

Markus Rudack

**Zum Selbsttest von Verbindungsnetzen
für Systems-on-Chip**

Cuvillier Verlag Göttingen

Zum Selbsttest von Verbindungsnetzen für Systems-on-Chip

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

1. Aufl. - Göttingen : Cuvillier, 2007
Zugl.: Hannover, Univ., Diss., 2007

978-3-86727-161-5

© CUVILLIER VERLAG, Göttingen 2007
Nonnenstieg 8, 37075 Göttingen
Telefon: 0551-54724-0
Telefax: 0551-54724-21
www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2007

Gedruckt auf säurefreiem Papier

978-3-86727-161-5

Zum Selbsttest von Verbindungsnetzen für Systems-on-Chip

Von der Fakultät für Elektrotechnik und Informatik
der Gottfried Wilhelm Leibniz Universität Hannover
zur Erlangung des akademischen Grades
Doktor-Ingenieur

genehmigte Dissertation von
Dipl.-Ing. Markus Rudack
geboren am 3. August 1971 in Soest

2007

1. Referent: Prof. Dr.-Ing. Joachim Mucha

2. Referent: Prof. Dr.-Ing. Hartmut Grabinski

Tag der Promotion: 14. Dezember 2006

Danksagung

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter am Institut für Theoretische Elektrotechnik und Hochfrequenztechnik sowie am Laboratorium für Informationstechnologie der Gottfried Wilhelm Leibniz Universität Hannover.

An erster Stelle möchte ich mich bei Herrn Prof. Dr.-Ing. Joachim Mucha bedanken. Er hat diese Arbeit betreut und mir die Möglichkeit gegeben, in einem äußerst interessanten Themenumfeld zu arbeiten und zu promovieren. Weiterhin danke ich Herrn Prof. Dr.-Ing. Hartmut Grabinski für die Übernahme des Korreferats und die Zusammenarbeit in der Abteilung Design & Test des Laboratoriums für Informationstechnologie. Mein besonderer Dank gilt Prof. Dr.-Ing. Erich Barke, weil er mir ermöglicht hat, diese Arbeit am Institut für Mikroelektronische System abzuschließen.

Weiterhin möchte ich mich bei allen Kolleginnen und Kollegen bedanken, die mir mit Diskussionen, Anregungen und Hinweisen bei dieser Arbeit geholfen haben. Spezieller Dank gilt meinem Bruder Dr.-Ing. Michael Rudack, der das Manuskript zu dieser Arbeit durchgesehen hat.

Meine Frau Susanne Rudack und meine Kinder Janosch und Johanna haben das Entstehen dieser Arbeit mit großer Anteilnahme verfolgt. Ohne ihre Unterstützung und ihr Verständnis wäre ich nie zum Ziel gekommen. Vielen Dank dafür.

München, im Januar 2007

Markus Rudack

Kurzfassung

Die Entwicklung von Systems-on-Chip, höchstintegrierten Schaltungen, die auf einem einzigen Chip vollständige elektronische Systeme enthalten, wird zunehmend von busbasierten Architekturen dominiert. Die Gründe liegen zum einen in der Wiederverwendung von Teilschaltungen, sogenannten IP-Cores, die über Busse miteinander kommunizieren, und zum anderen in der Notwendigkeit, Leitungssysteme hierarchisch anzulegen, um eine hohe Leitungsdichte mit geringen Signallaufzeiten auf langen Inter-Core-Leitungen zu vereinbaren.

Lange, parallel verlaufende Leitungen wie Busse sind mit zunehmender Strukturverkleinerung und höheren Taktraten von Störeffekten durch Leitungskopplung wie Signalverzögerung und Übersprechen zwischen benachbarten Leitungen betroffen. Inter-Core-Busse stellen daher in Zukunft einen Engpass bei der Leistungsfähigkeit von Core-basierten Systemen dar. Dennoch hat sich noch kein Testverfahren für Inter-Core-Busse etabliert, das Fehler durch Leitungskopplung angemessen erfasst.

In dieser Arbeit wird zunächst die Realisierung eines System-on-Chip für die Echtzeitkodierung von Bewegungsbildern hoher Auflösung beschrieben. Dieses System motivierte die Neuentwicklung von Tests auf dynamische Fehler auf Inter-Core-Bussen und von Selbsttestschaltungen für diese Busse.

Die neuen Leitungstests basieren auf dem bekannten Walking-1-Test. Sie werden erweitert, um Fehler durch kapazitive und induktive Leitungskopplung zu erfassen. Es werden Schieberegister-basierte Selbsttestrealisierungen dargestellt, die mit minimalem Overhead in den neuen Standard IEEE 1500 für den Test von eingebetteten Cores integriert werden können.

Für ein 16bit Bussystem wird eine Testschaltung für den Leitungsselbsttest präsentiert, die in einer $0,25\ \mu\text{m}$ -Technologie gefertigt worden ist. Die Auswirkung von Leitungskopplung kann für diese Testschaltung mit einem automatischen Testsystem für digitale Schaltungen gemessen werden. Im Fall von Testsignalen mit ungünstigster Leitungskopplung erreicht die Testschaltung nur etwa die halbe Betriebsfrequenz im Vergleich zum Fall günstigster Leitungskopplung. Dies unterstreicht die Notwendigkeit, Inter-Core-Busse durch einen geeigneten Test auf Kopplungsfehler zu prüfen. Die Messungen weisen eine gute Übereinstimmung mit Simulationen eines RLCM-Modell des Bussystems mit nicht linearem Treiber- und Lastmodell auf.

Schlagworte: Leitungstest, Leitungskopplung, Selbsttest, Systems-on-Chip

Abstract

Recent ultra large scale integration chip designs – so called Systems-on-Chip – are dominated by bus-based architectures. This is a result of intellectual property reuse with inter-core communication via on-chip busses. Achieving high interconnect density with low latency on long inter-core busses demands a hierarchical interconnect design, leading to bus-based architectures too.

With increasing clock rates and shrinking feature sizes long parallel busses are susceptible to delay and crosstalk of electromagnetic coupling of the bus lines. Thus, inter-core busses may be the limiting factor of future core based designs. However, no test approach aiming at coupling faults is established for inter-core busses.

This work is motivated by the design of a multi-processor system-on-chip for real-time coding of high resolution video images. Interconnect performance issues led to the development of interconnect tests for dynamic faults and of appropriate built-in self-test circuitry.

The interconnect tests are derived from the well-known walking-1 tests and are extended to cover capacitive and inductive coupling. Shift register based built-in self-test circuitry is presented that can be integrated with the new standard for embedded core test (IEEE 1500) with low area overhead.

Furthermore, a test circuit is presented consisting of a 16 bit parallel bus and self-test circuitry covering coupling faults which were manufactured in a 0,25 μm process. The chip is used to demonstrate the amount of combined driver and line delay that can be solely traced back to interconnect coupling.

The tests were conducted using a purely digital automatic test system. Using test patterns causing worst-case delay, the maximum frequency before coupling faults occur is only half as much as if using best-case pattern. The results outline the necessity to implement suitable test measures for inter-core busses. The measured results agree well with data obtained by simulation of interconnect lines using n -RLCM lumped models and non-linear bus driver and load models.

Keywords: Interconnect test, Crosstalk, Built-In Self-Test, Systems-on-Chip

Inhaltsverzeichnis

Inhaltsverzeichnis	ix
Abbildungsverzeichnis	xiii
Abkürzungsverzeichnis	xv
Liste der verwendeten Formelzeichen	xix
1 Einleitung	1
1.1 Systems-on-Chip	3
1.2 Globale Verbindungsnetze bei Systems-on-Chip	5
1.2.1 Die Dominanz von Leitungen in integrierten Schaltungen	5
1.2.2 Hierarchische Leitungssysteme	8
1.2.3 Leitungsmodelle	9
2 Test von Systems-on-Chip	13
2.1 Schwierigkeiten beim Test eingebetteter Cores	16
2.2 Testfreundlicher Entwurf eingebetteter Cores	17
2.2.1 Testmustererzeugung und Testauswertung	18
2.2.2 Selbsttest mit BILBOs	20
2.2.3 Selbsttest mit programmierbaren Mikroprozessor-Cores	22
2.2.4 Eingebetter deterministischer Test	23
2.3 IEEE 1500 Standard für den Test Core-basierter Systeme	24

3	Test von Inter-Core Verbindungsnetzen	27
3.1	Fehlermodelle für Verbindungsnetze	27
3.1.1	Statisches Fehlermodell für Leitungssysteme	28
3.1.2	Maximum Aggressor Fehlermodell	31
3.2	On-Chip Testmustererzeugung	36
3.3	On-Chip Testauswertung	39
4	Testimplementierung für ein Multiprozessorsystem	43
4.1	Systemarchitektur	43
4.1.1	Übersicht über die Architektur der Prozessorknoten	46
4.1.2	Fehlertoleranzkonzept	47
4.1.3	Testkonzept	48
4.2	Exemplarische Implementierung eines Selbsttests	50
4.2.1	Videoprozessorkern	50
4.2.2	Anwendungsspezifische Schnittstellen im Prozessorknoten	51
4.2.3	DRAM und Controller	51
4.2.4	Leitungssysteme	52
4.2.5	Fehlerabdeckung	53
4.3	Bewertung der Schaltungsentwicklung für die AxPe-GIS	54
5	Schieberegister-basierter Selbsttest von Inter-Core Verbindungsnetzen	57
5.1	Testverfahren für Verbindungsnetze mit induktiver Kopplung	57
5.2	On-Chip Testmustererzeugung mit Schieberegistern	60
5.3	On-Chip Testauswertung mit MISR	66
5.3.1	Aliasing durch die Testauswertung mit rückgekoppelten Schieberegistern	67
5.3.2	Fehlersimulation für Inter-Core Verbindungsnetze	69
5.4	Ablaufplan für den Selbsttest der Verbindungsnetze	72
5.4.1	Verbindungsgraph von Leitungssystemen	73
5.4.2	Kopplungsgraph von Leitungssystemen	78
5.5	IEEE 1500 Core-Test-Wrapper für den Bustest	82

6 Messung von Leitungskopplung mit einer digitalen Testschaltung	87
6.1 Testschaltung ACSelfTest	88
6.1.1 Testmustergenerator TPG	89
6.1.2 Testauswerter ORA	91
6.1.3 Testbus BusK	92
6.1.4 Fehlerdiagnose mit dem Leitungsselbsttest	95
6.2 Modellbasierte Verifikation der verwendeten Tests	95
6.2.1 Schaltungs- und Leitungsmodell	96
6.2.2 Simulationsergebnisse	100
6.3 Ergebnisse des Hardwaretest	103
6.3.1 Messergebnisse	105
6.3.2 Berechnung des Kopplungseinflusses aus Messdaten	108
6.4 Bewertung der Messergebnisse	111
7 Zusammenfassung	115
Literaturverzeichnis	118
Wissenschaftlicher Werdegang	127

Abbildungsverzeichnis

1.1	Verzögerungsfehler	2
1.2	System-on-Chip (SOC) mit <i>IP-Cores</i>	4
1.3	Einfaches kapazitives Leitungsmodell	6
2.1	Testdauer beim Scan-Test mit 500 MHz	15
2.2	Linear rückgekoppeltes Schieberegister (LFSR)	18
2.3	Linear rückgekoppeltes Schieberegister mit Mehrfacheingängen (MISR)	19
2.4	Selbsttest mit LFSR und MISR (STUMPS)	20
2.5	BILBO-Architektur für den Selbsttest von eingebetteten Cores	21
2.6	Aufbau eines BILBOs	21
2.7	Selbsttest von Inter-Core-Bussen mit BILBOs	22
2.8	Eingebetter deterministischer Test	23
2.9	Architektur eines IEEE 1500 Core Test Wrappers	25
3.1	Modell für zwei gekoppelte Leitungen mit mehreren RLCM-Elementen	32
3.2	Mögliche Signalverzerrungen bei Leitungskopplung	32
3.3	Testmustergenerator für Walking-1-Test mit $N = 4$	37
3.4	Testmustergenerator für einen vollständigen Test bezüglich des Maximum Aggressor Fehlermodells	39
3.5	Testauswertung mit Komparator beim Selbsttest von Bussen	41
4.1	Vereinfachtes Blockdiagramm für ein AxPe-Multiprozessorsystem	44
4.2	Vereinfachter Floorplan von AxPe-Subsystem und AxPe-GIS	45
5.1	Qualitative Darstellung von kapazitiver und induktiver Kopplung	59
5.2	Erweiterter Testmustergenerator für Tests auf Verzögerungsfehler	62
5.3	Steuersignale des erweiterten Testmustergenerators	62
5.4	Testmustergenerator für einen vollständigen Test auf dynamische Leitungsfehler	65
5.5	Steuersignale des Testmustergenerators für dynamische Leitungsfehler	65
5.6	Leitung mit vier Ports und zugehöriger ungerichteter Graph	74
5.7	Verbindungsgraph der Leitung mit vier Ports	74
5.8	Verbindungsgraph für das Bussystem eines AxPe-Subsystems	76
5.9	Kopplungsgraph für das Bussystem eines AxPe-Subsystems	80
5.10	Architektur eines Core Test Wrappers für den Leitungsselbsttest	82
5.11	Wrapperzellen für den Leitungsselbsttest	83

6.1	Blockschaltbild der Testschaltung ACSelfTest	88
6.2	Layout von Testmuster-generator und Testauswerter.	90
6.3	Signalverläufe bei der Testauswertung	92
6.4	Layout des Testbus, Übersicht	93
6.5	Schematischer Querschnitt des Testbus BusK	94
6.6	Schematische Darstellung der simulierten Schaltungsumgebung für den Testbus	96
6.7	Signalverlauf im <i>Even Mode</i> und im <i>Odd Mode</i>	98
6.8	Zusätzliche Signalverzögerung auf der Leitung 0 mit zunehmender Anzahl von Aggressorleitungen	101
6.9	Zusätzliche Signalverzögerung auf der Leitung 7 mit zunehmender Anzahl von Aggressorleitungen	101
6.10	Glitch auf der Leitung 0 mit zunehmender Anzahl von Aggressorleitungen	102
6.11	Digital-Tester HP 82000	104
6.12	Detailansicht des Digital-Tester: <i>Loadboard</i> mit offenem Testadapter . .	105
6.13	Verteilung der minimalen Taktperiode im Testmodus <i>Odd Mode</i>	106
6.14	Verteilung der minimalen Taktperiode im Testmodus <i>Even Mode</i>	106
6.15	Schematische Darstellung des Signalpfades für den Test einer Leitung . .	108
6.16	Verteilung $\Delta\tau_{Kopplung}$	111

Abkürzungsverzeichnis

Al	Aluminium
AMBA	Advanced Microcontroller Bus Architecture, engl. für eine Architektur eines Microcontroller-Bus
ATPG	Automatic Test Pattern Generation, engl. für Automatische Testmuster-generierung
BILBO	Built-In Logic Block Observer, engl. für Register zur Steuerung und Beobachtung von integrierter Logik
BIST	Built-In Self-Test, engl. für integrierter Selbsttest
CA	Cellular Automaton, engl. für zellulärer Automat
CPU	Central Processing Unit, engl. für zentrale Recheneinheit, Prozessor
CTL	Core Test Language, engl. für eine standardisierte Core Test Sprache
DDR	Double Data Rate, engl. für doppelte Datenrate
DFT	Design for Testability, engl. für testfreundlicher Entwurf
DMA	Direct Memory Access, engl. für direkter Speicherzugriff
DRAM	Dynamic Random Access Memory, engl. für dynamischer Speicher mit wahlfreiem Zugriff
eDRAM	embedded DRAM, engl. für eingebettes DRAM
FIFO	First In - First Out, engl. für „zuerst hinein - zuerst heraus“
GIS	Großflächig integrierte Schaltung
HDTV	High Definition Television, engl. für hochauflösendes Fernsehen
I/O	Input/Output, engl. für Eingabe/Ausgabe
IEEE	The Institute of Electrical and Electronics Engineers, engl. für Internationaler Berufsverband für Ingenieure der Elektrotechnik und Elektronik

