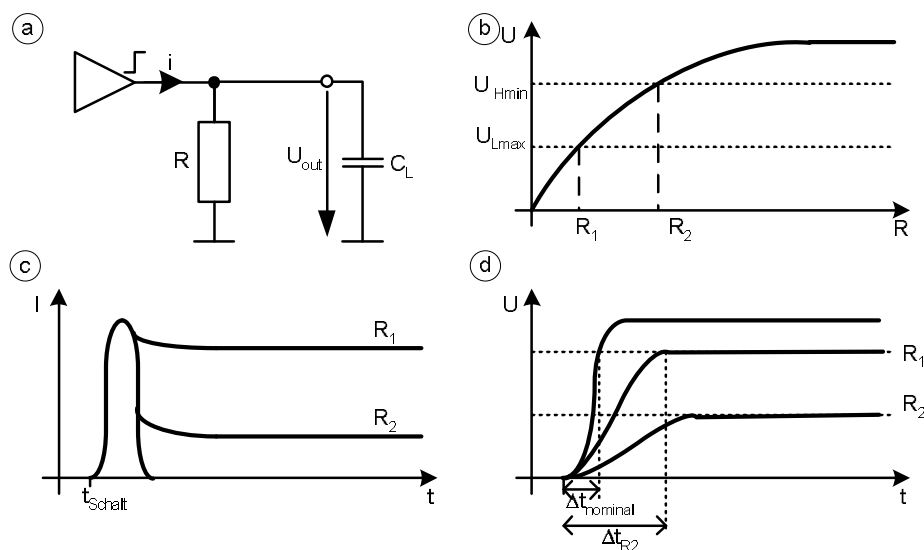


Abbildung 7: Auswirkungen des resistiven Schlusses (R) einer Signalleitung nach Masse

gungsleitung und einer Signalleitung (Kurzschluss nach Masse oder Versorgungsspannung) kann diese auf einen festen Wert ziehen. Die logische Funktion der Schaltung wird entsprechend verändert⁸. Zusätzlich kommt es zu einem erhöhten Stromfluss am Fehlerort, sollte das die Leitung treibende Gatter versuchen, den fehlerhaften Knoten auf den dem Fehlerwert gegensätzlichen Wert zu treiben. Handelt es sich bei dem Kurzschluss um einen resistiven Kurzschluss (Abbildung 7 (a)), hängt seine Auswirkung vom Wert des eingebrachten Widerstandes sowie der Treiberstärke des Gatters ab, welches die Signalleitung treibt. Ist der Widerstand ausreichend hoch, wird sich durch den Stromfluss ein Spannungsabfall über den defektbedingt eingebrachten Widerstand einstellen, die Signalleitung kann einen Spannungswechsel vornehmen (Abbildung 7 (b)). Der Spannungshub ist jedoch eingeschränkt, zudem stellt sich ein ungewollter, zusätzlicher Stromfluss mit einhergehender Verlustleistung ein (Abbildung 7 (c)). Reicht der Widerstandswert des Kurzschlusses aus, die Signalleitung beide definierten Logikpegel erreichen zu lassen, ist der Einfluss auf die statische Funktion der Schaltung nicht mehr gegeben, das dynamische Verhalten der Schaltung wird jedoch weiterhin verändert (Abbildung 7 (d)). Die Stromüberhöhung geht mit zunehmendem Widerstandswert zurück, bleibt aber über einen großen Widerstandsbereich meßbar erhalten.

⁸Es wird davon ausgegangen, dass die Signalleitung aufgrund der Schaltungsfunktion den dem Fehlerwert entgegengesetzten logischen Pegel annehmen kann.

2.5.1 Statische Fehlermodelle

Aus den obigen Überlegungen ist ersichtlich, dass die Auswirkungen eines Defektes auf die Schaltungsfunktion unterschiedlich sein können und diesem durch unterschiedliche Fehlermodelle Rechnung getragen werden muss.

Jedes Fehlermodell verwendet eine Beschreibung der betrachteten Schaltung als Grundlage für die Beschreibung möglicher Defektauswirkungen. Die Netzlistenbeschreibung kann auf unterschiedlichen Abstraktionsebenen geschehen. Je abstrakter der Grad der Schaltungsbeschreibung wird, desto weiter muss sich das Fehlermodell von dem realen Verhalten des Defektes entfernen. Auf der anderen Seite steigt die Zahl der modellierten Fehler mit sinkendem Abstraktionsgrad. Im Sinne des Tests muss das Ziel der Fehlermodellierung sein, einen optimalen Kompromiss aus zwei gegensätzlichen Anforderungen zu finden. Zum einen sollen die Auswirkungen der adressierten Defekte möglichst genau beschrieben werden, zum anderen ist eine möglichst geringe Zahl an modellierten Fehlern zu erreichen, um einen möglichst vollständigen Test unter Anwendung einer minimalen Anzahl an Testmustern zu ermöglichen.

Bei Fehlermodellen wird grob in statische und dynamische Fehlermodelle unterschieden. Zu den statischen Fehlermodellen gehören das stuck-open und stuck-short Fehlermodell sowie das Haftfehlermodell.

Stuck-open / Stuck-short fault model Stuck-open oder stuck-short Fehler [16, 133]⁹ gehen von einer Schaltungsbeschreibung auf Transistorebene aus. Es wird angenommen, dass sich die in der Schaltungsimplementierung vorliegenden Defekte in einem stets geöffneten (stuck-open) oder stets geschlossenen (stuck-short, stuck-on) Zustand eines Transistors äußern. Die Menge der modellierten Fehler enthält $2 * (\#Transistoren)$ Fehler. Die relativ geringe Abstraktion in der Betrachtung der Schaltung führt im Vergleich zu Fehlermodellen auf Gatterebene zu einer höheren Zahl modellierter Fehler, kann jedoch andererseits Fehler modellieren, die ein sequentielles Verhalten der Schaltung bewirken und nur mit einer geeigneten Sequenz an Testvektoren erkannt werden können [34].

Haftfehlermodell (stuck-at fault model) Das weit verbreitete Haftfehlermodell (Ständigfehlermodell, stuck-at fault model)¹⁰ [43] basiert auf einer Gatternetzlistenbeschreibung der

⁹in der Literatur alternativ auch mit *stuck-off* und *stuck-on* bezeichnet

¹⁰Es wird zwischen dem single stuck-at fault model und dem multiple stuck-at fault model unterschieden, die sich in ihrer Einzelfehler- bzw. Mehrfachfehlerannahme unterscheiden. Der Produktionstest verwendet das *single*

Schaltung. Zugrunde gelegt wird die Annahme, dass ein Schaltungsknoten (defektbedingt) statisch auf einer logischen Eins (stuck-at-1) oder einer logischen Null (stuck-at-0) festhängen kann. Als Folge kann der entsprechende Schaltungsknoten nicht auf den komplementären Wert umgeschaltet werden, was zu einer Veränderung der (statischen) Schaltungsfunktion führt¹¹.

Die Angabe zu den modellierten Fehlerorten unterscheidet sich in der Literatur zwischen den Schaltungsknoten einerseits und den Gattereingängen und Gatterausgängen andererseits. Werden Haftfehler an Moduleingängen und Modulausgängen separat als *Pin Faults* definiert [14], kann der vom Haftfehler angenommene Fehlerort auf den Schaltungsknoten angenommen werden.

Brückenfehlermodell (bridging fault model) Das Brückenfehlermodell (bridging fault model) [88] modelliert Defekte, die zu einem (widerstandslosen) Schluss zweier Leitungen innerhalb einer Schaltung führen. Im Unterschied zum Haftfehlermodell ist der resultierende Fehlerwert nicht technologieunabhängig, da die Fehlfunktion in der Schaltung von mehreren Parametern (z.B. den Treiberstärken der betroffenen Leitungen oder den Signalwerten auf den Leitungen) abhängig ist. Die Zahl der modellierten Fehler ergibt sich bei Annahme eines Schlusses zwischen zwei Leitungen und n innerhalb der Schaltung vorhandenen Leitungen zu $\frac{n(n-1)}{2}$ und wächst damit quadratisch mit der Schaltungsgröße. Liegt das Layout der zu testenden Schaltung vor, können viele der modellierten Fehler unberücksichtigt bleiben, da die betreffenden Leitungen physikalisch zu weit voneinander entfernt realisiert wurden, um einen Leitungsschluss zu ermöglichen. Werden mögliche Brückenfehler zwischen mehr als zwei Leitungen berücksichtigt, wächst die Fehlermenge schnell auf eine nicht mehr handhabbare Größe an.

Brückenfehler können mit Testmustern für Haftfehler detektiert werden, wenn diese den Brückenfehler aktivieren¹² und der sich auf dem Knoten einstellende Wert dem adressierten Haftfehler entspricht [136]. Empirische Untersuchungen [1] zeigen, dass Haftfehlerwertesätze mit 95% Fehlerabdeckung im Mittel auch 83% der modellierten Brückenfehler detektieren. Neben speziellen Brückenfehlertests werden auch I_{DDQ} -Tests (\rightarrow 2.5.3) zur Detektion eingesetzt [14, 84].

stuck-at fault model. Erfahrungen zeigen, dass auf der Einzelfehlerannahme basierende Testmuster mit hoher Wahrscheinlichkeit auch Schaltungen mit mehreren stuck-at Fehlern als defekt erkennen [59]. Im Folgenden wird von der Einzelfehlerannahme ausgegangen.

¹¹Es wird vorausgesetzt, dass der Knoten durch die Schaltungsfunktion beide logische Werte annehmen kann, sodass sich ein Festhängen in einer Veränderung der Schaltungsfunktion auswirkt.

¹²Ein Brückenfehler wird aktiviert, wenn auf den beteiligten Leitungen unterschiedliche Werte eingestellt werden

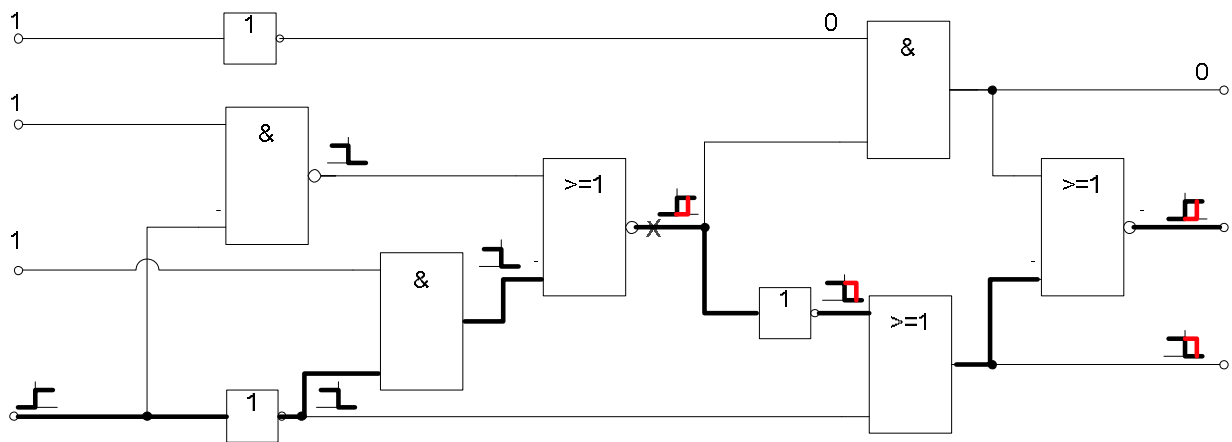


Abbildung 8: Der Übergangsfehler an x ist an 2 Schaltungsausgängen beobachtbar

2.5.2 Dynamische Fehlermodelle, Verzögerungsfehler

Im Unterschied zu statischen Fehlermodellen beschreiben dynamische Fehlermodelle die defektbedingte Veränderung des Laufzeitverhaltens einer Schaltung, in deren Folge nicht mehr alle Signalfanken innerhalb einer Taktperiode die Ausgänge erreichen; zum nachfolgenden Taktzeitpunkt werden an den Ausgängen falsche Werte übernommen. Wird der Schaltung über eine Verringerung der Taktrate ausreichend Zeit gegeben, alle Signale zu den Ausgängen zu propagieren, wird die Schaltungsantwort fehlerfrei sein. Daraus folgt, dass die durch das statische Haftfehlermodell modellierten Fehler als Untermenge der Verzögerungsfehler aufgefasst werden können, da ein Haftfehler das Umschalten eines Schaltungsknotens verhindert und damit zu einem unendlich großen Verzögerungsfehler für Signale durch diesen Schaltungsknoten führt.

Bei den dynamischen Fehlermodellen kann zwischen abstrakterer Fehlermodellierung und einer näher an den physikalischen Gegebenheiten liegenden unterschieden werden. Das Ziel der Abstraktion ist, den in der Schaltung vorliegenden Defekt mit möglichem geringem Aufwand zu detektieren. Eine Einschränkung erfährt die Abstraktion, wenn der Zielfehler die Auswirkung des Defektes nicht mehr modelliert. Im Falle dynamischer Fehlermodelle stellt das Übergangsfehlermodell die abstrakteste Beschreibung dar, während das Pfadverzögerungsfehlermodell den physikalischen Gegebenheiten am nächsten kommt. Diese Annäherung geht zu Lasten der Anzahl modellierter Fehler, die bereits für kleine Schaltungen zu groß werden können, um einen vollständigen Test zu ermöglichen [137].

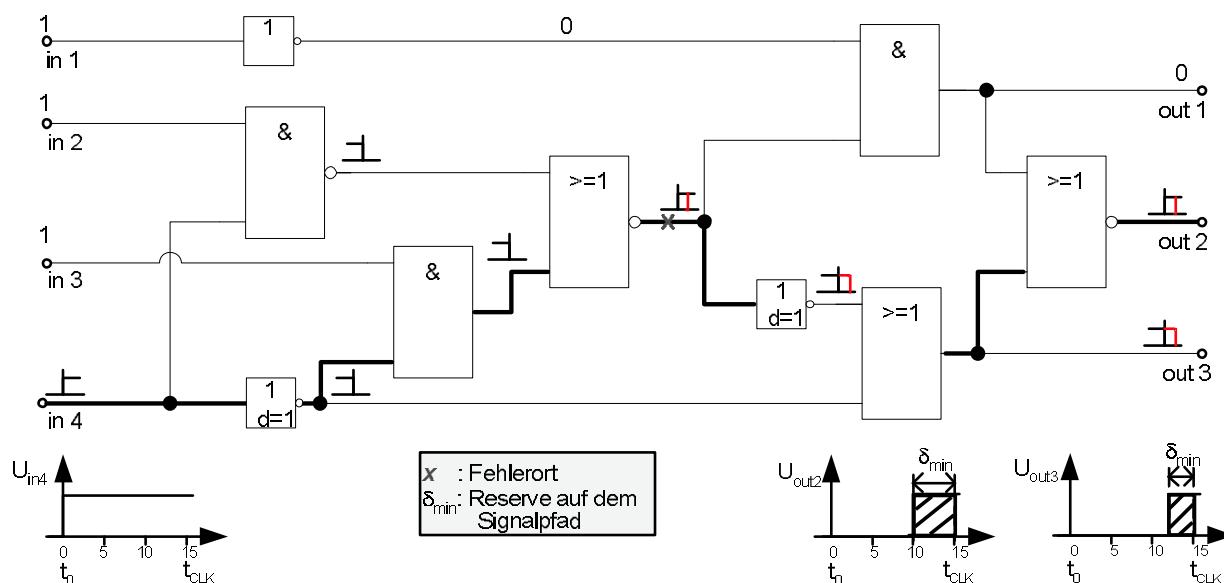


Abbildung 9: Auf längeren kritischen Pfaden verbleibt eine geringere Reserve

Übergangsfehlermodell (transition delay fault model, groß delay fault model) Das Übergangsfehlermodell [6, 57, 74, 122, 134] modelliert die Defektauswirkungen als eine lokal vorliegende Verzögerung einer fallenden (im folgenden mit “slow-to-fall fault” bezeichnet) oder einer steigenden Signalflanke (slow-to-rise fault). Als Fehlerorte werden die Gattereingänge und -ausgänge innerhalb der Schaltung angenommen und ihnen jeweils ein slow-to-rise und ein slow-to-fall fault zugeordnet.

Die durch den Verzögerungsfehler eingebrachte zusätzliche Verzögerungszeit wird als ausreichend groß angenommen, das Signal keinen der von ihm erreichbaren Ausgänge innerhalb der Taktzeit erreichen zu lassen und damit zum Taktzeitpunkt an jedem dieser Ausgänge detektierbar zu sein. Diese Annahme sorgt zusammen mit der lokalen Fehlerannahme für den höchsten Abstraktionsgrad innerhalb der Klasse der dynamischen Fehlermodelle. Als Folge der lokalen Fehlerannahme entspricht die Zahl der modellierten Fehler dem Doppelten aller Gatterein- und Gatterausgänge und ist damit linear von der Schaltungsgröße abhängig.

Gatterverzögerungsfehlermodell (gate delay fault model) Das Gatterverzögerungsfehlermodell [15, 63, 111] geht aus dem Übergangsfehlermodell hervor, wenn die detektierbare Fehlergröße berücksichtigt wird¹³. Ein Gatterverzögerungsfehler nimmt Einfluss auf die Schal-

¹³Die Laufzeit eines Signals entlang eines Signalpfades wird durch die Summe der Zeiten bestimmt welche die Gatter auf dem Pfad benötigen, um eine Wertänderung am Eingang an den Ausgang fortzuschalten.

tung, sobald die zusätzlich eingebrachte Verzögerung δ das beeinflusste Signal an zumindest einem Ausgang nach dem Taktzeitpunkt eintreffen lässt. Da die Taktfrequenz F der Schaltung (und damit die Taktzeit $t_{CLK} = \frac{1}{F}$) eine konstante Größe ist, kann über die Nominallaufzeit $t_{Nominal}$ eines beliebigen Signalpfades die Reserve (slack) s dieses Signalpfades definiert werden: $s = t_{CLK} - t_{Nominal}$. Der kleinste detektierbare Gatterverzögerungsfehler kann folglich über den längsten Signalpfad durch den Fehlerort detektiert werden, weil hier die zu übertreffende Reserve minimal wird. Im Beispiel aus Abbildung 9 kann ein Gatterverzögerungsfehler der Größe $]3..5]$ Zeiteinheiten an x nur am mittleren Schaltungsausgang erkannt werden.

Die Menge der vom Gatterverzögerungsfehlermodell modellierten Fehler entspricht der des Übergangsfehlermodells.

Leitungsverzögerungsfehlermodell (line delay fault model) Eine Abwandlung des Gatterverzögerungsfehlermodells stellt das Leitungsverzögerungsfehlermodell [82] dar. Bei Annahme der Leitungen als Fehlerort ist ein Test gesucht, welcher den Fehler auf dieser Leitung über den längsten sensitivierbaren Pfad durch diesen Fehlerort propagiert. Mit diesem Ansatz soll sichergestellt werden, dass die minimale Fehlergröße detektiert wird, weil der längste sensitivierbare Pfad jenen darstellt, der von allen zur Propagation des Fehlereffekts geeigneten Pfaden die geringste Reserve hat. Ferner wird jede Leitung der Schaltung auf eine zusätzliche, fehlerhafte Verzögerung getestet, während gleichzeitig die Fehlermenge im Vergleich zum Gatterverzögerungsfehlermodell auf das Doppelte der in der Schaltung enthaltenen Leitungen sinkt. Bei der Realisierung eines Tests nach dem Leitungsverzögerungsfehlermodell stellt sich die Bestimmung des längsten sensitivierbaren Pfades durch einen Schaltungsknoten als Problem heraus.

Durch die lokale Fehlerannahme zielen das Übergangs-, Gatter- und Leitungsverzögerungsfehlermodell auf sogenannte Spot-Defekte, also punktuell in der Schaltung vorliegende Abnormalitäten, ab, die zumeist in einer Verunreinigung begründet sein dürften. Das Pfadverzögerungsfehlermodell hingegen adressiert über die Schaltung verteilte Fehler, die ihren Grund eher in parametrischen Schwankungen haben dürften.

Pfadverzögerungsfehlermodell (path delay fault model) Das 1985 von G.L. Smith eingeführte Pfadverzögerungsfehlermodell [124] modelliert den zu detektierenden Fehler als verteilte, entlang eines Pfades eingebrachte zusätzliche Verzögerungen. Diese Verzögerungen können beliebig klein ausfallen und müssen als einzelne Verzögerung nicht ausreichen, eine

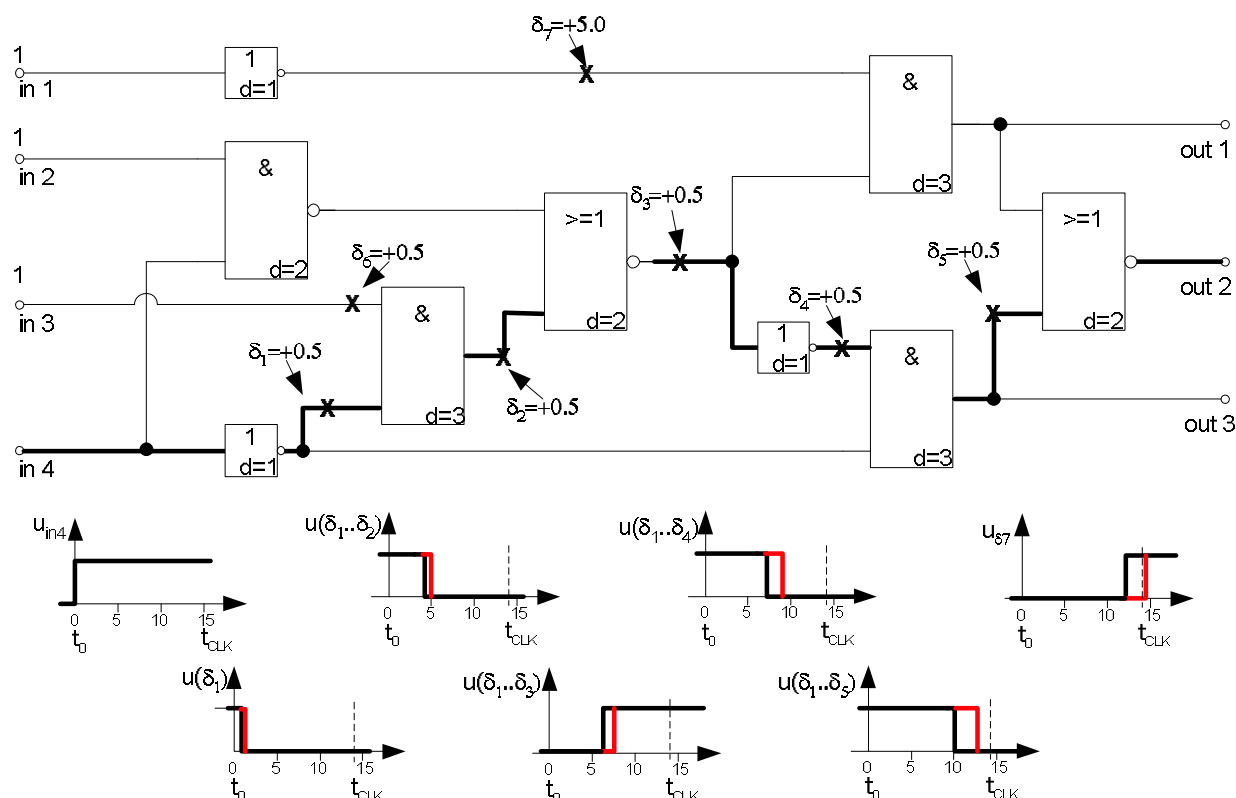


Abbildung 10: Verteilt eingebrachte Verzögerungen $\delta_1 - \delta_5$ führen zum Pfadverzögerungsfehler

Fehlfunktion der Schaltung zu bewirken. Durch die Summe der entlang eines Signalpfades eingebrachten fehlerhaften Verzögerungen kommt es jedoch zu einem verspäteten Eintreffen eines Signals am Pfadausgang. Die Menge der modellierten Fehler entspricht dem doppelten der Zahl aller strukturellen Pfade¹⁴ innerhalb der Schaltung, da für jeden Pfad eine steigende sowie eine fallende Signalfanke am Eingang angenommen werden kann.

Durch seinen im Vergleich zu anderen Verzögerungsfehlermodellen geringen Abstraktionsgrad ist die Menge der modellierten Fehler im Vergleich zu den anderen Verzögerungsfehlermodellen sehr groß und kann exponentiell mit der Zahl der in der Schaltung vorhandenen Gatter ansteigen [71]. Dadurch kann schon schon für Schaltungen mit einer relativ geringen Anzahl an Gattern ein vollständiger Test nach dem Pfadverzögerungsfehlermodell unmöglich werden. Es gibt verschiedene Ansätze, die Menge der zu testenden Pfade zu reduzieren.

Unter Berücksichtigung der Pfadlänge¹⁵ und der Taktzeit kann die zu testende Menge der Pfade

¹⁴Der Begriff *struktureller Pfad* beschreibt einen möglichen Weg, der von einem Schaltungseingang in Richtung der Ausgänge laufend zu einem Schaltungsausgang führt

¹⁵Die Länge eines Pfades definiert sich über die Zeit die ein Signal benötigt, ihn zu durchqueren. Da diese Verzö-

verringert werden. Über die Annahme einer Schranke kann der Test auf alle Pfade beschränkt werden, deren Länge diese Schranke übertrifft. Auf allen anderen Pfaden wird angenommen, dass etwaige zusätzliche Verzögerungen die Signale nicht über den Abtastzeitpunkt hinaus verzögern und damit kein Fehlverhalten der Schaltungsfunktion verursachen. Wenngleich das Pfadverzögerungsfehlermodell von geringen, über den Pfad verteilten Verzögerungen ausgeht, birgt dieser Ansatz die Gefahr, eine außergewöhnlich hohe Verzögerung auf einem dieser kürzeren Pfade nicht zu detektieren, wenn diese auf Leitungen eintritt, die nicht auch Teil der getesteten Pfade sind.

Ein anderer Ansatz der Pfadmengenreduzierung versucht, ausgehend von den längsten Pfaden innerhalb der Schaltung weitere, möglichst lange Pfade in die Menge der zu testenden Pfade aufzunehmen, welche Leitungen enthalten, die bislang nicht Teil eines zu testenden Pfades waren. Gelingt es, für jeden dieser Pfade einen Test zu finden, sind alle Leitungen zumindest einmal getestet worden. Es bleibt die Gefahr, dass Fehler nicht erkannt werden, die sich erst über eine andere, nicht berücksichtigte Leitungskombination auswirken.

Segmentverzögerungsfehlermodell (segment delay fault model) Das Segmentverzögerungsfehlermodell [54] kann als parametrierbares Verzögerungsfehlermodell betrachtet werden, dass in den beiden möglichen Extremen in das Pfadverzögerungsfehlermodell oder das Übergangsfehlermodell übergeht. Die Verzögerungen werden in Segmenten betrachtet, welche sich über die Anzahl L der Gatter auf dem längsten Pfad des Segmentes parametrisieren lassen, die sie enthalten. Die Verzögerung auf den Leitungen innerhalb der Segmente wird als ausreichend groß angenommen, um auf allen Pfaden durch das Segment messbar zu sein.

Für $L=1$ enthalten die Segmente Pfade mit maximal einem Gatter, was dem Übergangsfehlermodell entspricht. Für L_{max} = Anzahl der Gatter auf dem längsten Pfad der Schaltung sind alle Pfade der Länge 1 bis L_{max} enthalten, was dem Pfadverzögerungsfehlermodell entspricht. Ziel des Segmentverzögerungsfehlermodells ist es, die Vorteile der beiden Fehlermodelle zu vereinen, ohne deren Nachteile zu beinhalten.

gerung eine Funktion der auf dem Pfad liegenden Gatter ist, finden sich in der Literatur auch Textstellen, welche die Länge eines Pfades über die Anzahl der Gatter definieren, die ihn bilden. Soweit nicht anders angegeben, wird nachfolgend der Begriff der *Pfadlänge* und der *Nominallaufzeit* auf diesem Pfad synonym verwendet.

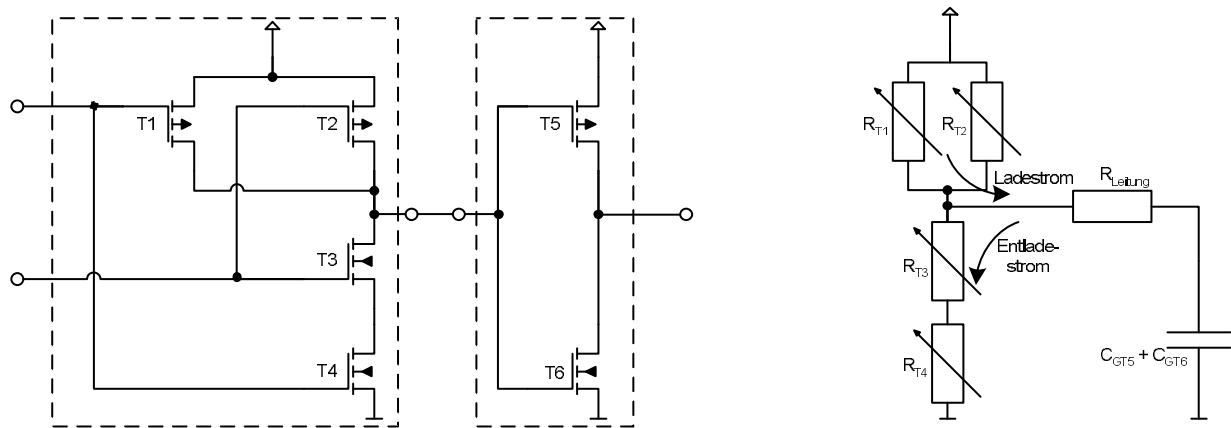


Abbildung 11: NAND-Gatters als treibendes Gatter und Inverter als Last: Transistornetzliste (links) und Ersatzschaltbild (rechts)

2.5.3 Strombasierte Testmethoden

Die bislang angeführten Fehlermodelle nutzen die Spannung als Messgröße zur Fehlerdetektion. Darüber hinaus kann auch der Versorgungsstrom als Messgröße dienen. Die I_{DDQ} -Messung [75] setzt auf der für (ideale) CMOS-Logik charakteristischen Eigenschaft auf, dass im Ruhezustand keine Querströme in der Schaltung fließen. Defekte, die beispielsweise zu resistiven Kurzschlüssen führen oder sich in einem nicht vollständig schließenden Transistor äußern können, je nach Schaltungszustand, zu ungewollten Leckströmen führen. Diese Leckströme stellen auch dann einen Qualitätsmangel dar, wenn sie die (logische) Schaltungsfunktion nicht beeinträchtigen. Die ungewollte Leistungsaufnahme kann für eine verkürzte Betriebsdauer leistungskritischer Anwendungen (Mobiltelefone, Pocket Computer, Herzschrittmacher) sorgen, zum anderen erfährt die Schaltung einen ungewollten Stress am Fehlerort, der zu einer verkürzten Lebensdauer führen kann. Die Testmuster-generierung für I_{DDQ} -Fehler geht von einem Pseudo-Haftfehlermodell aus. Nähere Diskussionen finden sich in [2].

Die dynamischen Varianten der Strommessmethode sind die I_{DDT} -Testmethoden [125], die sich vor allem in den Ansätzen der Meßwertauswertung unterscheiden [35].

2.6 Signallaufzeiten

Die Laufzeiten der Signale auf den Pfaden innerhalb einer Schaltung sind von vielen Parametern abhängig. Wird von einer in CMOS-Technologie realisierten Schaltung ausgegangen, können die

internen Schaltvorgänge als Lade- bzw. Entladevorgänge der Gatekapazitäten der zu treibenden Gatter aufgefasst werden. Die Signallaufzeiten entlang eines Pfades ergeben sich aus den Schaltzeiten der einzelnen Gatter auf diesem Pfad und sind damit direkt von den Umladezeiten der Gatekapazitäten abhängig. Ein Umladevorgang ist umso schneller abgeschlossen, je niedriger der Lade- bzw. Entladewiderstand sowie die umzuladene Kapazität ist. Dieser Umladewiderstand setzt sich zum einen aus dem Widerstand der Leitung, zum anderen aus den Widerständen der Transistoren, die am Umladevorgang beteiligt sind (Abbildung 11), zusammen.

- Der Widerstand einer Leitung ergibt sich aus dem spezifischen Widerstand des verwendeten Materials sowie ihrer geometrischen Abmaße. Sollte die Leitung über mehrere Verdrahtungsebenen verlaufen addieren sich die Übergangswiderstände der Vias¹⁶ hinzu.
- Die Widerstände, welche die am Umladevorgang beteiligten Transistoren einbringen, nehmen ebenfalls Einfluss auf die Schaltzeiten des Gatters, weshalb die Umschaltgeschwindigkeit eines logischen Gatters auch eine Funktion seines Eingangsmusters ist [89].
- Die Größe der umzuladenen Kapazitäten steigt mit der Anzahl der zu treibenden Gatter, da alle umzuladenen Gatekapazitäten parallel liegen.

Eine genauere Bestimmung der Signallaufzeiten kann erst mit dem Wissen um das Layout der Schaltung geschehen, da Parameter wie Leitungslänge oder parasitäre Kapazitäten sich mit dem Layout verändern. Bei einer Testmuster generierung liegen diese Daten zumeist nicht vor, weswegen einige Kunstgriffe unternommen werden, um die anzunehmenden Laufzeiten zu beschreiben. Eine erste Näherung besteht in der Annahme einer Einheitsverzögerung für jedes Gatter [102]. Diese Annahme kann nur als grobe Näherung dienen. Eine Verfeinerung bietet eine Gewichtung, welche die Einheitsverzögerung mit Anzahl der zu treibenden Gatter gewichtet, um den Einfluss verschiedener Kapazitäten zu modellieren. Für die Ermittlung der mit dem Pfadverzögerungsfehlermodell zu untersuchenden Pfade werden häufig statische Timinganalytoren eingesetzt [81], die auf eine Bibliothek des verwendeten Prozesses zurückgreifen, in welcher charakteristische Schaltzeiten der zur Verfügung stehenden Gatter abgelegt sind. Wenngleich der Einsatz statischer Timinganalytoren durch die Beachtung prozeßspezifischer Eigenschaften eine hohe Genauigkeit erlangt, kann auch ihr Einsatz weder die layoutbedingten noch die durch Eingangssignalkombinationen bedingten Laufzeiten genau modellieren [49].

¹⁶Mit Via werden Kontaktübergänge zwischen metallenen Leiterbahnebenen bezeichnet. Mit steigender Integrationsdichte nimmt die Anzahl dieser Übergänge zwangsläufig zu.

2.7 Testmustergenerierung und Fehlersimulation

Neben dem Zielfehler Φ deckt ein Testmuster T weitere Fehler ab. Im Falle des Haftfehlermodells sind entlang des kritischen Pfades P^k alle Haftfehler Φ_k^H getestet, die dem Sollwert v_k am entsprechenden Knoten entgegengesetzt sind. Weiterhin können für einen beliebigen Zielfehler in einer Schaltung verschiedene Testmuster existieren, weshalb die Aufgabe der Testmustergenerierung umso schwieriger wird, je weniger Testmuster für den Zielfehler existieren. Andererseits steigt die Wahrscheinlichkeit, dass ein Fehler von einem beliebigen Testmuster erkannt wird, mit der Zahl der unterschiedlichen Testmuster, die für ihn existieren. Das Ziel einer optimalen Testmustererzeugung muss demnach sein, die schwer detektierbaren Fehler zuerst zu adressieren, um die leichter zu detektierenden Fehler über die berechneten Testmuster mit abzudecken.

Die Bestimmung eines Testmusters für einen Zielfehler ist Aufgabe des Testmustergenerators, während ein Fehlersimulator für die Bestimmung der Fehlerabdeckung eines gegebenen Testmusters eingesetzt wird. Ein Testmustergenerationssystem besteht aus beiden Komponenten, um nach einer erfolgreichen Testmustergenerierung die zusätzlich abgedeckten Fehler zu erkennen und aus der Fehlerliste streichen zu können.

Die Kombination aus Testmustergenerator und Fehlersimulator führt zu einer schnell wachsenden Fehlerabdeckung zu Beginn der Testmustergenerierung. Dieser Anstieg geht auf die durch die Fehlersimulation erkannten Fehler zurück. Mit fortschreitender Zahl an Testmustern sinkt der Teil der zusätzlich erkannten Fehler. Für die schwer detektierbaren Fehler, für deren Erkennung nur wenige Testmuster existieren, werden die Testmuster überwiegend durch eine gezielte Testmustergenerierung gefunden. Die Fehlersimulation erfordert, unabhängig vom Fehlermodell, im Vergleich zur Testmustergenerierung einen erheblich geringeren Berechnungsaufwand, da keine Entscheidungsfindung und Entscheidungsrücknahme durchzuführen ist. Aus diesem Grund kann eine Testmustergenerierung mit Zufallstestmustern beginnen und die erreichte Fehlerabdeckung per Simulation ermittelt werden. Über die Definition einer Schranke für den je Testmuster zu erreichenden Gewinn an Fehlerabdeckung ($\Delta FC \geq \text{Schranke}$) kann ein Punkt gefunden werden, ab dem die deterministische Testmustergenerierung einsetzt (Abbildung 12).

2.8 Testklassen dynamischer Fehlermodelle

Während ein Testmuster für einen statischen Fehler diesen detektiert, existiert bei dynamischen Fehlermodellen eine Abstufung der Testqualität. Grund hierfür sind mögliche Signalwechsel innerhalb der Schaltung während des Tests, die über die Seiteneingänge des kritischen Pfades Ein-

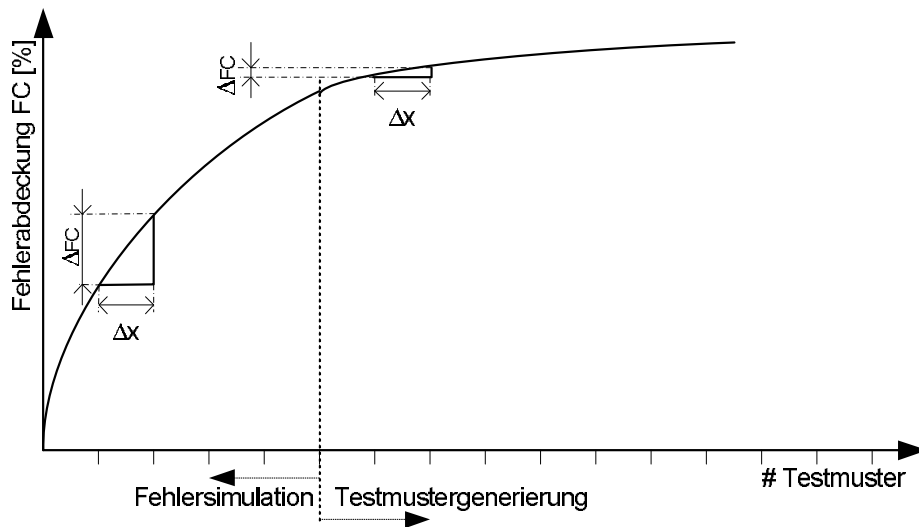


Abbildung 12: Die Fehlerabdeckung steigt zu Beginn der Testmustergenerierung stark an

fluss auf das Testergebnis nehmen können. Die meist verbreitete Einteilung unterscheidet Tests nach robusten und nicht robusten Tests.

Robuste und nicht robuste Seiteneingangsbelegungen

Für die Definition der Eigenschaften eines Signals an einem Seiteneingang des kritischen Pfades kann der Begriff kontrollierender bzw. nicht kontrollierender Wert eines Gatters definiert werden. Ein Wert v am Eingang eines Gatters heißt genau dann *kontrollierender Eingangswert*¹⁷ cv , wenn er den Wert am Ausgang des Gatters festlegt. Ein Eingangswert heißt genau dann *nicht kontrollierender Wert* ncv , wenn der Ausgang des Gatters weiterhin eine Funktion der übrigen Eingänge bleibt.

Für alle auf dem kritischen Pfad P^k befindlichen Gatter kann ein auf dem kritischen Pfad liegender Eingang angegeben werden, der im folgenden mit *Pfadeingang* bezeichnet wird. Alle anderen Eingänge werden als Seiteneingänge bezeichnet. Die Seiteneingänge eines kritischen Pfades teilen sich, abhängig von den Eigenschaften des Signals an diesem Eingang, in robuste oder nicht robuste Seiteneingänge auf. Ein Seiteneingang heißt genau dann robuster Seiteneingang, wenn er

- einen statischen ncv führt oder

¹⁷im folgenden wie in der Literatur üblich mit "kontrollierender Wert" bezeichnet

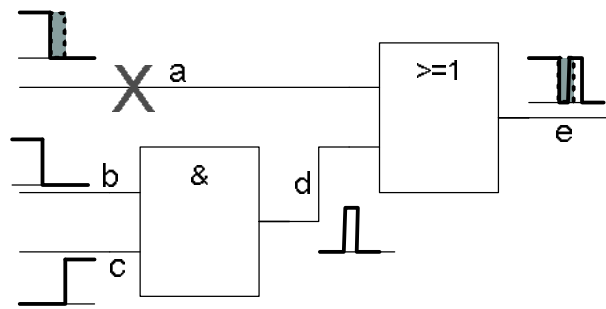


Abbildung 13: Das Eingangsmuster bildet ein robusten, aber hazardbehafteten Test für den Fehler an x

- im Falle einer Ausgangsflanke $cv \rightarrow ncv$ ebenfalls eine Flanke $cv \rightarrow ncv$ führt [71]¹⁸.

Ein Seiteneingang wird genau dann *nicht robuster Seiteneingang* genannt, wenn er eine Signalflanke $cv \rightarrow ncv$ führt, während der auf dem Pfad liegende Eingang des Gatters eine Signalflanke $ncv \rightarrow cv$ einnehmen soll [71].

2.8.1 Robuster Test

Ein Test wird *robuster Test* für einen Verzögerungsfehler Φ genannt, wenn er den Fehler unabhängig von Laufzeiten auf anderen Pfaden (und damit auch unabhängig von der Existenz weiterer Fehler) innerhalb der Schaltung detektiert [71]. Dies ist der Fall, wenn es sich bei allen Seiteneingängen des Pfades um nicht robuste Seiteneingänge handelt. Die generelle Aussage, dass ein robuster Test statische und dynamische Hazards an dem beobachteten Ausgang ausschließt [139], kann jedoch nur für eine Untermenge der robusten Tests gelten [121]. Das Eingangsmuster in Abbildung 13 ist ein robuster Test für den Verzögerungsfehler einer fallenden Flanke an Leitung a, obwohl er einen Hazard aufweist. Da der Hazard jedoch den Ausgang auf den im Fehlerfall erwarteten Wert einstellt wird ein Fehler erkannt, wenn zum Abtastzeitpunkt der Hazard auftritt; eine Fehlermaskierung tritt daher nicht auf.

¹⁸Es bleibt zu erwähnen, dass die Belegung der Seiteneingänge eines auf dem kritischen Pfad liegenden Gatters mit dem nicht kontrollierenden Wert auch als *fehlerleitende Belegung* bezeichnet wird, da allein der Wert am Pfadeingang den Ausgangswert des Gatters bestimmt.