IP	Intellectual Property, engl. für geistiges Eigentum
ISO	International Organization for Standardization, engl. für Internationale Vereinigung der Standardisierungsgremien
ITU-T	International Telecommunication Union, Telecommunication Standardization Bureau, engl. für Internationale Fernmeldeunion, Bereich für Fernmeldestandards
LFSR	Linear Feedback Shift Register, engl. für linear rückgekoppeltes Schieberegister
MAFM	Maximum Aggressor Fehlermodell
MISR	Multiple Input Signature Register, engl. für linear rückgekoppeltes Schieberegister mit Mehrfacheingang
MPEG	Moving Picture Experts Group, engl. für Expertengruppe für Bewegtbilder (Video)
<i>nG</i>	Negativer Glitch
ORA	Output Response Analyzer, engl. für Testauswerter
PCI	Peripheral Component Interconnect, engl. für einen Bus-Standard zur Anbindung von Peripheriegeräten
<i>pG</i>	Positiver Glitch
PLL	Phase-Locked Loop, engl. für phasengekoppelter Regelkreis
PN	Processing Node, engl. für Prozessorknoten
RISC	Reduced Instruction Set Computing, engl. für elektronisches Rechnen mit reduziertem Befehlssatz
ROM	Read-Only Memory, engl. für Festwertspeicher
SDRAM	Synchronous Dynamic Random Access Memory, engl. für synchroner dynamischer Speicher mit wahlfreiem Zugriff
SEC-DED	Single Error Correcting, Double Error Detecting, engl. für Einzelfehler korrigierend, Doppelfehler erkennend
Si	Silizium
SiO ₂	Siliziumdioxid
SMP	Symmetrisches Multiprozessorsystem
SOC	System-on-Chip
SRAM	Static Random Access Memory, engl. für statischer Speicher mit wahlfreiem Zugriff
STIL	Standard Test Interface Language, engl. für standardisierte Sprache zur Beschreibung von Testschnittstellen

STUMPS	Self-Test Using MISR and Parallel SRSG (Shift Register Sequence Generator), engl. für Selbsttest mit MISR und parallelem, Schieberegister basiertem Mustergenerator
TEM	Transversal elektromagnetisch
TPG	Test Pattern Generator, engl. für Testmustergenerator
UART	Universal Asynchronous Receiver-Transmitter, engl. für universeller Sende- und Empfangsbaustein
UDL	User Defined Logic, engl. für benutzerdefinierte Logik
V_{fF}	Verzögerung bei fallender Flanke
V_{stF}	Verzögerung bei steigender Flanke
WBR	Wrapper Boundary Register, engl. für Randregister der Core-Wrappers
WBY	Wrapper Bypass Register, engl. für Bypass-Register des Core-Wrappers
WIP	Wrapper Interface Port, engl. für Schnittstelle des Core-Wrappers
WIR	Wrapper Instruction Register, engl. für Befehlsregister des Core-Wrappers
WPI	Wrapper Parallel Input, engl. für paralleler Eingang des Core-Wrappers
WPO	Wrapper Parallel Output, engl. für paralleler Ausgang des Core-Wrappers
WSI	Wrapper Serial Input, engl. für serieller Eingang des Core-Wrappers
WSO	Wrapper Serial Output, engl. für serieller Ausgang des Core-Wrappers

Liste der verwendeten Formelzeichen

a	Aliasingwahrscheinlichkeit
A	Zustandsübergangsmatrix
A_{Die}	Chipfläche
a_i, b_i	i -tes Glied einer Folge (a_n) bzw. (b_n)
B	Anzahl der bidirektionalen Ports
C	Kapazität
C'	Kapazitätsbelag
C'_{i0}	Teilkapazitätsbelag von Leiter i zum Nullleiter (Erde)
C'_{ij}	Teilkapazitätsbelag von Leiter i zu Leiter j
$C(\tau)$	Autokorrelationsfunktion
C_{ABi}	Koppelkapazität der i -ten RLCM-Elemente von Leiter A und B
C_i	Kapazität des i -ten RLCM-Elements
C_L	Lastkapazität
E	Anzahl der Kanten eines Graphen
f	Betriebsfrequenz
f_{max}	Grenzfrequenz
f_W	Betriebsfrequenz eines Core-Test-Wrappers
$GF(2)$	Galois-Feld der Charakteristik 2
h	Leitungsdicke
h_i	Koeffizienten des charakteristischen Polynoms
k	Skalierungsfaktor für die laterale Strukturgrößen
K	Anzahl der Knoten eines Graphen
K_i	i -te Leitung eines Leitungssystems
k_{max}	Obergrenze der Testlänge
k_{min}	Minimale Testlänge
l	Skalierungsfaktor für die Fläche einer integrierten Schaltung
L	Leitungslänge

Liste der verwendeten Formelzeichen

l_c	Kritische Leitbahnlänge
L_i	Induktivität des i -ten RLCM-Elements
M_{ABi}	Gegeninduktivität der i -ten RLCM-Elemente von Leiter A und B
N	Anzahl der Leitungen eines Leitungssystems oder Anzahl der Speicherelemente (Flip-Flops) eines Registers
N_F	Anzahl der simulierten Fehler pro Fehlermodell
p	Periodenlänge einer periodischen Folge
p_k	Verhältnis der Anzahl der Sequenzen der Länge k zur Gesamtzahl der Sequenzen einer zufälligen Zahlenfolge
p_k^*	Verhältnis der Anzahl der Sequenzen der Länge k zur Gesamtzahl der Sequenzen einer periodischen Zahlenfolge
q_i	i -tes Steuersignal
R	Anzahl von Leseports
R	Elektrischer Widerstand
R'	Widerstandsbelag
R_i	Widerstand des i -ten RLCM-Elements
$R_{A,on}$	Innenwiderstand einer Ersatzspannungsquelle
\underline{s}	Signatur
S	Anzahl von Schreibports
\underline{s}_f	Signatur im Fehlerfall
\underline{s}_{gut}	Gutsignatur
\underline{s}_i	Zustandsvektor
t_{hold}	Notwendige Haltezeit eines Flip-Flops
t_i	i -ter Zeitschritt
$t_{k,l}$	Zeitschritt k in Gruppe l
$T_{Periode}$	Periodendauer eines Signals
t_r	Signalanstiegszeit
$T_{skew,ORA}$	Verschiebung des Taktsignals am Testauswerter
$T_{skew,TPG}$	Verschiebung des Taktsignals am Testmustergenerator
t_{slack}	Höchste zulässige Verzögerungszeit
U_1	Quellspannung einer Ersatzspannungsquelle
v	Phasengeschwindigkeit
\underline{r}_i	Testantwortvektor

w	Leitungsbreite
X	Unsicherer Logikwert
Z	Anzahl der Baumzweige eines Graphen
$\Delta\tau_{AB}$	Differenz der Signalverzögerungen τ_{AB} im <i>Even Mode</i> und im <i>Odd Mode</i>
$\Delta\tau_{linear}$	Differenz der Signalverzögerungen im <i>even mode</i> und <i>odd mode</i> bei linearem Treiber- und Lastmodell
ϵ	Permittivität
ϵ_r	Relative Permittivität
μ	Permeabilität
τ	Signalverzögerung
τ_{AB}	Signalverzögerung von einem Knoten A zu einem Knoten B
τ_{BusK}	Signalverzögerung auf einem Bus K
τ_{BusR}	Signalverzögerung auf dem Referenzbus
τ_{comp}	Signalverzögerung eines Komparators
τ_{gesamt}	Gesamtverzögerung des Signalpfades
τ_{TPG}	Signalverzögerung des Testmustergenerators
$\tau_{parametrischer\ Fehler}$	Signalverzögerung bei Vorliegen eines parametrischen Fehlers
τ_{soll}	Sollwert einer Signalverzögerung
0	Logisch 0
1	Logisch 1

1 Einleitung

Das exponentielle Wachstum der Schaltungskomplexität¹ ist die Triebfeder für den wirtschaftlichen Erfolg der Halbleiterindustrie. Es wird durch das *Moore'sche Gesetz* [Moo65] beschrieben, das eine Verdopplung alle 24 Monate vorhersagt. Das Moore'sche Gesetz gilt seit etwa 35 Jahren, für die Zukunft ist es allerdings durch folgende Hindernisse beschränkt oder zumindest gefährdet:

- **Physikalische Grenzen:** Bei Strukturen in der Größenordnung von einzelnen Atomen ist keine weitere Verkleinerung möglich. Bauelemente in atomaren Dimensionen haben außerdem vollständig andere Eigenschaften als klassische Halbleiterbauelemente, die die Basis heutiger integrierter Schaltungen bilden.
- **Technologische Grenzen:** Es werden Fertigungstechnologien benötigt, die letztendlich Strukturen in der Größenordnung von einzelnen Molekülen oder sogar Atomen herstellen oder bearbeiten können. Die Fertigung von Schaltungen mit Strukturgrößen von etwa 35 nm scheint jedoch gesichert.
- **Entwurfskomplexität:** Die Steigerung der Produktivität bei der Schaltungsentwicklung hält nicht mit der Entwicklung der Fertigungstechnologie mit (*Design Productivity Gap*). Dadurch kann die technologisch mögliche Anzahl von Bauelementen einer integrierten Schaltung nicht sinnvoll genutzt werden. Durch Innovationsschübe bei der Entwicklungsmethodik kann diese Lücke zeitweilig geschlossen werden. In jüngerer Vergangenheit hat sich beispielsweise die Wiederverwendung von größeren Teilschaltungen, so genannten *IP-Cores* (Intellectual Property Cores), zur Integration eines extrem komplexen Systems in einer einzigen integrierten Schaltung, einem *Systems-on-Chip*, durchgesetzt.
- **Testkosten:** Die Testkosten von integrierten Schaltungen beginnen, die Herstellungskosten zu dominieren, so dass deren Fertigung unwirtschaftlich werden kann. Die Testkosten entstehen im Wesentlichen durch die Investitionen für automatische Testsysteme und für die Testentwicklung. Die Ursachen für Kostensteigerungen sind einerseits mit höheren Taktraten steigende Ansprüche an die Genauigkeit der automatischen Testsysteme. Andererseits werden immer größere Testdatenmengen benötigt, um eine ausreichende Testqualität immer komplexerer Schaltungen zu erzielen, so dass die Testdauer für eine einzelne Schaltung kritisch wird.

¹ Die Anzahl der aktiven Bauelemente einer integrierten Schaltung.

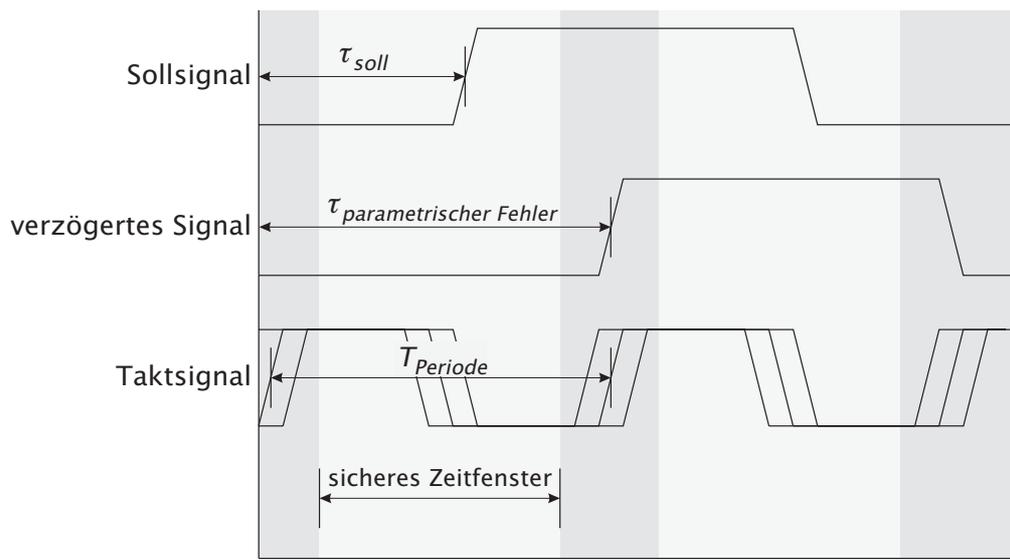


Abbildung 1.1: Verzögerungsfehler

Insbesondere die Hindernisse Entwurfskomplexität und Testkosten haben die Architektur integrierter Schaltungen in den vergangenen Jahren bereits stark geprägt und führen zu charakteristischen Eigenschaften, die in den folgenden Abschnitten kurz erläutert werden. Diese Eigenschaften führen auf das spezielle Problem, einen Selbsttest für Systeme von Leitungen zu entwickeln, die Teilschaltungen eines System-on-Chip miteinander verbinden. Die Lösung dieses Testproblems ist das Thema dieser Arbeit.

Mit dem Moore'schen Gesetz ist eine exponentielle Steigerung der Leistungsfähigkeit mikroelektronischer Systeme verbunden. Diese Steigerung wird durch die Integration von immer mehr und immer komplexeren Funktionen, insbesondere aber durch höhere Taktraten erzielt. Diese haben steilere Signalflanken zur Folge und bedeuten daher einen zunehmenden Hochfrequenzanteil bei den übertragenen Signalen.

Insgesamt werden die zulässigen Toleranzen für das zeitliche Verhalten einer integrierten Schaltung immer geringer. Demgegenüber steigt aber die Unsicherheit bei der Schaltungsentwicklung, weil mit hohen Signalfrequenzen der Einfluss von parasitären, d.h. ungewollten, Leitungsinduktivitäten und -kapazitäten auf den Signalverlauf innerhalb eines mikroelektronischen Systems zunimmt. Störsignale durch elektromagnetische Leitungskopplung, Reflexion an Leitungsenden und Dispersion, die in der Vergangenheit keine Rolle gespielt haben, können zu Fehlfunktionen führen. In digitalen Schaltungen wirken sich diese Störungen beispielsweise als eine Signalverzögerung aus.

Abbildung 1.1 zeigt, wie eine Signalverzögerung zu einem logischen Fehler führen kann. Die obere Zeile zeigt den nominellen Zeitverlauf eines Signals mit einer Verzögerung τ_{soll} gegenüber dem Taktsignal am Eingang eines flankengesteuerten Flipflops. Die mittlere Zeile zeigt das gleiche Signal, das aber durch einen nun angenommenen Fehler oder durch elektromagnetische Kopplung zweier Leitungen eine größere Verzögerung $\tau_{parametrischer\ Fehler}$ als das nominelle Signal in der oberen Zeile aufweist. Die untere Zeile zeigt das Taktsignal der Periode $T_{Periode}$ mit möglichen zeitlichen Schwankungen (so genannte *Jitter*). Durch diese Schwankungen wird der Zeitbereich verbreitert, in dem das Eingangssignal für ein sicheres Speichern des Signalwerts stabil bleiben muss. Dieser Bereich ist in Abbildung 1.1 dunkel unterlegt. Umgekehrt wird der Zeitbereich, in dem ein Signalwechsel erlaubt ist, schmaler. Durch die zusätzliche Verzögerung des fehlerhaften Signals verschieben sich dessen Signalwechsel in die dunkel unterlegten Bereiche, so dass das Flipflop den Signalwert nicht sicher speichern kann. Eine logische Fehlfunktion ist die Folge.