2.8.2 Nicht robuster Test

Ein Test wird *nicht robuster Test* für einen Fehler Φ genannt, wenn er diesen Fehler nur bei Abwesenheit etwaiger weiterer, in der Schaltung vorhandenen Fehler sicher detektiert [14]. Nach [27] ist ein Vektorpaar genau dann ein nicht robuster Test für einen (Pfad-) Verzögerungsfehler, wenn es die notwendigen Flanken auf dem (kritischen) Pfad einstellt und es sich bei zumindest einem Seiteneingang um einen nicht robuster Seiteneingang handelt, während alle anderen robuste Seiteneingänge sind. Robuste Tests bilden eine spezielle Untermenge der nicht robusten Tests (die Anwendung der notwendigen Bedingungen zur Berechnung eines nicht robusten Tests können auch zu robusten Testmustern führen, → Kapitel 6).

2.8.3 Testklassen

Während sich die Definitionen für robuste und nicht robuste Tests auf jedes dynamische Fehlermodell anwenden lassen, existieren weitere, speziell auf das Pfadverzögerungsfehlermodell abgestimmte Testklassen. Der Grund für die Entwicklung dieser speziellen Definitionen liegt in der großen Menge der durch das Pfadverzögerungsfehlermodell modellierten Fehler. Da ein vollständiger Test zumeist unmöglich ist stellt sich die Frage, welche der Pfade einen Test erfordern, um eine Aussage über die Fehlerfreiheit der Schaltung machen zu können.

Validatable non-robust testable path delay faults definiert [116]. Prinzipiell muss ein nicht robuster Pfadverzögerungsfehlerstest für einen Pfad P^K mindestens einen nicht robusten Seiteneingang i aufweisen. Ein *validatable non-robust testable path delay fault* [116] liegt vor, wenn (für jeden nicht robusten Seiteneingang von P^K) die Menge aller Pfade, die von den primären Schaltungseingängen zu i führen, über den verbleibenden Rest von P^K robust getestet sind, da auf ihnen damit kein Fehler mehr vorliegen kann, der den nicht robusten Test invalidieren kann.

Functional sensitizable path delay faults [25] treten nur im Zusammenspiel mehrerer Verzögerungsfehler in Erscheinung. Der bei fehlerfreier Schaltungsfunktion nicht funktionale Pfad wird in einen funktionalen Pfad überführt, wenn die Flanken an den Seiteneingängen verspätet eintreffen.

Robust dependent path delay faults [73] bezeichnen die Menge aller Pfade, auf denen ein Verzögerungsfehler zu einem Fehlverhalten der Schaltung führt.

Singly-testable path delay faults [45] enthalten alle Fehler, die unter der Einzelfehlerannahme sicher detektiert werden können.

Primitive path delay faults [68] enthalten alle Fehler, deren Test für Sicherstellung der korrekten Schaltungsfunktion notwendig sind.

2.9 Algorithmen zur Testmuster generierung

Die Testmuster generierung verfolgt, unabhängig vom Fehlermodell, zwei generelle Ziele. Zunächst muss der fehlerhafte Knoten -dem Fehler entsprechend- derart initialisiert werden, dass sich dort im fehlerhaften und fehlerfreien Fall unterschiedliche Werte einstellen. Das zweite Ziel ist die Fortschaltung des Fehlereffekts zu einem beobachtbaren Ausgang, um detektiert werden zu können. Die Testmuster generierung kann über verschiedene Ansätze verfolgt werden. Als akademisch interessant, aber für die Implementierung auf einem Computer ungeeignet, wird die Boolesche-Differenz Methode bewertet [61]. Der erste Versuch, den bis dahin ausschließlich verwendeten funktionalen Test durch strukturelle Testmethoden zu ersetzen, wird einem Paper aus dem Jahre 1959 [33] zugesprochen.

Die Algorithmen zur Testmuster generierung können, wenngleich sie mit Blick auf das Haftfehlermodell vorgeschlagen wurden, ebenso bei der Testmuster generierung für andere Fehlermodelle verwendet werden, wenn die veränderten Eigenschaften über die Verwendung einer modifizierten Logik Beachtung finden (vergl. auch 5).

2.9.1 D-Algorithmus

Der 1966 von J.P. Roth publizierte D-Algorithmus [117] bildet noch heute die Grundlage aller modernen Algorithmen der Testmuster generierung. Er verwendet eine 5-wertige Logik, die neben den Symbolen $\{0,1,X\}$ die beiden zusätzlichen Symbole $\{D, \bar{D}\}$ enthält. Diese in der Literatur auch als *discrepant value* bezeichneten Werte [129] unterscheiden sich im fehlerbehafteten und fehlerfreien Fall¹⁹ und ermöglichen damit die Erkennung eines Fehlereffektes. Über

¹⁹Ein Schaltungsknoten, der den Wert D führt, nimmt im Fehlerfall eine 0, im fehlerfreien Fall eine 1 an.

eine entsprechende Algebra wird es möglich, die in der Schaltung notwendigen Signale in Abhängigkeit der logischen Funktion eines Gatters zu berechnen. Der D-Algorithmus läuft generell in 3 Schritten ab. Nach der Injektion des Fehlers in die Schaltung wird eine Liste aller Gatter aufgebaut, über welche der Fehlereffekt an einen Ausgang propagiert werden kann (die sogenannte D-Front). Es folgt die Bestimmung eines kritischen Pfades vom Fehlerort zu einem Ausgang. Dazu wird ein Gatter aus der D-Front ausgewählt, fehlerleitend belegt und die D-Front anschließend aktualisiert. Der Vorgang wird solange fortgesetzt, bis ein kritischer Pfad gefunden wurde. Im dritten Schritt wird eine konsistente Eingangsbelegung der Schaltung gesucht, die auf jeder Leitung den Wert einstellt, der auf ihr durch die beiden vorhergehenden Schritte gefordert wurde. Diese Anforderungen ergeben sich immer dann, wenn für die Einstellung eines notwendigen Wertes am Ausgangsknoten eines Gatters mehrere mögliche Eingangsbelegungen existieren. Im dritten Schritt kommt es folglich zu Entscheidungen, die zu neuen Anforderungen an vor dem Gatter liegende Knoten führen. Führen die Entscheidungen zu nicht auflösbaren Konflikten, müssen sie zurückgenommen und andere Eingangsbelegungen gewählt werden. Gelingt die Einstellung aller Anforderungen, ist ein Test gefunden. Ist der gesamte Suchraum ohne eine Lösung durchlaufen worden, fällt der D-Algorithmus zum zweiten Schritt zurück und versucht, einen anderen kritischen Pfad zu finden. Lassen sich für keinen der kritischen Pfade alle Anforderungen auflösen, ist der Fehler nicht testbar.

2.9.2 PODEM

Der 1981 vorgestellte PODEM²⁰ Algorithmus [47] brachte neue Ideen in die Testmustergenerierung ein und konnte seine Stärke gerade bei den in der Speichertechnik notwendigen Fehlerkorrekturschaltungen ausspielen. Der PODEM-Algorithmus beschränkt seine Entscheidungen auf die Eingänge der Schaltung und ermittelt die daraus folgenden Signale über eine Logiksimulation des Eingangsmusters. Durch die ausschließlich an den Eingängen getroffenen Entscheidungen wird der Entscheidungsbaum im Vergleich zum D-Algorithmus stark verkleinert, bei dem vorher an jedem Schaltungsknoten eine Entscheidung möglich war.

2.9.3 FAN

Der 1983 publizierte FAN-Algorithmus [42] gehört bis heute zu den leistungsfähigsten Algorithmen. Seine Konzepte versuchen, Implikationen möglichst früh zu erkennen, um mögliche Widersprüche aufzulösen.

²⁰PODEM: Path Oriented DEcision Making

Sofortige Implikationen Aus notwendigen Wertsetzungen innerhalb der Schaltung folgen Anforderungen an Eingänge betroffener Logikgatter, um die gewünschten Signalwerte am Ausgang einzustellen. Der PODEM-Algorithmus versucht, eine Anforderung über eine Rückverfolgung der daraus folgenden Anforderungen bis an die Eingänge zu propagieren, um dort eine Entscheidung zu treffen. Mögliche Konflikte werden erst nach einer Simulation bemerkt. Im Unterschied dazu versucht der FAN-Algorithmus, eine Anforderung durch eine Wertsetzung an den betroffenen Schaltungseingängen zu erfüllen und führt in deren Folge die aus den Anforderungen entstehenden Implikationen sofort aus. Es zeigt sich, dass aus den Wertsetzungen folgende Widersprüche dadurch eher erkannt werden.

Multiple Backtrace Die durch die notwendigen Zuweisungen erhaltenen Anforderungen an Signalwerte auf Leitungen innerhalb einer Schaltung müssen durch eine Eingangsbelegung eingestellt werden. Zu diesem Zweck versuchen PODEM und FAN, diese Anforderungen an Gatterausgängen auf Anforderungen an den Gattereingängen abzubilden. Der FAN-Algorithmus bildet die Anforderungen dabei auf jeden Eingang eines Gatters ab, während PODEM möglichst schnell einen Eingang zu erreichen versucht. Die Abbildung der Anforderungen auf alle Gattereingänge kann zu verschiedenen Anforderungen an den Zweigen eines Verzweigungspunktes führen, sodass auch hier Konflikte früher erkannt werden.

Headlines Die Aufgabe der Testmustergenerierung besteht darin, die auf internen Knoten vorliegenden Signalwertanforderungen durch eine konsistente Eingangsbelegung zu erfüllen. Hierzu sind Entscheidungen zu fällen, welche der möglichen Optionen versucht werden soll. Um diese Entscheidungsfindung zu unterstützen, wurden für die Testmustergenerierung die zwei Eigenschaften “Beobachtbarkeit” und “Einstellbarkeit” definiert. Sie sind ein Maß, wie schwer sich der Signalwert einer Leitung beeinflussen oder beobachten lässt. Für die Bestimmung dieser Größen wurden verschiedene Algorithmen (CAMELOT [8], COMET [10], SCOAP [48], VICTOR [114]) veröffentlicht. Der allen Algorithmen zugrunde liegende Gedanke ist, dass sich ein Signal in der Schaltung umso leichter einstellen lässt, je weniger Schaltungseingänge auf den betroffenen Knoten Einfluss nehmen. Damit verbessert sich die Einstellbarkeit eines Knotens mit seiner Nähe zu den Schaltungseingängen. Umgekehrt ist ein Knoten umso leichter beobachtbar, je weniger Gatter zwischen ihm und einem beobachtbaren Ausgang transparent geschaltet werden müssen. Die Beobachtbarkeit eines Knotens steigt also mit seiner Nähe zu den Schaltungsausgängen.

Der FAN-Algorithmus verwendet das Konzept der Einstellbarkeit derart, dass er die in der Schal-

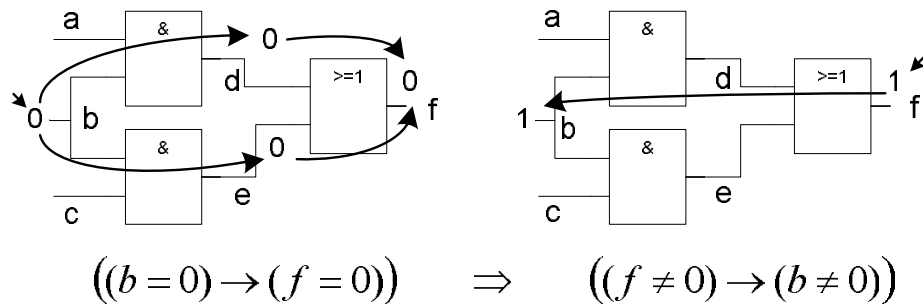


Abbildung 14: Das Kontrapositionsgesetz offenbart globale Implikationen

tung vorhandenen Leitungen in drei Klassen einteilt, die mit *Freelines*, *Headlines* und *Boundlines* bezeichnet werden. Jede Leitung, die einem primären Eingang folgt, ist von außen frei einstellbar. Gleiches gilt für alle Leitungen, die von einem primären Ausgang aus erreichbar sind, ohne vorher einen Verzweigungspunkt zu durchlaufen. Alle Leitungen, die diese Bedingung erfüllen, gehören zu der Menge der *Freelines*. Für die Einstellbarkeit aller Leitungen, die hinter einem Verzweigungspunkt liegen, müssen eventuelle Auswirkungen auf die über die anderen Verzweigungen beeinflussten Leitungen berücksichtigt werden. Da diese Leitungen damit nicht mehr unabhängig zu belegen sind, werden sie mit *Boundlines* bezeichnet. Die Leitungen, die den Übergang der Menge der *Freelines* zu der Menge der *Boundlines* bilden und noch frei einstellbar sind, werden *Headlines* genannt. Für die Testmustergenerierung sind diese *Headlines* von besonderer Bedeutung, da die in der Schaltung bestehenden Anforderungen nur bis zu diesen *Headlines* in Richtung der Eingänge abgebildet werden müssen. Als Folge kann die Testmustergenerierung versuchen, eine konsistente Belegung der *Headlines* zu erreichen, um diese in einem abschließenden Schritt durch eine Eingangsbelegung einzustellen.

Zusammengefasst zeigen experimentelle Ergebnisse, dass der FAN-Algorithmus zu einer geringeren Anzahl an Backtracks führt und folglich schneller zu einer Lösung gelangt als der PODEM-Algorithmus [42].

2.9.4 Recursive learning

1987 verwendete M. Schulz in seinem Testmustergenerator SOCRATES [123] einen als *Lernen* bezeichneten Ansatz, um sogenannte globale Implikationen zu bestimmen. In der in Abbildung 14 dargestellten Schaltung ergeben sich aus der Zuweisung $b=1$ lokale (über die Funktion des Gatters bestimmte) Implikationen ($d=1$, $e=1$, $f=1$). Eine weitere, globale Implikation folgt aus dem Kontrapositionsgesetz $a \rightarrow b \implies \bar{b} \rightarrow \bar{a}$. Für das Beispiel aus Abbildung 14 lautet die

globale Implikation $f=0 \rightarrow b=0$. Diese globale Implikation kann durch lokale Betrachtungen nicht gewonnen werden, weil aus $f=0$ keine weiteren lokalen Implikationen für d und e folgen.

Basierend auf diesem Ansatz des Lernens stellte Kunz einen rekursiven Lernansatz vor [72]. Von einer Anfangsbelegung innerhalb der Schaltung ausgehend wird gezeigt, dass sich bei einer genügenden Rekursionstiefe alle aus der Anfangsbelegung folgenden Implikationen bestimmen lassen. Das rekursive Lernen, als Ansatz zur Testmuster-generierung gedacht, hat sich in der Praxis durch den exponentiell mit der maximale Rekursionstiefe steigenden Zeitbedarf [14] nicht durchgesetzt, findet jedoch (mit begrenzter Rekursionstiefe) als Präprozess zur Verbesserung der Erkennung von Nichtlösungsgebieten im Suchraum Anwendung.

2.9.5 Decision diagrams

Decision diagrams bieten eine weitere Möglichkeit, eine Testmuster-generierung zu realisieren [7]. Dem Vorteil, dass es mit diesen Verfahren möglich ist, alle Testmuster zu berechnen, die für einen Fehler existieren, steht ihre Beschränkung auf moderate Schaltungsgrößen entgegen. Da kein BDD²¹ basiertes Verfahren bekannt ist, das sich auf Schaltungen aktueller Größe anwenden lässt, kommen sie im betrachteten Umfeld nicht weiter in Betracht.

2.9.6 Genetische Verfahren

Die Testmuster-generierung mit Hilfe von genetischen Verfahren versucht, ausgehend von einer Anzahl vorgegebener, möglicher Eingangskombinationen, über eine Simulation jene zu bestimmen, der die größte Erfolgchance eingeräumt werden kann. Während, ausgehend von diesem Muster, eine neue Zahl an Kandidaten aus diesem hervorgehen, werden die anderen Muster nicht weiter berücksichtigt. Der größte Nachteil der genetischen Verfahren ist, dass sie den Suchraum nicht systematisch durchlaufen. Folglich kann für den Fall, dass kein Testmuster für einen Fehler gefunden werden kann, nicht bewiesen werden, dass keines existiert.

2.10 Applikationsmodi

Für die Durchführung eines Tests auf Verzögerungsfehler sind verschiedene Abläufe möglich. Unter der Annahme eines *Full-Scan Designs*²² ist die Zuführung des pseudo primären Anteils

²¹BDD: *binary decision diagram*

²²Sind alle im Design verwendeten Speicherelemente als Scan Speicherelemente ausgeführt und in Scan-Ketten verschaltet, wird von einem Full-Scan Design gesprochen. Enthält das Design sowohl Scan-Ketten als auch

des Initialisierungsvektors V_I über Schiebeoperationen unproblematisch. Der zweite Testvektor muss dem ersten im nachfolgenden Takt folgen, um die für den Test erforderliche Flanke in der Schaltung einzustellen. Wird die Scan-Kette aus normalen Scan-Flip-Flops aufgebaut, ist ein serielles Eintakten des zweiten Testvektors nicht möglich, da sich während der Schiebeoperationen die Ausgänge der Speicherelemente verändern und diese Änderungen an die Schaltung gelangen. Die Verwendung sogenannter *Enhanced Scan-Flip-Flops* [29] ermöglicht ebenfalls ein serielles Einlesen des zweiten Testvektors. Die Verwendung dieser speziellen Speicherelemente ist jedoch für die Serienproduktion in der Industrie aufgrund ihres Flächenbedarfs nicht akzeptiert. Der Ersatz der für die Schaltungsfunktion notwendigen D-Flip-Flops durch für den Test geeignete Scan-Flip-Flops resultiert durch den eingebrachten Multiplexer bereits in einem zusätzlichen Flächenbedarf von etwa 20% je Flip-Flop [14] sowie zusätzlichem Routingaufwand. Der Einsatz von Enhanced Scan-Flip-Flops führt, je nach Realisierung, zu einem mehr als verdoppelten Flächenbedarf der Speicherelemente oder einer zusätzlichen Verzögerung im Signalpfad. Weil die Speicherelemente ausschließlich für den Test verwendet werden und im normalen Betrieb keinen Vorteil für die Funktion der Schaltung bieten, finden Enhanced Scan-Flip-Flops im allgemeinen keine Verwendung in der Serienfertigung.

Wird als Prüfling eine Schaltung im Standard-Scan-Design vorausgesetzt, bieten sich für den Test zwei Alternativen für die Zuführung des zweiten Testvektors V_P an, die mit *Scan-Shift-Modus*²³ [119, 120] bzw. *funktionale Einstellung*²⁴ [121] bezeichnet werden.

Scan shifting

Die erste Möglichkeit, den Testvektor V_P zu erhalten, besteht in einem weiteren Verschieben des bereits in der Scan-Kette befindlichen Testvektors V_I um eine Position (Abbildung 15, links). Die Verwendung des Scan-Shift Testmodus im Produktionstest stellt einige Anforderungen an den Testautomaten und die Schaltung. Die Speicherelemente haben die Verschiebeoperationen zu Beginn des zweiten Taktzyklus und das Speichern der Schaltungsantwort am Ende des Taktzyklus zu realisieren. Folglich müssen sie während des Taktzyklus vom Testmodus in den Normalmodus umgeschaltet werden, was hohe Anforderungen an Design und Testequipment stellt. Die Verwendung sogenannter *low cost tester* ist für den Scan-Shift Modus nicht möglich, weil der Test neben dem Taktsignal auch für das Testmode-Signal einen hochgenauen Kanal benötigt

einfache Speicherelemente, handelt es sich um ein *Partial-Scan Design*.

²³in der Literatur auch bezeichnet mit *skewed load, launch-from-shift*

²⁴in der Literatur auch bezeichnet mit *functional justification, broadside test, launch-from-capture*

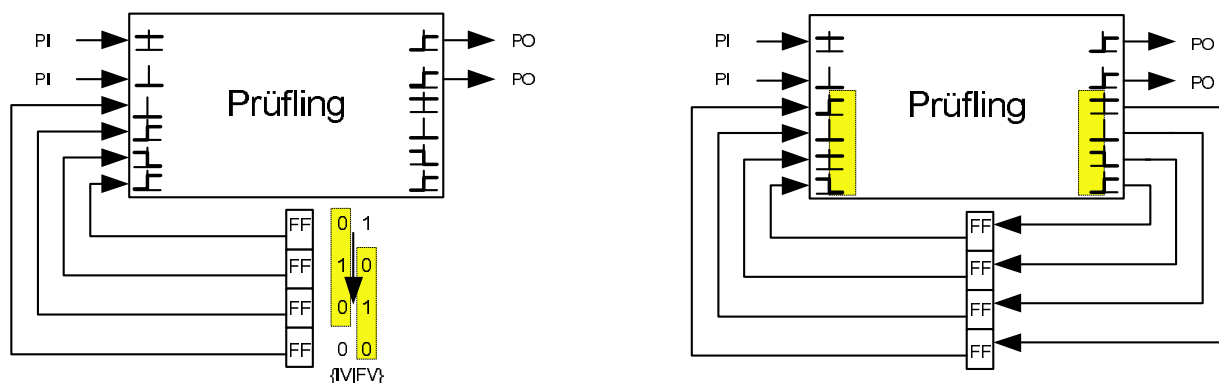


Abbildung 15: Scan-Shift Modus (links), funktionale Einstellung (rechts)

[64]. Dies ist mit low-cost Testautomaten nicht zu realisieren, weil die Kosten eines Testautomaten erheblich von der Zahl der bereitgestellten Breitbandkanäle abhängt [14].

Ein weiteres Problem ergibt sich bei der Frage der Fehlerabdeckung. Die zu testende Schaltung kann als Automat mit oder ohne primäre Eingänge vorliegen, wobei die Rückführungen in Abbildung 15 den internen Zustand des Automaten repräsentieren. Da dieser nicht notwendigerweise alle möglichen Zustände annehmen können muss, stellt sich in Bezug auf die Fehlerabdeckung von Tests im Scan-Shift-Modus die Frage, ob die getesteten Fehler (bzw. die in diesem Modus nicht testbaren Fehler) im bestimmungsgemäßen Betrieb der Schaltung vorkommen können und wie damit die Fehlerabdeckung des Tests zu bewerten ist (siehe auch [115]). Für den Fall, dass V_I an den Zustandsleitungen eine im normalen Betrieb nicht vorkommende Signalkombination einstellt, können beim Test Pfade aktiviert werden, die im Normalbetrieb keine Signalleitung ermöglichen. Detektiert der Test Laufzeitfehler auf diesen nicht funktionalen Pfaden, ist dies für die Funktion der produzierten Schaltung kein Mangel, sie könnte verkauft werden. Gleiches gilt für den Fall, dass der Eingangssignalwechsel an den pseudo primären Eingängen, wie ihn der um 1-Bit verschobene V_P erzeugt, im Normalbetrieb nicht auftreten kann.

Die letzte wichtige Eigenschaft des Scan-Shift-Tests ist, dass eine Abhängigkeit der Testmuster von der Reihenfolge, in der die Scan-Flip-Flops in der Scan-Kette verschaltet sind, besteht, wodurch auch die Testabdeckung und Testqualität beeinflusst werden. Ein Beispiel zeigt Abbildung 16. Unter Annahme des Pfadverzögerungsfehlermodells folgt aus einer gewünschten fallenden Flanke am oberen Eingang des UND-Gatters, dass im Initialisierungszyklus der Wert am Seiteneingang 0 sein muss. Da das Gatter im Propagationszyklus fehlerleitend zu belegen ist, ergibt sich am Seiteneingang, bedingt durch die Scan-Kette, notwendigerweise eine steigende Flanke. Im Beispiel ist der Pfadverzögerungsfehler im Scan-Shift Modus nicht testbar, da sich am

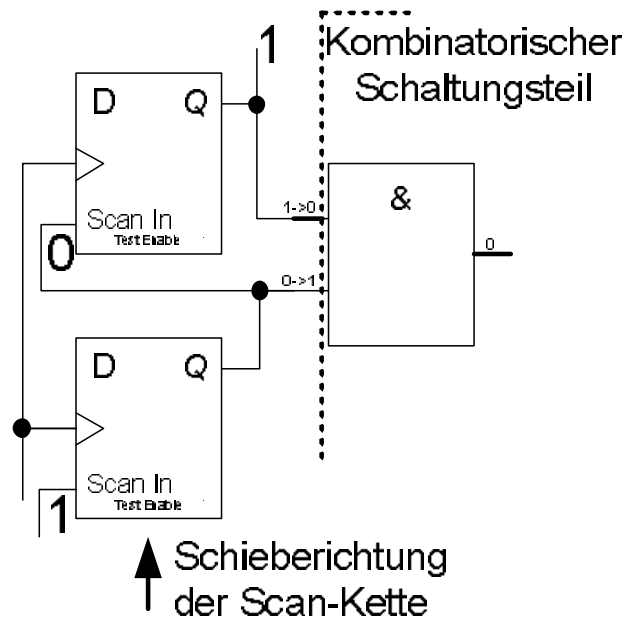


Abbildung 16: Die Testbarkeit ist von der Reihenfolge der Scan-Elemente in der Scan-Kette abhängig

Gatterausgang (aufgrund der Reihenfolge der Speicherelemente in der Scan-Kette) keine Flanke einstellen lässt.

Funktionale Einstellung

Eine weitere Möglichkeit, den Propagationsvektor V_P zu erhalten, besteht in der Verwendung der Schaltungsantwort für den pseudo primären Anteil des V_P (Abbildung 15). Dazu wird der Initialisierungsvektor V_I in die Scan-Kette eingetaktet, die Speicherelemente in den Normalmodus geschaltet und durch das Anlegen eines Taktsignals die Schaltungsantwort an den Eingänge übernommen. Das Ergebnis des Tests kann nach einem weiteren Takt in den Speicherelementen gehalten und ausgetaktet werden. Das Umschalten der Speicherelemente vom Test- in den Normalmodus ist in diesem Applikationsmodus zeitlich unkritisch, da dies nicht während des Propagationszyklus geschehen muss. Ferner erlaubt die Gewinnung des zweiten Testvektors V_P aus der Schaltungsantwort auf V_I nur den Test jener Fehler, die sich auch im Normalbetrieb einstellen können²⁵. Die Angabe einer Fehlerabdeckung kommt so den realen Gegebenheiten näher. Eine Abhängigkeit der Testmuster von der Reihenfolge der Speicherelemente in der Scan-Kette

²⁵unter der Annahme, dass im Normalbetrieb ein Eingangsvektor V_I auftreten kann

besteht ebenfalls nicht; der pseudo primäre Anteil des I_V lässt sich als beliebiger Vektor extern eintakten, während der Anteil des F_V ausschließlich von der Schaltungsfunktion abhängt.

Zusammengefasst reicht für die Testmuster generierung für Verzögerungsfehler die Gatternetzliste der zu testenden Schaltung als Eingangsdaten nicht aus, die Abhängigkeit der beiden Testvektoren voneinander ist ebenso zu beachten. Der gewählte Testmodus muss vor der Testmuster generierung bekannt sein, da er direkten Einfluss auf die Testmuster generierung nimmt. Für den Produktionstest sind bei der Auswahl des geeigneten Applikationsmodus weitere Anforderungen zu beachten, die aus den jeweiligen Modi folgen.

Mit Blick auf die Durchführung eines Produktionstests stellen die im Scan-Shift Modus erhaltenen Testmuster größere Anforderungen an die Testdurchführung. Zum Zeitpunkt der Messwertaufnahme (Capture Event) am Ende des Propagationszyklus müssen sich die Speicherelemente der Scan-Kette im Normal-Modus befinden. Da sie sich andererseits zu Beginn des Taktes noch im Shift-Modus befinden, um den Propagationsvektor zu erzeugen, müssen sie folglich innerhalb des Propagationszyklus umgeschaltet werden. Unter Beachtung der Tatsache, dass die Schaltung gerade im Propagationszyklus mit ihrer Nominalfrequenz betrieben werden muss, fällt diese Anforderung umso härter aus. Bei der Frage nach der erreichbaren Fehlerabdeckung haben Experimente an ISCAS89-Benchmarkschaltungen gezeigt, dass sich bei einer Verwendung von 100.000 pseudo-zufälligen Testmustern unter Verwendung des Scan-Shift Applikationsmodus eine im Schnitt um 6,2% höhere Übergangsfehlerabdeckung einstellte als bei der Verwendung der funktionalen Einstellung [121]. Für die Bewertung der Ergebnisse stellt sich jedoch, gerade mit Blick auf den Produktionstest, die Frage, in wie weit der Test die realen, im Betrieb vorliegenden Gegebenheiten widerspiegelt. Da die zu testende Schaltung in ihren Rückführungen interne Zustände codiert ist die Wahrscheinlichkeit, dass eine Eingangsvektorfolge, wie sie ein Scan-Shift Test erzeugt, im realen Betrieb vorkommt, eher gering. Folglich können Fehler, die ausschließlich bei der Verwendung des Scan-Shift Modus auftreten, nicht-funktionale Pfade sensitivieren, die im realen Betrieb der Schaltung nicht aktiviert werden und damit zu keinem Ausfall der Schaltung führen würden. Gerade für die Anwendung eines Verzögerungsfehlermodells im Produktionstest liegt hier die Gefahr, im Betrieb fehlerfrei arbeitende Schaltungen unnötigerweise auszusortieren [115]. Es darf jedoch mit Blick auf den defektorientierten Test nicht verschwiegen werden, dass diese Schaltungen einen Defekt aufweisen, da sie ansonsten auch im Scan-Shift Modus nicht ausfallen würden. Aus betriebswirtschaftlicher Sicht ist der Aspekt einer zu hohen Ausschussrate jedoch nicht akzeptabel.

Die oben geschilderten Probleme bestehen für einen Test unter Verwendung der funktionalen

Einstellung nicht. Mit Blick auf die für den Test verwendeten Taktschemata (→2.11) ist die Zuführung des Initialisierungsvektors zeitlich unkritisch. Den Speicherelementen kann folglich ausreichend Zeit gegeben werden, vom Test- in den Normal-Modus umzuschalten. Die beiden nachfolgenden Taktsignale, die den Propagationszyklus definieren, müssen mit der Nominalfrequenz der Schaltung erfolgen. Weil sich die Speicherelemente zu Beginn des Propagationszyklus schon im Normal-Modus befinden tritt das Problem, sie während des Zyklus umzuschalten, nicht auf.

Die theoretisch negative Eigenschaft der funktionalen Einstellung, im Mittel zu einer geringeren Fehlerabdeckung zu führen [121], wirkt sich positiv auf die Ergebnisse des Produktionstests aus. Weil der Propagationsvektor aus der Schaltungsantwort auf den Initialisierungsvektor hervorgeht, können ausschließlich Fehler zu einem Ausfall führen, die auch im Normalbetrieb der Schaltung auftreten und damit zu wirklichen Funktionsausfällen führen. Ferner kann den Ergebnissen aus [138] entnommen werden, dass die Fehlerabdeckung im Scan-Shift Modus stark von der Reihenfolge abhängt, in der die Speicherelemente in der Scan-Kette auftreten. Diese Einschränkung besteht bei der funktionalen Einstellung nicht, da die Schaltungsfunktion den zweiten Testvektor erstellt. Nachteilig wirkt sich die Verwendung der funktionalen Einstellung auf die Testmustererzeugung aus. Die Ergebnisse aus [138] und [130] geben übereinstimmend eine längere Rechenzeit je Fehler bei der Annahme der funktionalen Einstellung an.

2.11 Taktschemata

Für die Testdurchführung müssen weitere Eigenschaften der Schaltung und des Testequipments berücksichtigt werden. Verzögerungsfehler stellen sich bei hohen Taktfrequenzen ein. Folglich müssen die Testmuster mit der für die Schaltung spezifizierten Betriebsfrequenz der Schaltung zugeführt werden, wofür der Testautomat sogenannte high-speed Testerpins bereitstellen muss. Da die Kosten des Testautomaten überproportional mit der Zahl an high-speed Testerpins steigt [14], muss eine optimale Teststrategie möglichst wenige breitbandige Kanäle bedingen, um die Testkosten zu minimieren. Darüber hinaus bilden die angeschlossenen Testerpins im allgemeinen eine größere Last an den Schaltungseingängen, als sie bei normaler Verwendung der Schaltung auftreten würden. Eine Verschlechterung der Performance der Schaltung während des Tests ist die Folge [85].

Der eigentliche Test auf den angenommenen Verzögerungsfehler geschieht im Propagationszyklus, der einem normalen Taktzyklus entsprechen muss. Die Initialisierung und das Austakten

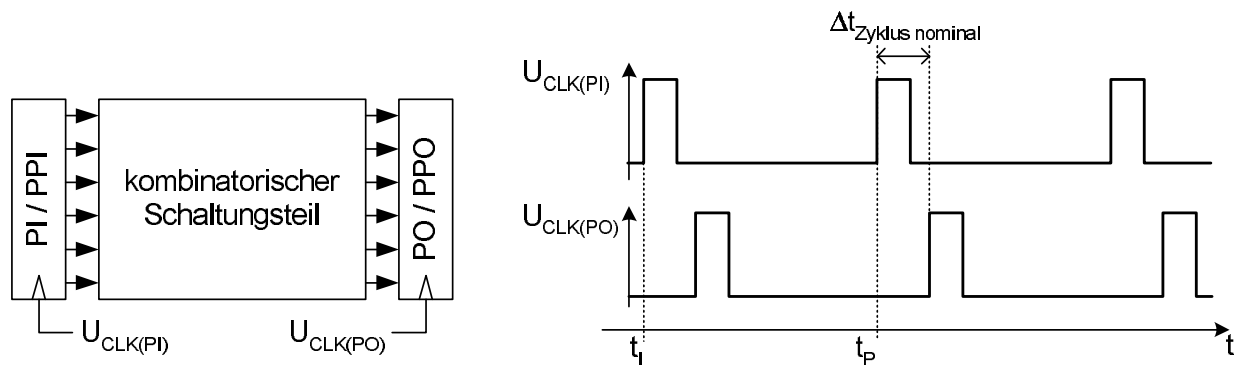


Abbildung 17: Durch Verschiebung gegeneinander kann aus langsameren Taktsignalen ein Propagationszyklus mit Betriebsfrequenz erreicht werden

der Schaltungsantwort kann mit geringerer Taktfrequenz geschehen²⁶. Je nach Art der Schaltung sind folgenden Strategien für die Testdurchführung möglich:

Verschobene, langsamere Taktsignale: Die zu prüfende Schaltung kann als rein kombinatorische Schaltung vorliegen, deren Ein- und Ausgänge wegen der Einbettung der Schaltung in eine Umgebung über Speicherelemente geführt werden. Können die Takte der Speicherelemente an den Eingängen (Erzeugen der Eingangssignalwechsel) und Ausgängen (Speichern der Schaltungsantwort) getrennt voneinander angesteuert werden, muss das Testequipment nicht in der Lage sein, ein Taktsignal mit Betriebsfrequenz bereitzustellen, wenn es stattdessen zwei langsamere Takte liefern kann, die um die gewünschte Zykluszeit Δt_{Zyklus} gegeneinander verschoben sind (Abbildung 17).

Normaler Scan-Test Für sequentielle Schaltungen mit Scan-Kette muss zwischen dem Applikationsmodus Scan-Shift oder funktionale Einstellung unterschieden werden. Es ergeben sich folglich Timingdiagramme wie in Abbildung 18.

Test sequentieller Schaltungen ohne Scan-Kette mit variablem Takt Der Test sequentieller Schaltungen erstreckt sich zumeist über mehrere Taktzyklen und erfordert einen hohen Aufwand für die Testmuster-generierung und die Testdurchführung. Die Schaltung muss im

²⁶wobei eine sinkende Taktfrequenz für die Schiebeoperationen die Testzeit und damit die Kosten des Tests erhöht. Andererseits wird die Verwendung günstigerer Tester ermöglicht, wodurch die Kosten sinken.

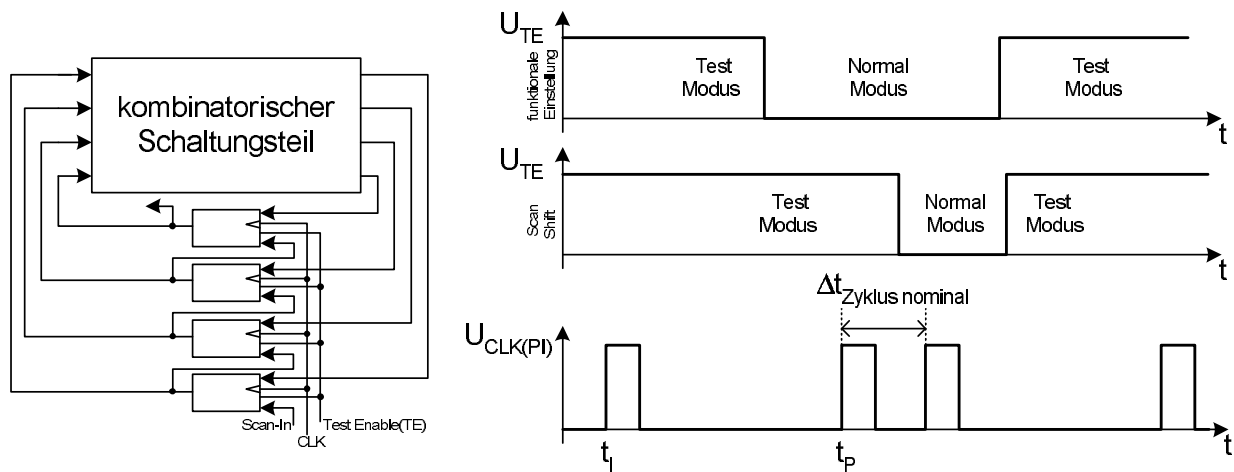


Abbildung 18: langsame Takte während des Ein- und Austaktens, der Test benötigt einen schnellen Takt

allgemeinen mehrere Takte durchlaufen, ehe sie initialisiert ist. Diese Initialisierungstakte werden bei geringerer Taktfrequenz vorgenommen, um keine möglichen Verfälschungen in der Initialisierung durch Timingprobleme auf anderen Signalpfaden zu riskieren. Der Propagationszyklus, der den Fehler testen soll, muss mit Betriebsfrequenz durchgeführt werden. Es schließen sich weitere Propagationszyklen an deren Aufgabe es ist, das Testergebnis an die primären Ausgänge zu leiten. Auch diese Propagationszyklen werden mit verringerter Taktfrequenz durchgeführt, um keine nachträglichen Verfälschungen zu riskieren.

Test sequentieller Schaltungen mit Betriebsfrequenz Ebenfalls möglich ist die Zuführung der Testvektoren mit Betriebsfrequenz, da keine pseudo primären Anteile zugeführt werden können und folglich alle beeinflussbaren Schaltungseingänge extern zugänglich sind. Der Zielfehler kann dann jedoch in verschiedenen Taktzyklen aktiviert (und das Ergebnis dadurch vielleicht invalidiert) werden.

2.12 Zusammenfassung

Ziel des Produktionstests ist die Detektion defektbedingter Fehler in produzierten Schaltungen. Weil sich verschiedene Defekte in unterschiedlichen Effekten äußern können, existieren verschiedene Fehlermodelle, welche die Basis des Tests sein können. Die Testmuster-generierung

geht im allgemeinen von einer kombinatorischen Schaltung aus, da der Aufwand für die Testmuster-generierung und Testdurchführung bei sequentiellen Schaltungen deutlich viel komplexer ausfällt. Aus diesem Grund hat sich das Full-Scan Design industrieweit durchgesetzt.

Der Test einer integrierten Schaltung kann als funktionaler Test oder struktureller Test durchgeführt werden. Der funktionale Test erfordert eine genaue Kenntnis der Schaltungsfunktion und ist als vollständiger Test nur für kleine Schaltungen möglich. Weil darüber hinaus keine Fehlerabdeckung angegeben werden kann ist die Bewertung eines nicht vollständigen funktionalen Tests schwierig. Strukturelle Testverfahren basieren auf einer Netzlistenbeschreibung der Schaltung, die auf einer beliebigen Abstraktionsebene geschehen kann. Die Gatterebene hat sich dabei als bester Kompromiss aus maximal möglicher Abstraktion und möglichst genauer Beschreibung der Defektauswirkung herauskristallisiert. Im Unterschied zu funktionalen Tests ermöglichen strukturelle Verfahren sowohl eine automatische Berechnung der Testmuster wie auch eine Angabe einer Fehlerabdeckung für einen nicht vollständigen Test.

Für die Detektion laufzeitbedingter Fehlereffekte ist die Verwendung mehrerer Testvektoren notwendig, da eine Signalflanke als Testsignal dient. Für die Testdurchführung folgen daraus Konsequenzen, die bereits bei der Testmuster-generierung bedacht werden müssen. Die Verwendung spezieller Enhanced Scan-Flip-Flops kann in diesem Zusammenhang den Test vereinfachen, ist jedoch wegen des erhöhten Flächenbedarfs der Speicherelemente in der industriellen Schaltungsfertigung nicht etabliert.

3 Anforderungen an den Test aktueller Schaltungen

In den vergangenen Jahren hat sich die *CMOS* Technologie in der Produktion mikroelektronischer Schaltung durchgesetzt (etwa 90% der in 2002 gefertigten Schaltungen wurden in *CMOS* realisiert [131]). Der Hauptgrund für die Verwendung der *CMOS*-Technologie liegt in ihrer typischen Eigenschaft, im statischen Fall idealerweise keine Leistungsaufnahme aufzuweisen, was aus zweierlei Sicht von Vorteil ist. Da in *CMOS*-Schaltkreisen nur bei Umschaltvorgängen Strom fließt, steigt die Betriebsdauer batteriebetriebener Anwendungen (Hörgeräte, mobile Computer, Mobiltelefone) je Batterieladung. Mit Blick auf die zunehmende Integrationsdichte und die damit steigende Verlustleistung je Flächeneinheit ist ein minimaler Leistungsbedarf ebenfalls von entscheidender Bedeutung. Der Preis des für einen Schaltkreis zu verwendenden Gehäuses wird auch durch seine Fähigkeit bestimmt, die im Chip entstehende Verlustleistung in Form von Wärme an die Umwelt abzugeben. Die Verwendung kostengünstigerer Gehäuse senkt demnach den Verkaufspreis des Produkts. Ferner ist die maximal mögliche Ableitung der Verlustleistungswärme einer der begrenzenden Faktoren für die Steigerung der Leistungsfähigkeit einer Schaltung. Kann die intern erzeugte Verlustleistung nicht ausreichend abgeführt werden, wird die Schaltung thermisch zerstört.

Für den Test aktueller Schaltungen bedeutet die Dominanz der *CMOS*-Prozesse, dass die für diese modernen Technologien typischen Fehlereffekte adressiert werden müssen. Dazu werden zunächst die typischen Defektmechanismen betrachtet und deren Auswirkungen bewertet. Empirische Arbeiten, die die Detektion fehlerhafter Schaltungen unter Anwendung verschiedener Fehlermodelle untersuchen, zeigen, dass der Produktionstest nicht mit *einem* Fehlermodell auskommt, um die Fehlerfreiheit der Schaltung sicherzustellen.

3.1 Defekte in aktuellen *CMOS* Schaltungen

Die Fertigung integrierter Schaltungen besteht aus einer Abfolge verschiedener, immer wiederkehrender Schritte, bei denen in jedem Schritt Defekte in die Schaltung eingebracht werden können. Unerwünschte Partikel (Punktdefekte) sind hierfür ebenso ein Beispiel wie Maskenfehler, Fehler bei der exakten Ausrichtung der Masken oder Abweichungen in der Zusammensetzung verwendeter Materialien (parametrische Defekte). Die Abwesenheit der möglichen Defekte mit möglichst geringem Aufwand sicherzustellen (und damit die Wahl der hierfür am geeignetsten erscheinenden Fehlermodelle) ist Ziel des Produktionstests.