Kritische parametrische Abweichungen durch Schwankungen im Fertigungsprozess bedeuten im Allgemeinen eine zusätzliche Verringerung des Zeitfensters für einen sicheren Signalwechsel innerhalb eines Taktzyklus. Das hat zur Folge, dass bei bestimmten extremen Signalverläufen, die ohne die Abweichungen nicht zu Fehlern führen, ein Signalwechsel nicht mehr in das reduzierte Zeitfenster fällt, so dass ein logischer Fehler auftritt. Extreme Signalverläufe sind Signalübergänge an einem oder mehreren Schaltungseingängen, die beispielsweise zu Übersprechen von Signalen auf benachbarten Leitungen oder zu kurzzeitigen Einbrüchen der Versorgungsspannung digitaler Schaltungen führen. Um die entsprechenden Fehler zu testen, müssen diese extremen Signalverläufe stimuliert werden. In dieser Arbeit wird untersucht, welche Eigenschaften diese Tests für lange, überwiegend parallel verlaufende Leitungen haben müssen.

1.1 Systems-on-Chip

Die technologische Entwicklung stellt durch steigende Integrationsdichte dem Schaltungsentwickler immer größere Ressourcen bezüglich der Anzahl integrierbarer Bauelemente zu Verfügung. Unter dem Druck immer kürzerer Entwicklungszyklen können diese Ressourcen nur durch die Verwendung von vorab entworfenen und verifizierten Teilschaltungen, den *IP-Cores*, ausgenutzt werden. Cores sind strukturell und funktionell eng verknüpfte Schaltungsteile, die über definierte Schnittstellen mit anderen Cores kommunizieren. Sie realisieren Funktionen wie Mikroprozessoren (z.B. eine RISC-CPU), Speicher (z.B. DRAM, SRAM oder ROM), Standardschnittstellen (z.B. PCI, UART oder DMA) oder auch spezialisierte Prozessoren für die Bearbeitung von Audio- oder Bilddaten (z.B. MPEG). Cores sind die Bausteine eines

komplexen integrierten Systems, nur noch geringe Teile werden für ein System neu entwickelt (*User Defined Logic*, UDL). Abbildung 1.2 zeigt ein derartiges System-on-Chip (SOC).

Dieser neue Entwurfsansatz, der auf der schwerpunktmäßigen Verwendung von *IP-Cores* beruht, unterscheidet sich vom „klassischen“ Entwurf dadurch, dass der Entwickler nur noch eine Verhaltensbeschreibung der verwendeten Teilschaltungen erhält, nicht aber detaillierte Informationen über die schaltungstechnische Realisierung.

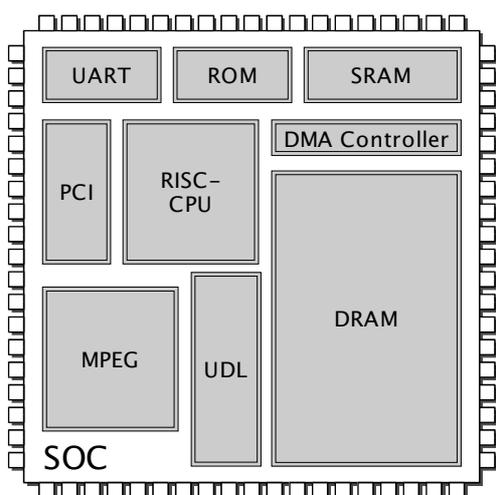


Abbildung 1.2: System-on-Chip (SOC) mit *IP-Cores*

Für einen erfolgreichen Entwurf ist es daher notwendig, dass sich die einzelnen Komponenten an kompatible Standards für die Interaktion halten. In der Regel interagieren die Cores eines solchen Systems über standardisierte Busse wie den PCI-Bus oder den AMBA-Bus von Arm Ltd. [Arm99]. Diese Systeme werden auch als Bus-basierte Systeme beschrieben. Die *Inter-Core*-Busse sind dabei von gleicher Bedeutung für die Funktion und Leistungsfähigkeit eines SOC's wie die Cores selbst.

Diese Busse zeichnen sich dadurch aus, dass ihre Länge in der gleichen Größenordnung liegen kann wie die Kantenlänge der integrierten Schaltung. Die Busse werden daher auch als global bezeichnet, weil sie einerseits die größte Länge besitzen und andererseits funktional gesehen Module auf der obersten logischen Hierarchieebene miteinander verknüpfen. Demgegenüber werden Leitungen als lokal bezeichnet, deren Länge in der Größenordnung einiger Dutzend Mikrometer liegt und die auf tieferer Ebene der logischen Hierarchie Schaltungsteile eines Cores miteinander verbinden.

Die Testentwicklung muss durch den Anbieter eines Cores erfolgen, weil diese auf detaillierten strukturellen Kenntnissen des Moduls beruht. Mittlerweile stehen

umfangreiche Bibliotheken von Cores für den Chipentwurf zur Verfügung. Diese enthalten im Allgemeinen auch verifizierte Tests für die einzelnen Komponenten. Die Tests müssen zu den einzelnen Cores transferiert werden. Dies soll in Zukunft durch den Standard IEEE 1500 [IEE05] geleistet werden, der neben einer standardisierten Testbeschreibungssprache (*Core Test Language*, CTL) eine standardisierte Testhülle (*Core Test Wrapper*) als Schnittstelle für den Test definieren soll [MKL⁺02]. Der Standard IEEE 1500 wird in Abschnitt 2.3 kurz dargestellt.

Auch die Spezifikation des AMBA-Busses legt ein Verfahren für den Test der Cores fest. Dabei wird der AMBA-Bus als breitbandige Schnittstelle zur Übertragung von Testdaten von der Peripherie zu den integrierten Cores verwendet. Insbesondere beim AMBA-Bus ist die Funktionsfähigkeit der Inter-Core-Busse für den Test der Cores notwendig. Inter-Core-Busse müssen daher sogar als erste Komponente des Systems getestet werden. Mit beiden Standards werden keine spezielle Methoden für den Test von Inter-Core-Leitungssystemen zur Verfügung gestellt. Diese Leitungssysteme stellen aber in Zukunft einen Engpass bei der Leistungsfähigkeit von Core-basierten Systemen dar und müssen daher auch beim Test gezielt berücksichtigt werden.

1.2 Globale Verbindungsnetze bei Systems-on-Chip

Im vorausgehenden Abschnitt 1.1 wurde dargestellt, wie die Notwendigkeit, die Entwurfskomplexität zu beherrschen, zu Systemen führt, die aus Cores zusammengesetzt sind, die über globale Busse miteinander interagieren. Aber auch die Veränderung der elektrischen Parameter von Verbindungsleitungen mit den Technologiegenerationen führt zu einem busbasierten Ansatz für die Schaltungsarchitektur.

1.2.1 Die Dominanz von Leitungen in integrierten Schaltungen

Verbindungsleitungen von integrierten Schaltungen haben einen stetig wachsenden Einfluss auf die Gesamtpformance, weil sie im Gegensatz zu aktiven Bauelementen bezüglich ihrer Signalverzögerung nicht von der allgemeinen Strukturverkleinerung profitieren. Diese Entwicklung kann bereits mit dem sehr einfachen längshomogenen Leitungsmodell nach Abbildung 1.3 erklärt werden [BRKO93, Boh95]. Der linke Teil der Abbildung stellt, stark vereinfacht, den Querschnitt zweier Leitungen auf einem Chip einer bestimmten Technologiegeneration dar. Der rechte Teil stellt die gleichen Leitungen in einer folgenden Technologiegeneration dar, wobei die lateralen Abmessungen um einen Faktor $k < 1$ reduziert werden, während die

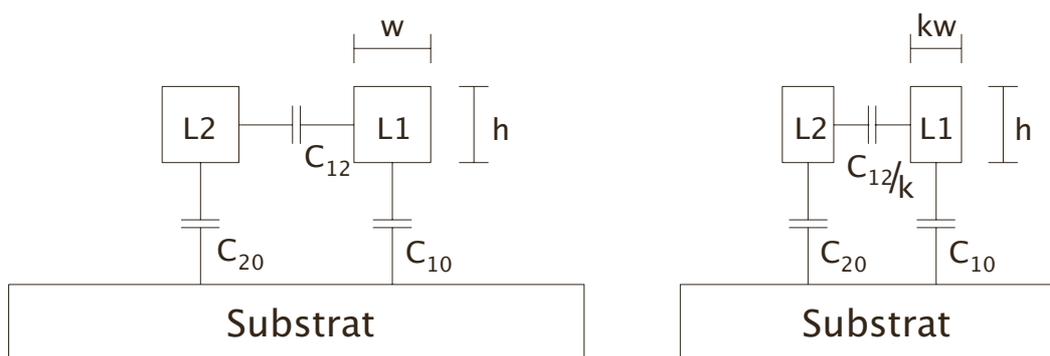


Abbildung 1.3: Einfaches kapazitives Leitungsmodell: nicht skaliert (links), laterale Dimensionen skaliert mit Faktor $k = 0,66$ (rechts).

vertikalen Abmessungen in diesem einfachen Beispiel als konstant angenommen werden².

Die Signalverzögerung τ einer Leitung ist in erster Näherung das Produkt ihres Widerstands R und ihrer Kapazität C , also $\tau = RC$. Widerstand R und Kapazität C skalieren mit einer Strukturverkleinerung. Bei länghomogenen Leitungen gibt man die Kapazität C und den Widerstand R bezogen auf die Leitungslänge L als Kapazitätsbelag $C' = C/L$ und Widerstandsbelag $R' = R/L$ an. Damit gilt für die Signalverzögerung $\tau = R'C'L^2$.

Eine Verringerung der Leitungsbreite w um einen Faktor $k < 1$ bedeutet eine entsprechende Reduktion des Leitungsquerschnitts, also letztendlich eine Vergrößerung des Widerstandsbelags R' um den Faktor k , wenn man annimmt, dass die Leitungsdicke h konstant bleibt.

Die Kapazitätsbeläge der Leitungen können nach Abbildung 1.3 in die Anteile C'_{10} bzw. C'_{20} zum Substrat und in die Koppelkapazität C'_{12} zur benachbarten Leitung aufgeteilt werden. Unter Annahme eines einfachen Parallelplattenmodells für die Kapazität nehmen die Beläge C'_{10} und C'_{20} um den Faktor k ab, während der Koppelungsanteil C'_{12} um den Faktor $\frac{1}{k}$ zunimmt, weil der Abstand von Leitungen wie die anderen lateralen Dimensionen um diesen Faktor abnimmt. In erster Näherung bleibt damit das Produkt von Widerstandsbelag und Kapazitätsbelag $R'C'$ einer Leitung günstigstenfalls konstant, wenn die Kapazität zum Substrat dominiert, steigt aber sonst quadratisch mit der Abnahme der Strukturgrößen.

Bei der Entwicklung der Leitungslänge L gibt es zwei gegenläufige Tendenzen. Zum einen kann die Länge von lokalen Leitungen, die benachbarte logische Zellen verbind-

² Die Skalierung von einer Technologiegeneration zur nächsten ist in Wirklichkeit sehr viel komplexer. Es erfolgt insbesondere ebenfalls eine Verringerung der vertikalen Abmessungen, die aber mit den einzelnen Schichten variiert, z.B. von den unteren zu den oberen Metalllagen. Außerdem können sich auch die Materialien für Leiter und für Dielektrika verändern.

den, mit deren lateralen Abmessungen verkürzt werden. Die Verzögerung lokaler Leitungen nimmt also bei konstantem Widerstands- und Kapazitätsbelag proportional zu k ab, und stellt für die Gesamtverzögerung eines Schaltungspfads bestehend aus Gattern und lokalen Leitungen kein Problem dar.

Mit jeder Technologiegeneration nimmt jedoch die Fläche integrierter Schaltungen um einen Faktor $l > 1$ zu, der von $\frac{1}{k}$ verschieden ist. Die maximale mögliche Länge für eine globale Leitung, die zwei Zellen an gegenüberliegenden Stellen der Schaltung verbindet, nimmt damit um den Faktor \sqrt{l} zu. Die Verzögerung globaler Leitungen nimmt also wenigstens proportional zu l zu.

Während die Signalverzögerungen auf Leitungen im Zuge der Skalierung also stagnieren (oder sogar zunehmen), geht man bei der intrinsischen Gatterverzögerung³ von einer Reduzierung um 15 % pro Jahr aus [ITR03]. Die Folge ist, dass die Signalverzögerung globaler Leitungen das zeitliche Verhalten einer integrierten Schaltung bei weiterer Skalierung dominiert [HMH01] und limitierend für die Schaltungsperformance wirkt. Damit kann ohne neue Leitungsarchitekturen die mit dem Wechsel der Technologiegenerationen verbundene Steigerung der Leistungsfähigkeit durch Erhöhung der Taktraten nicht weiter realisiert werden.

Um die durch die Dominanz der Leitungen gegebene Grenze zu überschreiten, hat die Halbleiterindustrie neue Materialien und Leitungsarchitekturen eingeführt. Die Verwendung von Kupfer als ein Leitungsmaterial mit höherer Leitfähigkeit und von isolierenden Materialien mit niedriger Permittivität ϵ sowie andere fertigungstechnische Fortschritte [HH00] verringern die Signalverzögerung auf Leitungen. Sie können den grundsätzlichen Trend aber nicht aufheben, sondern die Dominanz der Verzögerung langer Leitungen nur auf eine spätere Technologiegeneration verschieben. Der Grund für die Verwendung von Kupferleitungen ist daher eher in der größeren Widerstandsfähigkeit gegen Elektromigration zu sehen [BRKO93].

In dem dargestellten einfachen Verzögerungsmodell gilt für den Zusammenhang von Signalverzögerung τ und Leitungslänge L der Zusammenhang $\tau \sim L^2$. Daher werden lange Leitungen in mehrere Abschnitte aufgeteilt, die jeweils von starken Leitungstreibern getrieben werden. Bei optimaler Auslegung solcher Leitungssysteme hinsichtlich der Länge der Abschnitte und der Stärke der Treiber kann die Gesamtverzögerung gegenüber einer langen Einzelleitung deutlich verringert werden [LCL96, CDBS01].

³ Die intrinsische Gatterverzögerung ist die Signalverzögerung von einem Eingang eines logischen Gatters zu seinem Ausgang, die sich ergäbe, wenn keine Last am Ausgang vorhanden wäre.

1.2.2 Hierarchische Leitungssysteme

Neue Leitungsarchitekturen nutzen aus, dass die steigenden Signalverzögerungen nur die langen globalen Verbindungsleitungen betreffen. Dagegen skaliert die Verzögerung auf den kurzen lokalen Leitungen mit der Strukturverkleinerung, wenn auch nicht in dem gleichen Maße wie die Transistorschaltzeiten. Die lokalen Leitungen machen außerdem den Hauptanteil der Gesamtzahl aller Verbindungsleitungen aus, weil mit ihnen die Verbindungen innerhalb der strukturell eng verknüpften Cores realisiert werden. Daher werden lokalen Leitungen mit dem gleichen Faktor skaliert wie die aktiven Bauelemente, um die Dichte der Leitungen entsprechend der Dichte der Bauelemente zu vergrößern.