Die Fortschritte bei der minimal möglichen Auflösung der Fertigungstechnologien lassen eine

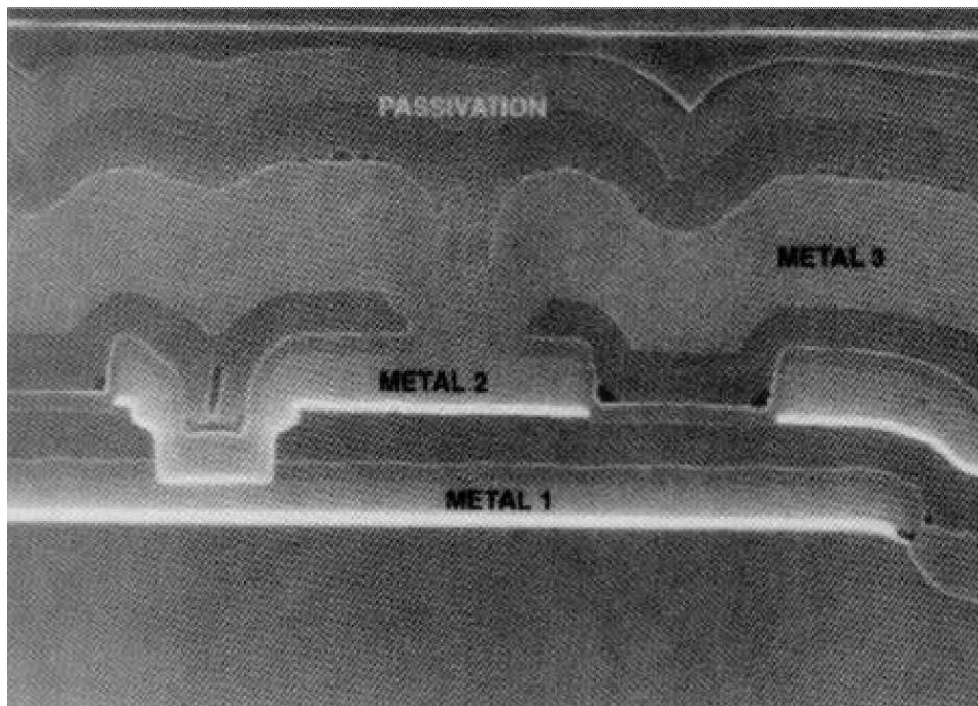


Abbildung 19: Unebenheiten aufgrund vergrabener Layer nehmen mit der Anzahl der Layer zu und erfordern in modernen Technologien Polierschritte zwischen der Herstellung der Leitungsebenen, um sie auszugleichen [132]

steigende Integrationsdichte zu, in deren Folge das Problem steigt, die zunehmende Anzahl an Transistoren -der Schaltungsfunktion entsprechend- auf geringerer Fläche zu verbinden. Wie von der Leiterplattenfertigung bekannt, muss mit der Integrationsdichte die Zahl der verwendeten Leiterbahnebenen steigen, um die Leitungsführung zu ermöglichen bzw. die verwendeten Leiterbahnen kurz zu halten. Bei der Schaltungsfertigung kommt es, wie in Abbildung 19 zu sehen, mit wachsender Zahl an Metallisierungsebenen zu dem Problem lokaler Unebenheiten, die in nachfolgenden Prozeßschritten zu Problemen führen können (Leiterbahnunterbrechungen durch Kanten im Untergrund, Fokussierungsprobleme in Lithographieschritten). Um dem entgegen zu wirken, schließen sich in modernen Fertigungsprozessen Planarisierungsschritte an die Herstellung einer Metallisierungsebene an. Da der Planarisierungsprozeß nicht den gesamten Wafer zur gleichen Zeit bearbeitet, bleibt die Möglichkeit einer Restwelligkeit über den Wafer und damit die vorgenannten potentiellen Probleme erhalten.

Durch die steigende Zahl an Metallisierungsebenen für die Verbindungsleitungen finden zudem vermehrt sogenannte Vias, also Kontaktübergänge zwischen einzelnen Metallisierungsebenen,

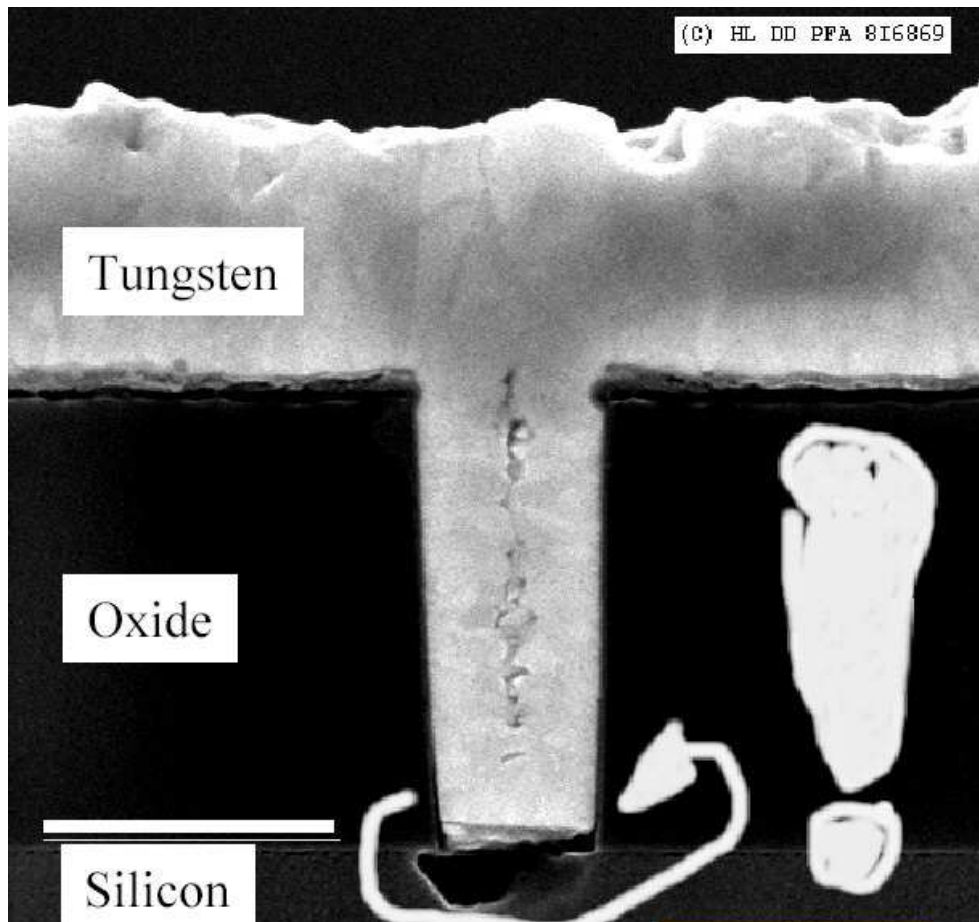


Abbildung 20: Das Problem unzureichender Kontaktierung besteht prinzipiell ebenso für Vias [132]

Anwendung. In der Praxis ist die Fertigung dieser Vias schwierig, weil sie (vergl. Abbildung 20) einen geringen Durchmesser bei relativ großer Tiefe besitzen, was das vollständige Ausfüllen erschwert [132][131] und im Fehlerfall zu einer Erhöhung des Übergangswiderstandes führt. Die dadurch bewirkten Probleme können sich neben der Erhöhung etwaiger Signallaufzeiten ($\rightarrow 2.5.2$) auch in Elektromigrationserscheinungen an den mangelhaften Übergängen äußern, weil das Material durch auftretende Stromdichten im Bereich von $10^5 \frac{A}{cm^2}$ abgetragen wird [131]. Wegen der steigenden Zahl an Metallisierungsebenen (vergl. Tabelle 10 im Anhang) wird erwartet, dass in kommenden Schaltkreisgenerationen Fehlereffekte zunehmen werden, die ihre Ursache in fehlerhaft gefertigten Vias haben [5]. In die gleiche Richtung gehen Ergebnisse von Intel. Dort wurde beim Übergang von einem $0,25\mu m$ -Prozess auf einen $0,18\mu m$ -Prozess ein erhöhtes Auftreten von hochohmigen Brückenfehlern bemerkt [93]. Auch hier ist eine Beeinträchtigung

des dynamischen Schaltungsverhaltens die Folge.

3.2 Fehlermodelle für aktuelle CMOS Schaltungen

Der Produktionstest einer Schaltung hat zum Ziel, über eine maximale Defektabdeckung in minimaler Zeit eine Aussage zur korrekten Funktion der Schaltung zu liefern. Vor diesem Hintergrund entscheiden mehrere Faktoren über die Anwendung eines Fehlermodells für den Produktionstest. Den Anforderungen an eine einfache Durchführbarkeit des Tests bei gleichzeitig hoher Fehlerabdeckung je Testmuster genügt der Haftfehlermodell (→2.5.1). MOS spezifische Fehlereffekte adressiert das stuck-open/stuck-short Fehlermodell (→2.5.1), sein geringer Abstraktionsgrad führt jedoch zu einer großen Zahl modellierter Fehler, die das Haftfehlermodell zu großen Teilen mit abdeckt. Mit der fortschreitenden Reduzierung minimal realisierbarer Schaltungsstrukturen, die zu veränderten Fehlermechanismen führt, sowie dem Trend zu weiter steigenden Taktfrequenzen muss der Produktionstest aktueller Schaltungen mögliche dynamische Fehlereffekte ebenfalls berücksichtigen. Die drei meist akzeptierten dynamischen Fehlermodelle sind das Übergangs-, Gatterverzögerungs- und Pfadverzögerungsfehlermodell (→2.5.2).

Die Anforderungen, die ein Fehlermodell für dynamische Fehlereffekte an den Test stellt, sind im Vergleich zu den statischen Fehlermodellen hoch. Für die Testmuster generierung ergeben sich zusätzliche Möglichkeiten bei der Belegung einer Leitung, da der Test über 2 Taktzyklen nun auch Signalwechsel zulässt. Zudem können mögliche Angaben zur erreichbaren Testqualität (→2.8) schon während der Testmuster generierung Beachtung finden. Die bei der Testdurchführung eingesetzten Testautomaten in der Lage sein, Taktimpulse mit einer hohen Genauigkeit bei gewünschter Betriebsfrequenz zu liefern. Weitere Anforderungen folgen aus der Zuführung der Testmuster an die Schaltung, da es sich um ein Testvektorpaar handelt, das in zwei aufeinander folgenden Taktzeitpunkten an die Schaltung anzulegen ist (→2.10). Wird ein Test nach dem Pfadverzögerungsfehlermodell erwogen, ergibt sich bei Schaltungen mit einer relativ geringen Anzahl an Gattern eine Fehlermenge, die sich nicht mehr im Arbeitsspeicher eines Computers halten lässt [137]. Die Verwendung des Übergangsfehlermodells könnte andererseits für den Produktionstest ungeeignet sein, weil es die auftretenden Verzögerungen nicht mit ausreichender Genauigkeit adressiert. Weiterhin stellt sich die Frage, ob die bislang während des Produktionstests berücksichtigten Fehlermodelle in der Praxis nicht bereits alle Schaltungen mit Verzögerungsfehlern als fehlerbehaftet detektieren.

3.3 Experimentelle Ergebnisse

Für die in Abschnitt 3.2 angeführten Fehlermodelle lässt sich eine Defektabdeckung bezüglich der in Abschnitt 3.1 angegebenen Defekte entwickeln, eine realistische Aussage über die Eignung entsprechender Testmuster kann jedoch erst durch eine praktische Anwendung an produzierten Schaltungen getroffen werden. In der Literatur finden sich in verschiedenen Arbeiten aus der Industrie ([49, 85, 86, 94]) Aussagen über die Eignung von Testmustern, die auf unterschiedlichen Fehlermodellen basieren. Darüber hinaus entstanden in den neunziger Jahren zwei Gemeinschaftsprojekte zwischen Universitäten und Industriepartnern, die sich explizit die Untersuchung der Eignung verschiedener Fehlermodelle und Testverfahren in der Anwendung an realisierten Schaltungen zum Ziel gesetzt haben. Das erste, als *Murphy-Projekt* bezeichnete Vorhaben [87], wurde 1991 unter Beteiligung verschiedener Hersteller von Testautomaten, Chip-Hersteller und Universitäten mit dem Ziel ins Leben gerufen, die Effektivität verschiedener Fehlermodelle und Tests unter verschiedenen Randbedingungen zu untersuchen. Im Rahmen des Murphy-Projekts entstand ein in $0,7\mu\text{m}$ CMOS-Technologie gefertigter Test-Chip [87], der 3 logische Blöcke und 2 Multiplizierer mit insgesamt 25.000 Gattern enthielt. Von den auf vier Wafern gefertigten Chips überstanden 5491 einen ersten Test auf Waferebene, der die Erkennung katastrophaler Fehler zum Ziel hatte. Sie bildeten die Grundlage der Experimente [18]. 1999 entstand ein Nachfolgechip, welcher in $0,35\mu\text{m}$ Technologie gefertigt [77] wurde und mit etwa 265.000 Gatteräquivalenten die zehnfache Größe erreichte.

Das zweite Projekt wurde 1994 vom SEMATECH Konsortium ins Leben gerufen. Es nutzt einen Graphikchip der Firma IBM als Grundlage seiner Messungen, der in einem $0,45\mu\text{m}$ CMOS-Prozess hergestellt wurde [94]. Die Ausrichtung dieses Projektes weicht vom Murphy-Projekt in soweit ab, dass es, von einer Verwendung verschiedener Fehlermodelle aber fest vorgegebener Testzeit ausgehend, die Frage adressiert, wieviel der zur Verfügung stehenden Testzeit anteilig jedem Test bereitzustellen ist, um ein optimales Testergebnis zu erzielen.

Die in den Projekten durchgeführten Experimente geben neben der Frage einer Defektabdeckung einzelner Fehlermodelle bzw. Testwertsätze auch Auskunft über den Einfluss möglicher Randbedingungen, unter denen ein Test durchgeführt werden kann [19, 20, 21, 50].

Alle Experimente kommen zu dem Ergebnis, dass jedes der angewendeten Fehlermodelle fehlerhafte Schaltungen erkennt, die jeden anderen Test unerkannt überstanden [85, 86, 94]. Ein anschauliche graphische Darstellung der Ergebnisse aus [94] bietet das Venn-Diagramm aus Abbildung 21.

Es zeigt sich, dass jeder Test nach einem Fehlermodell defekte Schaltungen erkennt, die bei kei-

nem anderen Test ausfallen. Der in der Graphik überproportionale Anteil der durch den I_{DDQ} -Test als fehlerhaft erkannten Schaltungen liegt an der geringen pass/fail-Schwelle des Experiments.

Wie in Abschnitt 2.5 geschildert, haben strukturelle Testverfahren gegenüber der Verwendung funktionaler Tests den Vorteil, dass sich die Testmuster algorithmisch berechnen lassen und mit Bezug auf die Zahl der modellierten Fehler eine Fehlerabdeckung der angewendeten Testmuster angegeben werden kann. Die Ergebnisse der Messungen zeigen jedoch, dass bislang nicht auf die Verwendung funktionaler Testmuster verzichtet werden kann, wenngleich sie nur eine geringe Zahl an defekten Schaltungen detektieren, die den strukturellen Testverfahren entgehen (im Experiment Maxwells 2% aller Chips / 6% in dem logischen Teil der Schaltung [85]). Weil keines der anderen Testverfahren diese Schaltungen als fehlerhaft erkennt, müssen sie Defekte beinhalten, welche die bisherigen strukturellen Fehlermodelle nicht abdecken²⁷.

Durch die wegen steigender Integrationsdichte ansteigenden Leckströme in den Schaltungen werden für die Verwendung strombasierter Testverfahren (\rightarrow 2.5.3) Probleme beim Einsatz für kommende Fertigungstechnologien erwartet [90, 131]. Wie in [93] beschrieben, ergibt sich bei der Stromaufnahme der verschiedenen Schaltungen durch Prozessvariationen eine Verteilungsfunktion. Um defekte Schaltungen von fehlerfreien unterscheiden zu können, sollten die Verteilungsfunktionen der fehlerhaften und fehlerfreien Schaltungen disjunkt sein (Abbildung 22). Wegen des steigenden Leckstromanteils bewegen sich jedoch beide Verteilungsfunktionen aufeinander zu, sodass sie sich überlappen und in diesem überlappenden Bereich eine Unterscheidung zwischen fehlerfreien und fehlerbehafteten Schaltungen unmöglich wird. Die Ergebnisse in [76] bestätigen eine Verteilung der Stromaufnahme gefertigter Schaltungen. Nach [93] bleibt der IDDQ-Test trotz dieser Probleme ein wichtiger Bestandteil des Tests. Daten aus dem SEMATECH Projekt zeigen, dass Schaltungen mit erhöhter Stromaufnahme eine bis zu 8fach erhöhte Wahrscheinlichkeit besitzen, bei einem Burn-In Test²⁸ auszufallen. Mit Bezug auf die Daten des Murphy-Projektes kommt [85] zu dem Ergebnis, dass 87 der fehlerhaften Schaltungen durch funktionale Tests erkannt werden, 73 hiervon fallen ansonsten nur bei der IDDQ-Messung aus. Ohne die Verwendung des IDDQ-Tests steigt damit die Abhängigkeit des Produktionstests von

²⁷Ein Beispiel hierfür wären die in Kapitel 8 betrachteten Probleme der Signalintegritätsverletzungen durch Koppungseffekte oder Schwankungen der Versorgungsspannung.

²⁸Statistiken zeigen, dass die Ausfallrate produzierter Schaltungen über die Produktlebenszeit zu Betriebsbeginn und am Ende der vorgesehenen Produktlebenszeit hoch sind. Während der Anstieg der Ausfallrate am Ende der Produktlebenszeit auf betriebsbedingte Abnutzungserscheinungen zurückzuführen ist, liegt der Grund der anfänglich erhöhten Ausfallrate in Qualitätsmängeln aufgrund von Prozessvariationen [65]. Beim Burn-In werden die produzierten Schaltungen eine Zeit lang unter erhöhten Temperaturen betrieben, was zu einer beschleunigten Alterung führt und damit jene Schaltungen erkennen lässt, die zu den Frühausfällen gehören.

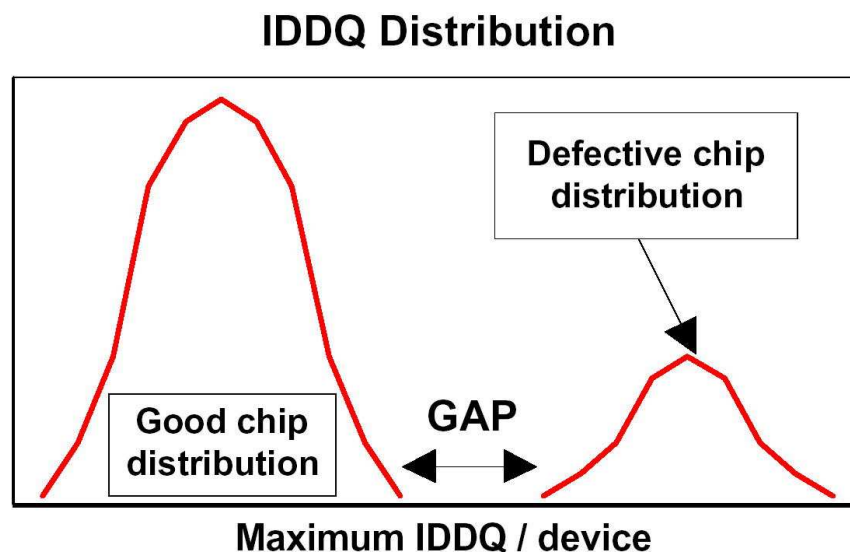


Abbildung 22: Die Anwendung von IDDQ-Tests wird erschwert, wenn die Menge der fehlerhaften und fehlerfreien Schaltungen ineinander übergehen [93]

funktionalen Testmustern.

Für Haftfehlertestmuster weist [79] zunächst darauf hin, dass ein Test T_1 mit einer Fehlerabdeckung von x nicht notwendigerweise eine Untermenge eines Tests T_2 mit einer höheren Fehlerabdeckung $x + \Delta$ sein muss²⁹. Damit erklärt sich, weshalb in den Versuchen eine Schaltung bei der Verwendung von Haftfehlertestwertsätzen T^{90} (Fehlerabdeckung: 90%) und T^{95} ausfiel, die Testwertsätze T^{98} , T^{99} und T^{100} überstand³⁰. Die Experimente zeigen, dass sich ein großer Anteil der Defekte in einem haftfehlerähnlichen Verhalten auswirken ($\frac{1}{3}$ [87] der defekten bzw. 57% der Schaltungen mit statischen Fehlern [39] im Murphy Projekt, über 47% der Schaltungen aus dem SEMATECH-Projekt [94]). Die Ergebnisse des Haftfehlertests können unter Verwendung von Testwertsätzen, die jeden Haftfehler mehrfach detektieren (im folgenden n-fach Haftfehlertests genannt), merklich verbessert werden. Nach [79] ist ein solcher Haftfehlertest in seinen Ergebnissen praktisch äquivalent zu der Verwendung aller den Fehler testenden Testmuster bei deutlich geringerer Zahl an Testvektoren. Laut [87] erkennen sie bei allen Taktfrequenzen eine größere Anzahl an fehlerhaften Schaltungen als Haftfehlertests mit 100% Fehlerabdeckung.

In allen Experimenten zeigt sich, dass ein Teil der fehlerhaften Schaltungen mit Laufzeitfehlern ausfällt. Aus [18, 39, 79] kann ersehen werden, dass die Zahl der den Test passierenden fehlerhaften Schaltungen mit abnehmender Taktfrequenz steigt. Nach [86] findet der AC-Scan-

²⁹ T_1 kann Vektoren enthalten, die nicht Teil von T_2 sind.

³⁰Die Vektoren aus T^{90} und T^{95} , bei denen die Schaltung ausfiel, waren in T^{98} , T^{99} und T^{100} nicht enthalten.

Test³¹ viele fehlerhafte Chips, die ansonsten nur beim funktionalen Test bei Betriebsfrequenz ausfallen. Bei Haftfehlertest mit 100% Fehlerabdeckung steigt die Fehlererkennungsrate, wenn sie mit Nominalfrequenz angelegt werden [86]. N-fach Haftfehlertests haben nach [18] eine hohe Übergangsfehlerabdeckung (im Experiment 89.11%-98,86%). Ferner ergeben sich bei Tests unter verminderter Betriebsspannung vermehrt Ausfälle, weil eine geringere Betriebsspannung die Umladevorgänge verlangsamt und damit Laufzeitfehler verstärkt [18]. Das Problem der Pfadauswahl für Pfadverzögerungsfehlertests bestätigen [39, 49, 86]. Während es in [49] für nur 7 der 100 längsten Pfade gelingt, einen Test zu generieren, führt die Anwendung von 148 Testmustern in [86] zu einer Erkennung von nur 18,5% aller fehlerhaften Schaltungen mit Verzögerungsfehlern. Als Grund werden Schwierigkeiten bei der Pfadauswahl angegeben. Nach [39] ergibt der Test für 100 kritische Pfade³² keine besseren Ergebnisse als ein Haftfehlertest bei Betriebsfrequenz. Es folgt daraus, dass der Pfadverzögerungsfehlertest für wenige, ausgesuchte Pfade nicht hinreichend für die Bestimmung der Fehlerfreiheit einer Schaltung ist.

3.4 Zusammenfassung

Die CMOS-Technologie dominiert bei der Fertigung heutiger Schaltungen, da sie im Ruhezustand im Idealfall keine Leistungsaufnahme aufweist. Sinkende Abmaße und der damit einhergehende Anstieg der Leckströme erschweren jedoch die Verwendung strombasierter Testmethoden, da sich defektbedingte Stromüberhöhungen vom Hintergrund der Leckströme nicht mehr abheben. Auf statischen Fehlermodellen wie dem Haftfehlermodell basierende Testverfahren modellieren viele CMOS-typische Fehlereffekte nicht, seine relativ geringe Menge an Testmustern lassen das Haftfehlermodell jedoch nach wie vor für den Produktionstest attraktiv erscheinen. Zudem zeigen praktische Untersuchungen, dass eine Reihe der vom Haftfehlermodell nicht adressierten Defekte Fehlereffekte hervorrufen, für die Haftfehlertests eine relativ hohe Detektionswahrscheinlichkeit haben [9]. Haftfehlertests unter Betriebsfrequenz können zusätzlich eine Reihe von dynamischen Fehlereffekten mit erkennen (wenngleich sie für diese als Zufallstestmuster angesehen werden müssen und der Test im Scan-Shift Applikationsmodus abläuft).

Bei den Verzögerungsfehlermodellen hat sich bislang keines als das geeignetste Fehlermodell etabliert. Die im Test geringe Erkennungsrate der Pfadverzögerungsfehlertests geht letztlich auf

³¹Es werden *AC-Scan-Test* und *At-Speed Stuck-At Test* unterschieden. Während der *At-Speed Stuck-At Test* als Zufallstestmuster für Verzögerungsfehlereffekte angesehen werden muss steht der Begriff *AC-Scan-Test* für alle verzögerungsfehlerbasierten Tests.

³²bestimmt unter der Annahme einer einheitlichen Verzögerung aller Gatter

die Beschränkung der Zahl zu testender Pfade zurück. Generell führt jedoch trotz der komplexeren Anforderungen kein Weg an verzögerungsfehlerbasierten Tests vorbei. Dies liegt zum einen an den engeren Timinglimits, die eine Folge der steigenden Taktfrequenzen sind. Zudem wird zunehmend eine Verschiebung der Anteile beobachtet, welche Leitungen und Gatter bei geringer werdenden Abmaßen zur Gesamtlaufzeit beitragen. Der Anteil der Laufzeiten auf den Leitungen wächst dabei mit abnehmender Strukturgröße [131]. Die durch das Routing bedingte steigende Anzahl an Verbindungsebenen und der damit einhergehende Anstieg von Kontaktübergängen (Vias) führt bei Fertigungsproblemen dieser Übergänge schnell zu einem überhöhten Widerstand. Erhöhungen der Signallaufzeiten sind die Folge.

Der Funktionaltest, dessen Testmuster generierung aufwendig ist, kann durch die strukturellen Testverfahren nicht ersetzt werden, als Grund gilt das Auftreten von Defekten in den Schaltungen, die diese Fehlermodelle nicht abdecken.

Die Verwendung mehrerer Testmuster für einen Fehler hat sowohl für statische wie auch dynamische Fehlermodelle [110] eine Qualitätsverbesserung gebracht, aus Sicht des Produktionstests ist der damit verbundene Mehraufwand vermutlich jedoch nicht zu bewältigen.

4 Fehlersimulation für Verzögerungsfehler

Ein vollständiges ATPG-System verwendet, wie in Abschnitt 2.7 angeführt, eine Kombination aus Testmuster-generator und Fehlersimulator. Die Liste zu Fehlersimulation entspricht einer Logiksimulation der Schaltung unter Verwendung der gegebenen Eingangsbelegung und der Annahme eines Fehlers am entsprechenden Fehlerort. Das Ergebnis der Logiksimulation der fehlerfreien Schaltung wird mit dem Ergebnis der Simulation der fehlerbehafteten Schaltung verglichen und über Unterschiede an den beobachtbaren Ausgängen auf eine Fehlerdetektion geschlossen.

Während sich die Aufgaben der Fehlersimulation für das Haftfehlermodell auf die Bestimmung der Fehlerabdeckung vorgegebener Testmuster beschränkt, kann eine Fehlersimulation für Verzögerungsfehler weitergehende Aussagen zu Testqualitäten oder erkennbaren Fehlergrößen liefern.

4.1 Fehlersimulation für Übergangsfehler

Übergangsfehler weisen die größte Gemeinsamkeit mit Haftfehlern auf (\rightarrow Abschnitt 2.5.2). Die vorhandenen Parallelen lassen sich auch für die Fehlersimulation nutzen. Eine Fehlersimulation nach dem Haftfehlermodell hat zur Aufgabe, die Initialisierung des Fehlerortes und die Propagation des Fehlereffekts zu mindestens einem beobachtbaren Schaltungsausgang zu überprüfen. Die gleichen Aufgaben hat eine Fehlersimulation nach dem Übergangsfehlermodell zu erfüllen. Folglich ist ein Testmuster $\langle V_I, V_P \rangle$ genau dann ein Testmuster für den Übergangsfehler $\Phi^{\ddot{u}}$ am Fehlerort k_Φ , wenn V_I den Fehlerort auf den Ausgangswert der angenommenen Flanke initialisiert, während V_P den Fehlerort auf den Endwert der Flanke umschaltet und den Fehlereffekt zu mindestens einem beobachtbaren Ausgang fortschaltet. Diese Eigenschaft weist ein Haftfehlermodell auf, der einen Haftfehler an k_Φ auf dem Ausgangswert der Flanke annimmt. Folglich kann der slow-to-rise (slow-to-fall) Übergangsfehler im Propagationszyklus als stuck-at-0 (stuck-at-1) Fehler aufgefasst werden. Soll über eine Fehlersimulation die Aussage gewonnen werden, ob das gegebene Testmuster den Fehler detektiert, kann eine Haftfehlersimulation des Propagationsvektors unter Annahme eines entsprechenden Haftfehlers am Fehlerort verwendet werden. Die zusätzliche Bedingung, dass der Initialisierungsvektor den Fehlerort initialisiert, überprüft eine Logiksimulation des Initialisierungsvektors.

Sind über die reine Fehlerdetektion hinaus Angaben über die erzielte Testqualität erwünscht, kann auf die Verwendung von Logiken nicht verzichtet werden, die auch Signalübergänge codie-

ren können [118, 122, 134, 135]. Unter Beachtung der Möglichkeit, diese zusätzlichen Informationen zu codieren, lassen sich die Einhaltung der robusten- bzw. nicht robusten Seiteneingangsbelegungen ($\rightarrow 2.8$) bis zum Ausgang verfolgen und somit Aussagen zur erreichten Testqualität gewinnen.

4.2 Fehlersimulation für Gatterverzögerungsfehler

Es existieren eine Reihe von Arbeiten zur Fehlersimulation für Gatterverzögerungsfehler [3, 15, 31, 36, 62, 96, 102, 112]. Die Simulatoren unterscheiden sich nach fehlerparallelem oder musterparallelem Ansatz, die Verwendung unterschiedlicher Logiken und die mögliche Unterstützung von Tristate-Gattern.

Musterparallele Simulatoren codieren in den im Rechner zur Verfügung stehenden Speicherworten mehrere Testmuster und ermitteln je Fehlersimulation für jeweils einen Fehler die Fehlerabdeckung durch die codierten Testmuster. Dem gegenüber stehen fehlerparallele Simulatoren, welche die Breite der zur Verfügung stehenden Speicherworte für die Simulation mehrerer Fehler unter Annahme des gleichen Testmusters verwenden. Der am Institut für Theoretische Elektrotechnik und Mikroelektronik der Universität Bremen realisierte Gatterverzögerungsfehlersimulator [100] arbeitet fehlerparallel und ist in der Lage, Schaltungen mit Tristate-Gattern zu berechnen [102]. Durch lokale, auf verzweigungsfreie Regionen beschränkte Fehlersimulationen gelingt die Bestimmung von Fehleräquivalenzen, wodurch die Menge der in der Schaltung modellierten Gatterverzögerungsfehler, für die eine Simulation notwendig ist, reduziert wird [102]. Die Simulation ist ereignisgesteuert und wird ebenenweise in Richtung der Ausgänge ausschließlich für jene Gatter vorgenommen, die von der Fehlerinjektion oder nachfolgenden Signaländerungen beeinflusst werden.

Im Unterschied zu anderen Arbeiten findet eine mit 101 Werten sehr umfangreiche Logik Anwendung. Der Grund für die deutlich komplexere Logik liegt in der Unterstützung von Tristate-Gattern, welche die Menge der zu berücksichtigenden Basiswerte erhöhen und damit zu einer deutlich höheren Zahl an Signalkombinationen führen. Aus den erhaltenen Informationen können Aussagen über die Qualität der vorgegebenen Tests bezüglich der modellierten Fehler gemacht werden. Ferner wird, basierend auf der Annahme einer Einheitsverzögerung, die minimal detektierbare Fehlergröße ermittelt. Diese kann bei einem gegebenen Fehler und Testmuster je nach Testklasse unterschiedlich ausfallen, wie Abbildung 23 zeigt. Der Gatterverzögerungsfehler am Fehlerort x kann bei einer Fehlergröße $\delta > 1,5$ robust an e erkannt werden. Für eine Fehlergröße $\delta > 0,5$ stellt das Testmuster einen nicht robusten Test an d dar.

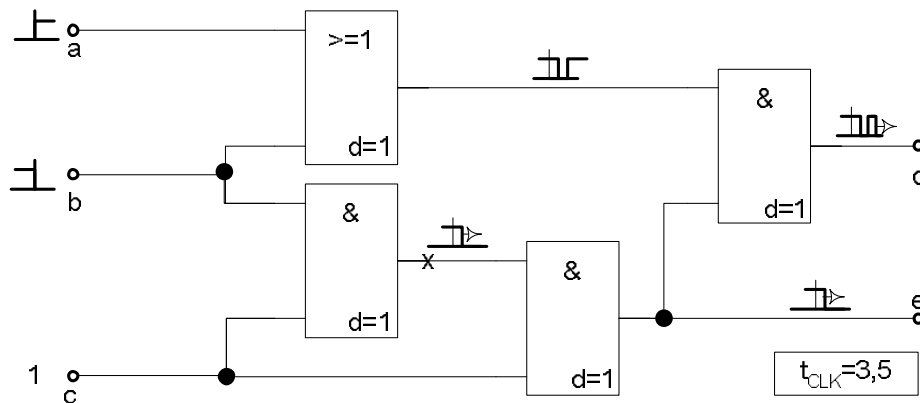


Abbildung 23: Ein Testmuster kann verschiedene Fehlergrößen mit unterschiedlicher Qualität detektieren

4.3 Fehlersimulation für Pfadverzögerungsfehler

In Abschnitt 2.5.2 wurde bereits angeführt, dass die Menge modellierter Pfadverzögerungsfehler schon für relativ geringe Schaltungsgrößen zumeist zu groß wird, um einen vollständigen Test zu ermöglichen. Die Größe der Fehlermenge stellt auch für die Fehlersimulation von Pfadverzögerungsfehlern ein Problem dar, sodass sich Pfadverzögerungsfehlersimulatoren in zwei Gruppen einteilen lassen; die *enumerativen* und *nicht enumerativen Verfahren*. Die enumerativen Verfahren [12, 17, 58, 67, 109, 124] verwenden eine Fehlerliste mit spezifizierten Pfadverzögerungsfehlern. Um den Speicherbedarf für die Beschreibung eines Fehlers so gering wie möglich zu halten, schlägt [12] eine Repräsentierung der Pfade durch eindeutige Nummern statt einer expliziten Angabe jeder Leitung auf dem fehlerhaften Pfad vor. Da aber auch dies nur für eine begrenzte Anzahl an Fehlern möglich ist, eignen sich enumerative Simulationsansätze nicht für große Schaltungen. Die nicht enumerativen Simulatoren [46, 53, 66, 108] versuchen, die Fehlerabdeckung eines gegebenen Testmusters über eine Zählung der getesteten Pfade abzuschätzen, ohne diese explizit in einer Liste zu führen.

Für die Anwendung des Pfadverzögerungsfehlermodells im Produktionstest kann das Problem der übergroßen Menge modellierter Pfadverzögerungsfehler außer acht gelassen werden, da im Regelfall eine begrenzte Anzahl an Pfaden vorgegeben wird. Die minimal detektierbare Fehlergröße ergibt sich aus der Nominallaufzeit des vorgegebenen Pfades, weswegen die Aufgabe eines Pfadverzögerungsfehlersimulators darin liegt, die Testqualität des gegebenen Testmusters für die in der Fehlerliste enthaltenen Fehler zu bestimmen. Da sich Informationen zur Testqualität mit Hilfe eines Gatterverzögerungsfehlersimulators, wie ihn z.B. [102] vorstellt, erlangen lassen,

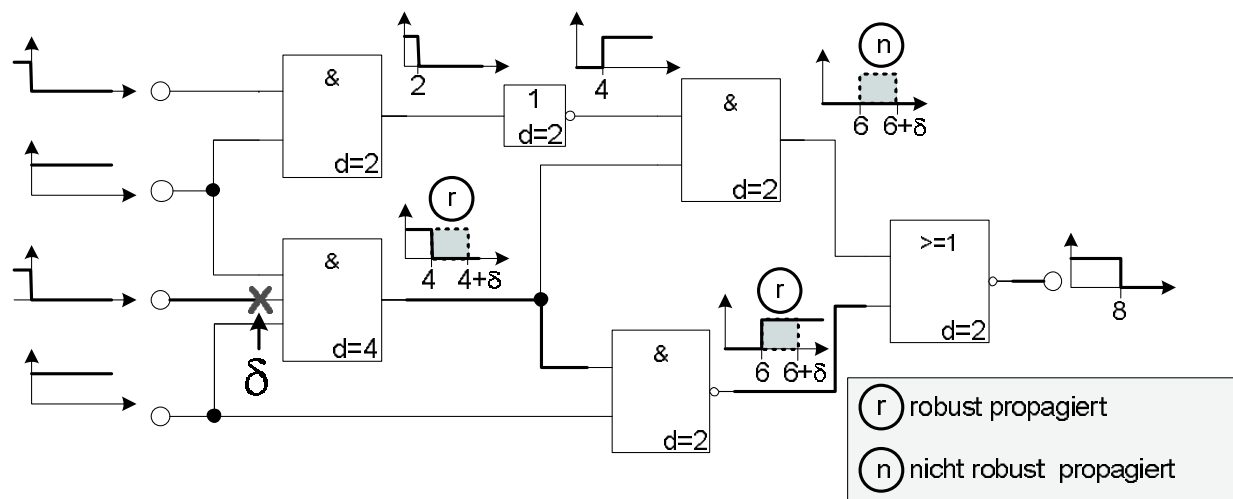


Abbildung 24: Degradierung der Testqualität durch Rekonvergenz

kann ein enumerativer Pfadverzögerungsfehlersimulator auf diesem aufbauend realisiert werden. Bei genauerer Betrachtung müssen hierbei 2 Fälle unterschieden werden.

1. Für den Fall, dass zwischen Pfadeneingang und Pfadausgang keine Rekonvergenz auftritt, trifft das Ergebnis bezüglich der Testqualität einer Gatterverzögerungsfehlersimulation bei einem angenommenem Gatterverzögerungsfehler am auf dem Pfad liegenden Schaltungseingang auch für den Pfadverzögerungsfehler zu, da dieser für den Gatterverzögerungsfehler einen kritischen Pfad zu einem Ausgang repräsentieren sollte.
2. Im Fall einer auftretenden Rekonvergenz kann der Einfluss der auf dem Pfad laufenden Flanke wieder über einen Seiteneingang auf den Pfad zurückwirken und damit Grund für eine Degradierung der Testqualität sein.

Wird auch beim Pfadverzögerungsfehlermodell von einer Einzelfehlerannahme ausgegangen, so wird angenommen, dass die Signalübergänge an den Seiteneingängen innerhalb der Spezifikation liegen. Folglich sollten die Fehlerattribute den Pfad nicht verlassen und eine Situation wie in Abbildung 24 dargestellt nicht auftreten.

Eine fehlerparallele Simulation kann das Problem vermeiden. Für die Bestimmung der zu simulierenden Fehler werden alle Gatterverzögerungsfehler am Ausgang jener auf dem Pfad liegenden Gatter gewählt, die einem Verzweigungspunkt folgen. Der Pfadverzögerungsfehler kann als robust getestet erkannt werden, wenn jeder injizierte Gatterverzögerungsfehler bis zum Ort des

5 Testmustergenerierung für Verzögerungsfehler

Im Unterschied zu statischen Fehlermodellen beschreiben dynamische Fehlermodelle den defektbedingten Einfluss auf die dynamischen Eigenschaften einer Schaltung. Im Falle der Verzögerungsfehlermodelle bewirkt der Fehler Φ eine Verzögerung der Signalflanke innerhalb der Schaltung. Ein Testmuster für Verzögerungsfehler besteht folglich aus (mindestens³³) zwei Testvektoren $\langle V_I, V_P \rangle$, die zu 2 aufeinander folgenden Taktzeitpunkten t_I, t_P an die Schaltung angelegt werden. Der Initialisierungsvektor (bzw. die ersten $(n-1)$ Initialisierungsvektoren aus einem Testmuster mit n Testvektoren) V_I hat die Aufgabe, die Schaltung dem Fehler entsprechend zu initialisieren, was primär die Einstellung des Ausgangswertes $iv(k_\Phi)$ ³⁴ am Fehlerort k_Φ zum Ziel hat. Der Propagationsvektor V_P des Testmusters muss den Signalwechsel am Fehlerort k_Φ hervorrufen. Darüber hinaus muss er sicherstellen, dass sich der Fehlereffekt Δ_Φ entlang eines kritischen Pfades P^k zu einem beobachtbaren Ausgang PO fortpflanzt.

5.1 Ansätze zur Testmustergenerierung für Verzögerungsfehlermodelle

Da Verzögerungsfehler eine Verzögerung von Signalübergängen bewirken, muss sich auch ihr Test über zwei Taktzyklen erstrecken (\rightarrow 2.5.2). Für die Bestimmung der Testmuster bieten sich verschiedene Möglichkeiten an, eine Berechnung durchzuführen. In [13, 40, 78, 99, 128] werden für die Berechnung spezielle Logiken entwickelt und angewandt, mit deren Hilfe sich die Signale mit ihren Eigenschaften in beiden Taktzyklen in jeweils einem Symbol beschreiben lassen. Die Testmustergenerierung wird danach unter Zuhilfenahme der in Abschnitt 2.9 vorgestellten und auf die jeweilige Logik angepassten Algorithmen realisiert.

Einen alternativen Ansatz bietet die getrennte Betrachtung der in der Schaltung vorkommenden Signale in zwei aufeinander folgenden Taktzyklen [26]. Das Problem der Testmustergenerierung für Verzögerungsfehler lässt sich damit in ein Problem der Berechnung von Testmustern für Haftfehler in sequentiellen Schaltungen überführen. Der in [118] und [44] vorgeschlagene Ansatz, eine Testmustergenerierung für robuste Pfadverzögerungsfehler durch eine Testmustergenerierung für Haftfehler in einer transformierten Schaltung zu ersetzen, ist für größere Schaltungen nicht

³³In [37, 104] wird gezeigt, dass ein aus 2 Vektoren bestehender Test nicht zu der maximalen Verzögerung auf dem untersuchten Pfad führen muss, da diese auch durch Initialisierungen vor dem ersten Taktzyklus beeinflusst werden. Eine Testmustergenerierung mit dem Ziel, mit 3 Vektoren für die Pfadverzögerungsfehlerdetektion diese zusätzlichen Erkenntnisse zu beachten, stellt [23] vor.

³⁴engl: initial value

praktikabel und wird damit für den Test aktueller Schaltungsgrößen nicht berücksichtigt. Ebenso wird wegen der aktuellen Schaltungsgrößen die Verwendung eines BDD-basierten Ansatzes (→ 2.9.5) ausgeschlossen.

5.1.1 Logiken zur Testmuster-generierung für Verzögerungsfehler

Der Test für Verzögerungsfehler benötigt eine Signalfanke durch den angenommenen Fehlerort, die im Fehlerfall entsprechend verzögert wird. Die in der Schaltung während der beiden Taktzyklen auftretenden Signale können damit sowohl rein statischer Natur sein, darüber hinaus jedoch auch einfache oder mehrfache Flankenwechsel (Hazards) beinhalten. Die Qualität eines Tests hängt maßgeblich von den auf den Fehlereffekt einwirkenden Signalen ab (→2.8), weshalb für die Bestimmung der Testqualität mögliche Mehrfachsignalwechsel nicht vernachlässigt werden können. Die Bestimmung der Testqualität kann zum einen mit Hilfe einer Fehlersimulation geschehen (→Kapitel 4), zum anderen, werden entsprechende Logiken verwandt, aber auch schon während der Testmuster-generierung beachtet werden. Ist dies das Ziel, müssen alle für die Bestimmung der Testqualität nötigen Eigenschaften in den Symbolen der verwendeten Logik beschrieben werden. Generell gilt dabei, dass mit einer zunehmenden Zahl der in der Logik enthaltenen Symbole die Vielfalt der codierbaren Informationen steigt, die auf dieser Logik aufsetzende Algebra und damit die Berechnungen jedoch an Komplexität zunehmen.

5.1.2 Sequentieller Ansatz

Eine Alternative zur Verwendung komplexerer Logiken bietet die Überführung des Testmuster-generierungsproblems für einen Verzögerungsfehler in das Problem einer Testmuster-generierung für einen Haftfehler in einer sequentiellen Schaltung [101]. Die zu testende Schaltung wird für die beiden beteiligten Taktzyklen getrennt betrachtet, was dem für sequentielle Schaltungen bekannten Abrollen um $n = 2$ Taktzyklen entspricht [69]. Der sequentielle Ansatz beschreibt die Signale zum Taktzeitpunkt, an dem sie einen definierten Wert aufweisen, führt also implizit eine Diskretisierung durch und ermöglicht so die Verwendung von Logiken, wie sie für statische Fehlermodelle Verwendung finden (Abbildung 26).

Werden die beiden für einen Verzögerungsfehler-test notwendigen Testvektoren auf ihre notwendigen Eigenschaften hin untersucht, ergeben sich Parallelen zu den Eigenschaften der Testvektoren klassischer Fehlermodelle. Die grundlegende Aufgabe des Initialisierungsvektors besteht darin, den Fehlerort auf den Ausgangswert zu initialisieren. Diese Initialisierung erfolgt, ab-

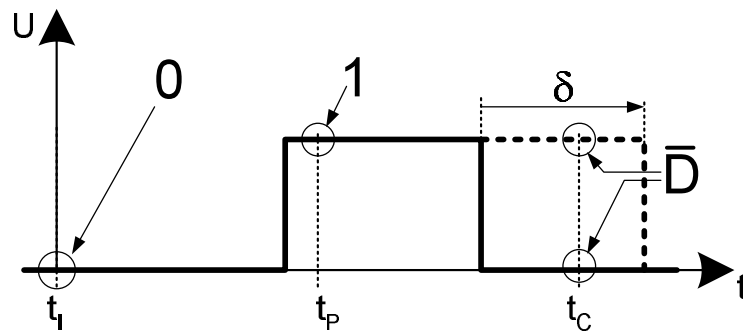


Abbildung 26: Die Betrachtung der Signalwerte zu den Taktzeiten führt implizit zu einer Diskretisierung

hängig vom Fehlermodell, an einem Knoten innerhalb der Schaltung (Übergangs- und Gatterverzögerungsfehlermodell) oder am Anfang eines Pfades (Pfadverzögerungsfehlermodell). Der Propagationsvektor hat, verglichen mit einem Testvektor für das Haftfehlermodell, die gleichen Aufgaben zu erfüllen. Dies umfasst zum einen, den Fehlerort auf den dem angenommenen Fehlerwert entgegengesetzten Wert umzuschalten, und zum anderen, den Fehlereffekt entlang des kritischen Pfades zu einem der Ausgänge zu propagieren. Ein Testmuster für Verzögerungsfehler kann demnach aus einer Kombination von Testmustern für Haftfehler bestehen.

Diese Analogien ermöglichen es, das Testmustergenerierungsproblem für einen Verzögerungsfehler in zwei Probleme aufzuspalten, die sich mit einem haftfehlerbasierten Testmustergenerator lösen lassen. Eine vollständige Trennung der beiden Aufgaben ist dabei jedoch nicht gegeben, da der Initialisierungsvektor, abhängig vom Applikationsmodus (\rightarrow 2.10), Einfluss auf den Propagationsvektor nimmt.

5.2 Testmustergenerierung

5.2.1 Testmustergenerierung für das Übergangsfehlermodell

Beim Übergang vom statischen Haftfehlermodell auf dynamische Verzögerungsfehlermodelle weist das Übergangsfehlermodell die größten Gemeinsamkeiten mit dem Haftfehlermodell auf (\rightarrow 2.5.2). Da keine Bewertung der Fehlergröße erfolgt und der Fehler über jeden kritischen Pfad durch den Fehlerort testbar ist, verhält er sich für die Dauer des Propagationszyklus wie ein Haftfehler am Fehlerort. Für einen zu detektierenden slow-to-rise (slow-to-fall) Fehler muss der Initialisierungsvektor den fehlerhaft angenommenen Knoten k_Φ auf den Ausgangswert der Flanke

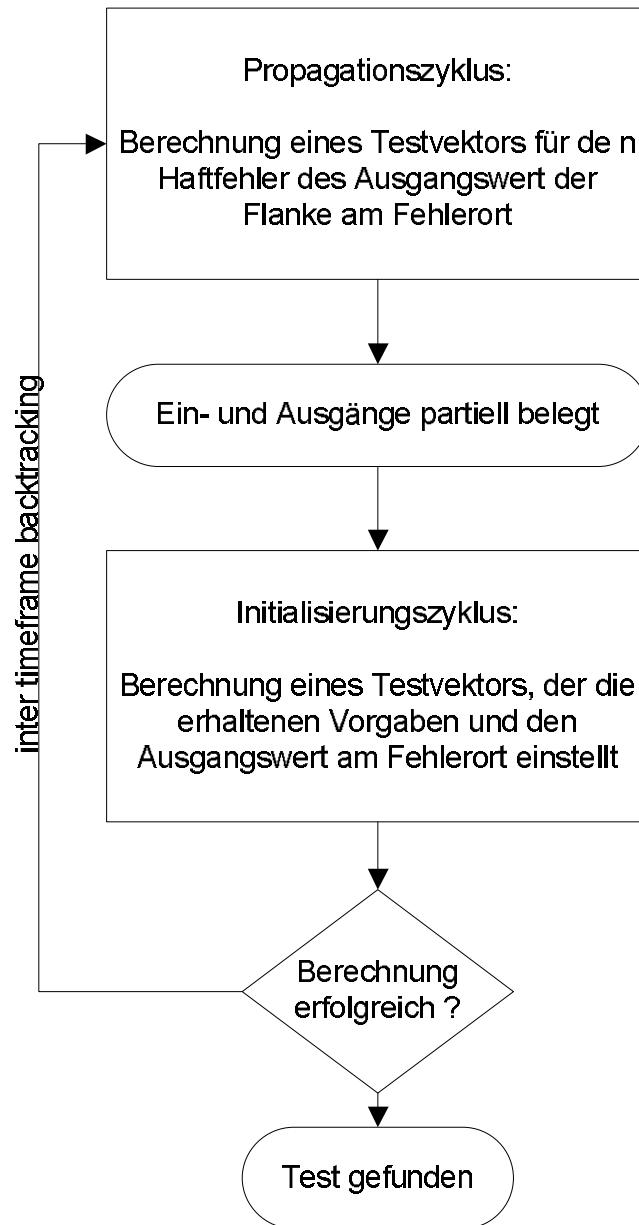


Abbildung 27: Ablaufdiagramm bei Berechnung von V_I und V_P in getrennten Zeiträumen

einstellen. Der Propagationsvektor muss diesen Knoten umschalten und die so erzeugte Flanke über einen (beliebigen) kritischen Pfad an einen beobachtbaren Ausgang fortschalten. Die gleichen Ziele hat eine Haftfehlertestmustergenerierung für einen angenommenen Haftfehler auf dem Anfangswert der Flanke.

Die naheliegendste Verwendung von Haftfehlertests für die Übergangsfehlerdetektion liegt in der Erkenntnis, dass im ersten Taktzyklus am Fehlerort der Ausgangswert der Flanke eingestellt werden muss. Dies gelingt durch den Einsatz eines Haftfehlertestmusters, welches am Fehlerort einen dem gewünschten Ausgangswert entgegengesetzten Haftfehler testet. Für den zweiten Taktzyklus wird ein Umschalten auf den Endwert durch ein Haftfehlertestmuster für den dem Endwert entgegengesetzten Haftfehler bewirkt. Unter Beachtung der durch die Testmusterzuführung eingebrachten Abhängigkeiten zwischen den beiden Testvektoren ($\rightarrow 2.10$) kann ein Testmustergenerator für Übergangsfehler unter Verwendung eines Testmustergenerators für Haftfehler wie in Abbildung 27 dargestellt realisiert werden. Die Berechnung beginnt mit der Bestimmung des Propagationsvektors V_P . Nach einer erfolgreichen Testmustergenerierung für einen Haftfehler am angenommenen Fehlerort im 2. Taktzyklus sind die Eingangswerte für den Propagationsvektor teilweise bestimmt. Abhängig vom Applikationsmodus ergeben sich damit Randbedingungen, die das für den ersten Taktzyklus zu bestimmende Testmuster erfüllen muss. Für einen Test im Scan-Shift Modus sind die Eingangswerte der Schaltung im 1. Taktzyklus bereits partiell bestimmt. Wird für den Test die funktionale Einstellung vorausgesetzt, muss V_I für den Haftfehler im ersten Taktzyklus zusätzlich über die Schaltungsfunktion die vorbestimmten Ausgänge einstellen.

Die Ergebnisse der Testmustergenerierung über 2 Haftfehlertestvektoren ist unter Umständen erfolglos, obwohl ein Testmuster für den adressierten Fehler existiert. Grund hierfür ist eine zu starke Einschränkung des Suchraums durch die Verwendung eines Haftfehlertestvektors für V_I . Der Initialisierungsvektor hat grundsätzlich die Aufgabe, den Fehlerort auf den Ausgangswert zu initialisieren. Ein Testmuster nach dem Haftfehlermodell muss zusätzlich einen kritischen Pfad zu einem der Ausgänge sensitivieren, über den der angenommene Fehler sichtbar wird. Neben dem damit erhöhten Aufwand für die Testmustergenerierung im ersten Initialisierungszyklus sinkt der Freiheitsgrad für die Testmustergenerierung. Kann ein Haftfehlertestmuster nur deshalb nicht gefunden werden, weil eine Propagation des Fehlereffekts nicht gelingt, obwohl sich der als fehlerhaft angenommene Knoten auf den gewünschten Wert einstellen lässt, kann ein mögliches Testvektorpaar nicht als Test erkannt werden.

Ein weiteres Problem stellt die getrennte Betrachtung der beiden Taktzyklen dar. Gelingt die Testmustergenerierung für einen Haftfehler im Propagationszyklus, während die anschließende

Suche im Initialisierungszyklus erfolglos bleibt, kann dies in den aus dem Propagationsvektor erhaltenen Vorgaben für den ersten Taktzyklus begründet sein. Die einzige Lösung wäre die Berechnung eines alternativen Testmusters für den im Propagationszyklus angenommenen Haftfehler, um danach im Initialisierungszyklus unter den veränderten Vorgaben eine erneute Berechnung durchzuführen. Experimente mit dem Prototypen eines Testmustergenerators für das Übergangsfehlermodell haben ergeben, dass dieses mit *inter-timeframe-backtracking* beschriebene Problem in der Praxis häufig auftritt. Für Fehler, die zu einem erhöhten Inter-timeframe-backtracking führen, kann ohne eine Berechnung aller möglichen Propagationsvektoren und den damit verbundenen Berechnungsversuchen für den Initialisierungsvektor keine Aussage über eine Redundanz gemacht werden. Es ist leicht einzusehen, dass eine fehlgeschlagene Berechnung des Propagationsvektors³⁵ durch die aus dem Propagationsvektor folgenden Vorgaben und damit die funktionale Abhängigkeit der beiden Testmuster voneinander bedingt ist. Um die Testmustergenerierung effektiver zu gestalten muss folglich versucht werden, diese Abhängigkeiten der Testmustergenerierung zugänglich zu machen, um sich einstellende Widersprüche möglichst früh zu erkennen. Dies kann nur durch das Auflösen der Grenze zwischen den beiden Taktzyklen geschehen, also dem Aufheben der strikten Trennung zwischen der Darstellung der Schaltung in den beiden Taktzyklen. Werden beide Schaltungsrepräsentationen parallel im Arbeitsspeicher gehalten und der gewählte Applikationsmodus über eine entsprechende Verknüpfung realisiert, können mögliche Abhängigkeiten zwischen den Taktzyklen bereits bei der Berechnung des V_P erkannt werden. Die zwischen den beiden Schaltungsrepräsentationen einzufügenden Verbindungsleitungen führen, je nach gewählten Applikationsmodus, entweder von den pseudo primären Ausgängen PPO_i zu den korrespondierenden pseudo primären Eingängen PPI_i (funktionale Einstellung, Abbildung 28) oder vom pseudo primären Eingang PPI_{i,T_I} im ersten Taktzyklus zum in der Scan-Kette benachbarten PPI_{i+1,T_F} in der Darstellung des Propagationszyklus (Scan-Shift-Modus, Abbildung 29). Ein weiterer Vorteil der Aufhebung einer strikten Trennung zwischen den Schaltungsrepräsentationen liegt in der Möglichkeit, eine Haftfehler-testmuster-generierung für den Fehlerort im Initialisierungszyklus zu vermeiden. Stattdessen fließt die notwendige Anforderung, diesen Knoten auf den Ausgangswert der Flanke einzustellen, als Randbedingung in die auf der Gesamtrepräsentation der Schaltung basierende Testmuster-generierung ein.

³⁵voraussetzend, dass der adressierte Haftfehler im ersten Taktzyklus nicht redundant ist

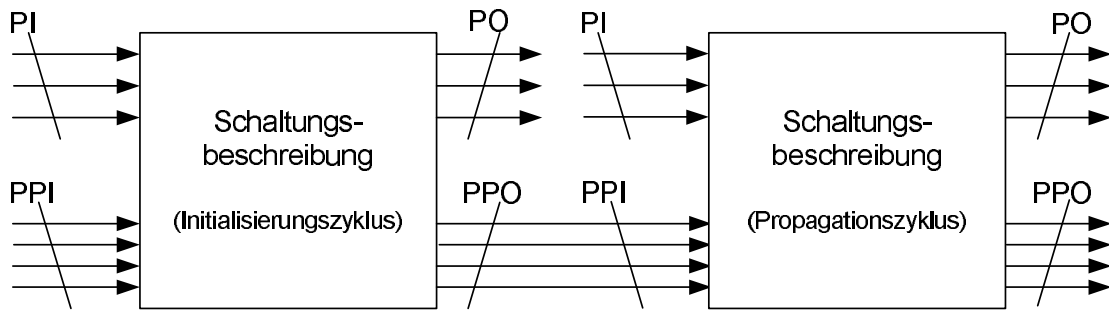


Abbildung 28: Netzlistenverknüpfungen bei Applikationsmodus “funktionale Einstellung”

5.2.2 Testmustergenerierung für das Gatterverzögerungsfehlermodell

Das Gatterverzögerungsfehlermodell folgt aus dem Übergangsfehlermodell, wenn die detektierbare Fehlergröße beachtet wird. Wie in Abschnitt 2.5.2 beschrieben, ergibt sich die detektierbare Fehlergröße über einen beliebigen kritischen Pfad P^k als Differenz aus der Nominallaufzeit des Signals auf diesem Pfad und der Taktzeit. Findet die *minimal* detektierbare Fehlergröße bei der Testmustergenerierung Berücksichtigung, führt dieses, die Suche nach einem optimalen Test vorausgesetzt, zu einer Explosion des Aufwands für die Testmustergenerierung.

Bei der Testmustergenerierung für Haftfehler werden bei der Suche nach konsistenten Eingangsbelegungen Heuristiken verwandt, die auf Messgrößen wie die Einstellbarkeit und die Beobachtbarkeit zurückgreifen (→2.9.3). Für die Testmustergenerierung für einen Haftfehler an einem Schaltungsknoten k_Φ resultiert die Verwendung der Heuristiken in dem Versuch, einen möglichst kurzen kritischen Pfad zu finden, da hier die geringste Anzahl an Gattern transparent zu belegen ist. Während die beste Einstellbarkeit tendenziell dem Eingang zukommt, der den geringsten “Abstand” zu den Eingängen hat wird die beste Beobachtbarkeit jenem kritischen Pfad ab dem Fehlerort zukommen, auf dem die geringste Anzahl an Gattern transparent zu schalten ist. Die Forderungen nach einem Pfad maximaler Länge steht beiden Interessen entgegen, resultieren daraus doch eine maximale Anzahl an einzustellenden Gattern bzw. eine maximale Anzahl an geforderten Signalwerten. Es ist leicht einzusehen, dass die Wahrscheinlichkeit, dass sich zwei Signalwerte aufgrund logischer Verknüpfungen durch die Schaltungsfunktion gegenseitig ausschließen, mit der Anzahl der geforderten Signale innerhalb der Schaltung steigen wird.

In der Literatur werden einige Testmustergeneratoren für das Gatterverzögerungsfehlermodell vorgestellt. In der ersten, 1988 vorgestellten Arbeit [62, 63] versucht der Algorithmus, einen Test für einen möglichst langen Pfad durch den Fehlerort zu realisieren. Um eine möglichst hohe Robustheit zu erreichen wird über Heuristiken versucht, die Anzahl der Signalwechsel in

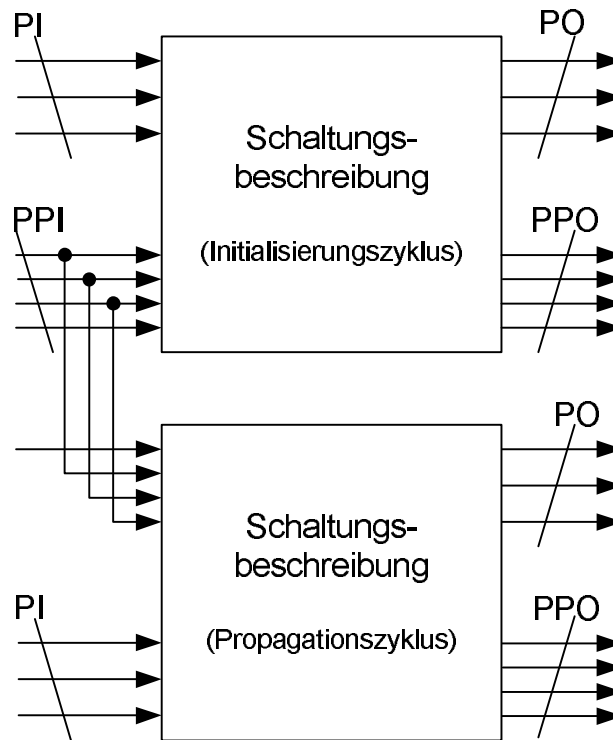


Abbildung 29: Netzlistenverknüpfungen bei Applikationsmodus "Scan-Shift"

der Schaltung zu minimieren, da diese den Grund für eine mögliche Invalidierung bilden. Eine anschließende Gatterverzögerungsfehlersimulation des erhaltenen Testmusters kann über die Beachtung der Laufzeiten in der Schaltung eine Aussage über die erreichte Testqualität und die minimal erkennbare Fehlergröße machen. Ein Vergleich ihres Ergebnisses zeigt, ob die gestellten Vorgaben erreicht wurden, andernfalls wird versucht, einen alternativen Test zu generieren. Eine zweite Arbeit von Park et al. [97, 98] generiert zunächst für alle Fehler der Fehlerliste Testmuster unter Verwendung des Übergangsfehlermodells. Für den erhaltenen Testwertsatz Θ wird anschließend einer Fehlersimulation nach dem Gatterverzögerungsfehlermodell durchgeführt. Verbleiben Fehler, die von Θ nicht mit ausreichender Qualität detektiert werden, wird für diese eine Testmustergenerierung unter Beachtung des längsten Pfades durch den Fehlerort durchgeführt, das erhaltene Testmuster T dem Testwertsatz Θ hinzugefügt und Θ erneut simuliert. Dies wiederholt sich für jeden Fehler, bis alle Fehler mit ausreichender Qualität für jeden der verbliebenen Fehler ein Testmuster nach dem Gatterverzögerungsfehlermodell berechnet wurde. Eine dritte Arbeit stellt [80] vor. Das Ziel der Testmustergenerierung ist dort die Generierung eines optimalen Gatterverzögerungsfehler-tests. Der Algorithmus arbeitet cone-orientiert und hält eine Liste aller noch möglichen Pfade mit deren Länge vor, um frühzeitig Nichtlösungsgebiete zu

erkennen, die in der minimal erkennbaren Fehlergröße begründet sind.

Zusammengefasst ist eine Testmustergenerierung für Gatterverzögerungsfehler mit dem Ziel, einen optimalen Test zu erhalten, für Schaltungen heutiger Größe eine Aufgabe, die in vertretbarer Rechenzeit nicht zu bewerkstelligen ist. Sind trotzdem Aussagen über die detektierbare Fehlergröße gewünscht, kann diese über eine nachträgliche Fehlersimulation vorgegebener Testmuster unter Verwendung eines Fehlersimulators für das Gatterverzögerungsfehlermodell (→4.2) erreicht werden.

5.2.3 Testmustergenerierung für das Pfadverzögerungsfehlermodell

Die in Abschnitt 5.2.2 erwähnten Probleme treten bei der Testmustergenerierung für Pfadverzögerungsfehler nicht auf, da der kritische Pfad Teil des Fehlermodells ist; der zu testende Pfad ist damit über den aktuellen Zielfehler bereits definiert. Die Testmustergenerierung hat demnach die Aufgabe, diesen Pfad zu sensitivieren und gewünschte Flanke am Pfadeingang einzustellen.

In der Literatur finden sich verschieden Arbeiten mit dem Ziel der Testmustergenerierung für das Pfadverzögerungsfehlermodell. Der Schwerpunkt der Arbeiten besteht dabei zumeist in der Definition von Testqualitäten sowie der Entwicklung und Verwendung geeigneter Logiken, die eine Testmustergenerierung für diese Testqualitäten ermöglichen. Eine Testmustergenerierung für robuste Pfadverzögerungsfehlertests adressieren unter anderen [11, 22, 23, 27, 30, 40, 41, 78, 124, 99, 128, 130]. Während [124] bei der Definition des Pfadverzögerungsfehlermodells für die robuste Testmustergenerierung 6 Symbole verwendet, kommen [78, 99] mit 5 Symbolen aus. Der 1991 vorgestellte Testmustergenerator DYNAMITE [40] verwendet für die robuste Testmustergenerierung eine 10-wertige Logik. Die Autoren zeigen auf, dass die zuvor verwandten Logiken unvollständig waren, weshalb ihre Verwendung in einigen Fällen die Erkennung von Implikationen verhinderte. Der Testmustergenerator realisiert unter Verwendung dieser 10-wertigen Logik einen FAN-orientierten Algorithmus (→ 2.9.3) und nutzt für die Erkennung globaler Implikationen Lernverfahren (→2.9.4). Der in [130] vorgestellte Testmustergenerator ist ebenfalls FAN-basiert, nutzt wegen der Unterstützung von Tristate-Gatter jedoch eine 20-wertige Logik für die Berechnung. FASTPATH [128] berücksichtigt ebenfalls Gatter mit hochohmigem Ausgangszustand, nutzt jedoch eine 29-wertige Logik für die Berechnung und geht bei der Belegung der Eingangssignale PODEM-orientiert vor.

In [41] gelingt es erstmalig, auch für die ISCAS Schaltung c6288 eine Fehlerabdeckung anzugeben. Der in der Arbeit vorgestellte Testmustergenerator RESIST unterscheidet sich von vorherigen Ansätzen in der Erkenntnis, dass viele Pfade eine Untermenge an gemeinsamen Teilpfadab-

schnitten teilen. Diese werden ein einziges Mal sensitiviert und das Ergebnis für alle betroffenen Pfade verwendet. Als Folge werden viele Sensitivierungsvorgänge eingespart, was den Generator effektiver erscheinen lässt.

Die in [30] und [11] vorgestellten Testmustergeneratoren verwenden den in Abschnitt 2.9.5 angesprochenen Ansatz binärer Entscheidungsbäume und sind damit in der maximalen Größe der zu bearbeitenden Schaltung beschränkt.

Ein zu testender Pfadverzögerungsfehler Φ^P ist durch den strukturellen Pfad³⁶ P^s und die gewünschten Flanken auf den einzelnen Leitungen dieses Pfades gegeben. Um einen nicht robusten Test (\rightarrow 2.8.2) für Φ^P zu realisieren, können die grundsätzlich zu erfüllenden Anforderungen bestimmt werden [25]:

1. Alle Seiteneingänge des Pfades P^k nehmen im Propagationszyklus den nicht kontrollierbaren Wert an, was einer fehlerleitenden Belegung entspricht.
2. Am Pfadeingang wird durch die Testvektoren $\langle V_I, V_P \rangle$ eine Flanke initiiert

Die beiden Vorgaben lassen für den zu testenden Pfadverzögerungsfehler Φ^P mit dem sequentiellen Ansatz aus 5.1.2 erfüllen.

5.3 Testmustergenerierung für Pfadverzögerungsfehler unter Verwendung des sequentiellen Ansatzes

Der größte Nachteil bei der Verwendung von Algorithmen und Logiken, die auf statische Fehlermodelle abgestellt sind, für den Bereich der Testmustergenerierung für dynamische Fehlermodelle (\rightarrow 5.1.2) liegt in der Unmöglichkeit, die Einhaltung der Randbedingungen für robuste Tests zu garantieren. Sind robuste Testmuster das vorgegebene Ziel der Testmustergenerierung, entfällt folglich die Wahlmöglichkeit nach Abschnitt 5 für die Realisierung des Testmustergenerators. Auf der anderen Seite kann die in der Literatur überwiegend vertretene Meinung, robuste Tests seien, da sie nicht invalidiert werden können, generell den nicht robusten Tests vorzuziehen, ohne Einschränkung nicht gelten. In [105, 38] wird gezeigt, dass unterschiedliche Testmuster für den gleichen kritischen Pfad in unterschiedlichen Signallaufzeiten resultieren. Mit Bezug auf die

³⁶Der Begriff ‘struktureller Pfad’ beschreibt einen topologischen Pfad von einem Schaltungseingang zu einem Schaltungsausgang. Jedem strukturellen Pfad werden 2 Pfadverzögerungsfehler (steigende und fallende Flanke am Pfadeingang) zugeordnet.

Qualität der Tests ergibt sich, dass in bestimmten Fällen nicht robuste Testmuster zu einer größeren Signallaufzeit führen als robuste [103], weswegen die Anwendung der robusten Testmuster in diesen Fällen nicht zur Detektion der minimal erkennbaren Fehlergröße führen bzw. nicht den Worst-Case in der Schaltung einstellen. Da die Erfahrung ferner zeigt, dass im Regelfall für die meisten Pfade kein robuster Test existiert, muss die Berechnung robuster Testmuster für den Produktionstest nicht vorrangiges Ziel sein.

Auf den Einsatz von Verzögerungsfehler-tests im Produktionstest haben Faktoren Einfluss, die bei einer akademischen Betrachtung des Testproblems keine oder eine untergeordnete Rolle spielen. Treibender Faktor für den industriellen Einsatz sind die Kosten, die für den Test aufzuwenden sind. Diese werden zum einen durch die Anforderungen an das zu verwendende Testequipment bestimmt (\rightarrow 2.10), zum anderen durch die Zahl zusätzlicher Vektoren, die dem Testwertsatz durch das neue Fehlermodell hinzugefügt werden und welche die Testzeit verlängern. Folglich besteht -wie bei den statischen Fehlermodellen- eine der Hauptanforderungen an den Test in einer möglichst hohen Fehlerabdeckung der einzelnen Testmuster. Weitere wirtschaftliche Faktoren für den Aufbau und die Anwendung eines Testmuster-generators ist der zu seiner Erstellung notwendige Aufwand, die reibungslose Integration in den vorhandenen Toolflow sowie ein möglichst geringer Wartungsaufwand. Die Wiederverwendung bereits bestehenden Codes kann jeden dieser Faktoren positiv beeinflussen. Die Herstellungskosten sinken durch fehlende Reimplementierung und Fehlersuche, die Integration des Generators in den Toolflow wird durch die vorhandenen Schnittstellen erleichtert und der Wartungsaufwand verringert, da der Support wiederverwendeter Module mehreren Produkten zugute kommt. Ferner wirken sich Optimierungen an diesen Codesegmenten auf mehrere Produkte leistungssteigernd aus.

Einen Überblick über die Ein- und Ausgabedaten zeigt Abbildung 30.

5.3.1 Fehlerlisten und Pfadbeschreibung

Wie aus Abbildung 30 ersichtlich werden zu testende Pfade dem Testmuster-generator extern vorgegeben. Für die Auswahl der Pfade bestehen mehrere Möglichkeiten. Ist sich der Designer der Schaltung der laufzeitkritischen Pfade bewusst, können diese von Hand selektiert und vorgegeben werden. Andernfalls werden im allgemeinen Timinganalysatoren eingesetzt, um computerunterstützt einen Satz möglicherweise kritischer Pfade zu ermitteln [49, 83]. Da bei der Verwendung statischer Timinganalysatoren eine Unterscheidung nach funktionalen und nicht

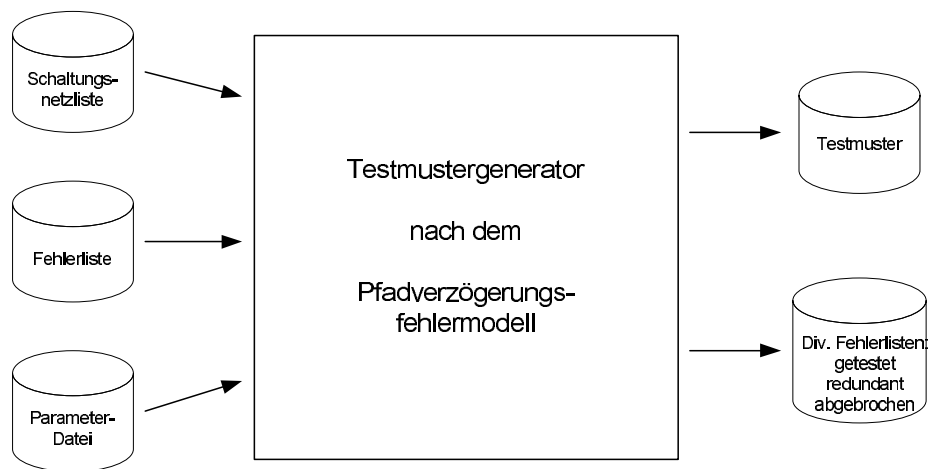


Abbildung 30: Überblick zum Testmustergenerator

funktionalen Pfaden³⁷ unmöglich ist kann die Menge der erhaltenen Pfade auch solche enthalten, die zu redundanten Pfadverzögerungsfehlern führen.

Liegt eine externe Definition der zu untersuchenden Pfade nicht vor, kann intern eine Fehlerliste aufgebaut werden. Die Größe der Liste ist über Auswahlkriterien beeinflussbar. Der Aufbau einer Liste *aller* Pfade ist wegen des Speicherbedarfs und der später aufzuwendenden Testzeit am Prüfling nur für kleinere Schaltungen sinnvoll. Wird eine prozentuale Schwelle mit Bezug auf den längsten strukturellen Pfad der Schaltung angegeben, kann die Fehlerliste aus allen Pfaden bestehen, deren Länge mindestens dieser Schwelle entspricht. Für Schaltungen, die eine große Anzahl an Pfaden mit annäherungsweise maximaler Länge aufweisen³⁸, kann dieser Ansatz weiterhin zu übergroßen Listen führen und damit nicht praktikabel sein. Eine dritte Möglichkeit, die Pfadliste aufzubauen, bietet die Verwendung der n längsten Pfade jedes Eingangs, womit die Menge der Pfade maximal dem n -fachen der Schaltungseingänge entspricht. Soll die Testmuster-generierung neben den durch das Fehlermodell adressierten kleinen, verteilten Fehlern auch die Abwesenheit von großen, lokalen Verzögerungsfehlern auf allen Leitungen sicherstellen, kann die aufzustellende Fehlerliste aus der Menge der längsten Pfade bestehen, die durch jede Leitung innerhalb der Schaltung laufen³⁹.

In Abschnitt 4.3 wurde bereits erwähnt, dass Pfade innerhalb einer verzweigungsfreien Region

³⁷Im englischen "false path" genannt. Es ist bekannt, dass die längsten strukturellen Pfade innerhalb einer Schaltung zumeist keine funktionalen Pfade sind [71].

³⁸was zum Beispiel bei laufzeitoptimierten Schaltungen der Fall ist

³⁹Die Erstellung der internen Fehlerlisten basiert auf der Annahme einer Einheitsverzögerung je Gatter, da in den verwendeten Netzlisten keine Timinginformationen vorhanden sind.

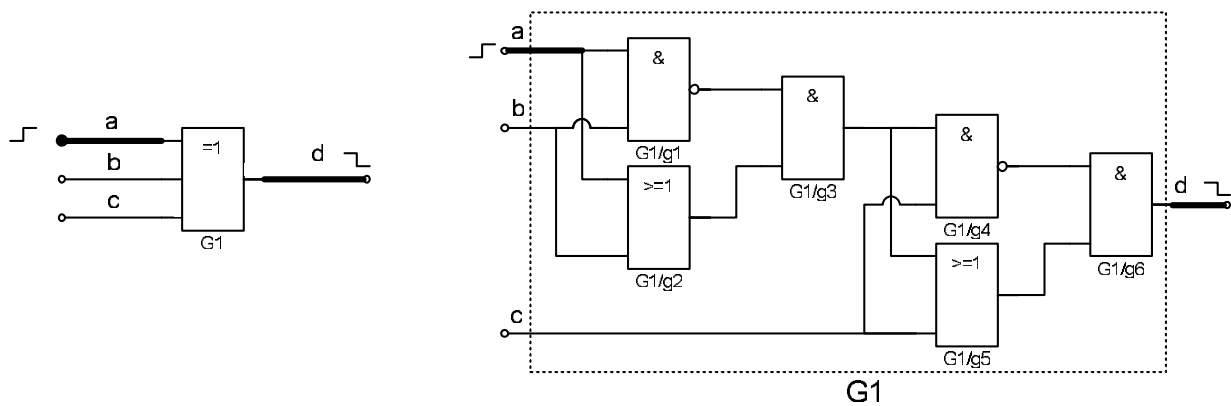


Abbildung 31: Der Pfad $\{(a, \uparrow), (d, \downarrow)\}$ ist in der Ersatzschaltung nicht mehr eindeutig definiert.

durch den Eingang in diese Region eindeutig definiert sind. Aus diesem Grund kann, um den Speicherplatzbedarf für die Pfadrepräsentierung minimal zu halten, der zu testende Pfadverzögerungsfehler als Liste jener Leitungen beschrieben werden, über welche er in die durchquerten verzweigungsfreien Regionen eintritt. Die Beschreibung bleibt damit bei deutlich verringerten Speicherbedarf eindeutig.

Probleme bei externer Pfadvorgabe Bei externer Vorgabe der zu untersuchenden Pfade kommt es in der Praxis zu Problemen, die ihren Ursprung in der unterschiedlichen Repräsentierung der Schaltungsnetzliste haben können. Die Testmustergenerierung basiert im allgemeinen auf einer möglichst technologieunabhängigen Netzlistenbeschreibung, während für die Bestimmung der kritischen Pfade im Design möglichst genaue Laufzeitinformationen und damit eine technologienahe Schaltungsbeschreibung gewünscht ist. Da verschiedene Fertigungsprozesse unterschiedliche Gattertypen unterstützen, werden diese für die Testmustergenerierung in funktionale Schaltungsblöcke identischer Funktion umgewandelt, die aus den vom Testmustergenerator unterstützten Basiselementen gebildet werden. Für ein Beispiel sei angenommen, dass die für eine Schaltung gewählte Fertigungstechnologie ein Exklusiv-Oder-Gatter bereitstelle (Abbildung 31). Für die Testmustergenerierung resultiert das nicht unterstützte Exklusiv-Oder Gatter in einer Ersatzschaltung aus unterstützten UND- und ODER-Gattern. Als Folge der Netzlisten-transformation können innerhalb der nachgebildeten Schaltungsfunktionen neue Verzweigungspunkte entstehen, die im Falle einer Rekonvergenz zu weiteren Pfaden in der Schaltung führen. Der ehemals eindeutige, extern vorgegebene Pfad kann daher durch die Transformation seine Eindeutigkeit verlieren. Die Testmustergenerierung erlangt damit weitere Freiheitsgrade, da sie einen Test für *einen* der (in der ihr vorliegenden Netzlistenbeschreibung) möglichen Pfadverzö-

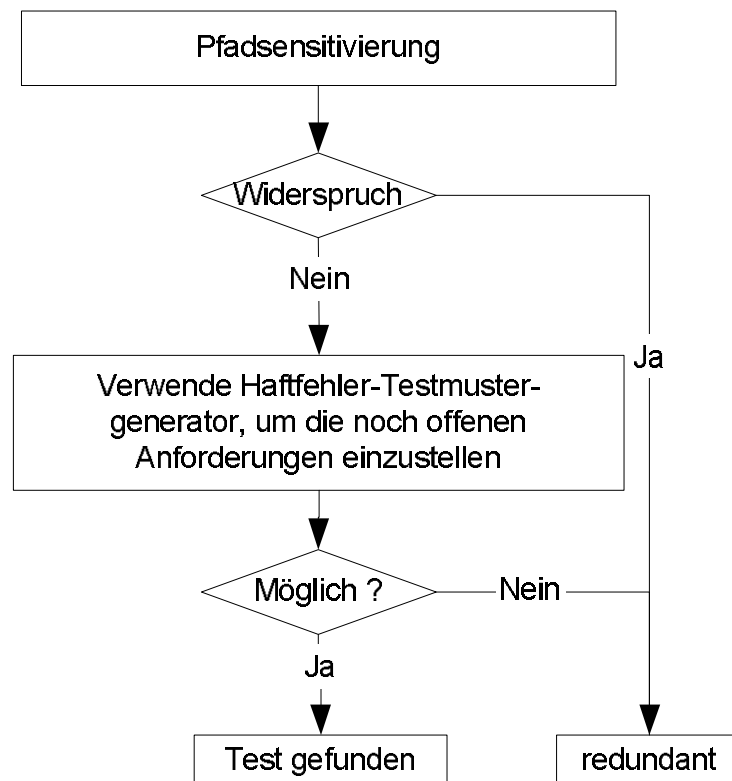


Abbildung 32: Flussdiagramm des Testmustergenerators für Pfadverzögerungsfehler

gerungsfehler zu finden hat.

5.3.2 Algorithmus zur Testmustergenerierung für Pfadverzögerungsfehler

Unter den zu berücksichtigenden Vorgaben (Seite 65) wurde ein Algorithmus für die Testmustergenerierung zur Detektion von Pfadverzögerungsfehlern entwickelt. Die Berechnung der Testmuster läuft in 2 Schritten ab (Abbildung 32). Im ersten Schritt werden auf den Leitungen entlang des Pfades die notwendigen Werte gesetzt und damit der Pfad sensitiviert, während der 2. Schritt die noch einzustellenden Signalleitungen konsistent belegt.

5.3.3 Pfadsensitivierung

Die Pfadsensitivierung wird für jedes Pfadsegment entlang des Pfades durchgeführt, solange es nicht (bedingt durch lokale Implikationen) zu einem Widerspruch kommt. Für das erste Pfadsegment sind zwei Schritte durchzuführen: Im Propagationszyklus wird dem Pfadsegment der

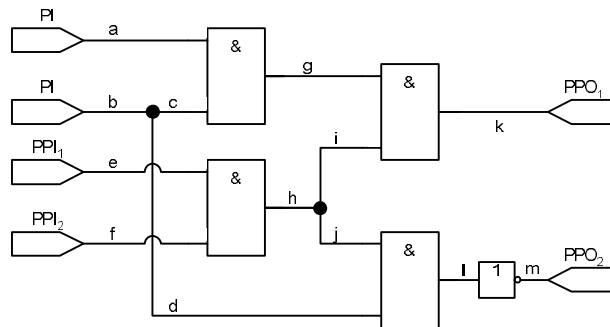


Abbildung 33: Eine einfache Beispielschaltung

Wert $D(\bar{D})$ zugewiesen und eine lokale Implikation durchgeführt. Die lokalen Implikationen führen zu Signalbelegungen innerhalb der verzweigungsfreien Region⁴⁰, in welcher sich die aktuelle Leitung befindet, werden den Fehlereffekt also bis zum nächsten Verzweigungspunkt propagieren und die notwendigen Seiteneingangsbelegungen vornehmen. Die Verwendung der vom D-Algorithmus bekannten D-Notation im Propagationszyklus scheint wegen des vorgegebenen Pfades nicht notwendig, da keine D-Front aufgebaut wird. Andererseits wird jedoch bei der Bestimmung notwendiger Seiteneingangsbelegungen der auf den Pfad liegenden Gatter die Erkennung, welche der Leitungen dem Pfad zugerechnet werden muss, erleichtert. Im Anschluss wird auf dem Pfad Eingang im Initialisierungszyklus der Ausgangswert der Flanke eingestellt und eine lokale Implikation durchgeführt. Tritt hierbei kein Widerspruch auf, ist die Signalflanke am Pfad Eingang eingestellt. Für alle weiteren Leitungen auf dem Pfad wird im Propagationszyklus der entsprechende Endwert in D-Notation (D für eine steigende, \bar{D} für eine fallende Flanke auf der betroffenen Leitung) gesetzt und lokale Implikationen durchgeführt. Ergeben sich während der Pfadsensitivierung Widersprüche, ist die Redundanz des Pfadverzögerungsfehlers bewiesen, da alle Signalbelegungen notwendige Wertsetzungen darstellen.