Für den geringen Anteil der langen globalen Leitungen wird eine inverse Skalierung durchgeführt. Das bedeutet, dass Leitungsdicke, -breite und -abstand in verschiedenem Maße vergrößert anstatt verringert werden, so dass Signalverzögerung auf einer Leitung und die elektromagnetische Kopplung zwischen mehreren Leitungen reduziert werden [KMSS98]. Inverse Skalierung bedeutet jedoch auch, dass für die globalen Verbindungsleitungen eine wesentlich größere Fläche benötigt wird. Da die benötigte Fläche direkt die Herstellungskosten beeinflusst, muss ein Kompromiss zwischen Leitungsperformance und Kosten gefunden werden [DM98, Mei03].

Insgesamt lässt sich die Signalverzögerung der globalen Leitungen durch inverse Skalierung, Optimierung von Leitungstreibern und Anwendung neuer Materialien soweit verringern, dass sie zumindest für die Technologiegenerationen der kommenden zehn Jahre kein Hemmnis für die Leistungsfähigkeit integrierter Schaltungen darstellen. Eher scheint die zur Verfügung stehende Verdrahtungsfläche ein Hindernis zu werden [The00].

Inverse Skalierung führt zu einer – im einfachsten Fall zweistufigen – Hierarchie der Leitungen einer integrierten Schaltung: Eine Ebene lokaler kurzer Leitungen, die mit den übrigen lateralen Dimensionen skalieren, und eine Ebene globaler, langer Leitungen, die überproportional große Leitungsbreiten und -abstände aufweisen, und deren Verzögerung wie oben beschrieben optimiert wird.

Diese Hierarchie der Leitungen passt zu der in Abschnitt 1.1 beschriebenen Entwurfsmethodik, die auf der Verwendung von Cores beruht. Die enge interne Verknüpfung innerhalb der Cores erfolgt auf der Ebene der lokalen Leitungen, die daher als Intra-Core-Leitungen bezeichnet werden und deren Verzögerung durch Skalierung ihrer Länge reduziert werden kann. Die Verbindung mit anderen Cores erfolgt über die Ebene globaler, langer Leitungen, die daher als Inter-Core-Leitungen bezeichnet werden.

Diese Verbindungen werden in der Regel als Busse realisiert, verbinden also mehrere Cores und bestehen aus mehreren Leitungen. Diese können über relativ große Distanzen parallel verlaufen. Das macht Inter-Core-Busse anfällig für Fehler durch Übersprechen oder andere Kopplungseffekte. Dabei genügt das in diesem Kapitel verwendete einfache Leitungsmodell nicht mehr zur Beschreibung der Leitungseigenschaften dieser Inter-Core-Busse [DCK⁺01]. Außerdem sind die Inter-Core-Busse wegen ihrer Länge und dem parallelen Verlauf besonders anfällig für Fertigungsfehler. Untersuchungen haben gezeigt, dass bis zu 80 % aller Defekte, die durch das Haftfehlermodell (siehe Abschnitt 3.1.1) beschrieben werden können, Inter-Core-Busse betreffen [KMV01].

1.2.3 Leitungsmodelle

Beim Schaltungsentwurf konnten Leitungen oder Verbindungsnetze anfangs als ideale Verbindungen zwischen idealisierten logischen Gattern betrachtet werden. Durch die fortschreitende Miniaturisierung und die damit verbundene Skalierung der Leitungen zu immer kleineren geometrischen Dimensionen sind die Übertragungseigenschaften von Leitungen nicht mehr zu vernachlässigen. Vielmehr treten Effekte wie endliche Signallaufzeiten, Dispersion, Reflexion und elektromagnetische Kopplung auf, die die Signale signifikant von ihrer idealen Form abweichen lassen. Für die Signalübertragung bedeuten diese Effekte kritische Auswirkungen wie Leitungsverzögerungen, Übersprechen oder mehrfaches Überschreiten der Schaltschwelle. Die Annahme idealer Verbindungen führt also zu Unsicherheiten oder sogar zu Fehlern beim Schaltungsentwurf.

Um die Funktionsfähigkeit einer integrierten Schaltung zu gewährleisten, muss beim Schaltungsentwurf die Signalintegrität erhalten werden, indem die unerwünschten Effekte innerhalb von Toleranzbereichen gehalten werden, die von der Schaltungstechnik abhängen. Aktuelle Entwurfswerkzeuge berücksichtigen daher die nicht idealen Eigenschaften von Leitungen, indem sie je nach den Anforderungen unterschiedlich genaue Modelle für verschiedene Synthese- und Analyseschritte verwenden.

Bei elektrisch kurzen Leitungen, bei denen die Bedingung erfüllt ist, dass die geometrische Leiterlänge kleiner als ein Zehntel der Wellenlänge der höchsten relevanten Signalfrequenz ist, genügen Leitungsmodelle mit konzentrierten Widerständen, Kapazitäten und zum Teil auch Induktivitäten [AN01]. Bei längeren Leitungen oder höheren Signalfrequenzen verwendet man in der Regel ein Modell mit verteilten Leitungsbelägen, die auf der Annahme der Ausbreitung einer Quasi-TEM-Welle auf der Leitung beruhen. Wegen des Auftretens des Skin-Effekts und des Proximity-Effekts müssen die Leitungsbeläge bei noch höheren Frequenzen als frequenzabhängig behandelt werden, um zu einer ausreichenden Modellierungsgenauigkeit zu kom-

men [Gra91]. Bei Schaltgeschwindigkeiten im Gigahertzbereich müssen schließlich die zweidimensionalen⁴ Leitungsmodelle fallengelassen werden, um Effekte wie Abstrahlung modellieren zu können.

Diese Arbeit behandelt Inter-Core-Leitungssysteme, deren Länge im Bereich von etwa 1 mm bis 2 cm liegt und auf denen Signale mit Frequenzanteilen bis 10 GHz übertragen werden. Für diese Leitungen ist die Modellierung mit verteilten Leitungsbelägen, insbesondere die Hinzunahme von Induktivitätsbelägen, und zum Teil die Berücksichtigung von frequenzabhängigen Leitungsparametern notwendig [NTO⁺98, GTA⁺99, DKR⁺97, DCK⁺01, CDBS01, BM02].

Typ	Jahr	Strukturgröße	Betriebsfrequenz f	Chipfläche A_{Die}
PentiumII	1998	0,25 μm	333 MHz	131 mm ²
PentiumIII	1999	0,18 μm	500 MHz	106 mm ²
Pentium4	2001	0,13 μm	1400 MHz	146 mm ²
Pentium4	2004	0,09 μm	2800 MHz	109 mm ²

Tabelle 1.1: Technologiedaten verschiedener Intel Mikroprozessoren für Desktop-Computer [Int04]

Typ	t_r	f_{max}	l_c	$\sqrt{A_{Die}}$
PentiumII	150 .. 300 ps	1,16 .. 2,33 GHz	6,5 .. 13 mm	11,4 mm
PentiumIII	100 .. 200 ps	1,75 .. 3,5 GHz	4,3 .. 8,6 mm	10,3 mm
Pentium4	35 .. 70 ps	5 .. 10 GHz	1,5 .. 3 mm	12,1 mm
Pentium4	18 .. 35 ps	10 .. 20 GHz	0,65 .. 1,3 mm	10,4 mm

Tabelle 1.2: Kritische Leitungslängen l_c für verschiedene Intel Mikroprozessoren

Eine einfache Abschätzung [AN01] für verschiedene Technologiegenerationen zeigt, dass Inter-Core-Verbindungsleitungen nicht als elektrisch kurz angesehen werden können. Als Beispiel seien vier Generationen von Intel Pentium Prozessoren gewählt⁵ [Int04]. Tabelle 1.1 enthält charakteristische Technologiedaten wie das Einführungsjahr, die minimale Strukturgröße, die Betriebsfrequenz f und die Chipfläche A_{Die} .

Tabelle 1.2 stellt von den Technologiedaten abgeleitete Größen dar wie die Signalanstiegszeit t_r bei der Betriebsfrequenz f , die signifikante Grenzfrequenz⁶ f_{max} für das Spektrum eines Signals mit der Betriebsfrequenz f , sowie die kritische Leitbahnlänge l_c , ab der eine Leitbahn nicht mehr als elektrisch kurz angesehen werden kann.

⁴ eine räumliche Dimension und die zeitliche Dimension

⁵ Intel Mikroprozessoren sind kein typisches Beispiel für Systems-on-Chip. Die Verbindungsnetze von Datenpfaden und integriertem Cache-Speicher können aber als Inter-Core-Verbindungsnetze verstanden werden.

⁶ Die Frequenzanteile eines Signals oberhalb der signifikanten Grenzfrequenz sind kleiner als 15 %.

Diese Größen werden folgendermaßen abgeschätzt:

$$t_r \approx 0,05..0,1 \cdot \frac{1}{f} \quad (1.1)$$

$$f_{max} \approx 0,35 \cdot \frac{1}{t_r} \quad (1.2)$$

$$l_c \approx \frac{1}{10} \lambda = \frac{1}{10} \frac{v}{f_{max}}. \quad (1.3)$$

Darin ist v die Phasengeschwindigkeit der sich auf der Leitung ausbreitenden elektromagnetischen Welle mit

$$v = \frac{1}{\sqrt{\mu\epsilon}}. \quad (1.4)$$

Die Leitung wird von Siliziumdioxid (SiO_2) mit einer relativen Permittivität $\epsilon_r \approx 4$ umgeben. Die Phasengeschwindigkeit v ist damit etwa halb so groß wie im Vakuum. Für die zweite Generation der Pentium4-Prozessoren kann man von einem Dielektrikum mit einem reduzierten $\epsilon_r \approx 3$ ausgehen. Dadurch steigert sich die Phasengeschwindigkeit um 15 %. Die kritische Leitbahnlänge l_c liegt damit bei allen relevanten Technologiegenerationen in derselben Größenordnung wie die Länge von globalen Leitungen ($\sqrt{A_{Die}}$). Daher muss die Ableitung und Bewertung der für den Test dieser Leitungen verwendeten Fehlermodelle anhand von Leitungsmodellen mit verteilten Leitbahnbelägen erfolgen.

Die Parameter der verschiedenen Leitungsmodelle können durch Messungen oder durch Berechnungen aus einer idealisierten Leitergeometrie und den Materialparametern gewonnen werden. Direkte Messungen an mikroelektronischen Schaltungen sind wegen der kleinen Strukturgrößen sehr schwierig. Mithilfe von Netzwerkanalysatoren und Waferprobern können beispielsweise Streuparameter für Einzel- oder Doppelleitungen gemessen werden, aus denen die Leitungsparameter berechnet werden können. Dabei ist es insbesondere kritisch, die Einflüsse des Messsystems zu minimieren. Dies wurde zum Beispiel erfolgreich in [AWW⁺00] für asymmetrische gekoppelte Doppelleitungen gezeigt.

Diese Messungen haben den Nachteil eines großen Messaufwands und können bisher nicht mehr als zwei gekoppelte Leitungen behandeln. Deshalb sind Verfahren entwickelt worden, die die Messung von Leitungseigenschaften wie Verzögerung oder Übersprechen durch Verwendung von auf einem Testchip integrierten Schaltungen erlauben.

In [CDBS01] wird eine Schaltung zur Messung von Signalen auf gekoppelten Leitungen beschrieben, die die Autoren als On-Chip-Oszilloskop bezeichnen. Diese Schaltung wird zusammen mit den zu untersuchenden Leitungsteststrukturen auf einem Chip integriert. Mit Hilfe einer Abtast- und Halte-Schaltung kann ein Messwert zu einem bestimmten Zeitpunkt ermittelt werden. Der zeitliche Verlauf eines Signals kann

rekonstruiert werden, indem durch ein externes Triggersignal der Messzeitpunkt verschoben wird. Damit können bei Leitungsteststrukturen mit bis zu fünf gekoppelten Leitungen Übersprechen und Leitungsverzögerung im Zeitbereich mit einer Auflösung von 10 mV und 5 ps gemessen werden. Diese Untersuchungen zielen auf die Charakterisierung und Optimierung der Leitungsgeometrie einer neuen Prozessgeneration ab. Sie erfordern die exakte Steuerung durch externe analoge Spannungsquellen und die externe Messung von analogen Spannungspegeln.

Auch die in Kapitel 6 behandelte Testschaltung liefert einen Beitrag zu den integrierten Messverfahren für Leitungskopplung. Diese Schaltung basiert auf dem in dieser Arbeit beschriebenen Leitungsselbsttest, so dass mit einem automatischen Testsystem für digitale Schaltungen (Digital-Tester) Verzögerungswerte für ein Bussystem mit realistischer Anzahl von Leitungen und relevanter Leitungslänge bestimmt werden können.

Das bessere Verständnis der Leitungseigenschaften, die stetige Verbesserung der Modelle und Simulationsmethoden und ihre Anwendung in der Schaltungsentwicklung ermöglichen es, trotz der mittlerweile eingetretenen Dominanz der Leitungseffekte bei aktuellen integrierten Schaltungen funktionsfähige Systeme zu entwerfen.

Wie andere Komponenten einer integrierten Schaltung sind Leitungen jedoch prinzipiell anfällig für physikalische Defekte, die bei der Fertigung nicht vollständig vermieden werden können. Der überwiegende Anteil von Brückenfehlern beispielsweise betrifft Busse [KMV01]. Neben diesen punktuellen Fehlern können physikalische Defekte die elektrischen Eigenschaften einer Leitung verändern, so dass ihr Verhalten vom spezifizierten Verhalten abweicht. Ist diese Abweichung ausreichend groß, tritt in der integrierten Schaltung ein logischer Fehler auf, indem in einer Zelle der Schaltung ein falscher Zustand gespeichert wird. Daher wird ein gezielter Test der Verbindungsnetze notwendig, der insbesondere auch prüft, ob die Verzögerungen trotz Schwankungen im Fertigungsprozess noch innerhalb der Spezifikation liegen.

Kapitel 2 gibt zunächst einen allgemeinen Überblick über die Problematik des Tests von Systems-on-Chip und stellt den Standard IEEE 1500 vor. Anschließend wird in Kapitel 3 auf den Test von Verbindungsnetzen eingegangen und der Stand der Technik dargestellt. In Kapitel 4 wird ein großflächig integriertes Multiprozessorsystems für die Videosignalverarbeitung beschrieben. Dieses motivierte die Neuentwicklung eines Testverfahrens für Inter-Core-Busse, das Signalübertragungseigenschaften wie Kopplung und Verzögerung besonders berücksichtigt. Ein solches Testverfahren, das sich speziell auf Inter-Core-Busse bezieht, wird in Kapitel 5 entwickelt und seine Integration als Selbsttest in den Standard IEEE 1500 beschrieben. Kapitel 6 stellt schließlich die Implementierung und messtechnische Untersuchung des neuen Testverfahrens dar.

2 Test von Systems-on-Chip

Der Test von integrierten Schaltungen ist notwendig, um die Qualität des Produkts „Integrierte Schaltung“ zu gewährleisten. Die Komplexität des Fertigungsprozesses und die Anzahl der hergestellten Strukturen sind so hoch, dass Fertigungsfehler nicht vollständig vermeidbar sind. Die Aufgabe eines Tests ist es daher, in der Gesamtmenge der gefertigten Schaltungen die Menge von Schaltungen zu finden, die hinsichtlich bestimmter Kriterien fehlerfrei sind [ABF95, BA00].

Solche Kriterien können von Fehlermodellen abgeleitet werden. Fehlermodelle beschreiben die Wirkung von Fertigungsfehlern, auch als Defekte oder physikalische Fehler bezeichnet, auf verschiedenen Abstraktionsebenen der Schaltungsbeschreibung. Diese Wirkung ist eine Abweichung vom spezifizierten Schaltungsverhalten, die als Fehler bezeichnet wird. Typische Fehlermodelle sind:

- Einzelhaftfehlermodell: Der physikalische Fehler wird modelliert, indem einer Signalleitung ein fester logischer Wert (0 oder 1) zugewiesen wird. Die Signalleitung führt zu einem Eingang eines Gatters oder eines Flip-Flops. Dieses Modell beschreibt auf der Gatterebene beispielsweise eine ungewollte leitende Verbindung zwischen einer Signalleitung und einer Versorgungsspannungsleitung.
- Brückenfehlermodell: Der physikalische Fehler wird auf Gatter- oder Transistorebene als Kurzschluss zweier Signalleitungen beschrieben, die den gleichen logischen Wert annehmen. Der logische Wert der kurzgeschlossenen Signalleitungen wird durch eine dominierende logische 1 (OR-Short) oder durch eine dominierende logische 0 (AND-Short) modelliert.
- Verzögerungsfehlermodell: Der physikalische Fehler wird auf Gatterebene dadurch modelliert, dass die Signalverzögerung der kombinatorischen Schaltung zwischen zwei Speicherelementen größer als die Taktperiode wird. Ein Signalwechsel wird daher mit einem Takt Verspätung übernommen. Dieser Fehlertyp wird bereits in der Einleitung beschrieben.

Für andere Schaltungstypen oder -komponenten wie Speicher oder Leitungen werden andere spezielle Fehlermodelle verwendet.

Die wichtigsten Tests für integrierte digitale Schaltungen sind naturgemäß digital. Dabei werden digitale Eingangsmuster, die sogenannten Testmuster, an die

zu testende Schaltung angelegt und die tatsächliche Antwort der Schaltung, die sogenannte Testantwort, an ihren Ausgängen gemessen und mit der erwarteten verglichen. Eine Differenz bedeutet, dass die Schaltung fehlerhaft ist. Das Erzeugen und das Auswerten von Testmustern und Testantworten wird mit Hilfe eines Digital-Testers durchgeführt.

Um einen Fehler in einer kombinatorischen Schaltung zu finden, müssen die Eingänge der Schaltung so eingestellt werden, dass im Fall eines Fehler ein Unterschied zwischen dem fehlerfreien und dem fehlerhaften Zustand am Fehlerort auftritt. Legt man zum Beispiel das Einzelhaftfehlermodell mit dem Wert 1 zugrunde, muss eine logische Null auf der Signalleitung erzeugt werden, um den Fehler zu steuern. Zugleich muss die Einstellung der Eingänge so erfolgen, dass der fehlerhafte Zustand an den Ausgängen der Schaltung beobachtet werden kann.

Die Bestimmung von Testmustern für kombinatorische Schaltungen ist ein NP-vollständiges Problem. Für diese Aufgabe sind ausgehend vom ursprünglichen D-Algorithmus [Rot66] Heuristiken entwickelt worden, die die automatische Testmusterberechnung auch für große kombinatorische Schaltungen mit einigen 100000 Gattern ermöglichen [BA00]. Ein Test heißt deterministisch, wenn die Testmuster ausgehend von Fehlermodellen berechnet werden.

Die meisten Schaltungen enthalten Speicherelemente und sind daher sequentiell. Für sequentielle Schaltungen ist die Testmusterberechnung noch komplexer, weil nicht nur ein Testmuster gefunden werden muss, das den Fehler steuert und beobachtbar macht, sondern weil zuvor die Schaltung in einen Zustand gebracht werden muss, in dem der Fehler überhaupt steuerbar ist. Für Fehler in sequentiellen Schaltungen muss also eine Sequenz von Testmustern gefunden werden. Eine weitere Sequenz von Testmustern ist notwendig, um den Fehlereffekt an den Ausgängen beobachtbar zu machen.

Eine wesentliche Vereinfachung des Testproblems für sequentielle digitale Schaltungen stellt die Einführung von Scan-Pfaden dar [ABF95, BA00]. Dabei werden die Speicherelemente durch Multiplexer erweitert und können dadurch im Testmodus zu einer Kette verbunden werden, deren Anfang und Ende nach außen zugänglich sind. Die sequentielle Schaltung wird für den Test in eine kombinatorische Schaltung umgewandelt, weil der Schaltungszustand von außen seriell gesteuert und beobachtet werden kann.

Ein Scan-Test besteht aus drei Phasen. In der ersten Phase wird der Scan-Pfad seriell mit einem Testmuster beschrieben und die Schaltung dadurch in einen bestimmten Zustand gebracht. In der zweiten Phase wird ein Schaltungstakt im normalen Betriebsmodus ausgeführt und dadurch die Antwort des kombinatorischen Schaltungsteils auf das Testmuster als neuer Schaltungszustand in der Schieberegisterkette gespeichert. In der dritten Phase wird der Registerinhalt seriell über den Scan-Pfad ausgelesen und mit dem erwarteten Registerinhalt verglichen. Zugleich kann das

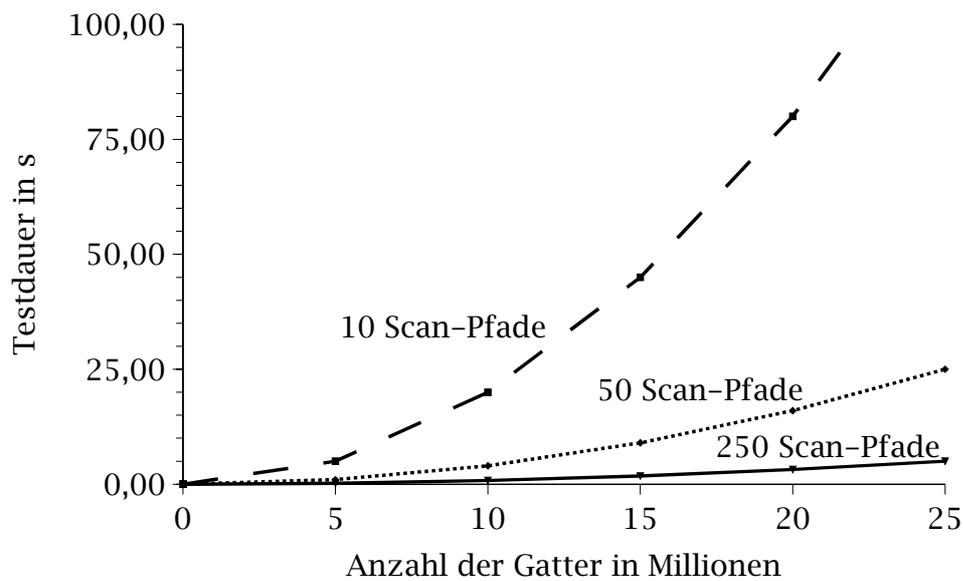


Abbildung 2.1: Testdauer beim Scan-Test mit 500 MHz

nächste Testmuster in den Scan-Pfad geschrieben werden. Zur Reduzierung der Testdauer kann ein langer Scan-Pfad in mehrere kurze Scan-Pfade aufgeteilt werden, die parallel verwendet werden. Die Anzahl der parallelen Scan-Pfade wird im Wesentlichen dadurch beschränkt, dass für jeden ein zusätzliches Paar von Schaltungsein- und -ausgang notwendig ist.

Scan-Pfade in Verbindung mit automatischer Testmusterberechnung sind das heute am meisten genutzte Verfahren für den Test digitaler Schaltungen. Es gibt verschiedene kommerzielle Software-Werkzeuge, die eingebettet in einen Design-Flow die Integration der Scan-Pfade und die Testmusterberechnung voll automatisiert durchführen.

Die Anwendung von Scan-Pfaden als primäres Testhilfsmittel ist aber bei steigender Schaltungsgröße durch die Testdauer und die Testdatenmenge beschränkt. Abbildung 2.1 zeigt die Abhängigkeit der Testdauer von der Schaltungscomplexität in Gatteräquivalenten bei einer angenommenen Taktrate von 500 MHz für verschiedene Längen der Scan-Pfade. Dabei wird eine einfache Abschätzung zugrunde gelegt, nach der erstens jedes zehnte Gatter ein Flip-Flop ist, das ein Element der Scan-Pfade wird, und zweitens die Anzahl der notwendigen Testmuster ein Hundertstel der Anzahl der Gatter beträgt. Unter diesen Annahmen ist die Testdauer proportional zur Anzahl der Gatter zum Quadrat. Daher beträgt die Testdauer für große Schaltungen selbst bei hoher Anzahl von Scan-Pfaden einige Sekunden und ist daher unerschwinglich hoch. Ebenso steigt die Testdatenmenge mit der Anzahl der Gatter zum Quadrat. Sie erreicht bei dem gezeigten Beispiel nach Abbildung 2.1 500 MByte und liegt in aktuellen Schaltungen weit darüber [BBD⁺01, KW99].

Der Scan-Test hat den weiteren Nachteil, dass er zunächst nicht für den Test auf Verzögerungsfehler geeignet ist, weil zwischen der Anwendung von zwei Testmustern jeweils eine Testphase zum seriellen Laden und Entladen der Scan-Pfade nötig ist. Für einen Test auf Verzögerungsfehler ist jedoch die Erzeugung eines bestimmten Signalübergangs notwendig, der von zwei Testmustern in direkt aufeinander folgenden Schaltungstakten erzeugt wird. Eine Möglichkeit, mit Scan-Pfaden auf Verzögerungsfehler zu testen, besteht darin, Testmuster-Paare zu berechnen, bei denen das zweite Testmuster aus dem ersten durch eine Verschiebung um ein Bit entsteht. Eine hohe Testabdeckung kann dabei aber nicht garantiert werden [BA00].

Wegen der genannten Einschränkungen ist der Scan-Test bei aktuellen Systems-on-Chip nicht praktikabel. Für den Test von Systems-on-Chip bestehen noch weitere Schwierigkeiten für die Testentwicklung und Testanwendung, die im folgenden Abschnitt kurz beschrieben werden. Die weiteren Abschnitte dieses Kapitels geben einen Überblick über Testhilfen, die einen wirtschaftlichen Test von eingebetteten Cores auf Systems-on-Chip ermöglichen.

2.1 Schwierigkeiten beim Test eingebetteter Cores

Die Entwicklung von System-on-Chip ist nur deshalb möglich, weil das Problem der Entwurfskomplexität durch die Partitionierung des Systems in funktional abgeschlossene, wiederverwendbare Cores vereinfacht wird [Cro99]. Die Entwicklung der Cores liegt in der Regel in anderen Händen als die Integration des Systems-on-Chip. Oftmals werden Cores von Fremdanbietern zugekauft und auf einem System-On-Chip integriert. Um die Rechte des Core-Anbieters zu schützen, werden dabei nicht alle detaillierten Information zur Core-Realisierung geliefert, sondern nur notwendige Daten wie ein funktionales Simulationsmodell und eine vereinfachte Layoutbeschreibung¹.

Zwangsläufig gehört zu den notwendigen Daten auch eine Beschreibung des Testverfahrens für einen Core sowie die Testmuster, weil dem Systemintegrator die notwendigen Daten zur Testentwicklung fehlen. Die Testentwicklung teilt sich daher ebenso hierarchisch auf den Systemintegrator — also den Core-Anwender — und den Core-Anbieter auf wie die Systementwicklung selbst. Der Core-Anbieter ist verantwortlich für die Entwicklung eines Test für den Core, während der Core-Anwender verantwortlich für die Integration der Tests für alle Cores seines Systems-on-Chip zu einem System-Test bleibt.

¹ Das vollständige Layout wird dagegen in einer verschlüsselten Form geliefert, die keine Rückschlüsse auf Schaltungsdetails zulässt.

Die für einen Core entwickelten Tests können jedoch nicht ohne weiteres für ein System-on-Chip genutzt werden, weil die Ein- und Ausgänge des Cores nicht mit den Ein- und Ausgängen des System-on-Chip übereinstimmen und daher nicht für die direkte Ansteuerung und Auswertung durch einen Digital-Tester zugänglich sind. Eine wesentliche Schwierigkeit beim Test von eingebetteten Cores besteht daher darin, eine große Menge von Testdaten über andere Cores zu den Core-Eingängen und von den Core-Ausgängen zu transferieren.

Zugleich sind die Core-Ausgänge fest mit den Eingängen anderer Cores verbunden. Beim Test eines Cores können ungültige Befehle und Daten erzeugt werden, die andere Cores beeinflussen. Cores müssen daher für den Test isoliert werden. Durch den Testdatentransfer über gleiche Ressourcen zu unterschiedlichen Cores und die Notwendigkeit der Testisolierung können nicht alle Cores gleichzeitig getestet werden. Ziel des Systementwicklers ist es daher, eine Reihenfolge und Zuordnung der Core-Tests zu finden, die die Testkosten minimiert [ICM02, LP02, Kor02]. Dabei sind die Reduzierung der Testdauer und die Verringerung des Schaltungsoverheads für Testhilfen sich widersprechende Optimierungsziele.

Wegen der Gesamtkomplexität von Systems-on-Chip sind etablierte Testhilfen wie Scan-Pfade praktisch nicht nutzbar, weil die Einschränkungen bezüglich Testdauer und Testdatenmenge besonders schwerwiegend sind.

2.2 Testfreundlicher Entwurf eingebetteter Cores

Um ein System-on-Chip trotz der oben beschriebenen Schwierigkeiten wirtschaftlich testbar zu machen, müssen zwei Voraussetzungen erfüllt werden: Bereitstellung eines Zugriffsmechanismus auf den Core für den Test und Reduktion der zum Core transferierten Testdatenmenge bei gleichbleibender Testqualität.

Die Schwierigkeiten können umgangen werden, indem die Testmustererzeugung und Testauswertung im System-on-Chip selbst und nicht durch einen Digital-Tester durchgeführt werden.

Die Vorteile des integrierten Selbsttests (BIST, Built-In Self-Test) sind vielfältig und machen ihn daher zur ersten Wahl für den Test von eingebetteten Cores. Erstens wird keine hohe Bandbreite² für den Datentransfer vom Digital-Tester zum Core benötigt. Damit können auch Cores mit sehr schlechter Zugänglichkeit ausreichend getestet werden. Außerdem können kostengünstige Digital-Tester mit beschränkter Leistungsfähigkeit verwendet werden. Zweitens existieren keine Differenzen bei der

² hier Datenmenge pro Zeit

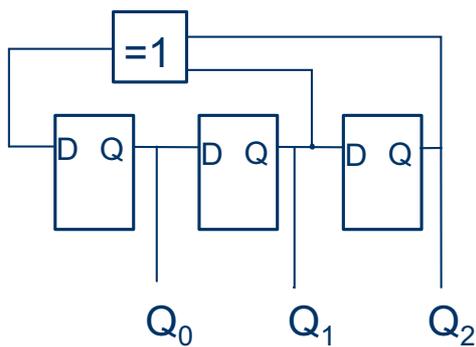


Abbildung 2.2: Linear rückgekoppeltes Schieberegister (LFSR)

Geschwindigkeit und Genauigkeit von Testobjekt und Tester. Solche Differenzen sind potenzielle Quellen für Qualitätseinbußen beim Test. Insbesondere ist ein Test auf Verzögerungsfehler bei maximaler Betriebsfrequenz möglich.