5.3.4 Einstellen der noch einzustellenden Knoten

Gelingt die Pfadsensitivierung ohne Widerspruch, verbleiben in der Schaltung Signalanforderungen an Leitungen, die in einem zweiten Schritt einzustellen sind. Für die Auflösung der Anforderungen können die aus den Algorithmen zur Testmuster generierung bekannten Verfahren (\rightarrow 2.9) verwendet werden. Für die Realisierung des Prototypen wurde auf einen im Sourcecode vorhandenen Testmuster generator für Haftfehler [129] zurückgegriffen, der den FAN Algorithmus

⁴⁰Da sich innerhalb einer Fanout-freien Region keine Verzweigung (Fanout) befindet ist der kritische Pfad innerhalb dieser Region eindeutig bestimmt

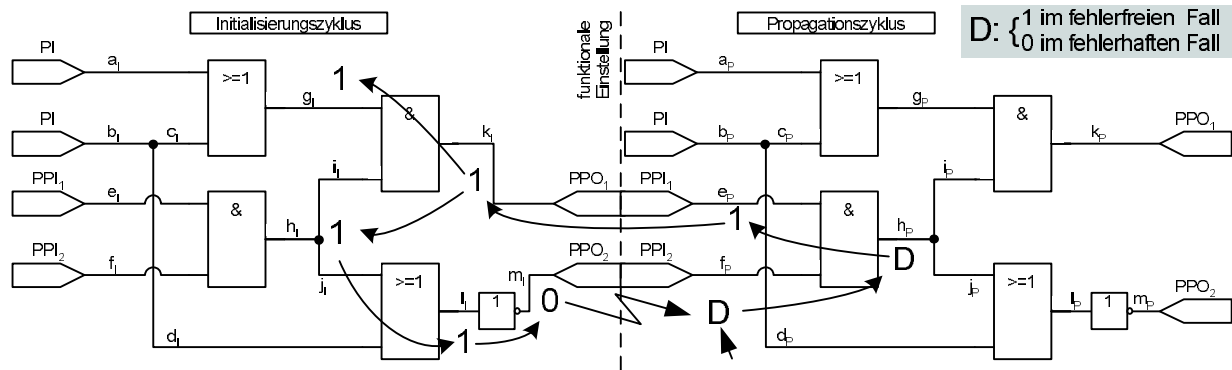


Abbildung 34: Die Betrachtung der Schaltung in beiden Taktzyklen unter Annahme funktionaler Einstellung

implementiert. Gelingt die Einstellung der Anforderungen, ist ein Pfadverzögerungsfehler test gefunden, andernfalls ist der Fehler redundant.

Beispiel 1: Für den Pfadverzögerungsfehler $\{f_r, j_r\}$ ⁴¹ der in Abbildung 33 gegebenen Schaltung ist ein Test gesucht.

Pfadsensitivierung: Unter Annahme des Applikationsmodus der funktionalen Einstellung ergibt sich zunächst $k_I = e_P, m_I = f_F$ (Abbildung 34). Die Pfadsensitivierung setzt zunächst die erste Leitung des Pfades im Propagationszyklus (f_P) auf D und führt lokale Implikationen durch. Die Vorwärtsimplikation innerhalb der verzweigungsfreien Region setzt $h_P = D$. Durch die fehlerleitende Belegung des UND-Gatters ergeben sich Implikationen, die zu einem Widerspruch an f_P führen (Tabelle 2). Der Fehler $\{f_r, j_r\}$ ist als redundant erkannt.

Tabelle 2: Implikationen aus Beispiel 1

Wertsetzung	Implikation
$f_P = D$	$h_P = D; e_P = k_i = 1, g_I = 1, h_I = 1, j_I = 1, l_I = 1, m_I = f_P = 1^*$

*)Wert bereits gesetzt und widersprüchlich

Beispiel 2: Für den Pfadverzögerungsfehler $\{f_f, j_f\}$ kann ein Test gefunden werden.

⁴¹Im folgenden werden die Indizes r für eine steigende (rising) und f für eine fallende Flanke verwendet

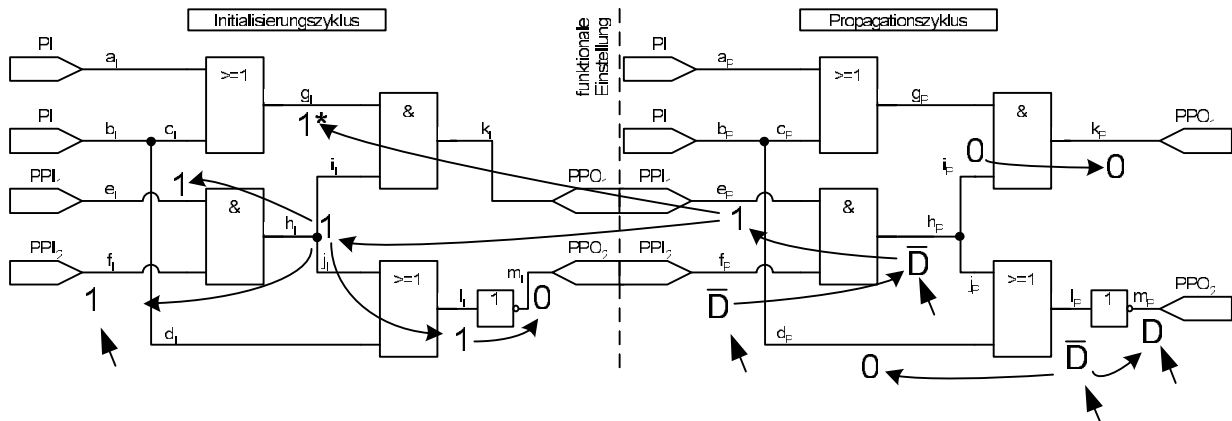


Abbildung 35: Für den Pfadverzögerungsfehler $\{f_f, j_f\}$ kann ein Testmuster gefunden werden

Tabelle 3: Pfadsensitivierung für Beispiel 2

Wertsetzung	Implikationen
$f_P = D$	$h_P = D; e_P = k_i = 1, g_I = 1, h_I = 1, j_I = 1, l_I = 1, m_I = 1,$ $f_P = 1^{**}, e_I = 1, f_I = 1$
$f_I = 1^{**}$	-
$h_P = D^{**}$	-
$j_P = D$	$l_P = D, m_P = D; d_P = 0, b_P = 0, c_P = 0, i_P = 0, k_P = 0$
$l_P = D^{**}$	-
$m_P = D^{**}$	-

***) Wert bereits gesetzt, kein Widerspruch

Pfadsensitivierung: Die Pfadsensitivierung setzt zunächst $f_P = \bar{D}$. Die Implikationen ergeben keinen Widerspruch, weil die Symbole (D, \bar{D}) ausschließlich zur Kennzeichnung der auf dem Pfad liegenden Leitungen verwendet werden. Ist der Leitung bereits ein Wert (D, \bar{D}) zugewiesen worden und entspricht der auf dieser Leitung neu zu setzende Wert dessen Wert für den fehlerfreien Fall, liegt kein Konflikt vor. Die Flanke wird mit $f_I = 1$ eingestellt. Danach wird der Pfad im Propagationszyklus sensitiviert (\rightarrow Tabelle 3)

Einstellen der verbleibenden Anforderungen War die Pfadsensitivierung erfolgreich, werden im zweiten Schritt des Algorithmus die noch einzustellenden Schaltungsknoten belegt. Im Beispiel (Abbildung 35) verbleibt $g_I = 1$ noch einzustellen.

Den Pfadverzögerungsfehler $\{f_f, j_f\}$ testet unter Annahme der funktionalen Einstellung das

Tabelle 4: Einstellung der Anforderungen aus Beispiel 2

Ziel	Wahlmöglichkeiten	Wahl	Implikationen
$g_I = 1$	$a_I = 1; c_I = 1$	$c_I = 1$	$g_I = 1^{**}; b_I = 1, d_I = 1, l_I = 1^{**}$

***) Wert bereits gesetzt, kein Widerspruch

Testmuster $\langle \{X,1,1,1\}, \{X,0,1,0\} \rangle$.

6 Pfadverzögerungsfehlerbasierter Test

Die mit dem vorgestellten Ansatz erhaltenen Testmuster basieren auf den in der Literatur gegebenen notwendigen Bedingungen zur Berechnung nicht robuster Testmuster. Für eine nähere Bestimmung der Testqualität wurden Berechnungen in verschiedenen ISCAS85- und ISCAS89-Benchmarkschaltungen vorgenommen, deren GröÙte maximal fast 1,5 Millionen Pfadverzögerungsfehler enthalten. Die Zahl der modellierten Fehler entspricht dem Doppelten der in der Schaltung enthaltenen Pfade. Die in den Tabellen angegebenen prozentualen Anteile der Tests beziehen sich auf die Gesamtfehlermenge. Im Vergleich zu aktuellen Designs haben die ISCAS-Benchmarkschaltungen nur eine geringe GröÙe. Sie eignen sich jedoch für eine vollständige Berechnung aller modellierten Fehler, sodass die Ergebnisse nicht von Pfadauswahlen beeinflusst und als aussagekräftige Basis für die Analyse verwendet werden können. Für die Klassifizierung der Testmuster wurde eine Fehlersimulation (→4.3) der erhaltenen Testwertsätze durchgeführt, bei der ein modifizierter Gatterverzögerungsfehlersimulator [102] zum Einsatz kam. Die Ergebnisse der Testmustererzeugungsläufe sind tabellarisch im Anhang aufgeführt (und im Text referenziert), um die Lesbarkeit nicht zu beeinträchtigen.

Tabelle 5: Definition der Testmodi

Testmodus	Initialisierung der Pfadsegmente	Forderung eines <i>ncv</i> an jedem Seiteneingang in beiden Taktzyklen
0	Nein	Nein
1	Ja	Nein
2	-	Ja

Aus den Ergebnissen der Experimente werden zwei Verbesserungsvorschläge abgeleitet und deren Effektivität nachgewiesen. Eine Definition dreier Testmodi (Tabelle 5) ermöglicht den Vergleich der Ergebnisse. Das Kapitel schließt mit der Betrachtung der Ergebnisse aus Berechnungen für kompakte Testmuster.

6.1 Testqualität

Die Ergebnisse der Testmusterberechnung, die auf den in [27] angegebenen notwendigen Bedingungen für die Realisierung eines nicht robusten Tests basiert, zeigt Abbildung 36. Die Schaltungen sind von links nach rechts mit steigender Zahl modellierter Fehler sortiert. Alle Ergebnisse

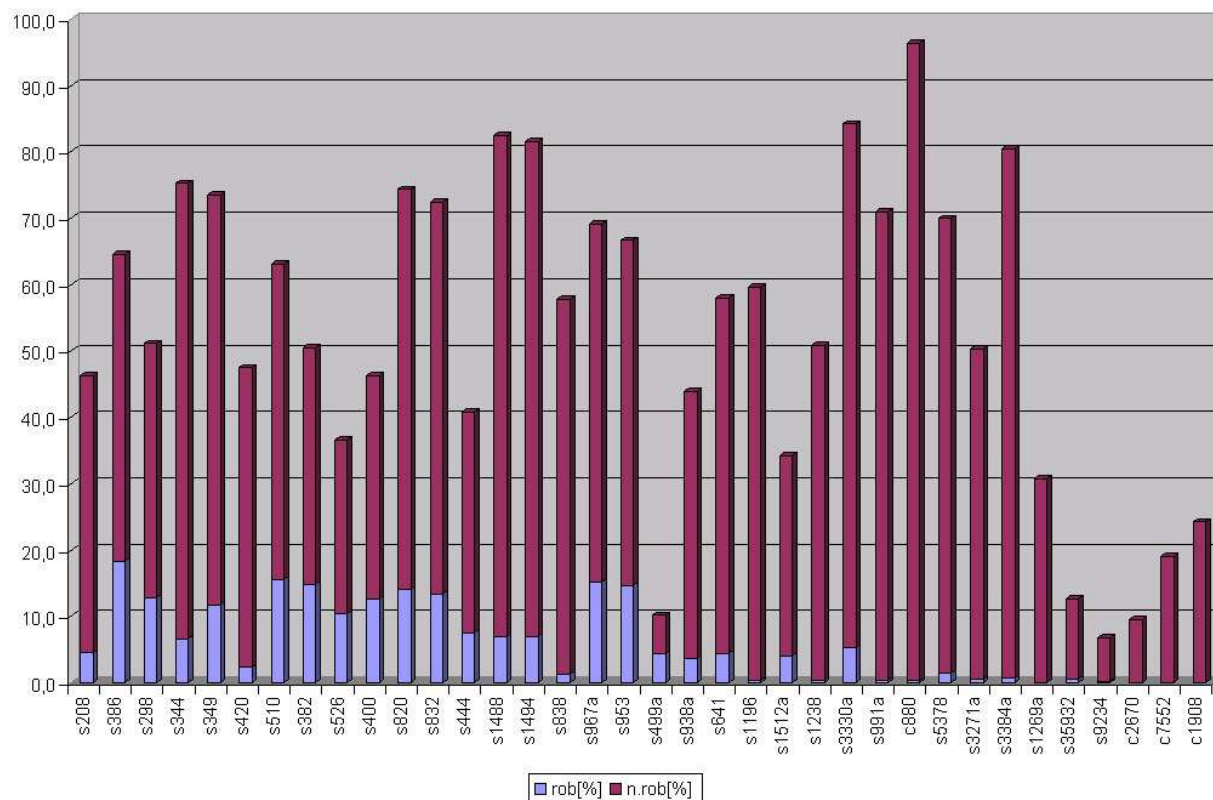


Abbildung 36: Maximal erreichbare Fehlerabdeckung; robuster und nicht robuster Anteil

beziehen sich auf den Applikationsmodus der funktionalen Einstellung, der für den Produktionstest den geeignetsten Modus darstellt (→2.10). Da die für die Testmustergenerierung verwandten Randbedingungen den minimal erforderlichen Ansprüchen an einen Verzögerungsfehlertest entsprechen, stehen die (in der Abbildung auf die Gesamtfehlermenge der jeweiligen Schaltung normierten) prozentual angegebenen Fehlerabdeckungen für das erreichbare Maximum⁴². Es zeigt sich, dass für keine Schaltung eine vollständige Fehlerabdeckung durch den Test erreicht wird. Die Zahl der Fehler, für die ein Test gefunden werden kann, variiert zwischen 6,8% und 96,3% und hängt damit stark von der jeweiligen Schaltung ab.

Für Verzögerungsfehlertests existieren verschiedene Testqualitäten (→2.8). Die Verteilung der robusten und nicht robusten Testmuster innerhalb der berechneten Testwertsätze ist in Abbildung 36 ebenfalls angegeben. Es zeigt sich, dass die Zahl der robusten Testmuster (heller Anteil)

⁴²Wie nachfolgend angeführt, ergaben Fehlersimulationen bei einigen Schaltungen, dass die aus der Testmustergenerierung erhaltenen Testmuster für den Zielfehler keinen Test darstellten. In diesen Fällen kann die erreichbare maximale Fehlerabdeckung etwas höher ausfallen als in Abbildung 36 angegeben.

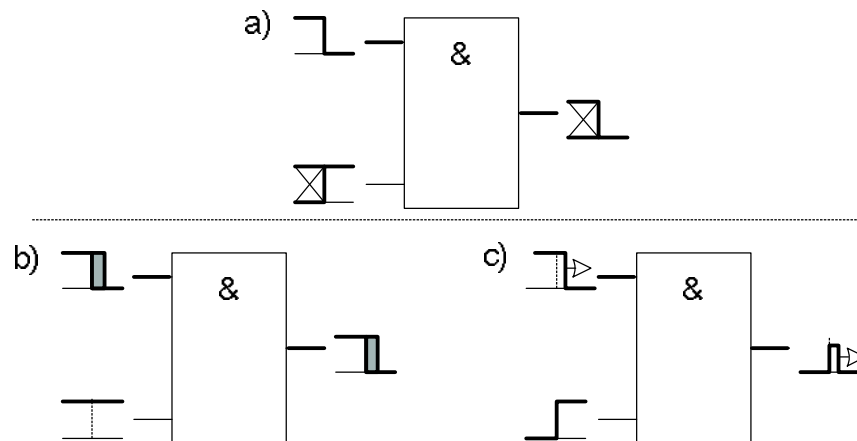


Abbildung 37: Nicht robuste Eingangsbelegung

tendenziell mit steigender Schaltungsgröße sinkt. Ferner überwiegt für alle Schaltungen die Zahl nicht robuster Testmuster im Testwertsatz.

Für einige Schaltungen unterscheiden sich die Ergebnisse der Fehlersimulation und Testmuster-generierung in der Beurteilung, ob das erhaltene Muster einen Test für den Zielfehler darstellt. Eine nähere Betrachtung der Eigenschaften der erhaltenen Testmuster zeigt, dass bei vielen Test-mustern die Detektion des Zielfehlers in Zweifel gezogen werden muss.

6.1.1 Nicht robuste Tests

Für die näheren Untersuchungen wird nachfolgend ein UND-Gatter betrachtet, sie gelten ana-log jedoch für jedes andere Standardlogikgatter mit mehreren Eingängen. Die in Kapitel 5.2.3 angegebenen notwendigen Bedingungen zur Realisierung eines nicht robusten Tests (nach dem Pfadverzögerungsfehlermodell) bedingen zum einen eine Flanke am Pfadeingang, zum anderen für den Propagationszyklus die Belegung der Seiteneingänge entlang des Pfades mit den nicht kontrollierenden Werten der Gatter. Beide Bedingungen erfüllt das Eingangsmuster aus Abbil-dung 37 (a), das an dem (unteren) Seiteneingang für den Initialisierungszyklus zwei mögliche Werte zulässt.

Für den ersten Fall seien der Initialisierungswert und der Endwert am Seiteneingang identisch (Abbildung 37 (b)). Es liegt in beiden Taktzyklen folglich ein nicht kontrollierender Wert am Seiteneingang, ein verzögertes Eintreffen der Flanke auf dem oberen Pfadeingang führt auch am Gatterausgang zu einem verzögerten Umschalten.

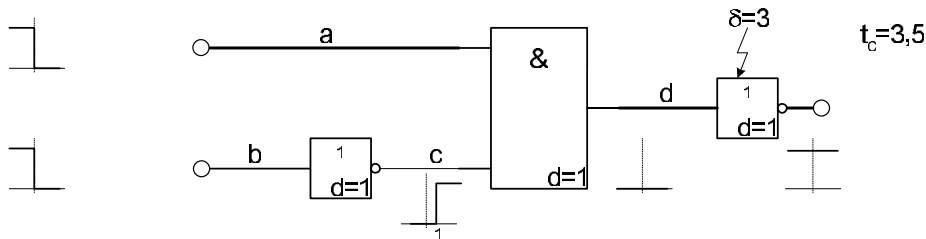


Abbildung 38: Nominallaufzeiten invalidieren den nicht robusten Test

Die Situation ändert sich mit dem ebenfalls möglichen Testmuster aus Abbildung 37 (c). Die nun am Seiteneingang auftretende Flanke kann, wenn sie später als der Signalwechsel auf dem Pfad eingang eintrifft, den Test invalidieren. Der Ausgangswert des Gatters wird im Initialisierungszyklus durch den Seiteneingang bestimmt. Im Propagationszyklus soll der Signalwechsel am Pfad eingang für eine Flanke am Gatterausgang sorgen. Im Idealfall führt die Konstellation zu einem hazardbehafteten statischen Ausgangssignal (Abbildung 37 (c)). Die Breite Δt_{Hazard} des Hazards wird durch die zeitliche Differenz des Eintreffens der Flanke am Seiteneingang und der auf dem Pfad bestimmt. Der Hazard verschwindet, sobald $t_{\text{Pfad}} \leq t_{\text{Seiteneingang}}$ wird. Dieser Sachverhalt kann durch eine weitere, *fehlerbedingte* Verzögerung des Signals am Seiteneingang gegeben sein, ebenfalls möglich ist jedoch auch eine größere *nominale* Laufzeit des Signals am Seiteneingang, wie sie beispielsweise durch eine Schaltungsstruktur nach Abbildung 38 gegeben sein kann. Das Beispiel zeigt, dass die vielfach in der Literatur vertretene Annahme, ein nicht robuster Test würde ausschließlich durch weitere Fehler innerhalb der Schaltung invalidiert werden [71], nicht zutrifft. Es sei an dieser Stelle darauf hingewiesen, dass der im Beispiel adressierte Pfadverzögerungsfehler zu keinem *nicht funktionalen* Pfad gehört, obwohl er, bedingt durch das Eingangsmuster, auch im fehlerfreien Fall keine Signaländerung propagiert. Tatsächlich ist die Signaländerung nur durch die Flanke am Seiteneingang maskiert; würde der Eingang *b* einen statischen High-Pegel führen, bekäme der zu testende Pfad signalleitende Eigenschaften und die hinter dem UND-Gatter angenommene Verzögerung bekäme Einfluss auf die Schaltungsfunktion. Ferner werfen die Überlegungen die Frage auf, ob die Anwendung der in [25] gegebenen notwendigen Bedingungen für die Generierung nicht robuster Tests ohne eine Beachtung der Laufzeiten innerhalb der Schaltung ausreichend bzw. sinnvoll ist, da die geschilderten Probleme nur durch eine Beachtung der Signallaufzeiten an den Seiteneingängen vermieden werden können.

Die erhaltenen Ergebnisse sind jedoch auch in einer anderen Hinsicht interessant. Wie bereits ausgeführt, ergibt sich Breite des sich einstellenden Hazards als Differenz der Laufzeiten an den

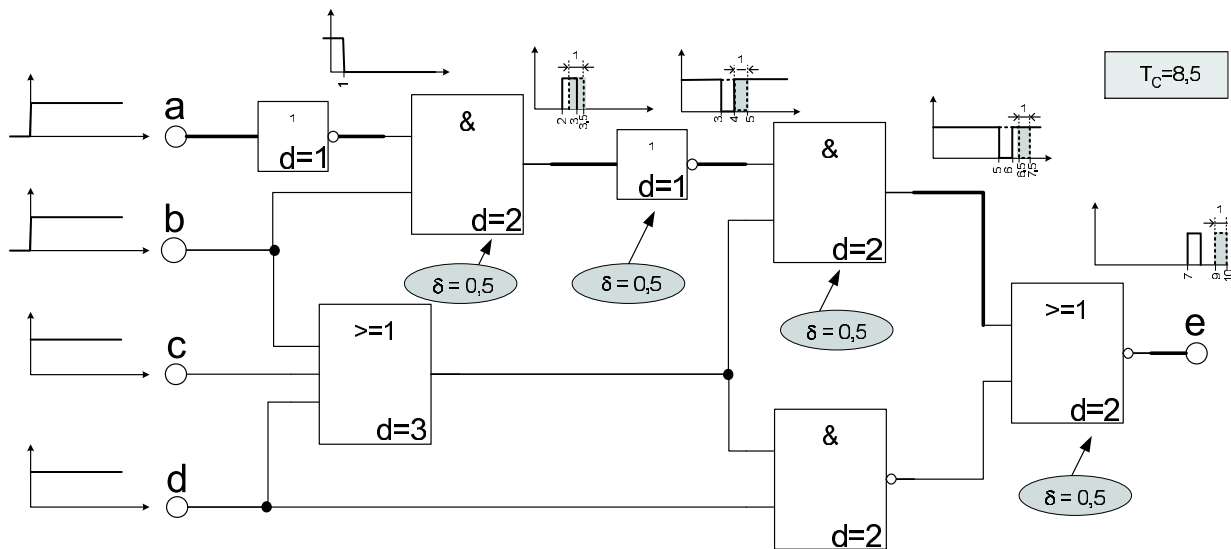


Abbildung 39: Verteilte Fehler verschieben den Hazard, verbreitern ihn aber nicht

Signaleingängen: $\Delta t_{\text{Hazard}} = t_{\text{Pfad}} - t_{\text{Seiteneingang}}$. Es folgt, dass eine Verzögerung der Flanke auf dem Pfadeingang des Gatters zu einer Verbreiterung des Hazards führt. Weil das Pfadverzögerungsfehlermodell von einer verteilten Fehlerannahme ausgeht ($\rightarrow 2.5.2$) stellt sich die Frage, welchen Einfluss weitere, auf dem Pfad liegende Verzögerungsfehler auf das zu messende Signal haben. Wird von einer Fehlergröße ausgegangen, die steigende und fallende Flanke gleichermaßen beeinflusst, ergibt sich, dass weitere Verzögerungsfehler den Hazard zeitlich verschieben, in seiner Breite aber unverändert lassen. Dieses Ergebnis ist problematisch, da nun auch größere Verzögerungen auf dem Pfad die Wahrscheinlichkeit einer Fehlerdetektion nicht mehr erhöhen, wenn sie erst hinter dem den Hazard erzeugenden Gatter liegen. Ein Beispiel zeigt Abbildung 39. Die vier zusätzlich eingebrachten Verzögerungen von jeweils 0,5 Zeiteinheiten verschieben den Hazard um 2 Zeiteinheiten, seine ursprüngliche Breite von einer Zeiteinheit bleibt trotz der zusätzlichen Verzögerungen unverändert. Zum Taktzeitpunkt ($T_C = 8,5$) liegt das erwartete low-Signal am Ausgang an, der Hazard erreicht ihn, fehlerbedingt, erst nach 9 Zeiteinheiten, der Pfadverzögerungsfehler wird nicht erkannt. Würde Eingang b stattdessen einen konstanten High-Pegel, würde die dann über Eingang a auf den Pfad geschaltete Flanke um die zusätzlichen 2 Zeiteinheiten verzögert und am Ausgang ihr Anfangswert gemessen werden.

Wird darüber hinaus davon ausgegangen, dass die in Abbildung 37 dargestellte Situation sich am Anfang des Pfades befindet und die bis dahin auftretende Fehlergröße nicht ausreicht, die Flanke auf dem Pfadeingang später als die am Seiteneingang eintreffen zu lassen (Abbildung 38), wird

der Ausgang des Gatters nicht umschalten; die zum Test eines Verzögerungsfehlers notwendige Flanke erlischt. Für den Test bedeutet dies, dass ein Pfad mit einem Verzögerungsfehler beliebiger Größe nicht mehr als solcher erkannt werden kann, sobald die zum Test notwendige Flanke auf dem Pfad erlischt.

Zusammengefasst sind die in der Literatur angegebenen Bedingungen für die Realisierung eines Tests, der den Zielfehler in Abwesenheit weiterer, fehlerbedingter Verzögerungen detektiert, nicht ausreichend. Es stellt sich die Frage, welche zusätzlichen Bedingungen zu erfüllen sind, um eine Fehlerdetektion zu gewährleisten. Prinzipiell bieten sich hierzu 2 Lösungsansätze an.

6.1.2 Erhöhung der Qualität nicht robuster Tests

Das Auftreten eines Hazards am Ausgang eines Gatters ist von den zeitlichen Relationen zwischen den Seiteneingangssignalen und dem Signal am Pfadeingang abhängig. Eine Testmuster-generierung, die auf die in Abschnitt 5.3 gegebenen Vorgaben aufsetzt, muss folglich zusätzlich die Ankunftszeiten der Flanken an den Eingängen der auf dem Pfad liegenden Gatter berücksichtigen. Da dies zum einen in der Regel nicht möglich ist (\rightarrow 2.6), zum anderen der Aufwand für die Testmuster-generierung dadurch stark erhöht wird, erscheint diese Lösung nicht praktikabel.

Die Alternative liegt in der Definition zusätzlicher Bedingungen, welche die Verwendung eines Hazards als Flanke für den Test ausschließen. Algorithmisch kann dies verhältnismäßig einfach erreicht werden, wenn eine Initialisierung aller auf dem Pfad befindlichen Leitungen im Initialisierungszyklus durchgeführt wird. Als Folge werden im Falle einer $ncv \rightarrow cv$ Flanke am Pfadeingang die Seiteneingänge in beiden Taktzyklen mit dem nicht kontrollierenden Wert belegt (im Propagationszyklus ist der Seiteneingang fehlerleitend zu belegen, im Initialisierungszyklus bedingt die Initialisierung der Ausgangsleitung, dass alle Gattereingänge den ncv führen müssen). Im umgekehrten Fall einer $cv \rightarrow ncv$ Flanke am auf dem Pfad liegenden Eingang des Gatters bestimmt der Ausgangswert an diesem Eingang den Ausgangswert des Signals am Gatterausgang. Folglich sind sowohl nicht kontrollierende wie kontrollierende Werte als Ausgangswert der Signale an den Seiteneingängen möglich, womit sich $cv \rightarrow ncv$ Flanke an einem Seiteneingang einstellen kann. Diese wird den Fehler jedoch nicht maskieren, da eine Verzögerung dieser Flanke auch die Flanke am Gatterausgang verzögern wird, da dieser erst umschaltet, wenn alle Eingänge den nicht kontrollierenden Wert angenommen haben.

Die Ergebnisse der Testmuster-generierung mit Initialisierung der auf dem Pfad liegenden Leitungen in beiden Taktzyklen sind in Tabelle 13 zusammengefasst. Im Vergleich zu den Ergebnissen

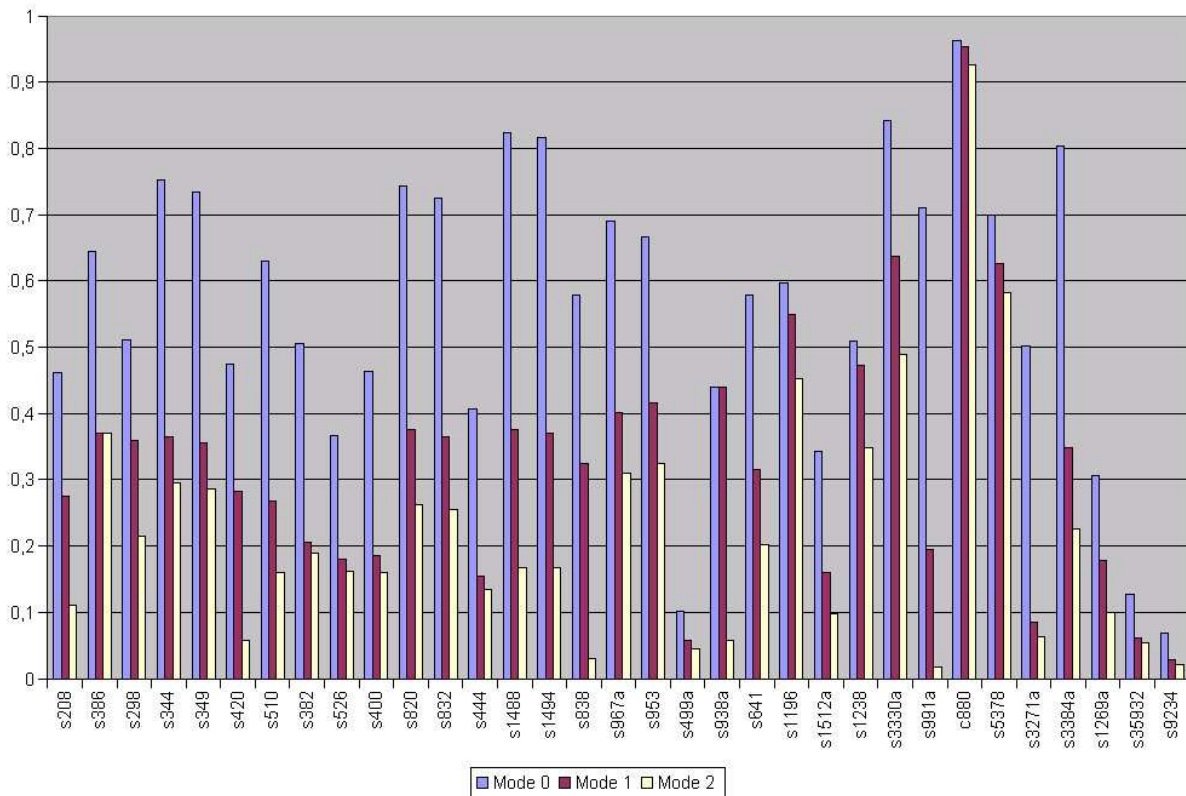


Abbildung 40: Die Zahl testbarer Fehler sinkt mit steigenden Anforderungen.

aus Tabelle 12 (siehe auch Abbildung 40, Mode 1 im Vergleich zu Mode 0) zeigt sich, dass die Zahl der Fehler, für die ein Test gefunden werden kann, sinkt. Aus dieser Beobachtung lässt sich ableiten, dass alle Fehler, für die ausschließlich unter den Originalanforderungen eine Testmusterberechnung gelingt, zwangsläufig einen Hazard als für den Test notwendige Flanke bedingen. Für alle anderen Fehler sollten Testmuster, die unter Einbeziehung der Pfadinitialisierung gewonnen wurden, bevorzugt werden, da sie schaltungsinternen Laufzeiten gegenüber unempfindlicher sind.

Mit Blick auf die Robustheit der erhaltenen Tests zeigt sich, dass nicht nur der Anteil der robusten Tests im Testwertsatz sondern auch deren absolute Zahl zumeist signifikant steigt (Abbildung 41, Mode 1 im Vergleich zu Mode 0). Begründet wird dieses Ergebnis durch die in der Schaltung zusätzlich definierten Signale. Allen Untersuchungen liegen die aus der Testmuster-generierung erhaltenen Testmuster zugrunde, welche alle undefinierten Eingangssignale zu X setzt. Die Einschätzung der Testqualität ist damit für jene Tests pessimistisch, die im Initialisierungszyklus an mindestens einem Seiteneingang ein X erscheinen lassen und einen nicht robusten Test er-

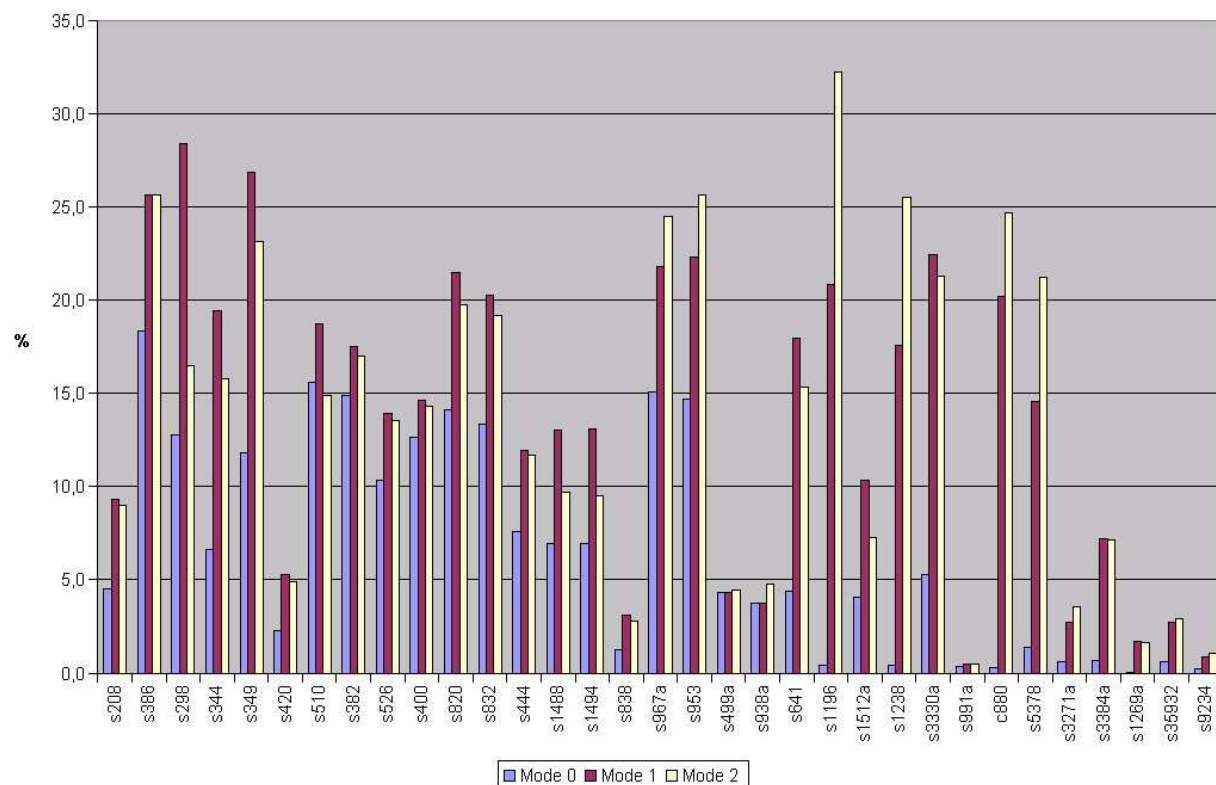


Abbildung 41: Anteil der robusten Muster im Testwertsatz bei verschiedenen Testmodi

möglichen. Die Initialisierung des Pfades in beiden Taktzyklen stellt alle Seiteneingänge auf einen definierten Wert ein und eliminiert dadurch bei einigen Tests die vorher zur nicht robusten Einschätzung führende alternative Eingangsbelegung. Folglich geht ein Teil der vorher nicht robusten Tests nun in einen robusten Test über.

Abbildung 42 zeigt die prozentuale Verteilung der nicht robusten und robusten Testmuster im Testwertsatz bei Anwendung der Initialisierung. Es wird ersichtlich, dass bei vielen Schaltungen die Mehrzahl der erhaltenen Testmuster innerhalb eines Testwertsatzes weiterhin nicht robust ist und diese Eigenschaft tendenziell mit der Größe der Schaltung zunimmt. Der Grund hierfür liegt in der Verwendung des sequentiellen Ansatzes für die Testmuster generierung, der sich einer Logik für statische Fehlermodelle bedient und je Taktzyklus nur einen diskreten Wert je Taktzyklus (der zum Taktzeitpunkt angenommen wird, →5.1.2) beschreiben kann. Als Folge ist es nicht möglich, etwaige Schaltvorgänge während der Taktzyklen zu beschreiben, die den Grund für eine mögliche Invalidierung bilden. Ein Beispiel zeigt Abbildung 43. Durch die Initialisierung der Pfadsegmente stellt sich am unteren Seiteneingang in beiden Taktzyklen der nicht kontrol-

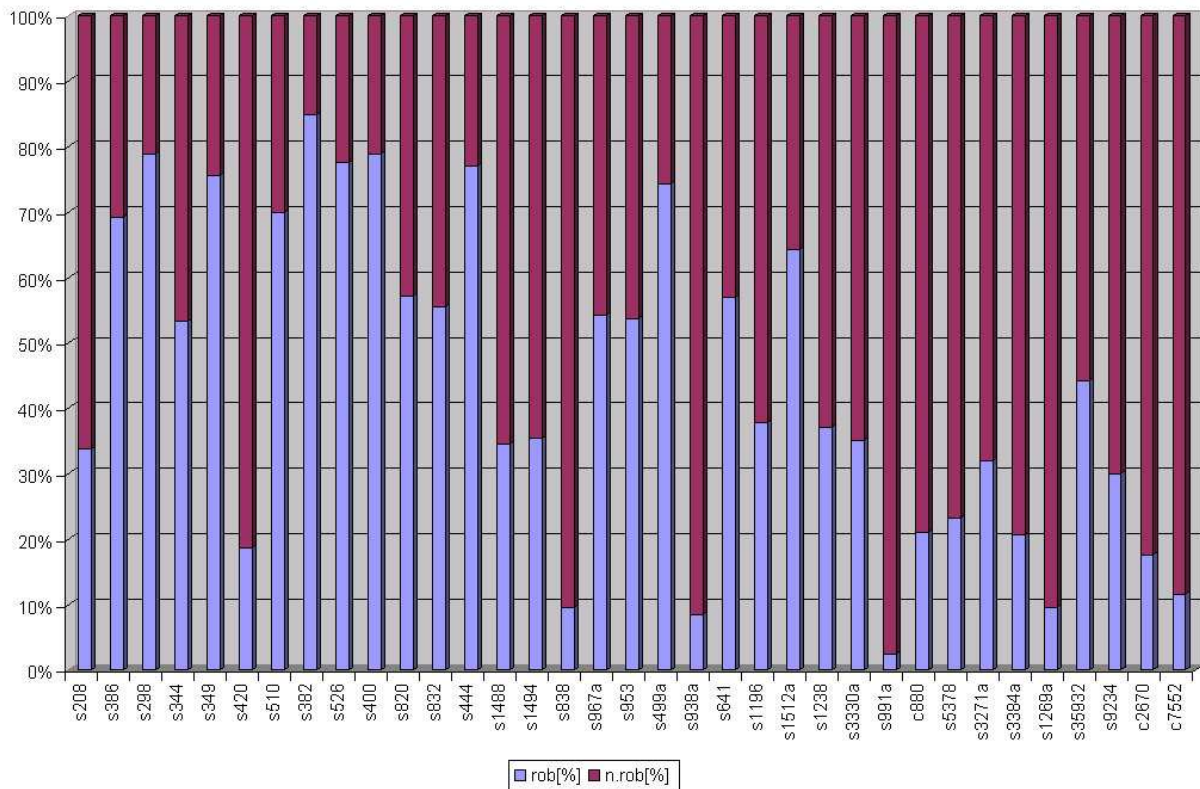


Abbildung 42: Verteilung robuster und nicht robuster Testmuster im Testwertsatz im Modus 1

lierende Wert ein. Wie diese beiden Werte durch die vor den Seiteneingängen liegende Logik eingestellt werden wird jedoch nicht vorgegeben, sodass eine Situation wie in Abbildung 43 dargestellt möglich ist. Ob es sich bei dem am Seiteneingangssignal um ein statisches Signal oder einen statischen Hazard handelt, hängt damit von den Laufzeiten der Signale am Eingang des davor liegenden Gatters ab. Der tendenziell größere Anteil nicht robuster Testmuster bei steigender Schaltungsgröße lässt sich leicht erklären, da mit der Länge des Pfades die Wahrscheinlichkeit steigt, dass mindestens ein Seiteneingang eine nicht robuste Seiteneingangsbelegung führt.

Aus diesen Überlegungen kann weiterhin abgeleitet werden, dass der sequentielle Ansatz (\rightarrow 5.1.2) konzeptbedingt nicht in der Lage ist, eine robuste Testmuster generierung sicherzustellen. Da robuste Tests jedoch eine Untermenge der nicht robusten bilden (\rightarrow 2.8) ist es möglich, dass ein berechnetes Testmuster für den Zielfehler “zufällig” auch einen robusten Test bildet.

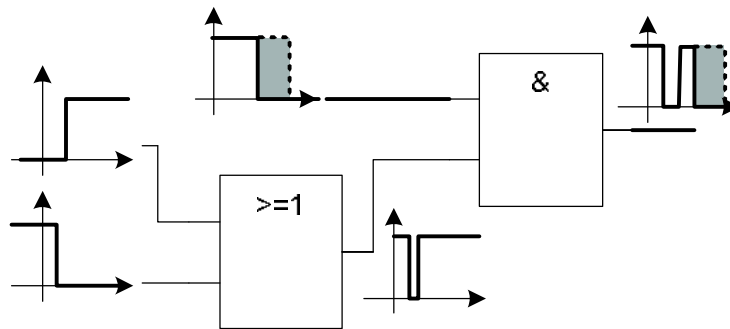


Abbildung 43: Beispiel für einen im sequentiellen Ansatz nicht beschreibbaren statischen Hazard

6.1.3 Nicht robuste Tests hoher Qualität

Die Untersuchungen der Testmuster aus Abschnitt 6.1.1 ergaben Hazardeffekte auf dem Pfad als Hauptproblem der invalidierten Testmuster. Weil trotz der Anwendung der Pfadinitialisierung für viele der erhaltenen Tests weiterhin die Gefahr einer Invalidierung besteht, stellt sich auch bei ihnen die Frage nach möglichen Invalidierungsmechanismen. Prinzipiell sind Signaländerungen an den Seiteneingängen des Pfades Grund für die Invalidierung eines Testmusters. Dabei umfassen Signaländerungen sowohl Flanken als auch Hazardeffekte. Um einen Eindruck darüber zu bekommen, welche Effekte welchen Anteil an der Degradierung der Testqualität haben, kann eine Simulation der nicht robusten Testmuster durchgeführt werden. Tabelle 6 enthält die Ergebnisse für einige Beispielschaltungen.

Tabelle 6: Invalidierungsmechanismen

	nicht robust	Hazard	Flanke	Hazard & Flanke
s1196	3669	10.36 %	80,85 %	8,79 %
s1238	3592	11.62 %	80,11%	8.26 %
s5378	18563	6.91 %	69,46 %	23.63 %
s9234	32238	21.73 %	73,96 %	4.31 %

Es zeigt sich, dass für einige Schaltungen ein großer Anteil der nicht robusten Testmuster *ausschließlich* durch Signalfanken an den Seiteneingängen invalidiert werden kann (Tabelle 6, Spalte 3), während statische Hazards hier nur einen verhältnismäßig kleinen Einfluss auf den Testwertsatz ausüben (Tabelle 6, Spalten 4 & 5). Eine Erhöhung des Anteils robuster Testmuster im Testwertsatz kann demnach erwartet werden, wenn zusätzlich zur Flanke am Pfadeingang in

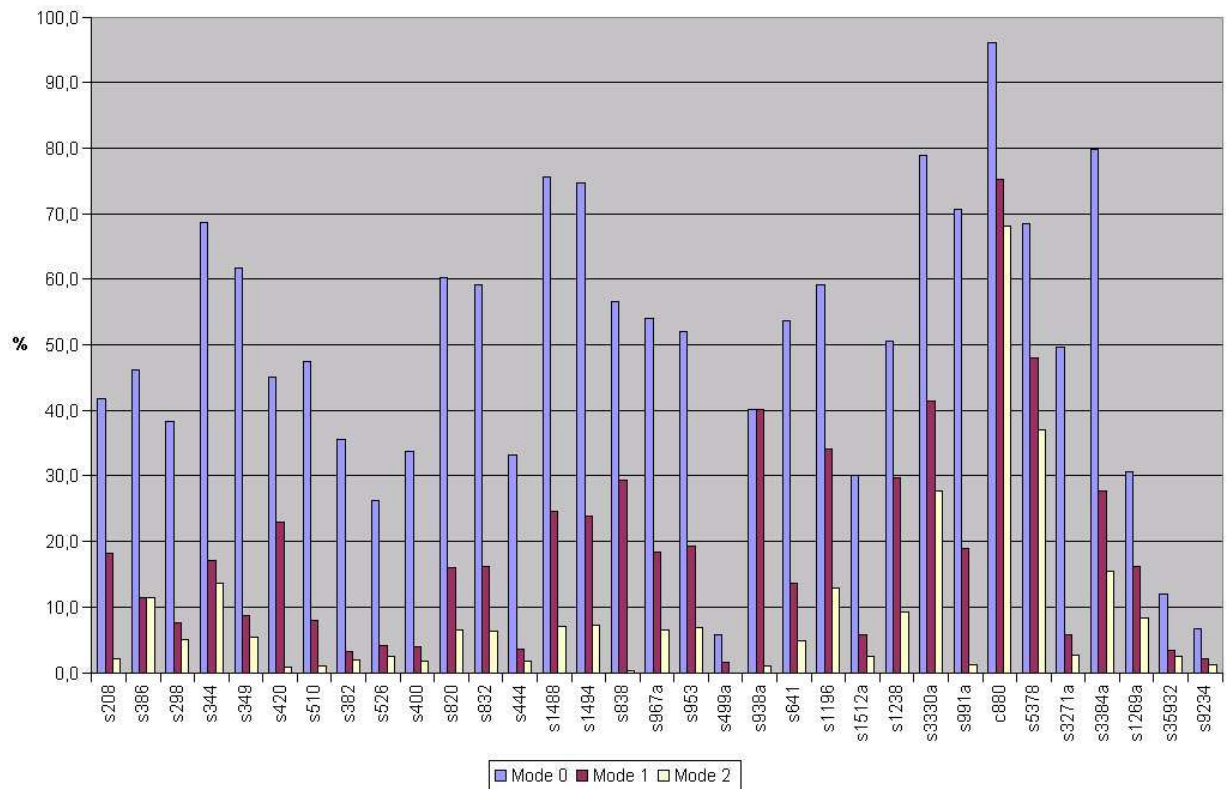


Abbildung 44: Anteil der nicht robusten Muster in den Testwertsätzen

beiden Taktzyklen eine Initialisierung aller Seiteneingänge des Pfades mit dem *ncv* des jeweiligen Gatters erfolgt. Dies entspricht im Ansatz der Forderung nach dem Test eines transparenten Pfades. Die Ergebnisse der Testmuster generierung unter diesen Vorgaben sind in Tabelle 14 aufgeführt.

Abbildung 44 zeigt den Anteil der nicht robusten Testmuster im Testwertsatz über die drei Testmodi. Der Vergleich zeigt, dass ihre absolute Zahl mit der Strenge der an den Test gestellten Anforderungen sinkt. Ist der Rückgang an nicht robusten Testmustern im Vergleich zwischen Mode 0 und Mode 1 noch für alle Schaltungen signifikant, hängt sein Ausmaß im Vergleich der Modi 1 und 2 in starkem Maße von der Schaltung ab. Im Unterschied hierzu zeigt Abbildung 41 für viele Schaltungen unter Mode 2 einen Rückgang der absoluten Zahl robuster Testmuster. Dies begründet sich durch die harten Anforderungen, die die fehlerleitende Belegung der Seiteneingänge an den Test stellt. Innerhalb des Testwertsatzes ist jedoch erneut ein deutlicher Gewinn an robusten Testmustern zu verzeichnen.

Für die unter diesen strengsten Vorgaben gewonnenen nicht robusten Testmuster gilt, dass sie

ausschließlich durch das Auftreten eines Hazards an mindestens einem Seiteneingang invalidiert werden können, da dieser Hazard auch im Ausgangssignal des betroffenen Gatters zu einem Hazard führen kann. Der für die Bestimmung der Testqualität verwendete Fehlersimulator verwendet für die Ermittlung möglicher Hazardeffekte das Einheitsverzögerungsmodell, dass jedem Gatter unabhängig von seiner Funktion und der Anzahl seiner Eingänge eine einheitliche Verzögerung zuweist. Anhand dieser Verzögerungen werden jedem Gatter in einem Präprozess Werte für das frühest mögliche und spätest mögliche Eintreffen einer Signalfanke zugewiesen. Als Folge sind in der Simulation verwandten Werte nicht auf die Laufzeiten der Signale aufgrund des aktuellen Testmusters bezogen, sondern verwenden die durch die beiden Eckwerte begrenzte Zeitspanne für die Bestimmung eines möglichen Hazards. Die Bewertung der Testmuster muss damit als pessimistisch angesehen werden, da nicht alle möglichen Hazards zwangsläufig an den Eingängen erscheinen müssen. Somit haben auch die nicht robusten Testmuster ein relativ geringes Potential invalidiert zu werden.

6.1.4 Zusammenfassung

Die Experimente haben gezeigt, dass eine Testmuster generierung für Verzögerungsfehlermodelle mit dem sequentiellen Ansatz zu verwendbaren Ergebnissen führt. Die aus der Literatur entnommenen Bedingungen für die Generierung nicht robuster Testmuster erweisen sich bei genauerer Betrachtung als nicht ausreichend, um einen Test zu realisieren, weshalb für die industrielle Anwendung eine Testmuster generierung im Testmodus 1 vorgeschlagen wird.

Die Verwendung von Testmustern aus dem Testmodus 0 kann nur dann empfohlen werden, wenn

- für den Zielfehler kein Testmuster unter den strengeren Anforderungen gefunden werden kann und
- die Länge des zu testenden Pfades möglichst nahe an der Länge des längsten Pfades liegt.

Die Beschränkung der zu testenden Pfade auf die längsten Pfade innerhalb der Schaltung verringert die Wahrscheinlichkeit, dass Flanken an den Seiteneingängen später als die auf dem Pfad laufende Flanke eintreffen und der notwendige Hazard erlischt. Das Problem der fehlerbedingten Hazardverschiebung (→Abbildung 39) bleibt jedoch bestehen.

Testmuster, die für jeden Seiteneingang in beiden Zeitrahmen den nicht kontrollierenden Wert fordern, bieten qualitativ die beste Grundlage für einen Test. Da dies jedoch, gerade in größeren

Schaltungen, nur für wenige Pfade realisierbar ist, wird für die Anwendung im Produktionstest eine zu große Fehlermenge ungetestet verbleiben.

Die vorangegangenen Experimente haben für steigende Testmodi eine Abnahme der nicht robusten Tests ergeben, während die absolute Zahl robuster Testmuster sich schaltungsabhängig veränderte (Abbildung 41). Tabelle 15 zeigt die Ergebnisse einer Analyse, wie sich die Qualität der Testmuster beim Übergang von Testmodus 1 nach Testmodus 2 ändert. Spalte 2 enthält die Zahl der Fehler, für die unter beiden Testmodi robuste Muster berechnet wurden, während Spalte 3 beispielsweise die Anzahl der Fehler beziffert, für die unter Testmodus 1 ein robustes Testmuster erhalten wurde, während sie mit dem Muster aus Testmodus 2 nicht robust getestet sind. Für fast jede Schaltung existieren Fehler, für die unter Testmodus 1 ein robuster Test berechnet wird, während sie unter strengeren Vorgaben nicht testbar sind (Spalte 4). Weiterhin existieren Fehler, deren Testqualität sich von nicht robust nach robust ändert (Spalte 5). Folglich ergibt sich mit Blick auf die Gewinnung robuster Testmuster in keinem Testmodus ein optimales Ergebnis. Die maximal erreichbare Zahl robuster Testmuster ergibt sich aus der Summe der in beiden Modi robust getesteten Fehler sowie der in einem der beiden Modi robust getesteten (Spalten 2, 3 und 5).

Eine erstaunliche Tatsache folgt aus Spalte 3. Entgegen der Annahme, dass eine fehlerleitende Belegung der Gatter zu beiden Taktzyklen die Testqualität verbessern müsste, kommt es, je nach Schaltung, sogar zu einer merklichen Zahl an Fehlern, deren Testqualität sich bei der Anwendung strengerer Anforderungen verschlechtern. Der Grund für die Änderung der Testqualität hing in den untersuchten Fällen jedoch nur indirekt mit den strengeren Anforderungen zusammen. Wie in Kapitel 2.7 bereits angeführt, können für einen Zielfehler mehrere Testmuster existieren, die, bezogen auf den Zielfehler, unterschiedliche Eigenschaften aufweisen. Durch die in den höheren Modi zusätzlich geforderten Signale innerhalb der Schaltung kann ein Teil der vorher möglichen Testmuster als Lösung ausgeschlossen werden, da sie nicht mehr alle der gestellten Anforderungen erfüllen. Dies gilt auch für robuste Tests, welche $cv \rightarrow ncv$ Flanken an den Seiteneingängen zuließen, die den Test jedoch nicht invalidierten. Für den Fall, dass das neue Testmuster (durch die Forderung des ncv an allen Seiteneingängen in beiden Taktzyklen) zu einem statischen Hazard an einem Seiteneingang führt, der den Test invalidieren kann, geht der vormals robuste Test durch die härteren Anforderungen in einen nicht robusten über.

Zusammengefasst kann der Ansatz, die Testmuster generierung für Verzögerungsfehler über eine Transformation auf ein Testmuster generierungsproblem für Haftfehler in einer sequentiellen Schaltung abzubilden, die Gewinnung robuster Testmuster nicht sicherstellen. Für die Klassifizierung der Tests ist daher ein geeigneter Fehlersimulator notwendig, der neben den Informa-

tionen über zusätzliche Fehlerabdeckungen auch Aussagen über die Qualität eines Tests treffen kann. Es ist gezeigt worden, dass über die Definition zusätzlicher Bedingungen die Wahrscheinlichkeit, einen robusten Test für den Zielfehler zu erhalten, erheblich gesteigert werden kann. Generell steigt die Qualität der erhaltenen Tests mit dem Testmodus.

6.2 Testmusterkompaktierung

Das Ziel des Produktionstests ist, eine hohe Fehlerabdeckung mit einer möglichst geringen Zahl an Testvektoren zu erreichen (\rightarrow 2.2), weshalb Bedarf an möglichst kompakten Testmustern besteht. Neben dem klassischen Ansatz, eine Kompaktierung der Testmuster bereits bei der Testmustergenerierung durchzuführen, haben sich in letzter Zeit auch Vorschläge gefunden, die Testmusterkompaktierung hardwareunterstützt zu realisieren (z.B. [113]). Für die Testmusterkompaktierung auf Seiten der Testmusterberechnung existieren mit der dynamischen und der statischen Kompaktierung zwei grundsätzliche Ansätze.

Die statische Testmusterkompaktierung geht von einem gegebenen Testwertsatz aus und hat zum Ziel, diesen zu minimieren. Um dies zu erreichen, wird ein Fehlersimulator eingesetzt, der die Fehlerabdeckung einer Testmustersequenz⁴³ bestimmt. Führt ein Testmuster in der Simulation zu keiner zusätzlichen Fehlerabdeckung, müssen alle von ihm detektierten Fehler durch die vorangegangene Sequenz bereits abgedeckt worden sein, das aktuelle Testmuster kann aus dem Testwertsatz entfernt werden. Da die Ergebnisse von der bereits zugeführten Sequenz abhängen, werden im Regelfall mehrere, unterschiedliche Sequenzen aus den Testmustern des Testwertsatzes gebildet.

Die dynamische Testmusterkompaktierung versucht bereits bei der Testmustergenerierung, eine möglichst hohe Fehlerabdeckung zu realisieren. Da nach einer erfolgreichen Testmustergenerierung für einen primären Fehler die Schaltungseingänge nur partiell belegt sind, kann unter Beachtung der bereits erhaltenen Eingangsbelegung versucht werden, für weitere, zusätzliche Fehler einen Test zu berechnen.

Eine aktive Kompaktierung kann für den in Abschnitt 5.3 durch leichte Modifikationen erreicht werden. Dazu wird der Algorithmus nach einer erfolgreichen Testmustergenerierung iterativ für alle verbliebenen Fehler ausgeführt (Abbildung 45). Für die zusätzlichen Zielfehler ist der Suchraum der Testmustergenerierung eingeschränkt, da durch die bereits vorgegebenen Signale innerhalb der Schaltung zahlreiche Entscheidungsmöglichkeiten entfallen. Das größte Problem bei der

⁴³Der Begriff "Testmustersequenz" beschreibt hier die Reihenfolge, in der die Testmuster des Testwertsatzes der Schaltung zugeführt werden.

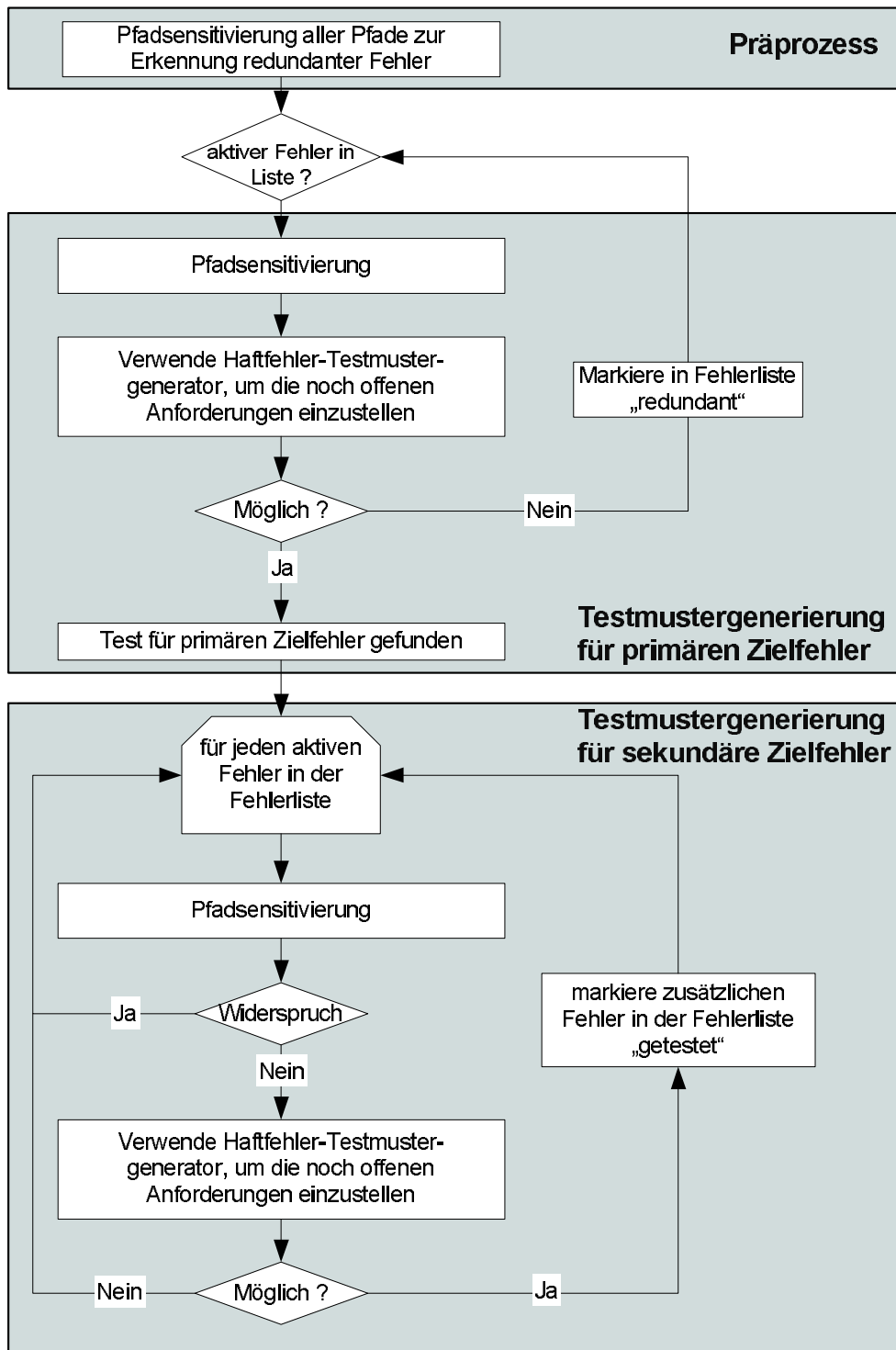


Abbildung 45: Ablaufdiagramm der Testmusterberechnung mit aktiver Kompaktierung

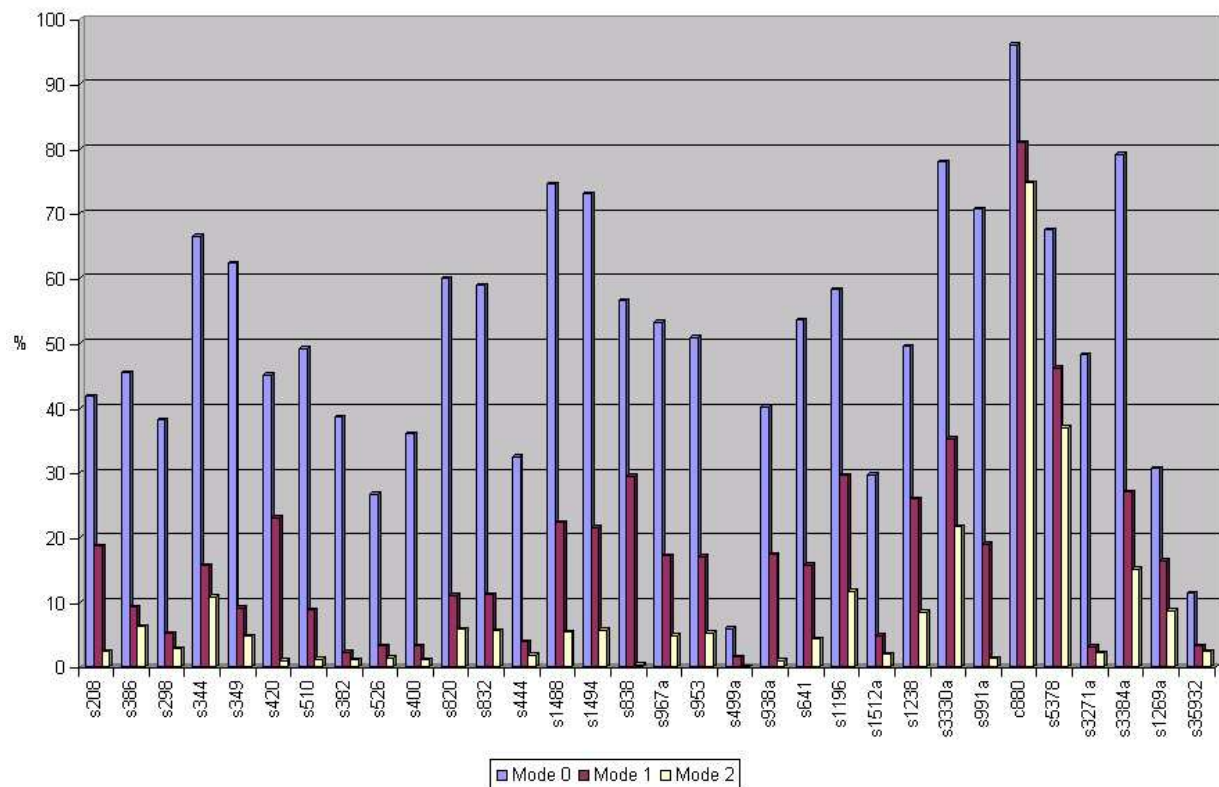


Abbildung 46: Anteil nicht robuster Tests bei aktiver Kompaktierung

aktiven Testmusterkompaktierung bilden die in der Fehlerliste enthaltenen redundanten Fehler, die aufgrund von Laufzeitbeschränkungen für die Testmustersuche nicht als redundant erkannt werden. Da diese Fehler in der Fehlerliste verbleiben und für jeden weiteren primären Zielfehler als Kandidaten für die zusätzlichen Zielfehler betrachtet werden müssen, wird ein Großteil der Rechenzeit für diese Fehler aufgewendet werden. Eine möglichst frühzeitige Erkennung redundanter Fehler ist somit in höchstem Maße wünschenswert. Die Erkennung redundanter Fehler ist kein triviales Problem, doch können einige einfache Zusammenhänge nützlich sein. Stehen Informationen zur Redundanz der in der Schaltung modellierten Haftfehler zur Verfügung, kann auf die Redundanz eines Verzögerungsfehlers geschlossen werden. Die Redundanz eines Haftfehlers Φ_k^H am Knoten k bedeutet, dass dieser nicht beide Signalpegel einnehmen oder ein Unterschied an keinem Ausgang beobachtet werden kann. Alle Verzögerungsfehler, deren Test einen kritischen Pfad durch diesen Knoten erfordern, sind demnach ebenfalls redundant.

Für das Pfadverzögerungsfehlermodell kann, da der kritische Pfad dem zu testenden Pfad entspricht, weiterhin versucht werden, vor einer Testmuster-generierung die zu testenden Pfade zu

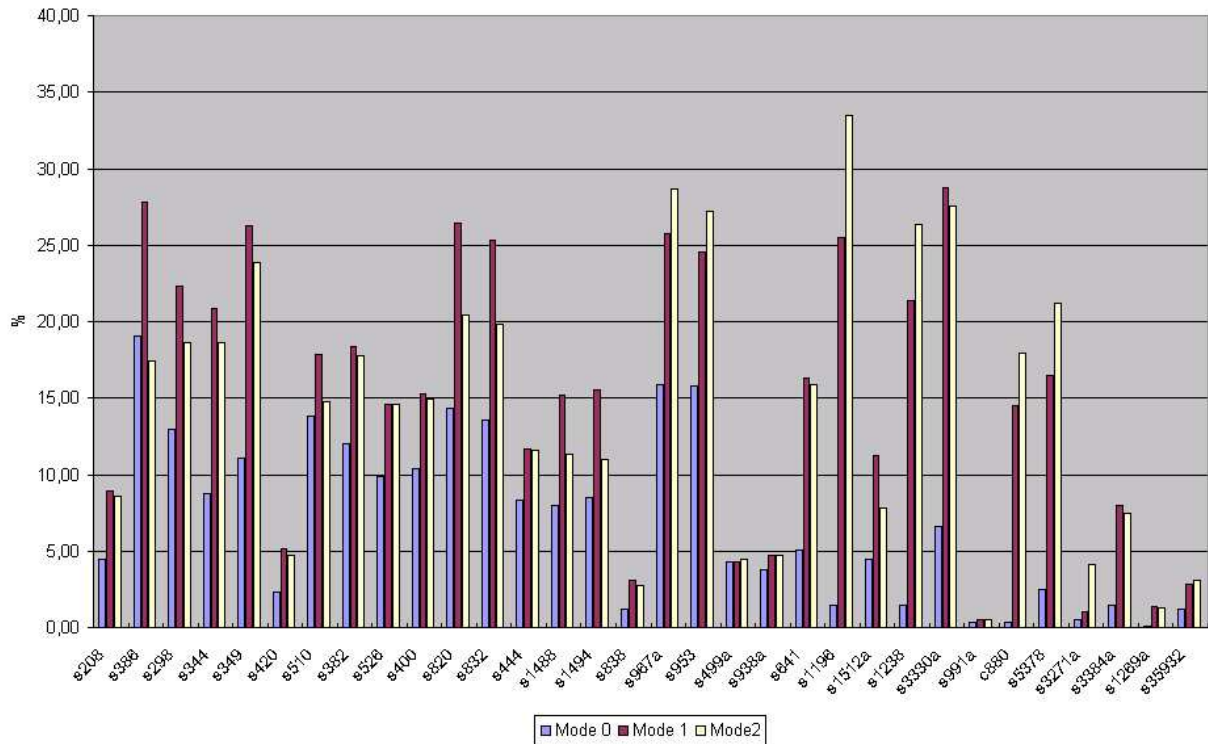


Abbildung 47: Anteil robuster Tests im Testwertsatz bei aktiver Kompaktierung

sensitivieren (\rightarrow 5.3). Kommt es infolge der lokalen Implikationen zu Widersprüchen, ist die Redundanz des Fehlers bewiesen, da alle sich in diesem Schritt ergebenden Signalwerte zwangsläufig aus der Definition des Pfadverzögerungsfehlers folgen. Weil im Normalfall die bei einer Testmustergenerierung adressierten Pfade nur einen Bruchteil der modellierten Fehler darstellen, ist die Effizienz dieses Ansatzes nur schwer zu bewerten. Für die untersuchten Schaltungen, die den Test aller modellierten Fehler zum Ziel hatten, zeigt sich, dass ein Großteil der redundanten Fehler bereits während der vorgeschalteten Sensitivierung erkannt werden.

Die Ergebnisse der Testmustergenerierungsläufe finden sich in den Tabellen 16, 17 und 18. Die Ergebnisse bezüglich der Tendenz an robusten und nicht robusten Tests sind mit denen der nicht kompakten Läufe vergleichbar. Der Anteil nicht robuster Testmuster sinkt kontinuierlich mit den strengeren Anforderungen (Abbildung 46), während der Anteil robuster Muster im Testwertsatz steigt, die absolute Zahl robust testbarer Fehler jedoch schaltungsabhängig variiert (Abbildung 47).

Abbildung 48 zeigt graphisch den Unterschied an robusten Testmustern im Testwertsatz zwischen kompakter und nicht kompakter Testmustergenerierung. Die dargestellten Werte beziehen

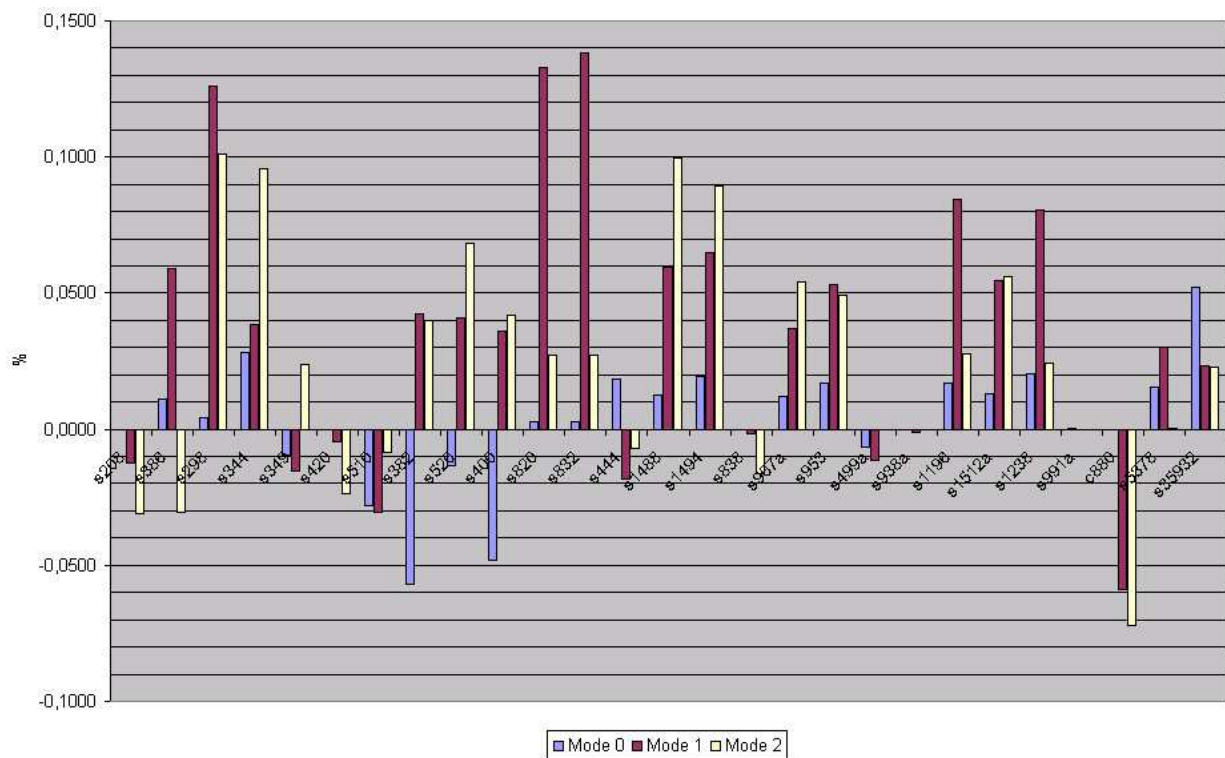


Abbildung 48: Veränderung der Zahl robuster Muster bei normaler und kompakter Testmuster-generierung

sich auf die bei der nicht kompakten Testmuster-generierung erhaltenen Testmuster, sodass positive Werte einen Gewinn an robusten Testmustern im Testwertsatz der kompakten Testmuster-generierung bedeuten. Es zeigt sich, dass keine allgemein verbindliche Aussage getroffen werden kann, obwohl die kompakte Testmuster-generierung tendenziell zum Erhalt einer höheren Zahl robuster Testmuster führt. Die erreichbaren Kompressionsraten liegen in den Beispielen zwischen und 1,35 und 23,04 Fehlern/Testmuster und sinken mit steigenden Anforderungen. Eine genauere Betrachtung zeigt, dass für größere Schaltungen tendenziell kompaktere Testmuster erzielt werden können. Das Ergebnis erklärt sich aus der größeren Anzahl an Schaltungseingängen, die das parallele Testen mehrerer Pfade unterstützen.

6.3 Vergleiche mit Arbeiten aus der Literatur und Abschätzung der Leistungsfähigkeit

Der Vergleich der erreichten Ergebnisse mit anderen Arbeiten aus der Literatur ist nur eingeschränkt möglich. 11 der 13 vorliegenden Arbeiten treffen bei der Angabe der Ergebnisse keine Aussage zum verwendeten Applikationsmodus ($\rightarrow 2.10$). Für die gezeigten Ergebnisse muss daher davon ausgegangen werden, dass keine Abhängigkeiten $FV = f(IV)$ berücksichtigt wurden, wodurch die Testmuster generierung mehr Freiheitsgrade erlangt. Die Anwendung der erhaltenen Testmuster setzt jedoch die Verwendung von Enhanced-Scan-Flip-Flops voraus. Zwei Arbeiten haben ihren Schwerpunkt in der Berechnung von Tests, die besonderen Ansprüchen gerecht werden. So beschränkt sich [30], auf die Berechnung von Testmustern, die einen hazardfreien robusten Test darstellen, während [27] explizit die Berechnung möglichst hochwertiger nicht robuster Tests für jene Fehler versucht, für die kein robuster Test existiert. Die Arbeit von Underwood et. al [128] benennt die Testmuster generierung für einen Motorola-Prozessor als Umfeld und verwendet die funktionale Einstellung als Applikationsmodus, den vorgestellten experimentellen Ergebnissen liegt jedoch eine Menge von nur 1000 ausgewählten Pfaden zugrunde.

Als einzige vergleichbare Arbeit verbleibt damit [52]. Der Autor beschreibt darin die Unterschiede und Einflüsse der Verwendung der beiden Applikationsmodi. Tabelle 7 und Tabelle 8 zeigen einen Vergleich der Ergebnisse bei Verwendung des Applikationsmodus der funktionalen Einstellung.

Aus Tabelle 7 lässt sich ableiten, dass der sequentielle Ansatz selbst bei strengsten Vorgaben (vorletzte Spalte) die robuste Fehlerabdeckung eines Ansatzes mit angepasster Logik (wie ihn [52] verwendet) nicht erreichen kann. Die Zahl der erhaltenen robusten Tests liegt, je nach Schaltung,

Tabelle 7: Vergleich mit den Ergebnissen aus [52] für robuste Tests

	# Fehler	[52]	Modus 0	Modus 1	Modus 2	Modus 2 in % von [52]
s3330	9530	5386	503	2135	2028	37,65
s5378	27084	16218	373	3946	5744	35,41
s3271	38388	2235	233	1053	1367	61,16
s3384	39582	8217	264	2846	2833	34,48
s35932	394282	13504	2280	10732	11604	85,93
s9234	489708	8454	1066	4277	5150	60,92

Tabelle 8: Vergleich mit den Ergebnissen aus [52] für nicht robuste Tests

	# Fehler	[52]	Mode 0	Δ	Mode 1	Mode 2
s3330	9530	8065	8022	-43 (0,53%)	6073	4667
s5378	27084	19044	18936	-108 (0,57%)	16967	15758
s3271	38388	19292	19289	-3 (0,02%)	3287	2400
s3384	39582	31941	31834	-7 (0,02%)	13793	8975
s35932	394282	49786	49786	0 (0,00%)	24221	21341
s9234	489708	33397	33304	-93 (0,28%)	14263	10634

zwischen 35% und 86%. Tabelle 8 vergleicht die Ergebnisse der nicht robusten Testmuster-generierung aus [52] mit der Zahl der Fehler, für die der in dieser Arbeit vorgestellte Testmuster-generators einen Test finden konnte. Die Ergebnisse sind für beide Arbeiten praktisch identisch, wird dem Vergleich die Testmuster-generierung unter schwächsten Vorgaben zugrunde gelegt. Die geringfügigen Abweichungen (Spalte Δ) dürften ihren Ursprung in möglichen kleineren Fehlern in der Implementierung des vorgestellten Generators oder der Referenz haben.

Vergleichsläufe des entwickelten Testmuster-generators mit einem industriell erhältlichen Testmuster-generators an drei Schaltungen der Größe 370.000, 500.000 und 750.000 Gatter zeigten weiterhin, dass eine auf dem sequentiellen Ansatz beruhende Testmuster-generierung bei entsprechenden Optimierungen bzgl. zu verwendender Abbruchkriterien in Bezug auf Laufzeit und Fehlerabdeckung zu vergleichbaren Ergebnissen führt, sich also auch für Schaltungen aktueller Größe eignet. Der Prototyp wurde darüber hinaus bereits für den Produktionstest einer industriell gefertigten Schaltung der Größenordnung 2 Millionen Gatter bei etwa 150.000 Flip-Flops angewandt und erzielte dabei zufriedenstellende Ergebnisse.

7 Anwendungen verzögerungsfehlerbasierter Testmuster

Der im Rahmen dieser Arbeit untersuchte Testmustergenerator wurde mit der Maßgabe entwickelt, die in aktuellen Technologien immer häufiger auftretenden defektbedingten Laufzeitverletzungen in den Produktionstest integrieren zu können. Darüber hinaus sind weitere Anwendungen der erhaltenen Testmuster möglich, zum Beispiel in der Analyse von Problemen in der Ramp-Up Phase einer Chip-Produktion oder in der Klassifizierung der aus der Produktion erhaltenen Schaltkreise.

7.1 Produktionstest

Die durch die veränderten Gegebenheiten bei den Fertigungstechnologien neu eingebrachten Fehlermechanismen führen im Produktionstest zu einer steigenden Komplexität der Testabläufe sowie einer wachsenden Datenmenge an Testvektoren. Als Folge steigen die Kosten des Tests in Folge der benötigten Testzeit und des Speicherbedarfs der Testerhardware. Da einige Fehlermodelle zusätzlich erhöhte Anforderungen an den von der Testerhardware zu realisierenden Datendurchsatz stellen, erhöhen sich deren Kosten, die wiederum einen Teil zu den Testkosten je Chip beitragen. Aus diesem Grunde stehen für den Test aktueller Schaltungen verschiedene Alternativen bereit, aus denen ein Testablauf zusammengestellt werden kann.

Die in Kapitel 3.3 beschriebenen Experimente weisen nach, dass auf den Test von Verzögerungsfehlermodellen im Produktionstest nicht verzichtet werden kann, eine Aussage über das geeignetste Verzögerungsfehlermodell geben sie jedoch nicht. Mit Blick auf die Testdatenmenge bietet sich an, bereits im Testwertsatz vorhandene Testvektoren bei Betriebsfrequenz an die Schaltung zu legen, um die Testdatenmenge gering zu halten und Verzögerungsfehlereffekte parallel zu den eigentlich adressierten Fehlern anzugehen⁴⁴. Diese Lösung kann jedoch nur eine ergänzende Strategie zur deterministischen Suche nach Verzögerungsfehlern sein, da sie für die Verzögerungsfehler eher den Charakter zufällig ausgewählter Testmuster haben. Die Fehlerabdeckung eines solchen Tests ist über eine Fehlersimulation bestimmbar. Soll dem Verzögerungsfehlerstest ein Fehlermodell zugrunde gelegt werden, stellt sich die Frage nach einer Eignung der einzelnen Fehlermodelle.

⁴⁴im Kapitel 3.1 auch als *at-speed stuck-at Test* bezeichnet

Das Pfadverzögerungsfehlermodell geht von einer verteilten Fehlerannahme aus, kann wegen der großen Zahl modellierter Fehler aber keinen vollständigen Test ermöglichen und wird sich daher eher auf einige ausgewählte Pfade im Design anwenden lassen. Da es sich bei diesen Pfaden vornehmlich um die längsten Pfade in der Schaltung handelt, bietet sich ein -abseits des defektorientierten Testansatzes- liegendes Anwendungsgebiet für Tests nach dem Pfadverzögerungsfehlermodell in der Bestimmung der maximal erreichbaren Betriebsfrequenz des Prüflings (speed binning). Insgesamt kann das Pfadverzögerungsfehlermodell für den Produktionstest wegen seiner großen Fehlermenge jedoch nur unterstützenden Charakter haben.

Die Übergangs- und Gatterverzögerungsfehlermodelle bieten im Unterschied dazu durch ihre lokale Fehlerannahme prinzipiell eher die Möglichkeit eines vollständigen Tests. Kleine, verteilte Fehler werden sie ebenfalls finden, wenn diese sich entlang der für die lokalen Fehler gewählten kritischen Pfade zu einer messbaren Größe aufsummieren. Da diese im Falle des Gatterverzögerungsfehlermodells möglichst lang sein sollen, beinhaltet ein vollständiger, optimaler Gatterverzögerungsfehlerertest auch den Test der längsten kritische Pfade der Schaltung und deckt die Pfadverzögerungsfehler auf ihnen mit ab.

Mit Blick auf die Datenmenge des gesamten Testwertsatzes ergibt sich zwangsläufig, dass -einen vollständigen Test nach dem Übergangs- oder Gatterverzögerungsfehlermodell voraussetzender Anteil der verzögerungsfehlerbasierten Testmuster deutlich größer ist als der der statischen Fehlermodelle. Dies begründet sich durch die notwendigen zwei Testvektoren je Testmuster sowie die durch die komplexeren Bedingungen schlechtere Komprimierbarkeit dieser Muster. In Tabelle 9 sind Ergebnisse der Testmustererzeugungen nach dem Haftfehlermodell (SA) und dem Übergangsfehlermodell (TD) gegenüber gestellt. Bei den Übergangsfehlern wurde der Applikationsmodus der funktionalen Einstellung verwendet. Ferner wurden die primären Eingänge während des Tests konstant gehalten, was den Freiheitsgrad bei der Testmustererzeugung einschränkt. Die Ergebnisse zeigen, dass im Vergleich zum Haftfehlerertest eine größere Anzahl an übergangsfehlerbasierten Testvektoren notwendig ist, um eine ansprechende Fehlerabdeckung zu erhalten. Folglich ist die Kompressionsrate, wie in Abschnitt 6.2 bereits für Testmuster nach dem Pfadverzögerungsfehlermodell gezeigt, auch für das Übergangsfehlermodell deutlich schlechter als für das Haftfehlermodell. Die zusätzlichen Restriktionen (Applikationsmodus, konstante Werte an den primären Eingängen) verhindern zudem das Erreichen einer vergleichbaren Fehlerabdeckung. Der Testwertsatz ist in jedem Fall bei der Beachtung dynamischer Fehlereffekte von deren Testvektoren dominiert, obwohl auch hier nur ein Testvektor (*IV*) je primären Fehler zuzuführen ist, da *FV* aus *IV* gebildet wird. Die Ergebnisse zeigen weiterhin, dass sich eine

auf dem sequentiellen Ansatz (\rightarrow 5.1.2) basierende Testmuster generierung für die Berechnung industrieller Schaltungen aktueller Schaltungsgröße eignet.

Den akademisch interessanten Überlegungen, dass in bestimmten Fällen erst die Verwendung von drei Testvektoren die Detektion des minimal detektierbaren Fehlers ermöglicht [38], stehen im Produktionstest aktuelle Schaltungsgrößen gegenüber, die bereits bei der Berechnung zweier Testvektoren zu hohen Rechenzeiten führen. Da nur der erste *IV* in die Scan-Kette eingetaktet werden kann, müssen die folgenden beiden aus diesem folgen. Die Wahrscheinlichkeit, einen entsprechenden Testvektor zu finden, dürfte damit für die meisten Fehler gering sein. Ferner stellt sich die Frage, ob die Kompressionsraten solcher Tests nicht nochmals geringer ausfallen, sodass die Zahl der einzutaktenden Testvektoren weiter steigt, um eine akzeptable Fehlerabdeckung zu erreichen.