Drittens besteht die Möglichkeit, jede beliebige Leitung direkt zu steuern oder zu beobachten. Dadurch kann die Testqualität gesteigert werden. Viertens können integrierte Selbsttests nicht nur für den Produktionstest, also direkt nach der Fertigung, sondern auch nach der Montage auf einer Leiterplatte genutzt werden. Dadurch kann die Zuverlässigkeit eines elektronischen Systems in regelmäßigen Abständen während des Betriebs geprüft werden.

Die Integration von Selbsttest ist mit höheren Kosten verbunden. Dies betrifft sowohl die Fertigungskosten durch den höheren Flächenbedarf einer Schaltungen mit Selbsttest als auch die Entwicklungskosten durch den höheren Aufwand für die Testentwicklung und Testverifikation im Vergleich zum Scan-Test. Die oben genannten Vorteile führen aber zu der Entscheidung, den Leitungstest ebenfalls als integrierten Selbsttest zu entwerfen.

2.2.1 Testmustererzeugung und Testauswertung

Die Testmustererzeugung kann effektiv mit einem linear rückgekoppelten Schieberegister (LFSR) oder mit einem zellularen Automaten (CA) erfolgen. Unter bestimmten Voraussetzungen erzeugen diese Schaltungen pseudo-zufällige Testmuster. Diese Testmuster haben Eigenschaften von echten Zufallsmustern, wie z.B. eine Gleichverteilung von Nullen und Einsen, werden aber deterministisch erzeugt [Gol82].

Das in Abbildung 2.2 gezeigte LFSR besteht aus $N = 3$ Flip-Flops und heißt daher 3-stufig. Es kann maximal $2^N - 1 = 7$ verschiedene Zustände annehmen, die als Testmuster genutzt werden. Für eine hohe Testqualität ist es sinnvoll, LFSRs mit

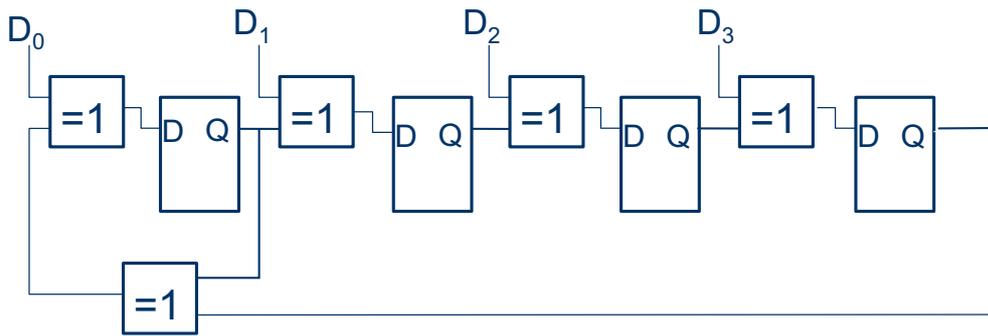


Abbildung 2.3: Linear rückgekoppeltes Schieberegister mit Mehrfacheingängen (MISR)

maximaler Anzahl von Zuständen zu implementieren. Das ist der Fall, wenn die Rückkopplung durch ein so genanntes primitives Polynom beschrieben werden kann.

Beim Test mit pseudo-zufälligen Mustern erweisen sich nur wenige Fehler als schwer zu testen, so dass bereits mit wenigen tausend Testmustern ein Anteil von weit mehr als 90% der Fehler gefunden werden kann. Die Anzahl von automatisch berechneten Testmustern ist deutlich geringer, um die gleiche Fehlermenge zu finden. Außerdem können gezielt Test für die schwer zu testenden Fehler generiert werden. Die Qualität von pseudo-zufälligen Testmustern muss durch Fehlersimulation geprüft werden [BA00]. Beim Selbsttest mit pseudo-zufälligen Mustern fallen enorm hohe Datenmengen für die Testauswertung an. Daher müssen die Testantworten der Schaltung komprimiert werden. Eine effektive Lösung dafür stellen wiederum lineare Schieberegister dar, bei denen jeder neue Zustand durch XOR-Verknüpfung der Testantworten mit dem aktuellen Registerzustand entsteht. Abbildung 2.3 zeigt ein 4-stufiges linear rückgekoppeltes Schieberegister mit Mehrfacheingang (Multiple Input Signature Register, MISR).

Die Testauswertung erfolgt durch Vergleich des Registerzustandes nach Ablauf des Tests, der so genannten Testsignatur, mit dem erwarteten Registerzustand, der so genannten Gutsignatur. Diese muss vorab durch eine Schaltungssimulation ermittelt werden. Die Komprimierung der Testantworten zu einer einzelnen Testsignatur ist mit Informationsverlust verbunden, durch den ein Fehler, der in den Testantworten erkennbar ist, zu einer Testsignatur führt, die identisch mit der Gutsignatur ist. Dieser Fall wird als Aliasing bezeichnet. Die Wahrscheinlichkeit für das Auftreten von Aliasing tendiert für N -stufige MISR zu 2^{-N} für große N , und ist mithin sehr gering [WDGS88].

Abbildung 2.4 zeigt die einfachste Selbsttestarchitektur für eingebettete Cores, die so genannte STUMPS-Architektur³ [BM82]. Die Eingänge der Scan-Pfade des Cores

³ Self-Test Using MISR and Parallel SRSG (Shift Register Sequence Generator)

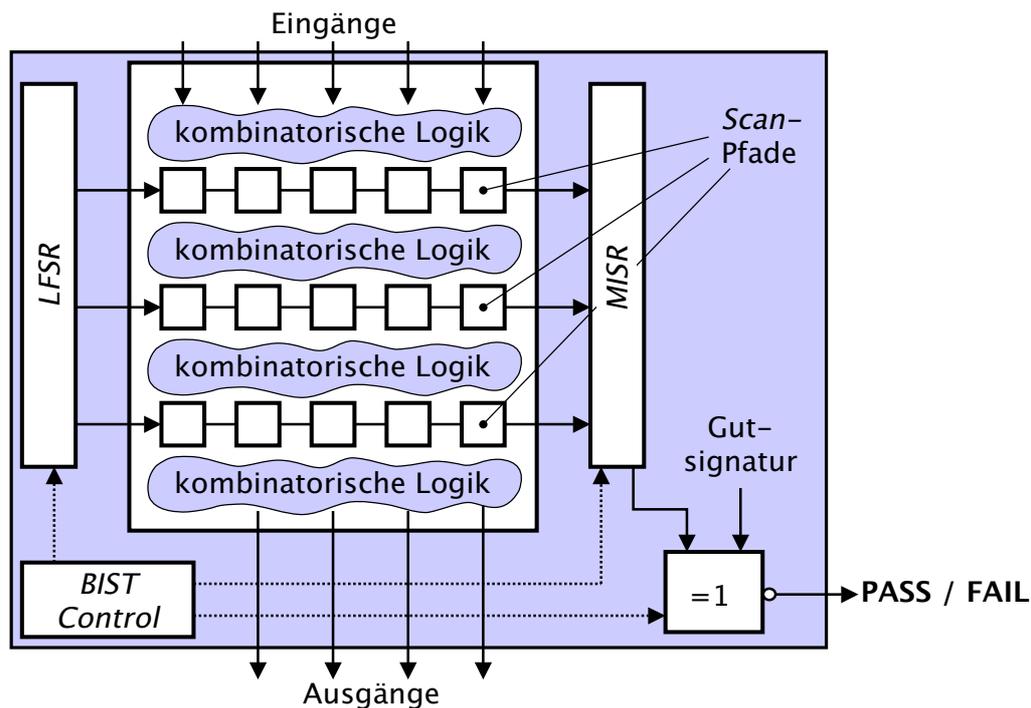


Abbildung 2.4: Selbsttest mit LFSR und MISR (STUMPS)

werden von einem LFSR gesteuert, während die Ausgänge mit einem MISR verbunden sind. Diese Architektur beseitigt die Schwierigkeit des Testzugriffs auf den Core, nicht aber das Problem der Testdauer.

2.2.2 Selbsttest mit BILBOs

Eine wesentliche Verbesserung der Testdauer, aber auch der Testqualität wird durch die BILBO-Architektur nach Abbildung 2.5 erreicht. Dabei werden alle Schieberegister durch so genannte BILBOs⁴ ersetzt [KMZ80]. BILBOs sind erweiterte Schieberegister, die durch wenige zusätzliche Gatter die Funktionen paralleles Register, Schieberegister, LFSR oder MISR ausführen können. Den Aufbau eines BILBOs zeigt Abbildung 2.6.

Bei dieser Architektur stimulieren die Testmuster parallel die Eingänge der kombinatorischen Schaltungsteile. Dadurch wird mit jedem Taktzyklus ein neues Testmuster an die Schaltung angelegt, so dass ein Test auf Verzögerungsfehler prinzipiell möglich ist. Die Testauswertung erfolgt durch serielles Auslesen der BILBO-Signatur und Vergleich mit der Gutsignatur.

⁴ Built-In Logic Block Observer

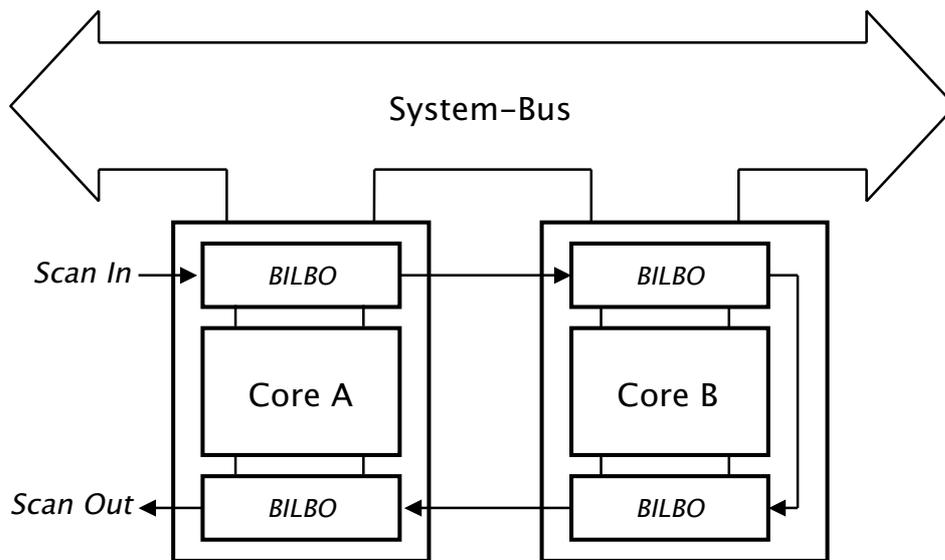


Abbildung 2.7: Selbsttest von Inter-Core-Bussen mit BILBOs

Mit Hilfe der BILBO-Architektur ist auch ein pseudo-zufälliger Test der Inter-Core-Busse bei Systems-on-Chip möglich. Nach Abbildung 2.7 werden die Ein- und Ausgänge der Cores mit BILBOs versehen. Während genau ein Core pseudo-zufällige Muster erzeugt, werden die Signale am Ende der Leitungen von den anderen Cores komprimiert, so dass schließlich eine Testsignatur über alle Cores seriell ausgelesen werden kann.

Die Implementierung der BILBO-Architektur verursacht im Vergleich der beschriebenen Verfahren den höchsten Schaltungsoverhead. Außerdem sind keine Werkzeuge kommerziell verfügbar, die eine Implementierung voll automatisiert durchführen.

2.2.3 Selbsttest mit programmierbaren Mikroprozessor-Cores

Viele Systems-on-Chip enthalten programmierbare Mikroprozessor-Cores. Durch geeignete Programmierung können diese zur Erzeugung von pseudo-zufälligen oder vorab berechneten Testmustern sowie zur Auswertung der Testantworten genutzt werden [RT97]. Die Testentwicklung ist zeitintensiv, weil die Programmierung in der Regel manuell erfolgen muss. Der Test des Mikroprozessor-Cores selbst muss auf andere Art gelöst werden.

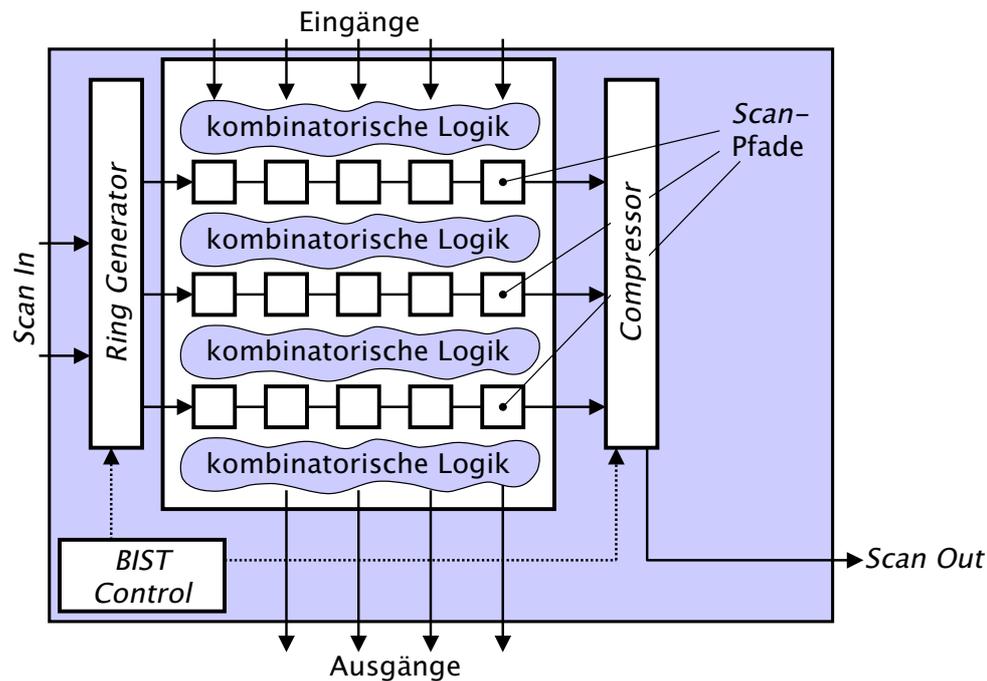


Abbildung 2.8: Eingebetter deterministischer Test

2.2.4 Eingebetter deterministischer Test

Ein Zwitter zwischen den beschriebenen Selbsttest-Architekturen, die pseudo-zufällige Testmuster benutzen und dem externen Test mit deterministischen Testmustern stellt der so genannte eingebettete deterministische Test nach Abbildung 2.8 dar [RTK⁺02]. Diese Testarchitektur ist prinzipiell nur eine Variante der STUMPS-Architektur. Anstelle eines LFSR wird jedoch ein so genannter Ring-Generator zur Testmustererzeugung verwendet. Ein Ring-Generator ist ein zum Ring geschlossenes Schieberegister mit nicht linearer Rückkopplung. Der Ring-Generator erzeugt vorab berechnete, also deterministische Testmuster. Aus den deterministischen Testmustern sowie der Rückkopplung lässt sich ein Gleichungssystem aufstellen, das die Berechnung der notwendigen Eingangssignale des Ring-Generators erlaubt.