Primäre Eingänge

Wie bereits erwähnt sind die in Tabelle 9 erhaltenen Verzögerungsfehlertestmuster unter der Randbedingung konstanter primärer Eingänge entstanden. Die Werte auf primären Eingängen unterliegen, im Gegensatz zu den pseudo primären Eingängen, keinen strukturell bedingten Abhängigkeiten (\rightarrow 2.10). Da sie extern direkt zugänglich sind, werden sie jedoch durch den Tester eingangsseitig eine höhere parasitäre Lastkapazität durch die Verbindungsleitung, Pads und Bondleitungen aufweisen (gleiches Problem beschreibt auch [126]). Weil die Anforderungen an die Genauigkeit des Umschaltens auch für die primären Eingänge gelten, ergeben sich bei der Applikation der Testmuster möglicherweise Probleme. Aus diesem Grund wird oftmals darauf verzichtet, die für die Testmuster generierung wertvollen weil leicht einstellbaren primären Eingänge während des Tests umzuschalten. Als Folge sinkt die Zahl der Fehler, für die ein Test gefunden werden kann, weil der Freiheitsgrad für die Testmuster generierung zurück geht.

Die aus der Literatur zu gewinnenden Aussagen weisen darauf hin, dass die Notwendigkeit eines Verzögerungsfehlertests für den Produktionstest gegeben ist, die Anwendung jedoch vielfach über einen Übergangsfehlertest oder einen Haftfehlertest bei Betriebsfrequenz nicht hinaus geht. Findet das Pfadverzögerungsfehlermodell Anwendung, wird dies nur für eine kleine Zahl ausgesuchter Pfade Anwendung finden. Es steht zu erwarten, dass die dadurch erreichte zusätzliche Defektabdeckung eher gering ist und nur zusätzliche Gewissheit erreicht wird, dass die als kritisch erachteten Pfade der Schaltung nicht ausfallen. Über den Einsatz von Gatterverzögerungsfehlertests wird kaum berichtet. Grund hierfür dürfte zum einen die mangelhafte Verfügbarkeit geeigneter Testmuster generatoren sein, zum anderen die in Abschnitt 5.2.2 angeführten Probleme.

Tabelle 9: Ergebnisse von Testmustererzeugungsläufen in industriellen Designs

	Fehlermodell	# Gatter	# Flip-Flops	# Scan-Ketten	# Testvektoren	Fehlerabdeckung
P044	<i>SA</i>	44830	2173	23	1697	96.33
	<i>TD</i>				9694	51.62
P088	<i>SA</i>	88460	4302	18	926	98.99
	<i>TD</i>				514	70.63
P092	<i>SA</i>	92813	3877	8	249	99.83
	<i>TD</i>				452	94.70
P140	<i>SA</i>	140516	10502	24	575	94.37
	<i>TD</i>				1936	74.75
P262	<i>SA</i>	262451	16576	45	952	97.68
	<i>TD</i>				2915	89.49
P263	<i>SA</i>	263543	16576	45	927	90.49
	<i>TD</i>				2836	78.73
P278	<i>SA</i>	278783	17899	55	746	91.05
	<i>TD</i>				6841	82.59
P284	<i>SA</i>	284767	17899	55	1134	95.27
	<i>TD</i>				2212	42.94
P330	<i>SA</i>	330014	17226	64	2249	97.25
	<i>TD</i>				14470	90.40
P387	<i>SA</i>	387518	24065	50	398	96.11
	<i>TD</i>				1036	84.36
P404	<i>SA</i>	404241	24065	50	762	96.26
	<i>TD</i>				880	78.16
P417	<i>SA</i>	417974	29205	64	609	88.04
	<i>TD</i>				861	57.21
P464	<i>SA</i>	464993	30731	76	2273	97.85
	<i>TD</i>				1523	88.58
P874	<i>SA</i>	874023	42076	59	993	95.92
	<i>TD</i>				2063	85.77
P950	<i>SA</i>	950751	104624	82	628	94.48
	<i>TD</i>				2835	90.19

me, die in der Schaltung vorliegenden Verzögerungen mit einer ausreichenden Genauigkeit zu beschreiben.

7.2 Analyse

Das Ziel der Analyse ist die Ermittlung von Defektmechanismen, welche die Schaltung ausfallen lassen. Während der Test seinen Schwerpunkt in der Anwendung auf Produkte aus eingefahrenen Prozessen haben dürfte, kommt der Analyse eine besondere Bedeutung während der Startphase⁴⁵ der Produktion einer neuen Schaltung zu. Über die Analyse können die Gründe für wiederholt auftretende, defektbedingte Ausfälle ermittelt und Maßnahmen ergriffen werden, die zur Beseitigung der Defekte führen. Die in der ramp-up Phase eines Produktes adressierten Defekte sind dabei in der Regel Defekte, die vom Schaltungslayout begünstigt werden und sich über eine geänderte Leitungsführung oder Bauteilplatzierung abstellen lassen [56].

Für die Analyse einer Schaltung sind, je nach Ausrichtung, sowohl das auf einer lokalen Fehlerannahme basierende Gatterverzögerungsfehlermodell wie auch das von einer verteilten Fehlerannahme ausgehende Pfadverzögerungsfehlermodell geeignet.

Beim Industriepartner durchgeführte Untersuchungen verwenden die erstellten Testmustergeneratoren für verschiedene Problemstellungen in der Analyse. In [5] wird das für in Nanotechnologie gefertigte Schaltkreise erwartete Problem der resistiven Vias (\rightarrow 3.1) adressiert und der Einfluss auf die Signallaufzeiten mit Hilfe eines Testmustergenerators für Gatterverzögerungsfehler [106] untersucht. Die Untersuchungen verwenden einen speziell für den Test hergestellten Chip (Vektor-Chip), der in $0,25\mu\text{m}$ -CMOS Technologie gefertigt wurde. Die Auswertungen der Daten integrierter Ring-Oszillatoren zeigen zunächst erhebliche Geschwindigkeitsunterschiede zwischen den Chips verschiedener Wafer. Daraus folgt, dass für die Detektion eines Verzögerungsfehlers Φ der Größe δ_1 in identischen, aber auf unterschiedlichen Wafern gefertigter Schaltungen unterschiedliche Timing-Limits gesetzt werden müssen. Ferner zeigen die Untersuchungen, dass 8 fehlerhafter Schaltungen aus einer Menge von 910 Schaltungen allein durch den verzögerungsfehlerbasierten Test als defekt erkannt werden, während alle im Haftfehler-test ausfallenden Schaltungen auch unter Anwendung der Verzögerungsfehler-tests detektiert wurden. Die generelle Aussage der Arbeit, der Verzögerungsfehler-test solle den Haftfehler-test als spannungsbasierten Test ersetzen, kann jedoch so nicht gelten. Der im Vergleich zum Haftfehlermodell hohe Testaufwand, die im Paper angesprochenen Probleme mit Prozessvariationen und die

⁴⁵wegen dem Ziel, möglichst schnell ein hohe Ausbeute zu erzielen, auch als *ramp-up phase* bezeichnet

einschränkenden Annahmen während der Testmuster generierung lassen ihn eher als sinnvolle Ergänzung denn als Ersatz erscheinen.

Mit dem Problem, dass der Test nach Verzögerungsfehlermodellen nicht alle Verzögerungsfehlereffekte in Nanotechnologieschaltungen erkennt, befasst sich [92]. Anhand von experimentellen Ergebnissen zeigt sich, dass die Anwendung einer Testwertsätze $\Theta^{\ddot{U}}$ von 1861 Übergangsfehler-testmustern im Vergleich zu Θ^H (1861 Haftfehler-testmustern) zu keiner höheren Erkennungsrate führt, obwohl sich die berechneten Fehlerabdeckungen beider Testwertsätze, bezogen auf das Übergangsfehlermodell, mit 97% ($\Theta^{\ddot{U}}$) und 40% (Θ^H) deutlich unterscheiden. Ferner fielen bei Tests unter verschiedenen Betriebsspannungen Schaltungen mit Laufzeitproblemen bei höherer Betriebsspannung aus, die unter niedriger Spannung fehlerfrei erschienen, während andere unter niedriger Betriebsspannung mit Laufzeitfehlern ausfielen, die bei höheren Betriebsspannungen verschwanden. Weil eine niedrigere Betriebsspannung die Umladevorgänge verzögert und damit die Schaltung verlangsamt, sind die dann ausfallenden Schaltungen über die bekannten Mechanismen erklärlich. Eine fehlerfreie Operation unter niedriger Spannung und verzögerungsbedingte Ausfälle bei höheren Betriebsspannungen lassen sich jedoch zunächst nicht erklären. Es folgt daraus, dass die Verzögerungsfehlermodelle nicht alle in Nanotechnologieschaltungen auftretenden Fehlereffekte modellieren (\rightarrow Kapitel 8).

In [83, 81] wird eine Kombination von Testmustern für das Übergangs- bzw. Pfadverzögerungsfehlermodell zur Diagnose fehlerhafter Devices verwendet. Zunächst wird eine Liste möglicher Fehlerorte aufgebaut, die sich bei Anwendung der Übergangsfehler-testmustern aus den ausfallenden Devices ergeben (vergl. Abbildung 3 auf Seite 4). Für diese Liste wird die Verwandtschaft des Übergangsfehlermodells mit dem Haftfehlermodell ausgenutzt, welche die Verwendung eines haftfehlerbasierten Diagnoseablaufes ermöglicht [55]. Für ein ausfallendes Testmuster T sind alle Knoten k potentielle Kandidaten, für die der FV einen entsprechenden Haftfehler-test bildet, während IV den Knoten k auf den Wert des Haftfehlers einstellt. Über die Kombination der Informationen über ausfallende und nicht ausfallende Vektoren kann diese Liste der möglichen Fehlerorte minimiert werden. Für die verbleibenden Fehlerorte werden im Anschluss Testmuster nach dem Pfadverzögerungsfehlermodell berechnet. Das Ziel ist, über die so erhaltenen Timinginformationen die potentiellen Fehlerorte weiter einzuschränken.

7.3 Speed binning

Abseits vom defektorientierten Testansatz können verzögerungsfehlerbasierte Testmuster auch dazu verwandt werden, Prozessvariationen auszumessen, um so die maximal mögliche Perfor-

mance jedes einzelnen Chips auszuloten. Zu diesem Zweck werden vorzugsweise pfadverzögerungsfehlerbasierte Testmuster verwandt, die die längsten funktionalen Pfade des Designs überprüfen. Mit den gewonnenen Informationen lassen sich die gefertigten Schaltungen in unterschiedliche Leistungsklassen einteilen und zu unterschiedlichen Preisen verkaufen.

8 Fehlereffekte in Nanotechnologien

Die durch die steten Anforderungen an Performancegewinne sowie technologische Fortschritte ermöglichten Anstiege bei den Betriebsfrequenzen digitaler Schaltungen haben gezeigt, dass Verzögerungsfehlermodelle nur die ersten Auswirkungen beschreiben, welche Defekte bei steigenden Systemfrequenzen auf die produzierten Schaltungen haben können [92]. Mit steigender Integrationsdichte verändern sich die Eigenschaften der Schaltungen. Sinkende Abmaße bei den Transistoren führen zu geringeren Gatekapazitäten, für ihre Umladung werden folglich weniger Ladungsträger benötigt, die Schaltgeschwindigkeit der Transistoren steigt. Da die Anzahl der zu integrierenden Transistoren steigt, nehmen auch die Probleme beim Routing zu, die Verwendung einer größeren Anzahl an Verbindungsebenen ist die zwangsläufige Folge. Die parasitären Widerstände der Leitungen steigen, bedingt durch die geringeren Abmaße sowie durch die durch die Vias eingebrachten Übergangswiderstände. Zusammengenommen führen diese beiden Effekte dazu, dass sich mit sinkenden Abmaßen die Anteile, die die Transistoren und die Leitungen zur Signallaufzeit beitragen, immer weiter in Richtung der Leitungen verschieben. Als Folge wird erwartet, dass in kommenden Technologien bezüglich der Schaltungsausfälle Defekte dominieren, welche die Eigenschaften der Leitungen beeinflussen, während parametrische Schwankungen im Verhalten der Transistoren auf die Gesamtlaufzeit auf den Signalpfaden einer Schaltung einen eher geringeren Einfluss nehmen.

Die steigenden Betriebsfrequenzen führen jedoch im Zusammenspiel mit den geringeren Leitungsabständen und feineren Strukturen noch zu weiteren Problemen, die weder die statischen noch die Verzögerungsfehlermodelle abdecken und die unter dem Schlagwort *Signalintegritätsverletzungen* zusammengefasst werden. Die auftretenden Fehlereffekte sind vielfältig und umfassen kopplungsbedingte Einflüsse ebenso wie Elektromigrationserscheinungen.

8.1 Elektromigrationserscheinungen

Der Begriff der Elektromigration beschreibt einen Materialabtrag infolge eines Stromflusses. Elektromigration tritt ein, wenn die Stromdichte in einem Leiter einen kritischen Wert übersteigt⁴⁶ [131]. Als Folge können die Ladungsträger durch Stoßeffekte Ionen aus dem Leiterbahnmateriale lösen und damit einen Materialabtrag bewirken, sodass der für die Stromleitung zur Verfügung stehende Leiterquerschnitt abnimmt. Als Folge steigt die Stromdichte weiter an, was den Effekt noch verstärkt. Es wird erwartet, dass der Einfluss von Elektromigrationserscheinungen

⁴⁶Als Richtwert für die maximal erlaubte Stromdichte gibt [131] $1\text{mA}/\mu\text{m}^2$ bei 125°C an.

für modernere Technologien wegen der geringeren Leiterquerschnitte ebenso steigen wird wie durch die Anzahl verwendeter Vias. Diese bergen im Falle einer mangelhaften Kontaktierung die Gefahr erhöhter Stromdichten, da der für die Signalleitung zur Verfügung stehende Querschnitt verringert wird.

Um Elektromigrationsprobleme zu erkennen werden gefertigte Schaltungen einem Burn-In Test unter erhöhter Betriebsspannung unterzogen.

8.2 Signalkopplungen

In Anbetracht der steigenden Betriebsfrequenzen wird Kopplungserscheinungen auf internen Signalleitungen in modernen Schaltungen vermehrt Beachtung geschenkt werden müssen. Signalkopplungen zwischen Leitern können durch induktive oder kapazitive Kopplung hervorgerufen werden [140], wobei in Bezug auf die sinkenden Abstände zwischen den Verbindungsleitungen vermehrt kapazitive Effekte in das Interesse des Tests rücken. Die Beachtung und Verhinderung möglicher schaltungsinterner Signalkopplungen wurden in der Vergangenheit dem Aufgabenbereich des Schaltungsdesigns zugeordnet. Technologische Unterstützung fand der Designer in der Verwendung anderer Werkstoffe für die Isolatoren, sogenannter high- κ Dielektrika, welche die über $C = \varepsilon_0 \varepsilon_r \frac{A}{d}$ gegebenen Kapazitäten herab setzen. Die Fertigung hochfrequent getakteter integrierter Schaltungen hat jedoch inzwischen erkennen lassen, dass kopplungsbedingte Probleme einzelner Bauteile als Folge von Produktionsabweichungen auftreten können und damit eine Berücksichtigung im defektorientierten Testen finden müssen.

Die defektbedingte Entstehung von Kopplungserscheinungen in integrierten Schaltungen kann in verschiedenen Prozeßschritten begründet sein, sobald diese die parasitäre Kapazität zwischen den Leitungen erhöhen. Als Beispiel kann eine nicht ausreichende Ätzung oder Belichtungsfehler, in deren Folge der Abstand zwischen Leiterbahnen geringer ausfällt als erwartet, ebenso eine Quelle für eine erhöhte Kapazität zwischen den Leitern sein wie Fehler in der Ausrichtung zweier übereinander liegender Layer, in deren Folge sich Leiterbahnen mit einer größeren Fläche als vorgesehen überlappen [89]. Untersuchungen in der Literatur zeigen, welche Auswirkungen kopplungsbedingte Mechanismen auf ein Nutzsignal haben können [28].

Abbildung 49 zeigt, dass sich die Signalverfälschungen in Über- bzw. Unterschwingern sowie temporären Signalschwankungen äußern können. Um die Einflüsse der Veränderungen auf die Schaltungsfunktion zu beurteilen, müssen die beteiligten Kopplungseffekte und die im realen Betrieb auftretenden Größen abgeschätzt und bewertet werden. Ferner müssen für den Test Wege gefunden werden, den Fehlereffekt dem Test zugänglich zu machen.

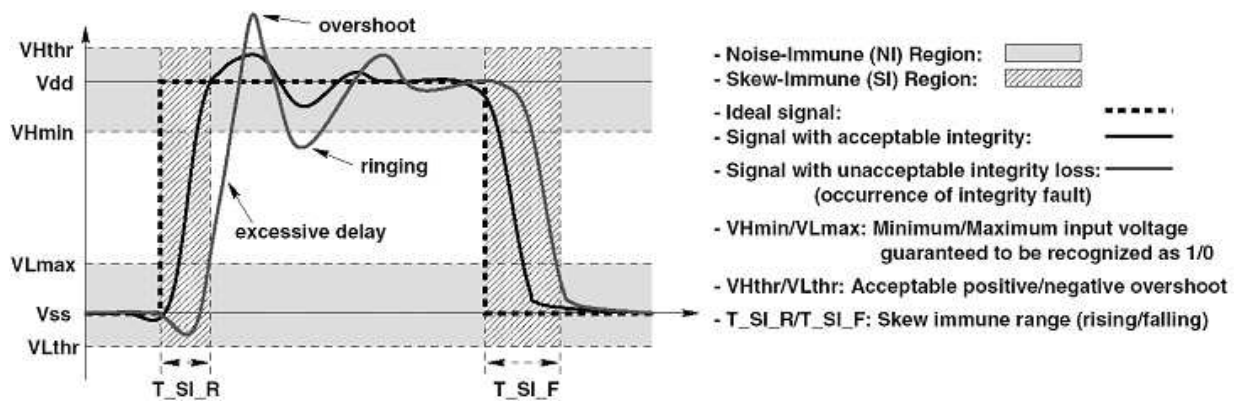


Abbildung 49: Das ideale Rechtecksignal kann in der Realität verschiedene Signalformen aufweisen [4]

8.2.1 Überschwinger

Als Überschwinger sind Spannungspulse definiert, die das Signal den als logisch "1" definierten Bereich nach oben verlassen lassen. Weil diese Signale am Gateanschluß der betroffenen Transistoren auftreten, können sie im Gateoxyd zu erhöhten Feldstärken führen und dieses schädigen. In der Literatur finden sich Hinweise, dass eine kurzzeitige Spannungsüberhöhung in der Regel keinen unmittelbaren Schaden anrichtet, sich bei wiederholtem Auftreten jedoch nachhaltige Veränderungen Schaltungseigenschaften einstellen können. Die möglichen Fehlereffekte betreffen dabei zum einen Ladungsträger, die durch die erhöhte Feldstärke soviel (kinetische) Energie gewinnen dass es ihnen gelingt, ins Gateoxyd einzudringen und sich dort abzulagern, eine Veränderung der Transistoreigenschaften ist die Folge [95] (*hot carrier effect*). Ebenfalls bekannt ist, dass fortwährende Überspannungsimpulse auf ein Gateoxyd das Material schwächen und damit Grund für einen sogenannten *zeitabhängigen Durchbruch des Dielektrikums* sein können, was wiederum zum vorzeitigen Ausfall der Schaltung führen kann [60]. Wegen der mit den modernen Technologien ebenfalls sinkenden Betriebsspannungen wird den Überschwängern jedoch technologiebedingt nur wenig Möglichkeit eingeräumt, zu Problemen in modernen Schaltungen zu führen. Die für den Hot-carrier Effekt notwendigen Energien von 3,2 eV (Elektronen) bzw. 3,8 eV (Löchern) lassen sich bei sinkenden Spannungen immer schwerer erreichen [131].

8.2.2 Unterschwinger

Als Unterschwinger werden Effekte bezeichnet, die das Potential eines Schaltungsknotens temporär unter den als logische "0" definierten Spannungspegel absinken lassen. Ein Schädigungs-

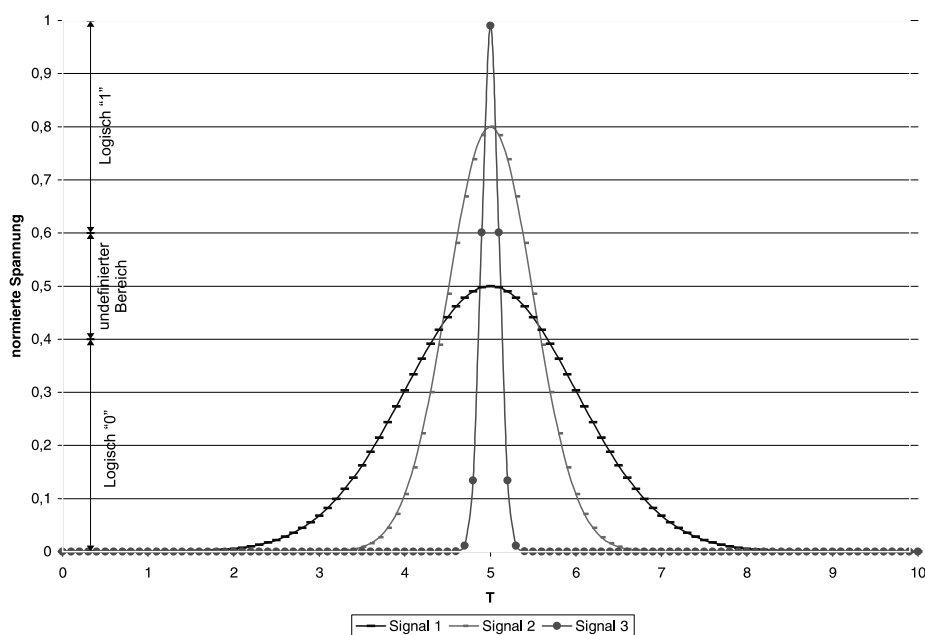


Abbildung 50: Der statische Hazard aufgrund von eingekoppelten Signalen kann unterschiedliche Form und Amplitude erreichen

potential bestünde für den Fall, dass es durch die umgekehrte Spannung zu einer ungewollten Stromleitung über eine nun in Durchlaßrichtung gepolte PN-Strecke kommt. Da kopplungsbedingte Einflüsse ausschließlich auf die Gates der Transistoren wirken und damit kein P- oder N-dotiertes Gebiet erreichen stellen Unterschwinger für in CMOS realisierte Schaltungen kein unmittelbares Problem dar.

8.2.3 Temporäre Signalschwankungen

Temporäre Signalschwankungen (Hazard-Effekte) können als Folge einer Energieeinkopplung auf statische Signale (Abbildung 50) als auch in Form von Einschwingvorgängen nach einem Umschaltvorgang entstehen. Um Auswirkungen auf das Schaltungsverhalten zu nehmen, muss der Signalpegel sich zunächst zumindest soweit in den nicht definierten Bereich bewegen, dass die nachfolgende Logik ihn als vom Endwert unterschiedlichen Signalwert interpretiert. Da die dabei auftretenden Umladevorgänge in den Basiskapazitäten der Transistoren eine gewisse Zeit benötigen filtert das Gatter den fehlerhaften Schaltimpuls, wenn dessen Verweildauer für diesen Umschaltvorgang nicht ausreicht. Im Beispiel aus Abbildung 50 sind die Signale 1 und 3 unproblematisch, da Signal 1 keine ausreichende Spannung erreicht während die Verweildauer von

Signal 3 nicht ausreicht, einen Umschaltvorgang hervorzurufen.

Mit Blick auf die Realisierung eines möglichen Tests lassen sich Hazardeffekte dem Test prinzipiell zuführen, da sie im Fehlerfall, digital betrachtet, zu unterschiedlichen Signalpegeln führen. Problematisch ist die Frage, wie sich ein durch Kopplungseffekte möglicherweise eingeführter Hazard an die Ausgänge leiten lässt. Da die am Fehlerort eingebrachten fehlerhaften Signalpegel möglicherweise nur den undefinierten Grenzbereich zwischen dem maximal als ‘0’ bzw. minimal als ‘1’ definierten Signalpegel erreichen stellt sich die Frage, ob verschiedene Wege vom Fehlerort in Richtung der Ausgänge das Signal unterschiedlich beeinflussen [51]. Der in [51] vorgestellte Ansatz nutzt Verstärkungsfaktoren einzelner Gatter als Entscheidungshilfe bei der Wahl des nächsten Gatters zur Propagation des Fehlereffektes zu den Ausgängen. Größtes Manko bei der Realisierung des dort vorgestellten Testmustergenerators liegt in der fehlenden Beachtung der Laufzeit, die das Fehlersignal vom Fehlerort bis zu den Ausgängen benötigt. Da die zu erwarteten, kopplungsbedingt eingebrachten Hazards eher von geringer Breite sein dürften ist ihre Detektion bereits schwierig, da der Ausgang exakt zu Zeit ihres Eintreffens beobachtet werden muss. Alles in allem scheint die deterministische Erkennung temporärer Signalschwankungen für den Produktionstest aus diesen Gründen mit den herkömmlichen Methoden nicht realisierbar zu sein.

8.2.4 Signalverzögerungen

Aus der Menge der kopplungsbedingten Signalveränderungen bleiben eingebrachte Signalverzögerungen damit der einzige Kandidat für die Detektion im Produktionstest. Diese sind durch die in Abschnitt 2.5.2 nicht modelliert, da diese den Fehlerort als einzigen Grund für die Signalverzögerung annehmen, während kopplungsbedingte Signalverzögerungen durch Signalwechsel auf den Aggressoren ausgelöst werden. Eine Verzögerung des Signals auf der Opferleitung kann immer dann geschehen, wenn die Signalwechsel auf der Opferleitung und den Aggressoren eine entgegengesetzte Richtung aufweisen und am Fehlerort möglichst zeitgleich auftreten [24]. Das in [28] definierte *Maximal Aggressor Model* geht von einer beliebigen Zahl an Leitungen als Aggressor aus, die sich bei einem zeitgleichen Signalwechsel auf allen Aggressoren untereinander nicht beeinflussen, während ihre Wirkung auf die Opferleitung maximal wird. Für den praktischen Einsatz lassen sich die Aggressorleitungen jedoch auf die direkt benachbarten Leitungen der Opferleitung reduzieren, da diese den größten Einfluss ausüben, während sie gleichzeitig die Opferleitung gegen fernere Abschirmen [140].

Die drei grundsätzlichen Aufgaben des Tests bestehen demnach darin,

1. eine Flanke auf den Aggressoren einzustellen
2. eine gegenläufige Flanke auf der Opferleitung zu erzeugen und
3. einen kritischen Pfad zu mindestens einem Ausgang zu finden, der den Fehlereffekt propagiert.

Für die Realisierung kann jeder verzögerungsfehlerbasierte Testmustergenerator verwandt werden, der einen Test für einen Verzögerungsfehler auf der Opferleitung berechnet und als Nebenbedingung die gewünschten Flanken auf den Aggressoren einstellt.

Die so gewonnenen Testmuster können nur als erster Ansatz für die Realisierung eines Tests für Signalintegritätsverletzungen verstanden werden, da sie die charakteristischen Eigenschaften und Relationen zwischen den Signalen kaum berücksichtigen. Aus Simulationen ergibt sich, dass sich der Einfluss des Aggressors auf das Opfersignal maximiert, wenn beide Flanke zeitgleich auftreten [24]. Die Beachtung dieser Forderung schränkt die Verwendung eines verzögerungsfehlerbasierten Testmustergenerators ein. Bei Verwendung eines Testmustergenerators für das Übergangsfehlermodell kann auf eine nachträgliche Simulation der erhaltenen Muster nicht verzichtet werden, um die Einhaltung der Gleichzeitigkeitsbedingung zu überprüfen. Bei der Verwendung eines Testmustergenerators nach dem Gatterverzögerungsfehlermodell wird die Beachtung von Laufzeiten in der Schaltung möglich, die in 2.5.2 angeführten Probleme der fehlenden Timinginformationen bleiben jedoch bestehen. Bei der Verwendung eines Testmustergenerators für das Pfadverzögerungsfehlermodell können die zu sensitivierenden Pfade extern vorgegeben und damit zum Beispiel mit Hilfe eines Timing-Analysator bestimmt werden.

Neben dem möglichst gleichzeitigen Auftreten der Flanken nimmt auch die Treiberstärke der Leitungen Einfluss auf die Größe der Verzögerung die umso größer ausfällt, je steiler der Flankenwechsel auf den Aggressoren ausfällt. Ferner begünstigt eine schwache Treiberleistung des Opfers die Beeinflussbarkeit des Signals. Die Treiberleistung wiederum kann, wenn die betroffenen Gatter von Logikgattern getrieben werden, über die Eingangsmuster bestimmt werden [89].

8.3 Einflüsse der Versorgungsspannung

Neben Einflüssen auf den Signalleitungen kann eine erhöhte Schaltungsaktivität zu temporären Spannungsabfällen an den Versorgungsleitungen infolge parasitärer Verluste, hervorgerufen durch den erhöhten Stromfluss, führen. Die Schaltung wird folglich zeitweilig mit einer geringeren Versorgungsspannung betrieben, was ihre Funktion verlangsamt und damit ebenfalls zu

Verzögerungseffekten führt. Ein erster Ansatz für den Test stellt [70] vor. Zum Einsatz kommt ein Testmuster-generator für Pfadverzögerungsfehler. Nach einer gelungenen Testmuster-generierung werden die noch unbestimmten Eingänge mit dem Ziel belegt, zu einer möglichst hohen Aktivität in der Schaltung zu führen.

9 Zusammenfassung

Die Entwicklungen der letzten Jahrzehnte haben die Mikroelektronik bis in sicherheitskritische Bereiche des täglichen Lebens vordringen lassen. Aufgrund der dadurch steigenden Anforderungen an die Zuverlässigkeit ausgelieferter Schaltungen sowie die mit den Fortschritten in der Technologieentwicklung neu eingebrachten Fehlermechanismen kommt dem Produktionstest eine wachsende Bedeutung bei der Fertigung integrierter Schaltungen zu. Die in den gefertigten Schaltungen auftretenden Fehlereffekte lassen sich nicht mit einem Fehlermodell ausreichend abdecken, die Auswertung verschiedener experimenteller Reihen zeigen für alle Fehlermodelle Schaltungen auf, die nur im Test nach diesem Fehlermodell ausfallen. In der Menge der im Produktionstest einzusetzenden Fehlermodelle haben sich in den letzten Jahren die Laufzeitfehler etabliert, begründet in der sich dem Schaltungsdesigner aufgrund sinkender parasitärer Größen anbietenden Möglichkeit, den Datendurchsatz der Schaltung durch die Verwendung höherer Taktfrequenzen zu realisieren.

Die Anforderungen an den Produktionstest werden vor allem durch die Kosten bestimmt, die er verursacht. Aus diesem Grunde kann eine Entwicklung hausinterner Testwerkzeuge dem Halbleiterhersteller Vorteile bieten. Aus Sicht eines Softwareherstellers ist die Verwendung bereits bestehender Codefragmente beim Aufbau neuer Testmustergeneratoren attraktiv, da diese Fragmente bereits im Betrieb befindlich sind und damit eine hohe Fehlerfreiheit aufweisen sollten. Weitere positive Aspekte bietet die leichtere Integration des neuen Werkzeuges in die bestehende Umgebung durch die bereits vorhandenen Schnittstellen, der geringere werkzeugspezifische Wartungsaufwand sowie die Optimierung einzelner Codeteile, die sich in mehreren Tools leistungsfördernd auswirken kann.

Vor diesem Hintergrund entstand im Rahmen der vorgelegten Arbeit ein Testmustergenerator nach dem Pfadverzögerungsfehlermodell, der konzeptionell auf einen Testmustergenerator nach dem Haftfehlermodell aufsetzt. Der Ansatz weicht von den in der Literatur zumeist verfolgten Ansätzen ab, einen der bekannten Algorithmen zur Testmustergenerierung (D-Algorithmus, PODEM, FAN) auf eine spezielle Logik anzupassen und zu reimplementieren. Die Zerlegung des Testmustergenerierungsproblems in eine Testmustergenerierung für 2 aufeinander folgende Taktzyklen verhindert die Beschreibung möglicher Signalwechsel, die zwischen den Taktzeitpunkten auftreten. Das sich daraus ergebende Problem, die robuste Testmustergenerierung während der Berechnung nicht sicherstellen zu können, erschien mit dem Wissen, daß robuste Tests nur für einen Bruchteil der testbaren Pfade einer Schaltung existieren, akzeptabel. Die Testmustergenerierung beginnt mit einer Sensitivierung des Pfades. Gelingt dies nicht, ist der Pfad nicht testbar.

Für alle untersuchten Schaltungen wurde der größte Teil der redundanten Pfade über diesen wenig aufwendigen Schritt erkannt.

Für die Bewertung der erhaltenen Testmuster wurde ein Fehlersimulator aufgebaut. Die Ergebnisse liessen Zweifel an der Qualität der Testmuster aufkommen, die ihren Ursprung in der Definition der für einen nicht robusten Test notwendigen Bedingungen hatten. Nach einer Analyse der Invalidierungsmechanismen konnten Verbesserungen angegeben werden, die die Qualität der erhaltenen Testmuster deutlich erhöhten. Vergleiche mit anderen Arbeiten zeigen unter gleichen Randbedingungen, daß die Zahl der Fehler, für die ein nicht robuster Test gefunden werden kann, praktisch identisch ist. Die Zahl robuster Testmuster erreicht jedoch auch unter strengsten Einschränkungen nicht die in der Literatur angegebenen Werte. Dieses Ergebnis war zu erwarten, wengleich die Fehlersimulation für die erhaltenen pessimistische Werte lieferte. Grund hierfür ist eine fehlende Beachtung der testmusterspezifischen Signallaufzeiten, in deren Folge Signale auch dann als hazardbehaftet klassifiziert werden, wenn dies wegen interner Laufzeiten praktisch nicht eintreten kann.

Durch die im Produktionstest geforderte minimale Zeit für die Testdurchführung besteht Bedarf an kompakter Testmustern. Die Ergebnisse der Testmuster-generierung mit aktiver Testmuster-kompaktierung haben in den meisten Fällen einen weiteren Anstieg robuster Testmuster ergeben, die Kompressionsrate ist dabei jedoch stark schaltungsabhängig. Ein Vergleich mit kommerziell erhältlichen Tools hat gezeigt, dass diese die Testmuster-generierung kompakter Testmuster für dynamische Fehlermodelle erst seit kurzer Zeit unterstützen.

Nach einer Darstellung möglicher Einsatzgebiete der Testmuster-generatoren und einzelner Arbeiten, in denen sie bereits Anwendung fanden schließt die vorgelegte Arbeit mit einem Ausblick auf die kommenden Problemstellungen ab. Dabei zeigt sich mit dem Erreichen des Zeitalters der Nanotechnologie, dass die auftretenden Fehlereffekte immer komplexere Formen annehmen. Im Bereich der Signalintegritätsverletzungen, die ein weites Feld an Fehlereffekten umfassen, äußern sich viele Effekte in Signalverzögerungen, die durch die Verzögerungsfehlermodelle nicht modelliert werden. Die Frage, ob und wie sich derartige Fehlereffekte dem Produktionstest zuführen lassen, ist Thema aktueller Forschungen. Erste Ideen, einen Testmuster-generator für derartige Problemstellungen zu realisieren, bilden den Abschluss dieser Arbeit.

Literatur

- [1] M. Abramovici and P. R. Menon. A Practical Approach to Fault Simulation and Test Generation for Bridging Faults. *IEEE Transactions on Computers*, C-34(7):658–663, July 1985.
- [2] Robert C. Aitken. Extending the Pseudo-Stuck-At Fault Model to Provide Complete ID-DQ Coverage. In *17th VLSI Test Symposium*, pages 128–134. IEEE, April 1999.
- [3] Y. Aizenbud, M. Leibowitz, P. Chang, and B. Koenemann. AC Test Quality: Beyond Transition Fault Coverage. In *International Test Conference*, pages 568–577. IEEE, 1992.
- [4] Amir Atthara and Mehrdad Nourani. Test Pattern Generation for Signal Integrity Fault on Long Interconnects. In *VLSI Test Symposium*, pages 336–344. IEEE, April 2002.
- [5] Keith Baker, Guido Gronthoud, Maurice Lousberg, Ivo Schanstra, and Charles Hawkins. Defect-Based Delay Testing of Resistive Vias-Contacts - A Critical Evaluation. In *International Test Conference*, pages 467–476. IEEE, October 1999.
- [6] Z. Barzilai and B. Rosen. Comparison of AC Self-Testing Procedures. In *International Test Conference*, pages 258–265. IEEE, 1983.
- [7] Bernd Becker. Testing with decision diagrams. *INTEGRATION - the VLSI Journal*, 26:5–20, 1998.
- [8] R.G. Bennetts, C.M. Maunder, and G.D. Robinson. CAMELOT: A Computer-Aided Measure for Logic Testability. *IEE Proc. E*, 128:177–189, 1981.
- [9] Brady Benware, Chris Schuermyer, Srenevasan Ranganathan, Robert Madge, Prabhu Krishnamurthy, Nagesh Tamarapalli, Kun-Han Tsai, and Janusz Rajski. Impact of Multiple-Detect Patterns on Product Quality. In *International Test Conference*, pages 1031–1040. IEEE, October 2003.
- [10] W.C. Berg and R.D. Hess. COMET: A Testability Analysis and Design Modification Package. In *International Conference on Test*, pages 364–378. IEEE, 1982.
- [11] Debashia Bhattacharya, Prathima Agrawal, and Vishwani D. Agrawal. Delay Fault Test Generation for Scan/Hold Circuits using Boolean Expressions. In *29th Design Automation Conference*, pages 159–164. IEEE, 1992.

- [12] S. Bose, P. Agrawal, and V.D. Agrawal. Path Delay Fault Simulation of Sequential circuits. *Transactions on VLSI Systems*, 1(4):453–461, December 1993.
- [13] Soumitra Bose, Prathima Agrawal, and Vishwani D. Agrawal. Deriving Logic Systems for Path Delay Test Generation. *Transactions on Computers*, 47(8):829–846, 1998.
- [14] Michael L. Bushnell and Vishwani D. Agrawal. *Essentials of Electronic Testing*. Number ISBN 0-7923-799-1-8 in *Frontiers in Electronic Testing*. Kluwer Academic Publishers, 2000.
- [15] J.L. Carter, V.S. Iyengar, and B.K. Rosen. Efficient Test Coverage Determination for Delay Faults. In *International Test Conference*, pages 418–427. IEEE, September 1987.
- [16] Glenn R. Case. Analysis of actual fault mechanisms in CMOS logic gates. In *The proceedings of the thirteenth design automation conference on Design automation*, pages 265–270, 1976.
- [17] T.J. Chakraborty, V.D. Agrawal, and B.K. Rosen. Path Delay Fault Simulation Algorithms for Sequential Circuits. In *1st Asian Test Symposium*, pages 453–457, November 1992.
- [18] Jonathan T.-Y. Chang, Chao-Wen Tseng, Chien-Mo James Li, Mike Purtell, and Edward J. McCluskey. Analysis of Pattern-Dependent And Timing-Dependent Failures In an Experimental Test Chip. In *International Test Conference*, pages 184–193. IEEE, October 1998.
- [19] J.T.Y. Chang and E.J. McCluskey. Quantitative Analysis of Very-Low-Voltage Testing. In *VLSI Test Symposium*, pages 332–337. IEEE, 1996.
- [20] T.Y.J. Chang and E.J. McCluskey. SHOrt Voltage Elevation (SHOVE) Test for Weak ICs. In *VLSI Test Symposium*, pages 446–451. IEEE, 1997.
- [21] T.Y.J. Chang, C.W. Tseng, Y.C. Chu, S. Wattal, M. Purtell, and E.J. McCluskey. Experimental Results for IDDQ and VLV Testing. In *VLSI Test Symposium*, pages 118–123. IEEE, 1998.
- [22] Chih-Ang Chen and Sandeep K. Gupta. A Satisfiability-Based Test Generator for Path Delay Faults in Combinational Circuits. In *33rd Design Automation Conference*, pages 209–214. IEEE, 1996.
- [23] L.-C. Chen, S.K. Gupta, and M.A. Breuer. High Quality Robust Tests for Path Delay Faults. In *VLSI Test Symposium*, pages 88–93. IEEE, May 1997.

-
- [24] Wei-Yu Chen, Sandeep K. Gupta, and Melvin A. Breuer. Test Generation for Crosstalk Induced Delay in Integrated Circuits. In *International Test Conference*, pages 191–200. IEEE, September 1999.
- [25] K.-T. Cheng and H.-C. Chen. Classification and Identification of Nonrobust Untestable Path Delay Faults. *Transactions on Computer Aided Design of Integrated Circuits and Systems*, 15(8):845–853, August 1996.
- [26] K.-T. Cheng, S. Devadas, and K. Keutzer. Delay-Fault Test Generation and Synthesis for Testability Under a Standard Scan Design Methodology. *IEEE Transactions on Computer Aided Design of Integrated Circuits And Systems*, 12(8):1217–1231, August 1993.
- [27] K.-T. Cheng, A. Krstic, and H.-C. Chen. Generation of High Quality Tests for Robustly Untestable Path Delay Faults. *IEEE Transactions on Computers*, 45(12), pp. 1379-1392, December 1996.
- [28] Michael Cuviallo, Sujit Dey, Xiaoliang Bai, and Yi Zhao. Fault modeling and simulation for crosstalk in system-on-chip interconnects. In *Proceedings of the 1999 IEEE/ACM international conference on Computer-aided design*, pages 297–303. IEEE Press, 1999.
- [29] B.I. Dervisoglu and G.E. Strong. Design for Testability: Using Scanpath Techniques for Path-Delay Test and Measurement. In *International Test Conference*, pages 365–374. IEEE, October 1991.
- [30] Rolf Drechsler. BiTeS: A BDD based Test Pattern Generator for Strong Robust Path Delay Faults. In *ITG-GI-GMM Workshop Testmethoden und Zuverlässigkeit von Schaltungen und Systemen (Duisburg)*, pages 19–24, 1994.
- [31] D. Dumas, P. Girard, C. Landrault, and S. Pravossoudovitch. An Implicit Delay Fault Simulation Method with Detection Threshold Determination. In *International Test Conference*, pages 705–713. IEEE, 1993.
- [32] E.B. Eichelberger and T.W. Williams. A Logic Design Structure for LSI Testability. In *14th Design Automation Conference*, pages 462–468. IEEE, June 1977.
- [33] R.D. Eldred. Test Routines Based on Symbolic Logical Statements. *Journal of the ACM*, 6(1):33–36, January 1959.

- [34] Engl, Hörbst, Ruehli, Ohtsuki, Williams and Goto, and Hartenstein. *VLSI Testing*, volume 5 of *Advances in CAD for VLSI*. Elsevier Science Publishers B.V., 2nd edition, 1986. ISBN 0 444 87895 5.
- [35] Thomas Fiebig and Ralph Wagner. Test von integrierten Schaltungen durch Beobachtung des transienten Versorgungsstroms: Übersicht und praktische Ergebnisse. In *13. ITG/GMM/GI-Fachtagung 'Testmethoden und Zuverlässigkeit von Schaltungen und Systemen'*. VDE, Februar 2001.
- [36] F. Fink, K. Fuchs, and M.H. Schulz. An Efficient Parallel Pattern Gate Delay Fault Simulator With Accelerated Detected Fault Size Determination Capabilities. In *European Test Conference*, pages 171–180, 1991.
- [37] P. Franco, W.D. Farwell, R.L. Stokes, and E.J. McCluskey. An Experimental Chip to Evaluate Test Techniques: Chip And Experiment. In *International Test Conference*, pages 653–662. IEEE, October 1995.
- [38] P. Franco and E.J. McCluskey. Three-Pattern Tests for Delay Faults. In *VLSI Test Symposium*, pages 452–456. IEEE, April 1994.
- [39] Piero Franco, Siyad Ma, Jonathan Chang, Yi-Chin Chu, Sanjay Wittal, Edward J. McCluskey, Robert L. Stokes, and William D. Farwell. Analysis and Detection of Timing Failures in an Experimental Test Chip. In *International Test Conference*, pages 691–700, October 1996.
- [40] K. Fuchs, F. Fink, and M.H. Schulz. DYNAMITE: An Efficient Automatic Test Pattern Generation System For Delay Faults. *IEEE Transactions on CAD*, 10(10):1323–1335, October 1991.
- [41] Karl Fuchs, Michael Pabst, and Torsten Rössel. Resist: A recursive test pattern generation algorithm for path delay faults. *IEEE Trans. on CAD*, 13:1550–1561, December 1994.
- [42] H. Fujiwara and T. Shimono. On the Acceleration of Test Generation Algorithms. *IEEE Transactions on Computers*, C-32(12):1137–1144, December 1983.
- [43] J.M. Galey, R.E. Norby, and J.P. Roth. Techniques for the Diagnosis of Switching Circuits Failures. In *2nd Annual Symposium on Switching Circuit Theory and Logical Design*, pages 152–160. IEEE, October 1961.

-
- [44] M.A. Gharaybeh, M.L. Buschnell, and V.D. Agrawal. Classification and Test Generation for Path-Delay Faults Using Single Stuck-Faults Tests. In *International Test Conference*, pages 139–148. IEEE, 1995.
- [45] M.A. Gharaybeh, M.L. Bushnell, and V.D. Agrawal. Classification and Modelling of Path Delay Faults and Test Generation Using Single Stuck-Fault Tests. In *International Test Conference*, pages 139–148. IEEE, October 1995.
- [46] M.A. Gharaybeh, M.L. Bushnell, and V.D. Agrawal. An Exact Non-Enumerative Fault Simulator for Path-Delay Faults. In *International Test Conference*, pages 276–285. IEEE, October 1996.
- [47] P. Goel. An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits. *IEEE Transactions on Computers*, C-30(3):215–222, March 1981.
- [48] L.M. Goldstein and E.L. Thigan. SCOAP: Sandia Controlability and Observability Analysis Program. In *17th Conference on Design Automation*, pages 190–196. IEEE, 1980.
- [49] Joel Grodstein, Dilip Bhavsar, Vijay Bettada, and Richard Davies. Automatic Generation of Critical-Path Tests for a Partial-Scan Microprocessor. In *ICCD*. IEEE, October 2003.
- [50] H. Hao and E.J. McCluskey. Very-Low-Voltage Testing for Weak CMOS Logic IC's. In *International Test Conference*, pages 275–284. IEEE, 1993.
- [51] H. Hashempour and F. Lombardi. Detection of Crosstalk Faults in Deep Sub-Micron Chips Using Stuck-At Tests and Tools. In *12th IEEE North Atlantic Test Workshop*, pages 9–14. IEEE, May 2003.
- [52] Manfred Henftling and Hannes Wittmann. Bit Parallel Test Pattern Generation for Path Delay Faults. In *European Design and Test Conference*, pages 521–525, 1995.
- [53] K. Heragu, V.D. Agrawal, M.L. Bushnell, and J.H. Patel. Improving a Nonenumerative Method to Estimate Path Delay Fault Coverage. *Transactions on Computer-aided Design of Integrated Circuits and Systems*, 16(7):759–762, July 1997.
- [54] K. Heragu, J.H. Patel, and V.D. Agrawal. Segment Delay Faults: A New Fault model. In *14th VLSI Test Symposium*, pages 32–39. IEEE, May 1996.
- [55] C. Hora, W. Beverloo, M. Lousberg, and R. Segers. On Electrical Fault Diagnosis in Full-Scan Circuits. In *International Workshop on Defect Based Testing*, pages 17–22, 2001.

- [56] Camelia Hora. *On Diagnosing Faults in Digital Circuits*. PhD thesis, Technische Universitat Eindhoven, November 2002. ISBN 90-386-1990-1.
- [57] E.P. Hsieh, R.A. Rassmussen, L.J. Vidunas, and W.T. Davis. Delay Test Generation. In *Design Automation Conference*, pages 486–491, 1977.
- [58] Y.-H. Hsu and S.K. Gupta. A Simulator for At-Speed Robust Testing of Path Delay Faults in Combinational Circuits. *Transactions on Computers*, 45(11):1312–1318, November 1996.
- [59] J. L. A. Hughes and E. J. McCluskey. An Analysis of the Multiple Fault Detection Capabilities of single Stuck-At Fault Test Sets. In *International Test Conference*, pages 52–58. IEEE, 1984.
- [60] Hunter. The Statistical Dependence of Oxide Failure Rate on Vdd and tox Variation, with Applications of Process Design, Circuit Design and End Use. In *37th Annual International Reliability Physics Symposium*, pages 72–81, 1999.
- [61] Stanley L. Hurst. *VLSI Testing*. The Institution of Electrical Engineers, London, Michael Faraday House, Six Hills Way, Stevenage, Herts. SG1 2AY, United Kindom, 1998. ISBN 0 85296 901 5.
- [62] V.S. Iyengar, B.K. Rosen, and I. Spillinger. Delay Test Generation 1 - Concepts and Coverage Metrics. In *Proceedings of IEEE International Test Conference*, pages 857–866. IEEE, September 1988.
- [63] V.S. Iyengar, B.K. Rosen, and I. Spillinger. Delay Test Generation 2 - Algebra and Algorithms. In *Proceedings of IEEE International Test Conference*, pages 867–876. IEEE, September 1988.
- [64] Vinay B. Jayaram. Experimental Study of Scan-Based Transition Fault Testing Techniques. Master’s thesis, Bradley Department of Electrical and Computer Engeneering, Blacksburg, Virginia, January 2003.
- [65] Nirja Jha and Sandeep Gupta. *Testing of Digital Systems*. Cambridge University Press, 2003. ISBN 0 521 77356 3.
- [66] D. Kagaris, S. Tragoudas, and D. Karayiannis. Improved Non-Enumerative Path-Delay Fault-Coverage Estimation Based on Optimal Polynomial-Time Algorithms. *Transactions*

-
- on *Computer Aided Design of Integrated Circuits and Systems*, 16(3):309–315, March 1997.
- [67] B. Kapoor. An Efficient Method for Computing Exact Path Delay Coverage. In *European Design and Test Conference*, pages 516–520, March 1995.
- [68] W. Ke and P.R. Menon. Delay-Verifiability of Combinational Circuits based on Primitive Faults. In *International Conference on Computer Design*, pages 86–90. IEEE, October 1994.
- [69] Marinus Hendrik (Mario) Konijnenburg. *Automatic Test Pattern Generation for Synchronous Sequential Circuits*. PhD thesis, Universiteit Delft, December 1998. ISBN 90-9012096-3.
- [70] Angela Krstic, Yi-Min Jiang, and Kwang Ting Cheng. Pattern Generation for Delay Testing and Dynamic Timing Analysis Considering Power-Supply Noise Effects. *IEEE Transactions on CAD of IC's and Systems*, 20(3):416–425, March 2001.
- [71] Angela Krstic and Kwang-Ting Chen. *Delay Fault Testing in VLSI Circuits*. Number 14 in Frontiers of Electronic Testing. Boston Kluwer Academic Publishers, 1998. ISBN 0-7923-8295-1.
- [72] W. Kunz and D.K. Pradhan. Recursive Learning - an Attractive Alternative to the Decision Tree for Test Generation in Digital Circuits. In *International Test Conference*, pages 816–825. IEEE, September 1992.
- [73] W.K. Lam, A. Saldanha, R.K. Brayton, and A.L. Sangiovanni-Vicentelli. Delay Fault Coverage, Test Set Size and Performance Trade-Offs. *Transactions on Computers*, C-29(C):235–248, March 1995.
- [74] Y. Leventel and P.R. Menon. Transition Faults in Combinational Circuits. In *International Fault Tolerant Computing Symposium*, pages 278–283. IEEE, July 1986.
- [75] M.W. Levi. CMOS is Most Testable. In *Proc. of the International Test Conference*, pages 217–220. IEEE, October 1981. Vorstellung des iddq-Testes.
- [76] C.-M.J. Li and E.J. McCluskey. IDDQ Data Analysis Using Current Signature. In *Int. Workshop on IDDQ Testing*, pages 37–42. IEEE, November 1998.

- [77] James C.M. Li, Jonathan T.-Y. Chang, Chao-Wen Tseng, and Edward J. McCluskey. ELF35 Experiment - Chip and Experiment Data. Technical Report TR-99/3, Center for Reliable Computing, Stanford University, Gates Bldg. 2A Room 236, 1999.
- [78] C.J. Lin and S.M. Reddy. On Delay Fault Testing in Logic Circuits. *IEEE Transactions on CAD of Integrated Circuits and Systems*, CAD-6(5):694–703, September 1987.
- [79] Siyad C. Ma, Piero Franco, and Edward J. McCluskey. An Experimental Chip to Evaluate Test Techniques - Experiment Results. In *International Test Conference*. IEEE, October 1995.
- [80] Udo Mahlstedt. *Deterministische Testgenerierung für Gatterverzögerungsfehler unter Berücksichtigung der minimal erkennbaren Fehlergröße*. PhD thesis, Universität Hannover, August 1995.
- [81] A.K. Majhi, G. Gronthoud, C. Hora, M. Lousberg, P. Valer, and S. Eichenberger. Improving Diagnostic Resolution of Delay Faults using Path Delay Fault Model . In *21st VLSI Test Symposium*. IEEE, April 2003.
- [82] A.K. Majhi, J. Jacob, L.M. Patnaik, and V.D. Agrawal. On Test Coverage of Path Delay Faults. In *Proc. 9th International Conference on VLSI Design*, pages 418–421, January 1996.
- [83] Ananta Majhi, Guido Gronthoud, Volker Meyer, Arne Sticht, and Juergen Schloeffel. Improving Diagnostic Resolution of Delay Faults using Path Delay Fault Model . In *15. ITG/GMM/GI Workshop Testmethoden und Zuverlässigkeit von Schaltungen und Systemen*, page 50. VDE, März 2003.
- [84] Y.K. Malaiya and R. Rajusman. *Bridging Faults and IDDQ Testing*. IEEE Computer Society Press, Los Alamitos, California, 1992.
- [85] Peter Maxwell, Ismed Hartanto, and Lee Bentz. Comparing Functional and Structural Tests. In *International Test Conference*, pages 400–407. IEEE, 2000.
- [86] Peter C. Maxwell, Robert C. Aitken, Kathleen R. Kollitz, and Allen C. Brown. IDDQ and AC Scan: The War Against Unmodeled Defects. In *International Test Conference*, pages 250–258. IEEE, IEEE, 1996.

-
- [87] Edward J. McCluskey and Chao-Wen Tseng. Stuck-Fault Tests vs. Actual Defects. In *International Test Conference*, pages 336–343. IEEE, October 2000. Zusammenfassung der Ergebnisse des Murphy-Chip Projektes.
- [88] K.C.Y. Mei. Bridging and Stuck-At Faults. *IEEE Transactions on Computers*, C-23(7):720–727, July 1974.
- [89] Volker H.-W. Meyer, Ajoy Palit, Walter Anheier, Arne Sticht, and Jürgen Schlöffel. Can Signal Integrity Faults Be Detected By Delay Tests ? In *12th IEEE North Atlantic Test Workshop*, pages 131–136. IEEE, May 2003.
- [90] misc. International Technology Roadmap for Semiconductors. <http://public.itrs.net/>, 2001.
- [91] misc. *Introduction to IC Technology*. 2003. <http://www.icknowledge.com>.
- [92] Will Moore, Guido Gronthoud, Keith Baker, and Maurice Lousberg. Delay-Fault Testing and Defects in Deep Sub-Micron ICs - Does Critical Resistance Really Mean Anything ? In *International Test Conference*, pages 95–104. IEEE, October 2000.
- [93] Phil Nigh and Anne E. Gattiker. Test Method Evaluation Experiments & Data. In *Proc. of IEEE International Test Conference*, pages 454–463, 2000.
- [94] Phil Nigh, Wayne Needham, Ken Butler, Peter Maxwell, Rob Aitken, and Wojchech Maly. So What IS an Optimal Test Mix? A Discussion of the SEMATECH Methods Experiment. In *International Test Conference*, pages 1037–1038. IEEE, October 1997.
- [95] M. Nourani and A. Attarha. Built-In Self Test For Signal Integrity. In *Design Automation Conference*, pages 792–797. IEEE, June 2001.
- [96] B.G. Oomman and S.B. Akers. Fault Simulation For Delay Faults. In *European Test Conference*, pages 328–335, 1989.
- [97] E.S. Park and M.R. Mercer. An efficient Delay Test Generation System for Combinational Logic. In *27th Design Automation Conference*, pages 522–528. IEEE, June 1990.
- [98] E.S. Park and M.R. Mercer. An Efficient Delay Test Generation System for Combinational Logic. *Transactions on Computer -Aided Design*, CAD-11(7):926–938, July 1992.
- [99] S. Patil and S.M. Reddy. A Test Generation System for Path Delay Faults. *International Conference on Computer Design*, pages 40–43, October 1989.

- [100] F. Pöhl and W. Anheier. Quality Determination for Gate Delay Fault Tests Considering Three-State Elements. *Journal of Electronic Testing - Theory and Application*, 14:47–53, 1999.
- [101] F. Pöhl, V. Meyer, and W. Anheier. ATPG für Verzögerungsfehler in Schaltungen mit konventionellem Prüfpfad unter Berücksichtigung von Tri-State-Gattern. In *11. ITG/GMM-Workshop "Testmethoden und Zuverlässigkeit von Schaltungen und Systemen"*(Potsdam), pages 40–43, Februar 1999.
- [102] Frank Pöhl. *Simulation von Verzögerungsfehlern in hochintegrierten Digitalschaltungen mit Tristate-Elementen*, volume 311 of *Fortschritts-Berichte, VDI Reihe 20*. VDI Verlag, Düsseldorf, isbn 3-18-331120-8 / issn 0178-9473 edition, November 2000. Dissertation Universität Bremen.
- [103] A. Pierzynska and S. Pilarski. Non-Robust versus Robust. In *International Test Conference*, pages 123–131, October 1995.
- [104] A. Pierzynska and S. Pilarski. Pitfalls in Delay Fault Testing. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(3):321–329, March 1997.
- [105] Alicja Pierzynska and Slawomir Pilarski. Quality Considerations in Delay Fault Testing. In *EURO-DAC*, pages 196–201. IEEE, September 1995.
- [106] F. Poehl, V. Meyer, and W. Anheier. Delay Fault Test Generation for Circuits with Standard Scan Design Considering Three-State Elements. In *IEEE International Test Synthesis Workshop*, 1999.
- [107] Fred Pollack. New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies. <http://www.intel.com/research/mrl/Library/micro32Keynote.pdf>, November 1999. Micro32 Conference, Haifa Israel.
- [108] I. Pomeranz and S.M. Reddy. An Efficient Nonenumerative Method to Estimate the Path Delay Fault Coverage in Combinational Circuits. *Transactions on Computer Aided Design of Integrated Circuits and Systems*, 13(2):240–250, February 1994.
- [109] I. Pomeranz and S.M. Reddy. SPADES-ACE: A Simulator for Path delay Faults in Sequential circuits with Extensions to Arbitrary Clocking Schemes. *Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(2):251–263, February 1994.

-
- [110] Irith Pomeranz and Sudhakar M. Reddy. On n-Detection Test Sets and Variable-n-Detection Test Sets for Transition Faults. In *Transactions On Computer Aided Design*, pages 372–383, March 2000.
- [111] A.K. Pramanick and S.M. Reddy. On the Detection of Delay Faults. In *International Test Conference*, pages 845–856, 1988.
- [112] A.K. Pramanik and S.M. Reddy. On the Fault Coverage of Gate Delay Fault Detecting Tests. *IEEE Transactions on CAD of ICs and Systems*, 16(1):78–94, January 1997.
- [113] Janusz Rajski, Jerzy Tyszer, Nilanjan Mukherjee, Rob Thompson, Kun-Han Tsai, Andre Hertwig, Nagesh Tamarapalli, Gregorz Mrugalski, and Jun Qian. Embedded Deterministic Test For Low Cost Manufacturing Test. In *International Test Conference*, pages 301–310. IEEE, October 2002.
- [114] I.M. Ratiu, A. Sangiovanni-Vicentelli, and D.O. Peterson. VICTOR: A Fast VLSI Testability Analysis Programme. In *International Conference on Test*, pages 397–401. IEEE, 1982.
- [115] J. Rearick. Too Much Delay Fault Coverage Is a Bad Thing. In *International Test Conference*, pages 624–633. IEEE, 2001.
- [116] S.M. Reddy, C.J. Lin, and S. Patil. An Automatic Test Pattern Generator for the Detection of Path Delay Faults. In *International Conference on Computer-Aided Design*, pages 284–287. IEEE/ACM, November 1987.
- [117] J.P. Roth. Diagnosis of Automata Failure: A Calculus and a Method. *IBM Journal of Research and Development*, 10:278–291, October 1966.
- [118] A. Saldanha, R.K. Brayton, and A.L. Sangiovanni-Vicentelli. Equivalence of Robust Delay-Fault and Single Stuck-At Fault Test Generation. In *9th Design Automation Conference*, pages 173–176, June 1992.
- [119] J. Savir. Skewed-Load Transition Test: Part I, Calculus. In *International Test Conference*, pages 705–713. IEEE, October 1992.
- [120] J. Savir. Skewed-Load Transition Test: Part II, Coverage. In *International Test Conference*, pages 714–722. IEEE, October 1992.

- [121] Jacob Savir and Srivinas Patil. On Broad-Side Delay Testing. In *VLSI Test Symposium*, pages 284–290. IEEE, April 1994.
- [122] M.H. Schulz and F. Brglez. Accelerated Transition Fault Simulation. In *Proceedings of 26th Design Automation Conference*, pages 237–243, June 1987.
- [123] M.H. Schulz, E. Trischler, and T.M. Serfert. SOCRATES: A Highly Efficient Automatic Test Pattern Generation System. *IEEE Transactions on Computer-Aided Design, CAD-7(1)*:126–137, January 1988.
- [124] G.L. Smith. Model For Delay Faults Based Upon Paths. In *International Test Conference*, pages 152–158. IEEE, 1985.
- [125] S.T. Su, R.Z. Makki, and T. Nagle. Transient Power Supply Current Monitoring - A New Test Method for CMOS VLSI Circuits. *Journal of Electronic Testing: Theory and Application*, 6(1):23–44, February 1995.
- [126] Nandu Tendolkar, Rajesh Raina, and Rick Woltenberg. Novel Techniques for Achieving High At-Speed Transition Fault Test Coverage for Motorola’s Microprocessor Based on Power-PC (TM) Instruction Set Architecture. In *VLSI Test Symposium*, April 2002.
- [127] Ken Tumin, Carmen Vargas, Ross Patterson, and Chris Nappi. Scan vs. Funtional Testing - A Comparative Effectiveness Study on Motorola’s MMC2107(TM). In *International Test Confrence*, pages 443–450. IEEE, 2001.
- [128] B. Underwood, W. Law, S. Kang, and H. Konuk. Fastpath: A Path-Delay Test Generator for Standard Scan Designs. In *International Test Conference*, pages 154–163. IEEE, 1994.
- [129] Hans van der Linden. *Automatic Test Pattern Generation for Three-State Circuits*. Thesis Technische Universiteit Delft, 1996. ISBN 90-9009585-3.
- [130] Prab Varma. On Path Delay Testing in a Standard Scan Environment. In *International Test Conference*, pages 164–173. IEEE, October 1994.
- [131] Harry J.M. Veendrick. *Deep Submicron CMOS ICs*. Kluwer Bedrijfsinformatie b.v., 1st english edition, 1998. ISBN 90 557 612 81.
- [132] versch. Autoren. 2. Dresdner Sommerschule Mikroelektronik. Schulungsunterlagen, September 2001.

-
- [133] R.L. Wadsack. Fault Modeling and Logic Simulation of CMOS and MOS Integrated Circuits. *Bell Systems Technical Journal*, 57(5):1449–1474, JuneMay/June 1978.
- [134] J.A. Waicukauski, E. Lindbloom, B. Rosen, and V. Iyengar. Transition Fault Simulation. *IEEE Design and Test of Computers*, 4(2):32–38, April 1987.
- [135] J.A. Waicukauski, E. Lindbloom, B.K. Rosen, and V. Iyengar. Transition Fault Simulation by Parallel Pattern Single Fault Propagation. In *International Test Conference*, pages 542–549. IEEE, 1986.
- [136] M.J.Y. Williams and J.B. Angel. Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic. *IEEE Transactions on Computers*, C-22(1):46–60, January 1973.
- [137] H. Wittmann and M. Henftling. Efficient Path Identification for Delay Testing - Time and Space Optimization. In *European Test Conference*, pages 513–517. IEEE, March 1994.
- [138] H. Wittmann and M. Henftling. Path Delay ATPG for Standard Scan Design. In *EURO-DAC with EUROVHDL*, pages 202–207, 1995.
- [139] Hans-Joachim Wunderlich. *Hochintegrierte Schaltungen: Prüfunggerechter Entwurf und Test*. Springer-Verlag Heidelberg-Berlin, 1991. ISBN 3-540-53456-3.
- [140] Brian Young. *Digital Signal Integrity: Modeling and Simulation with Interconnects and Packages*. Prentice Hall Modern Semiconductor Design Series. Prentice-Hall PTR, 2000. ISBN 0-13-028904-3.

Index

- Übergangsfehlermodell, 16
- BDD (binary decision diagrams), 30
- boundlines, 29
- Brückenfehlermodell, 14
- bridging fault model, 14
- broadside test, 31
- burn-in Test, 102
- burn-in test, 45
- CMOS, 39
- D-Algorithmus, 26
- D-Flip-Flop, 31
- Defekt, 7, 39
- Diagnose, 9
- Dielektrikum, zeitabhängiger Durchbruch, 103
- dynamische Fehlereffekte, 3
- DYNAMITE, 63
- Elektromigration, 41, 101
- enhanced scan Flip-Flop, 31, 91
- FAN, 27, 63, 69
- FASTPATH, 63
- Fehler, 7
- fehlerleitende Belegung, 24
- Fehlermodell, 7
- Fehlersimulation, enumerativ, 51
- Fehlersimulator, 22
- freelines, 28
- full-scan design, 30
- functional justification, 31
- funktionale Einstellung, 31, 36, 60, 74, 91
- gate delay fault model, 16, 42
- Gatterverzögerungsfehlermodell, 16
- groß delay fault model, 16
- Haftfehlermodell, 5, 13, 42
- headlines, 28
- hot carrier effect, 103
- IDDQ, 20
- Implikation, 63
- Implikation, global, 29
- Implikation, lokal, 29
- inter timefarme backtracking, 60
- Kontrapositionsgesetz, 29
- kontrollierender Eingangswert, 23
- kritischer Pfad, 23
- launch-from-capture, 31
- launch-from-shift, 31
- Leitungsverzögerungsfehlermodell, 17
- lernen (rekursiv), 29
- line delay fault model, 17
- low-cost Testautomat, 31
- Maximal aggressor model, 105
- multiple backtrace, 28
- Murphy Projekt, 43
- nicht robuster Test, 25
- partial-scan design, 30
- path delay fault model, 17, 42
- Pfad, nicht funktionaler, 76
- Pfadeingang, 23

- Pfadverzögerungsfehlermodell, 3, 17
- pin faults, 14
- Planarisierung, 40
- PODEM, 27, 63
- Produktionstest, 3, 34, 39, 51, 65
- pseudo primärer Ausgang, 10
- pseudo primärer Eingang, 10

- ramp up phase, 97
- RESIST, 63
- robuster Test, 24

- Scan-Kette, 10, 31, 36
- scan-shift Modus, 31, 32, 36, 47, 60
- Schaltung, kombinatorisch, 9
- Schaltung, sequentiell, 9
- segment delay fault model, 19
- Segmentverzögerungsfehlermodell, 19
- Seiteneingang, 23
- Signallaufzeiten, 21, 41
- skewed load, 31
- standard scan design, 3
- Stromdichte, 101
- struktureller Pfad, 64
- stuck-at fault model, 13
- stuck-on fault model, 13
- stuck-open fault model, 13, 42

- Test, 8
- Test, defektorientiert, 7
- Test, funktionaler, 7
- Test, robuster, 64
- Test, struktureller, 7
- Testmuster, 11
- Testvektor, 11
- Testwertsatz, 11

- Timinganalysator, 21
- transition delay fault model, 16, 42

- veqtor chip, 97
- Verzögerungsfehler, 15, 25
- Via, 21, 40

Anhang

Tabelle 10: Entwicklung der Mikroelektronik am Beispiel der Intel Prozessorfamilie [91]

Year	Product	Process type	Line width (μm)	Transistors (M)	Masks layer	Cell Construction				Layers		Die size (mm)
						Proc/bus (bits)	Clock (MHz)	Voltage (V)	Cache (Kbits)	Poly	Metal	
1971	4004	PMOS	10	0.0023	—	4	0.108	12	0	1	1	13.5
1972	8008	PMOS	10	0.0035	—	8	0.2	12	0	1	1	15.2
1974	8080	NMOS	6.0	0.006	—	8	2	12	0	1	1	20.0
1976	8085	NMOS	3.0	0.0065	—	8	0.37	5	0	1	1	20.0
1978	8086	NMOS	3.0	0.029	—	16	5-10	5	0	1	1	28.6
1979	8088	NMOS	3.0	0.029	—	16/8	5-8	5	0	1	1	28.6
1982	80286	CMOS	1.5	0.134	—	16	6-12	5	0	1	2	68.7
1985	80386DX	CMOS	1.5	0.275	10	32	16-33	5	0	1	2	104
1989	80486DX	CMOS	1.0	1.2	12	32	25-50	5	0	1	3	163
1992	80486DX2	CMOS	0.8	1.2	—	32	50-66	5	0	1	3	81
1993	Pentium	BiCMOS	0.8	3.1	18	32/64	60-66	5	0	1	3Al	264
1994	80486DX4	CMOS	0.5	1.6	—	32	75-100	5	0	1	3Al	—
1995	Pentium Pro	BiCMOS	0.35	5.5	20	32/64	150-200	3.3	0	1	4Al	310
1997	Pentium II	CMOS	0.35	7.5	16	32/64	233-300	2.8	0	1	4Al	209
1998	Celeron	CMOS	0.25	19	19	32/64	300-333	—	128	1	5Al	—
1999	Pentium III	CMOS	0.18	28	21	32/64	500-733	1.65	256	1	6Al	140
2000	Pentium 4	CMOS	0.18	42	21	32/64	1400 — 2000	1.7	256	1	6Al	224
2001	Pentium 4	CMOS	0.13	55	23	32/64	2000 — 2200	1.5	512	1	6Cu	146
2001	Itanium	CMOS	0.18	25	21	64/64	733-800	—	96	1	6Al	—
2002	Pentium 4	CMOS	0.13	55	23	32/64	2000 — 3000	1.5	512	1	6Cu	131
2002	Itanium	CMOS	0.13	220	23	64/64	900 — 1000	—	256/1500	1	6Al	421
2003	Pentium 4	CMOS	0.09	>55	25	32/64	>3000	1.2	>512	1	7Cu	—

Tabelle 12: Messwerte verschiedener ISCAS89-Benchmarkschaltungen

Name	# Fehler	robust	[%]	nicht robust	[%]	Summe	[%]
s208	290	13	4,5	121	41,7	134	46,2
s386	414	76	18,4	191	46,1	267	64,5
s298	462	59	12,8	177	38,3	236	51,8
s344	710	47	6,6	487	68,6	534	75,2
s349	730	86	11,8	450	61,6	536	73,4
s420	738	17	2,3	333	45,1	350	47,4
s510	738	115	15,6	350	47,4	465	63,0
s382	800	119	14,9	285	35,6	404	50,5
s526	820	85	10,4	215	26,2	300	36,6
s400	896	113	12,6	302	33,7	415	46,3
s820	984	139	14,1	592	60,2	731	74,3
s832	1012	135	13,3	598	59,1	733	72,4
s444	1070	81	7,6	355	33,2	436	40,7
s1488	1924	133	6,9	1453	75,5	1586	82,4
s1494	1952	135	6,9	1457	74,6	1592	81,6
s838	2018	25	1,2	1141	56,5	1166	57,8
s967a	2142	323	15,1	1157	54,0	1480	69,1
s953	2312	339	14,7	1201	51,9	1540	66,6
s499a	2948	128	4,3	172	5,8	300	10,2
s938a	3428	128	3,7	1378	40,2	1506	58,0
s641	3488	152	4,4	1869	53,6	2021	58,0
s1196	6196	26	0,4	3669	59,2	3695	59,6
s1512a	6972	283	4,1	2102	30,1	2385	34,2
s1238	7118	30	0,4	3592	50,5	3622	50,9
s3330a	9530	503	5,3	7519	78,9	8022	84,2
s991a	14920	52	0,3	10538	70,6	10590	71,0
c880	17284	53	0,3	16599	96,0	16652	96,3
s5378	27084	373	1,4	18563	68,5	18936	69,9
p8052	27648	1226	4,4	10986	39,7	12212	44,2
s3271a	38388	233	0,6	19056	49,6	19289	50,2
s3384a	39582	264	0,7	31570	79,8	31834	80,5
s713	43624	123	0,3	4270	9,8	4393	10,1
s1269a	79140	44	0,1	24228	30,6	24272	30,7
s35932	394282	2280	0,6	47506	12,0	49786	12,6
s9234	489708	1066	0,2	32238	6,6	33304	6,8
c2670	1359920	128728	9,5	226	0,0	128954	9,5
c7552	1452988	138	0,0	275734	19,0	275872	19,0
c1908	1458114	0	0,0	353222	24,2	353222	24,2

Tabelle 13: Ergebnisse für ISCAS89-Benchmarkschaltungen bei Testmustererzeugung mit Pfadinitialisierung

Name	# Fehler	robust	[%]	nicht robust	[%]	Summe	[%]
s208	290	27	9,3	53	18,3	80	27.6
s386	414	106	25,6	47	11,4	153	37.0
s298	462	87	18,8	40	8,7	127	27.5
s344	710	138	19,4	121	17,0	259	36.4
s349	730	196	26,8	63	8,6	259	35.4
s420	738	39	5,3	169	22,9	208	28.2
s510	738	138	18,7	59	8,0	197	26.7
s382	800	140	17,5	25	3,1	165	20.6
s526	820	114	13,9	33	4,0	147	17.9
s400	896	131	14,6	35	3,9	166	18.5
s820	984	211	21,4	158	16,1	369	37.5
s832	1012	205	20,3	164	16,2	369	36.5
s444	1070	128	12,0	38	3,6	166	15.6
s1488	1924	250	13,0	472	24,5	722	37.5
s1494	1952	256	13,1	467	23,9	723	37.0
s838	2018	63	3,1	593	29,4	656	32.5
s967a	2142	517	24,1	401	18,7	918	42.8
s953	2312	516	22,3	445	19,2	961	41.5
s499a	2948	128	4,3	44	1,5	172	5.8
s938a	3428	163	4,8	593	17,3	756	22.1
s641	3488	627	18,0	474	13,6	1101	31.6
s1196	6196	1289	20,8	2117	34,2	3406	55.0
s1512a	6972	721	10,3	399	5,7	1120	16.0
s1238	7118	1250	17,6	2115	29,7	3365	47.3
s3330a	9530	2135	22,4	3938	41,3	6073	63.7
s991a	14920	72	0,5	2828	19,0	2900	19.5
c880	17284	3484	20,2	13005	75,2	16489	95.4
s5378	27084	3946	14,6	13021	48,1	16967	62.7
s3271a	38388	1053	2,7	2234	5,8	3287	8.5
s3384a	39582	2846	7,2	10947	27,7	13793	34.9
s1269a	79140	1352	1,7	12742	16,1	14094	17.8
s35932	394282	10732	2,7	13489	3,4	24221	6.1
s9234	489708	4277	0,9	9986	2,0	14263	2.9
c2670	1359920	5866	0,4	27405	2,0	33271	2.4
c7552	1452988	22239	1,5	169886	11,7	192125	13.2
c1908	1458114	67958	4,7	253112	17,4	321070	22.1

Tabelle 14: Ergebnisse der Berechnungen für ISCAS89-Benchmarkschaltungen bei Forderung eines *ncv* an allen Seiteneingängen in beiden Taktzyklen

Name	# Fehler	robust	[%]	nicht robust	[%]	Summe	[%]
s208	290	26	9,0	6	2,1	32	11.1
s386	414	75	18,1	23	5,6	98	23.7
s298	462	76	16,5	23	5,0	99	21.5
s344	710	112	15,8	97	13,7	209	29.5
s349	730	169	23,2	40	5,5	209	28.7
s420	738	36	4,9	6	0,8	42	5.7
s510	738	110	14,9	8	1,1	118	16.0
s382	800	136	17,0	15	1,9	151	18.9
s526	820	111	13,5	21	2,6	132	16.1
s400	896	128	14,3	16	1,8	144	16.1
s820	984	194	19,7	64	6,5	258	26.2
s832	1012	194	19,2	64	6,3	258	25.5
s444	1070	125	11,7	19	1,8	144	13.5
s1488	1924	186	9,7	135	7,0	321	16.7
s1494	1952	185	9,5	140	7,2	325	16.7
s838	2018	56	2,8	6	0,3	62	3.1
s967a	2142	575	26,8	143	6,7	718	33.5
s953	2312	592	25,6	160	6,9	752	32.5
s499a	2948	132	4,5	0	0	132	4.5
s938a	3428	163	4,8	34	1,0	197	5.8
s641	3488	534	15,3	168	4,8	702	20.1
s1196	6196	1998	32,2	803	13,0	2801	45.2
s1512a	6972	506	7,3	172	2,5	678	9.8
s1238	7118	1817	25,5	662	9,3	2479	34.8
s3330a	9530	2028	21,3	2639	27,7	4667	49.0
s991a	14920	72	0,5	189	1,3	261	1.8
c880	17284	4259	24,6	11761	68,0	16020	92.6
s5378	27084	5744	21,2	10014	37,0	15758	38.2
s3271a	38388	1367	3,6	1033	2,7	2400	6.3
s3384a	39582	2833	7,2	6142	15,5	8975	22.7
s1269a	79140	1306	1,7	6636	8,4	7942	10.1
s35932	394282	11604	2,9	9737	2,5	21341	5.4
s9234	489708	5150	1,1	5484	1,1	10634	2.2
c2670	1359920	11507	0,8	19667	1,4	31174	2.2
c7552	1452988	40506	2,8	93726	6,5	134232	9.3
c1908	1458114	59314	4,1	168809	11,6	228123	15.7

Tabelle 15: Verschiebungen zwischen den Mengen der Testmusterqualitäten für Testmuster der Modi 1 und 2

Modus 1 Modus 2	robust robust	robust nicht robust	robust nicht getestet	nicht robust robust	nicht robust nicht robust	nicht robust nicht getestet
s208	6	0	21	20	6	27
s386	55	15	36	20	8	19
s298	57	10	20	19	13	8
s344	98	1	39	14	96	11
s349	144	10	42	25	30	8
s420	5	0	34	31	6	132
s510	73	2	63	37	6	16
s382	119	7	14	17	8	0
s526	88	14	12	23	7	3
s400	105	4	22	23	12	0
s820	114	35	62	80	29	49
s832	110	31	64	84	33	47
s444	100	6	22	24	14	0
s1488	66	50	134	120	85	267
s1494	75	53	128	110	87	270
s838	3	0	60	53	6	534
s967a	311	72	134	264	71	66
s953	311	69	136	281	91	73
s499a	102	0	26	30	0	14
s938a	67	0	96	96	34	463
s1196	734	302	253	1264	501	352
s1512a	333	108	280	173	64	162
s1238	663	245	342	1154	417	544
s3330a	871	815	449	1157	1824	957
s991a	0	0	72	72	189	2567
c880	1151	2275	58	3108	9486	411
s5378	1429	1938	579	4315	8076	629
s3271a	817	68	168	550	965	719
s3384a	839	1516	491	1994	4626	4327
s1269a	113	699	540	1193	5939	5612
s35932	5951	3362	1419	5653	6375	1461
s9234	31	198	32	180	1400	1200

Tabelle 16: Ergebnisse der kompakten Testmuster-generierung in Modus 0

Name	# Fehler	robust	%	nicht robust	%	# Test-muster	Summe	Kompressions-faktor
s208	290	13	4,5	121	41,7	46	134	2,91
s386	414	79	19,1	188	45,4	72	267	3,71
s298	462	60	13,0	176	38,1	35	236	6,74
s344	710	62	8,7	472	66,5	104	534	5,13
s349	730	81	11,1	455	62,3	98	536	5,47
s420	738	17	2,3	333	45,1	130	350	2,69
s510	738	102	13,8	363	49,2	103	465	4,51
s382	800	96	12,0	308	38,5	66	404	6,12
s526	820	81	9,9	219	26,7	64	300	4,69
s400	896	93	10,4	322	35,9	72	415	5,76
s820	984	141	14,3	590	60,0	198	731	3,69
s832	1012	137	13,5	596	58,9	200	733	3,67
s444	1070	89	8,3	347	32,4	65	436	6,71
s1488	1924	153	8,0	1433	74,5	211	1586	7,52
s1494	1952	166	8,5	1426	73,1	226	1592	7,04
s838	2018	25	1,2	1141	56,5	490	1166	2,38
s967a	2142	341	15,9	1139	53,2	457	1480	3,24
s953	2312	365	15,8	1175	50,8	455	1540	3,38
s499a	2948	126	4,3	174	5,9	102	300	2,94
s938a	3428	128	3,7	1378	40,2	759	1506	1,98
s641	3488	177	5,1	1867	53,5	454	2044	4,50
s1196	6196	89	1,4	3606	58,2	1223	3695	3,02
s1512a	6972	314	4,5	2071	29,7	348	2385	6,85
s1238	7118	103	1,4	3519	49,4	1241	3622	2,92
s3330a	9530	626	6,6	7431	78,0	413	8057	19,51
s991a	14920	53	0,4	10537	70,6	1307	10590	8,10
c880	17284	53	0,3	16599	96,0	5392	16652	3,09
s5378	27084	664	2,5	18272	67,5	822	18936	23,04
s3271a	38388	203	0,5	18493	48,2	1077	18696	17,36
s3384a	39582	588	1,5	31323	79,1	3552	31911	8,98
s1269a	79140	98	0,1	24247	30,6	6197	24345	3,93
s35932	394282	4875	1,2	44911	11,4	4243	49786	11,73

Tabelle 17: Ergebnisse der kompakten Testmuster-generierung in Modus 1

Name	# Fehler	robust	%	nicht robust	%	# Test-muster	Summe	Kompressions-faktor
s208	290	26	8,97	54	18,62	29	80	2,76
s386	414	115	27,78	38	9,18	51	153	3,00
s298	462	103	22,29	24	5,19	22	127	5,77
s344	710	148	20,85	111	15,63	71	259	3,65
s349	730	192	26,30	67	9,18	68	259	3,81
s420	738	38	5,15	170	23,04	53	208	3,92
s510	738	132	17,89	65	8,81	75	197	2,63
s382	800	147	18,38	18	2,25	49	165	3,37
s526	820	120	14,63	27	3,29	43	147	3,42
s400	896	137	15,29	29	3,24	55	166	3,02
s820	984	260	26,42	109	11,08	120	369	3,08
s832	1012	256	25,30	113	11,17	126	369	2,93
s444	1070	125	11,68	41	3,83	49	166	3,39
s1488	1924	293	15,23	429	22,30	158	722	4,57
s1494	1952	303	15,52	420	21,52	155	723	4,66
s838	2018	62	3,07	594	29,44	101	656	6,50
s967a	2142	551	25,72	367	17,13	302	918	3,04
s953	2312	567	24,52	394	17,04	298	961	3,22
s499a	2948	126	4,27	46	1,56	63	172	2,73
s938a	3428	162	4,73	594	17,33	180	756	4,20
s641	3488	570	16,34	551	15,80	191	1121	5,87
s1196	6196	1577	25,45	1829	29,52	1415	3406	2,41
s1512a	6972	782	11,22	338	4,85	205	1120	5,46
s1238	7118	1521	21,37	1844	25,91	1456	3365	2,31
s3330a	9530	2135	22,40	3938	41,32	452	6073	13,44
s991a	14920	72	0,48	2828	18,95	575	2900	5,04
c880	17284	2509	14,52	13980	80,88	6204	16489	2,66
s5378	27084	4455	16,45	12512	46,20	1816	16967	9,34
s3271a	38388	1053	2,74	2234	5,82	331	3287	9,93
s3384a	39582	2846	7,19	10947	27,66	2742	13793	5,03
s1269a	79140	1352	1,71	12742	16,10	5453	14094	2,58
s35932	394282	11291	2,86	12930	3,28	1400	24221	17,30

Tabelle 18: Ergebnisse der kompakten Testmuster generierung in Modus 2

Name	# Fehler	robust	%	nicht robust	%	# Testmuster	Summe	Kompressionsfaktor
s208	290	25	8,6	7	2,4	18	32	1,78
s386	414	72	17,4	26	6,3	46	98	2,13
s298	462	86	18,6	13	2,8	23	99	4,30
s344	710	132	18,6	77	10,8	55	209	3,80
s349	730	174	23,8	35		55	209	3,80
s420	738	35	4,7	7	0,9	26	42	1,62
s510	738	109	14,8	9	1,2	54	118	2,19
s382	800	142	17,8	9	1,1	55	151	2,75
s526	820	120	14,6	12	1,5	43	132	3,07
s400	896	134	15,0	10	1,1	58	144	2,48
s820	984	201	20,4	57	5,8	116	258	2,22
s832	1012	201	19,9	57	5,6	116	258	2,22
s444	1070	124	11,6	20	1,9	50	144	2,88
s1488	1924	218	11,3	103	5,4	110	321	2,92
s1494	1952	214	11,0	111	5,7	114	325	2,85
s838	2018	55	2,7	7	0,3	42	62	1,48
s967a	2142	614	28,7	104	4,9	284	718	2,53
s953	2312	629	27,2	123	5,3	307	752	2,45
s499a	2948	132	4,5	0	0,0	86	132	1,53
s938a	3428	163	4,8	34	1,0	127	197	1,55
s641	3488	554	15,9	152	4,4	178	706	3,97
s1196	6196	2076	33,5	725	11,7	1280	2801	2,19
s1512a	6972	544	7,8	134	1,9	151	678	4,49
s1238	7118	1877	26,4	602	8,5	1185	2479	2,09
s3330a	9530	2622	27,5	2060	21,6	591	4682	7,92
s991a	14920	72	0,5	189	1,3	193	261	1,35
c880	17284	3104	18,0	12916	74,7	7529	16020	2,13
s5378	27084	5747	21,2	10011	37,0	2485	15758	6,34
s3271a	38388	1584	4,1	816	2,1	419	2400	5,73
s3384a	39582	2971	7,5	6004	15,2	1548	8975	5,80
s1269a	79140	1030	1,3	6912	8,7	4037	7942	1,97
s35932	394282	12087	3,1	9254	2,3	1380	21341	15,46