Der Vorteil liegt in der drastisch reduzierten Datenmenge der Eingangssignale und Testantworten im Vergleich zur Datenmenge beim externen Test sowie in der Verwendung von deterministischen Testmustern im Gegensatz zu den echten Selbsttestverfahren. Der eingebettete deterministische Test kann jedoch nicht vollständig autonom ohne einen Digital-Tester ausgeführt werden. Für die Integration des eingebetteten deterministischen Tests steht ein kommerzielles Entwurfswerkzeug zur Verfügung. Daneben gibt es weitere vergleichbare Verfahren, die ebenfalls auf der Dekomprimierung von Testmustern auf dem Chip beruhen [CC01, KVMW01]. Außerdem können mit Hilfe nicht-linear rückgekoppelter Schieberegister deterministische

Testmuster ohne zusätzliche Stimulierung durch Eingangssignale erzeugt werden. Die notwendigen Rückkopplungsschaltungen können jedoch sehr aufwändig sein [Dae82].

Die dargestellten Testverfahren für eingebettete Cores stellen mögliche Lösungen für die Probleme Testdatenmenge und Testdauer dar, so dass der Schaltungstest nicht vorrangig die Komplexität von Systems-on-Chip einschränkt. Auf spezielle Testverfahren für eingebettete Speicher [van98, Nig03] oder analoge Schaltungen [BR01] soll hier nicht weiter eingegangen werden.

2.3 IEEE 1500 Standard für den Test Core-basierter Systeme

Die vorausgehenden Abschnitte haben gezeigt, dass es viele verschiedene Lösungen für den Test bzw. den Selbsttest von eingebetteten Cores gibt. Die Auswahl und Implementierung einer dieser Lösungen für einen bestimmten Core hängen oft von den Details seiner Implementierung selbst ab. Diese Details sind in der Regel nur dem Anbieter des Core, nicht aber dem Anwender bekannt. Daher muss der Core-Anbieter bei der Testentwicklung für ein System-on-Chip Unterstützung leisten. Im Idealfall liefert der Core-Anwender einen definierten Test für den Core an den Core-Anwender.

Das Ziel der Entwicklung des Standards IEEE 1500 [IEE05] für den Test Core-basierter Systeme ist es, den notwendigen Datenaustausch zwischen Core-Anbieter und Core-Anwender zu standardisieren. Dadurch soll es möglich sein, den Test für einen Core im Kontext vieler unterschiedlicher Systems-on-Chip wiederzuverwenden [MKL⁺02].

Der Standard IEEE 1500 definiert daher ein standardisiertes Modell für den Austausch von Testinformationen. Dazu wird die Core Test Language (CTL, IEEE Std. 1450.6) verwendet, die eine Erweiterung der Standard Test Interface Language (STIL, IEEE Std. 1450.0) [IEE99] ist. STIL ist eine generische Sprache, die für den Datenaustausch zwischen Digital-Tester und Entwurfsautomatisierungswerkzeugen entwickelt wurde. Sie erlaubt im Wesentlichen die Beschreibung von Testvektoren und Testsignalen. CTL fügt Konstrukte hinzu, die für den Test von eingebetteten Cores spezifisch sind.

Ein Core ist typischerweise in ein System-on-Chip eingebettet, d.h. seine Ein- und Ausgänge sind nicht nach außen über Gehäusepins zugänglich. Um einen Test des Cores durchzuführen, wird eine Testinfrastruktur benötigt, die die Quelle der Testdaten (BIST oder Digital-Tester) mit den Eingängen des Cores verbindet und die die

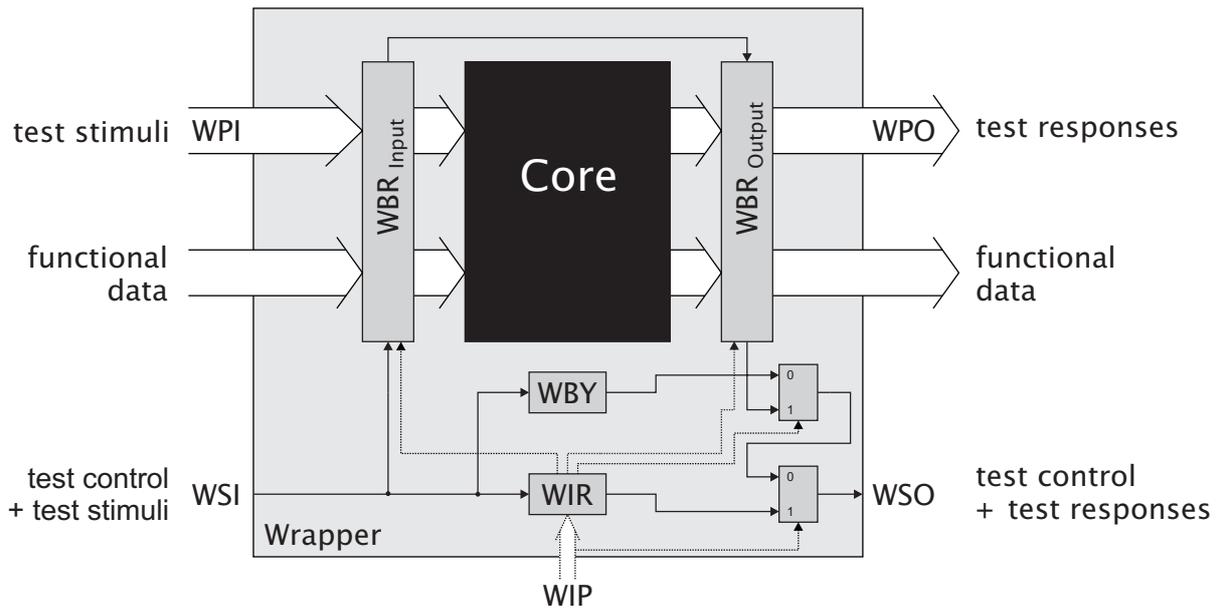


Abbildung 2.9: Architektur eines IEEE 1500 Core Test Wrappers nach [MKL⁺02]

Ausgänge des Cores mit der Testauswertung verbindet (BIST oder Digital-Tester). Diese Infrastruktur wird auf Seiten des Cores ebenfalls durch IEEE 1500 standardisiert. Sie ist vergleichbar mit dem Standard IEEE 1149.1 für den Leiterplattentest [IEE01]. Dabei wird jedoch nicht das Testverfahren für den Core und die Implementierung dieser Infrastruktur, also die Testhardware, standardisiert, sondern nur eine Menge von Operationen, die von der Hardware ausgeführt werden muss.

Als Testinfrastruktur für einen Core wird ein Core Test Wrapper definiert. Dieser Wrapper ist eine Schale um den Core und erlaubt drei Betriebsmodi:

- **Systembetrieb:** Der Wrapper ist transparent, d.h. der Core verhält sich, als ob der Wrapper nicht vorhanden wäre.
- **Nach innen gerichteter Test:** Über den Wrapper kann der Test des Cores durchgeführt werden.
- **Nach außen gerichteter Test:** Der Core ist isoliert und über den Wrapper können Leitungssysteme und Logik außerhalb des Cores getestet werden.

Abbildung 2.9 zeigt ein Blockschaltbild der Architektur eines Core Test Wrappers. Der Block Core stellt den eingebetteten Core dar, der zu testen ist. Der Wrapper umgibt diesen Core wie eine Schale. Er besitzt dieselben funktionalen Eingangs- und Ausgangsports wie der Core selbst. Weiterhin muss der Wrapper ein Paar eines 1 bit breiten Eingangs- und Ausgangsports WSI und WSO (Wrapper Serial Input und Wrapper Serial Output) besitzen. WSI wird verwendet, um seriell Testdaten zu laden, während

WSO genutzt wird, um Testdaten seriell vom Wrapper zu lesen. Optional besitzt ein Core Test Wrapper auch entsprechende parallele Ports WPI (Wrapper Parallel Input) und WPO (Wrapper Parallel Output, deren Breite vom Anwender bestimmt werden kann. Mit Hilfe dieser seriellen und parallelen Ports können in flexibler Weise Randbedingungen an die Testdatenbandbreite erfüllt werden.

Schließlich besitzt der Wrapper noch den Port WIP (Wrapper Interface Port), der für die Steuerung des Wrappers genutzt wird. Dieser Port besteht aus sechs Signalen, die exklusiv für die IEEE 1500 Wrapper verwendet werden müssen. Der WIP steuert einen Block WIR, das Wrapper Instruction Register. Werden in dieses Register Testinstruktionen geladen, wird ein entsprechender Testmodus aktiv. Im Falle eines nach innen gerichteten Tests muss der Wrapper beispielsweise die Steuerbarkeit der Core-Eingänge und die Beobachtbarkeit der Core-Ausgänge gewährleisten.

Diese Funktion wird durch die Blöcke WBR (Wrapper Boundary Register) gewährleistet. Diese Register umfassen alle Ein- und Ausgänge des Core. Für die Implementierung eines Selbsttests ist von Bedeutung, dass durch die WBR alle Ein- und Ausgänge mit einem Register versehen sind.

Die Architektur enthält außerdem einen Block WBY, das Wrapper Bypass Register. Dieses Register dient dazu, den seriellen Testzugriff über die WBR eines Cores zu umgehen, um so die Kette aller WBR der Cores auf einem System-on-Chip zu verkürzen.

Der Standard IEEE 1500 sieht zwei Möglichkeiten vor, einen Core standardkonform zu machen. Die erste Möglichkeit wird mit IEEE 1500 wrapped bezeichnet. Dabei beinhaltet der Core schon die vollständige Funktionalität des Wrappers. Die zweite Möglichkeit wird mit IEEE 1500 unwrapped bezeichnet. Dabei beinhaltet der Core noch keinen Wrapper, wird aber mit einem CTL Programm geliefert, das beschreibt, wie ein Wrapper zum Core hinzugefügt werden muss. Bei beiden Möglichkeiten muss der Test als CTL Programm beschrieben werden.

Der Standard IEEE 1500 beschreibt eine flexible Testinfrastruktur für Systems-on-Chip. Obwohl der Standard mit einem nach außen gerichteten Testmodus prinzipiell den Test von Inter-Core-Bussen ermöglicht, ist ein Test auf Verzögerungs- und Kopplungsfehler praktisch nicht möglich, weil dazu mit zwei aufeinander folgenden Testmustern bestimmte Signalübergänge bei der spezifizierten Systemtakttrate erzeugt werden müssten. Der nach außen gerichtete Testmodus erfordert jedoch nach jedem Testmuster eine Betriebsphase im Scan-Modus, um das folgende Testmuster zu laden.

In dieser Arbeit wird daher eine Erweiterung des Standards IEEE 1500 vorgestellt, die einen Selbsttest für Inter-Core-Busse ermöglicht, der Verzögerungs- und Kopplungsfehler abdeckt. Damit kann eine wesentliche Lücke des Standards geschlossen werden.

3 Test von Inter-Core Verbindungsnetzen

Wie in Abschnitt 1.2 beschrieben dominiert die Verzögerung der Signale auf Leitungen, insbesondere auf Inter-Core-Verbindungsnetzen, in Zukunft die Performance einer integrierten Schaltung. Daher gibt es mittlerweile intensive Aktivitäten, die Eigenschaften von Leitungen zu erfassen und durch Modelle ausreichend genau darzustellen. Während die Durchführung von Produktionstests, um das Auftreten von Verzögerungsfehlern in integrierten Schaltungen zu prüfen, bereits industrielle Praxis ist, sind für den entsprechenden Test von Inter-Core-Verbindungsnetzen erst wenige adäquate Verfahren veröffentlicht [BDR01, YHS⁺01, NA02]. Diese Verfahren sind die Grundlage für den im Rahmen dieser Arbeit entwickelten integrierten Leitungstest und werden in diesem Kapitel vorgestellt.

3.1 Fehlermodelle für Verbindungsnetze

Elektrische Leitungen einer integrierten Schaltung können von unterschiedlichsten physikalischen Defekten betroffen sein, die verschiedenste elektrische Auswirkungen haben [AC89]. Überschüssiges Leitungsmaterial beispielsweise kann zu ungewollten elektrischen Verbindungen zweier Leitungen führen. Ein logischer Fehler kann in diesem Fall immer dann auftreten, wenn auf den betroffenen Leitungen ein unterschiedlicher Signalpegel vorliegt¹. Andererseits kann eine Leitung unterbrochen sein, wenn zum Beispiel ein Kontaktloch (*Via*) von einer Metallebene zur nächsten nicht mit genügend leitendem Material aufgefüllt ist. In diesem Fall kann ein Signalwechsel am Anfang der Leitung in der Regel nicht an deren Ende beobachtet werden, was zu einem logischen Fehler führt.

Neben derartigen Defekten, die sich lokal begrenzt auswirken, gibt es weitere, die größere Regionen einer Schaltung betreffen, aber in ihrer Wirkung nicht so offensichtlich sind. Fehlerhafte Dotierprofile führen zu Abweichungen des elektrischen Verhaltens von aktiven Bauelementen, z.B. einem vergrößerten Innenwiderstand eines Transi-

¹ Das tatsächliche Verhalten eines solchen Defektes hängt stark vom Ort und dem elektrischen Widerstand der Verbindung sowie von den Leitungstreibern ab [EPRB03].

stors. Ein Überätzen oder Unterätzen während der zahlreichen Lithographieschritte bei der Chipfertigung führt zu Abweichungen bei den geometrischen Abmessungen von Leitungen oder auch aktiven Elementen, so dass zum Beispiel bei Leitungen Variationen ihrer elektrischen Parameter wie Widerstand oder Kapazität auftreten. Ein solcher physikalischer Defekt führt erst in der Summe seiner Auswirkungen auf viele Elemente einer Schaltung dazu, dass ein logischer Fehler tatsächlich auftritt. Die Signalintegrität auf langen, gekoppelten Leitungssystemen wird stark von solchen Schwankungen im Fertigungsprozess beeinflusst [CGB97].

Physikalische Defekte wie die beschriebenen sind bei der Fertigung nicht vollständig zu vermeiden. Daher muss an jeder hergestellten integrierten Schaltung ein Test durchgeführt werden, der das Vorhandensein solcher Fehler möglichst vollständig prüft. Wegen ihrer wachsenden Bedeutung, auch für die Leistungsfähigkeit einer integrierten Schaltung, müssen dabei besonders Fehler berücksichtigt werden, die Inter-Core-Verbindungsnetze betreffen. Die Auswirkungen von physikalischen Defekten können wegen ihrer Vielfalt und wegen der Komplexität aktueller Schaltungen nicht vollständig mit einem Schaltkreissimulator auf elektrischer Ebene behandelt werden. Daher müssen für die Entwicklung und Bewertung von Testverfahren Fehlermodelle auf logischer oder Verhaltensebene verwendet werden, die physikalische Defekte repräsentieren.

3.1.1 Statisches Fehlermodell für Leitungssysteme

Physikalische Defekte wie überschüssiges oder fehlendes Leitbahnmaterial wirken sich permanent und auch bei sehr niedrigen Frequenzen auf das elektrische Verhalten der Leitungen aus. Diese Defekte sollen daher als statisch bezeichnet werden. Sie können daher durch ein einfaches statisches Fehlermodell auf Schaltkreis- oder auf Gatterebene beschrieben werden. Die Fehlertypen, die von dem in dieser Arbeit verwendeten statischen Fehlermodell erfasst werden, sind in den folgenden Abschnitten beschrieben.

Brückenfehler

Brückenfehler werden durch ungewollte leitende Verbindungen auf einer Metallisierungsebene oder zwischen verschiedenen Metallisierungsebenen gebildet. Ursache können beispielsweise Ablagerungen des Leitungsmaterials, also überschüssiges Leitbahnmaterial, oder Fehler der isolierenden Schichten sein. Die ohmschen Widerstände solcher Verbindungen zwischen zwei Leitungen variieren von wenigen Ohm bis zu einigen Kiloohm und beeinflussen das Verhalten solcher Brückenfehler erheblich [EPRB03]. Aber nur ein geringer Prozentsatz von Verbindungen erreicht

Widerstände über 500Ω [RMBF96]. Daher sollen Brückenfehler hier als praktisch widerstandsfrei angenommen werden und können so charakterisiert werden, dass alle beteiligten Leitungen am Eingang der nächsten Logikstufe den gleichen logischen Wert annehmen. Welchen logischen Wert die Leitungen annehmen, wird durch die verwendete Technologie, insbesondere durch die Dimensionierung der Treiber der betroffenen Leitungen, sowie durch die Lage des Kurzschlusses bestimmt. Brückenfehler mit geringem ohmschen Widerstand lassen sich einem der folgenden Typen zuordnen.

- AND-Short: Der Brückenfehler verknüpft die Leitungen zu einem logischen UND-Gatter.
- OR-Short: Der Brückenfehler verknüpft die Leitungen zu einem logischen OR-Gatter.
- Brückenfehler zwischen starken und schwachen Treibern: Die Leitung mit dem stärkeren Treiber dominiert den Pegel der anderen Leitungen.
- Weak Short: Der Brückenfehler führt zu einem Pegel auf den Leitungen, der weder einer logischen Eins noch einer logischen Null zuzuordnen ist.

fehlerfrei		AND-Short		OR-Short		versch. Treiber		Weak Short	
K_0	K_1	K_0	K_1	K_0	K_1	K_0^*	K_1	K_0	K_1
0	0	0	0	0	0	0	0	0	0
0	1	0	0	1	1	0	0	X	X
1	0	0	0	1	1	1	1	X	X
1	1	1	1	1	1	1	1	1	1

Tabelle 3.1: Funktionstabelle der verschiedenen Modelle für Leitungskurzschlüsse (Die unter versch. Treiber mit * gekennzeichnete Leitung ist dominierend)

Das Verhalten der verschiedenen Brückenfehler ist in Tabelle 3.1 aufgeführt. Grundsätzlich gilt, dass die Fehler an der Leitungsausgangsseite nur beobachtbar sind, wenn an der Leitungseingangsseite unterschiedliche Logikpegel getrieben werden. Bei einem digitalen Test mit dem Muster 10 für die Leitungen K_0 und K_1 wird ein OR-Short am Leitungsende von K_1 detektiert, weil statt des zu erwartenden Werts 0 der Wert 1 beobachtet wird². Auch bei einem AND-Short ist der Fehler bei einem Test mit unterschiedlichen Logikpegeln immer auf einer der betroffenen Leitungen zu detektieren.

Beim Brückenfehler mit verschiedenen starken Treibern ist der Fehler nur auf der Leitung K_1 mit dem schwächeren Treiber zu erkennen. Die Leitung K_0 mit dem

² Für die Fehlerdiagnose ist dieser Test jedoch nicht ausreichend, weil dafür die Lokalisierung des Fehlers auf beiden vom Fehler betroffenen Leitungen notwendig ist. Um auch auf Leitung K_0 den Fehler zu lokalisieren, wäre daher auch der Test mit dem inversen Muster notwendig.

stärkeren Treiber liefert dagegen immer die korrekten logischen Werte am Leitungsausgang. Leitung K_1 stellt aber über die unerwünschte Verbindung eine zusätzliche Last für den Treiber der Leitung K_0 dar. Dadurch überschreitet die Signalverzögerung auf Leitung K_0 den spezifizierten Wert, so dass der Fehler auch auf Leitung K_0 beobachtbar wird, wenn während des Tests Signalwechsel im spezifizierten Systemtakt stattfinden. Dieses Verhalten wird aber nicht von dem statischen Fehlermodell erfasst.

Brückenfehler sind mit einem Testmuster pro zu testendem Leitungspaar vollständig zu testen. Die Obergrenze für die Testlänge bei einem N -Bit Leitersystem beträgt daher $k_{max} = \frac{1}{2}N!$. Die untere Grenze für die Testlänge ist $k_{min} = \lceil \log_2(N + 2) \rceil$ [Kau74]. Damit ist ein effektiver Test auf Brückenfehler möglich. In Abschnitt 3.2 wird ein Test der Ordnung $O(N)$ beschrieben, der Brückenfehler vollständig testet und besonders für die Implementierung eines Selbsttests geeignet ist.

Weak Shorts sind dagegen mit den in Tabelle 3.1 dargestellten Tests nicht sicher zu erkennen, da ein Spannungspegel zwischen Betriebsspannung und Erde (hier mit X bezeichnet) von der nachfolgenden Logikstufe abhängig von deren Schaltschwelle als logischer Wert 0 oder 1 interpretiert werden kann. Eine sichere Fehlererkennung wäre nur mit Hilfe einer Messung der tatsächlichen Spannung am Leitungsende möglich, die im Rahmen eines Selbsttests einer digitalen Schaltung nur mit sehr großem Aufwand implementiert werden kann.

Eine Fehlererkennung ist aber möglich, wenn die zulässige Signallaufzeit auf den betroffenen Leitungen K_0 und K_1 überschritten wird. Ursache ist die zusätzliche Last für die Treiber der betroffenen Leitungen, die die jeweils andere Leitung darstellt, wenn sie in entgegengesetzter Richtung getrieben wird. Fehler vom Typ Weak Short können für den Test daher so behandelt werden wie die in Abschnitt 3.1.2 beschriebenen Verzögerungsfehler, die durch kapazitive oder induktive Kopplung zweier Leitungen verursacht werden. Sie werden aber nicht von dem hier verwendeten statischen Fehlermodell erfasst.

Leitungshaftfehler

Das Haftfehlermodell ist das einfachste Fehlermodell, das zur Testentwicklung für integrierte Schaltungen verwendet wird. Haftfehler sind dadurch charakterisiert, dass betroffene Leitungen unabhängig vom am Leitungseingang getriebenen Logikpegel am Leitungsausgang einen konstanten Logikpegel halten. Es sind zwei Arten von Haftfehlern möglich:

- Stuck-at-0: Die Leitung nimmt fest den logischen Wert Null an.
- Stuck-at-1: Die Leitung nimmt fest den logischen Wert Eins an.

Haftfehler modellieren insbesondere physikalische Defekte wie Kurzschlüsse von Signalleitungen mit Versorgungsspannung oder Masse. Ein Test auf diese Haftfehler kann erfolgen, indem jede Leitung jeweils einmal mit dem logischen Wert Null und einmal mit dem logischen Wert Eins getrieben wird. Diese Bedingung ist beim Test auf Brückenfehler erfüllt. Der Test auf Brückenfehler beinhaltet damit den Test auf Haftfehler. Das Haftfehlermodell ist das wichtigste Fehlermodell für den Test integrierter Schaltungen. Der Anteil an physikalischen Defekten in den Logik-Teilen aktueller integrierter Schaltungen, die durch das Stuck-at-Fehlermodelle abgedeckt werden, beträgt ca. 80 Prozent [KMV01].

Unterbrechungsfehler

Leitungsunterbrechungen entstehen, wenn die Leitungsstrukturen zu geringe Querschnitte aufweisen oder vollständig getrennt sind, was durch Leitbahnabrisse an Kanten, fehlerhaft ausgebildete Kontaktlöcher oder auch durch Elektromigration während des Betriebs verursacht werden kann. Das so vom Leitungstreiber getrennte Leitungsende nimmt einen Pegel zwischen logisch Null und logisch Eins an. Das Leitungspotential wird durch Kopplungen mit benachbarten Leitungen bestimmt. Diese Unterbrechungsfehler werden von digitalen Tests erkannt, wenn auf der defekten Leitung der umgekehrte Pegel wie auf den benachbarten Leitungen getrieben wird. Bei ausreichender Kopplung nimmt das Potential auf dem abgetrennten Leitungsende den Wert der benachbarten Leitungen an, und der Fehler ist damit beobachtbar.

Die Kopplung der Leitungen ist aber für den fehlerfreien Betrieb einer Schaltung generell unerwünscht und wird daher so weit wie möglich reduziert. Im Allgemeinen reicht die Zeit bis zum nächsten Signalwechsel daher nicht aus, um einen sicheren Logikpegel zu erreichen. Durch das getrennte Leitungsende kann jedoch die Kapazität der benachbarten Leitungen gegenüber dem fehlerfreien Fall verändert werden. In diesem Fall kann ein Unterbrechungsfehler als Kopplungsfehler beschrieben werden. Diese werden in Abschnitt 3.1.2 behandelt. Das hier verwendete statische Fehlermodell erfasst daher keine Unterbrechungsfehler.

3.1.2 Maximum Aggressor Fehlermodell

Das in Abschnitt 3.1.1 beschriebene statische Fehlermodell ist nicht geeignet, Defekte zu beschreiben, die die Signalintegrität auf einer Leitung beeinträchtigen. Um die Leitungseigenschaften genauer zu beschreiben, wird ein Modell zweier benachbarter Leitungen A und B entsprechend Abbildung 3.1 verwendet. Das Modell besteht aus mehreren RLCM-Elementen mit den Widerständen R_{Ai} und R_{Bi} , den Kapazitäten C_{Ai} und C_{Bi} und den Induktivitäten L_{Ai} und L_{Bi} , sowie den Koppelkapazitäten

3 Test von Inter-Core Verbindungsnetzen

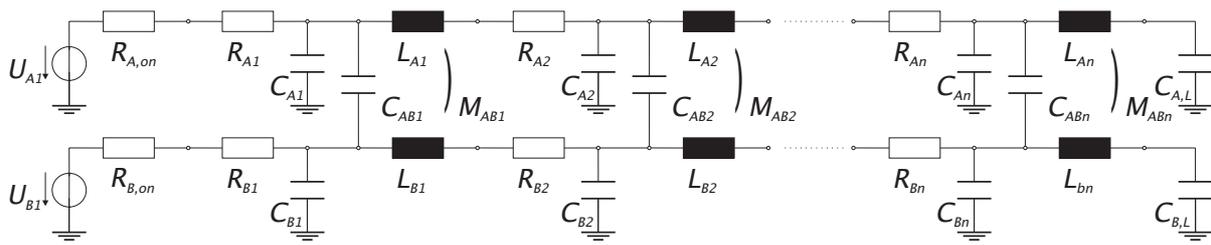


Abbildung 3.1: Modell für zwei gekoppelte Leitungen mit mehreren RLCM-Elementen

C_{ABi} und den Gegeninduktivitäten M_{ABi} (mit $i = 1, 2, \dots, n$) für die Leitungen. Die Leitungstreiber werden durch lineare Ersatzspannungsquellen mit den Innenwiderständen $R_{A,on}$ und $R_{B,on}$, sowie den Quellspannungen U_{A1} und U_{B1} modelliert, die Gattereingänge an den Enden der Leitungen durch lineare Lastkapazitäten $C_{A,L}$ und $C_{B,L}$.

Die Koppelkapazitäten C_{ABi} und Gegeninduktivitäten M_{ABi} können insbesondere in Technologien des tiefen Sub-Mikrometer-Bereichs nicht vernachlässigt werden, sondern führen zu einer Signalverzerrung auf einer Leitung, die von Signalwechseln auf einer oder mehreren benachbarten Leitungen verursacht werden. Die Größe dieser Verzerrungen hängt stark von Schaltungsparametern wie zum Beispiel Treiberstärke, repräsentiert durch R_{on} , Leitungslänge, Taktrate oder Anstiegsrate der Signale ab. Durch geeigneten Entwurf bzw. günstige Dimensionierung der Schaltung können die Signalverzerrungen durch Kopplung soweit verringert werden, dass die Schaltung unter gegebenen Bedingungen wie spezifiziert funktioniert.

Durch Fertigungsfehler kann die Kopplung jedoch so verstärkt werden, dass die Verzerrungen das erlaubte Maß überschreiten und eine Fehlfunktion auftritt. Dabei gibt es im Wesentlichen drei unerwünschte Effekte. Wenn die Koppelkapazität zwischen den Leitungen A und B signifikant wird, treten zwei mögliche Effekte auf. Angenommen, auf Leitung A finde ein Signalwechsel statt, während Leitung B einen konstanten Zustand haben soll. Dann wird wegen der Koppelkapazität Energie von Leitung A auf Leitung B übertragen, was zu einem kurzzeitigen Spannungsanstieg

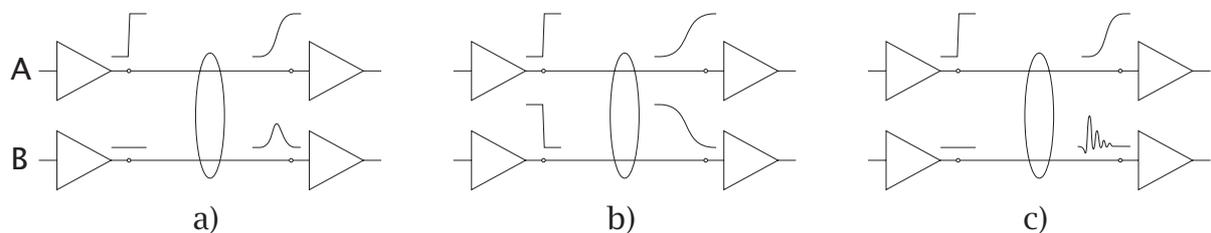


Abbildung 3.2: Mögliche Signalverzerrungen bei Leitungskopplung: a) Glitch, b) Verzögerung, c) Schwingungen

auf Leitung B führt. Diese Art der Signalverzerrung wird *Glitch* genannt und ist in Abbildung 3.2 a) dargestellt.

Jetzt sei angenommen, dass auf Leitung A ein Signalwechsel stattfindet, während auf Leitung B ebenfalls ein Signalwechsel stattfinden soll, aber in entgegengesetzter Richtung. Die Energieübertragung durch die Koppelkapazität führt in diesem Fall zu einer Verzögerung des Signalwechsel. Diese Art der Signalverzerrung wird *Verzögerung* genannt und ist in Abbildung 3.2 b) dargestellt. Bei Signalwechseln in gleicher Richtung ist außerdem eine kurzzeitige Spannungsüberhöhung (engl. overshoot) möglich. Diese können jedoch nicht zur Übernahme falscher logischer Pegel in nachfolgenden Gattern führen, verursachen also keine logischen Fehler. Sie sind vielmehr ein Problem für die Zuverlässigkeit und Lebensdauer der integrierten Schaltung und müssen daher ebenfalls minimiert werden. In dieser Arbeit soll jedoch nicht weiter darauf eingegangen werden.

Wenn neben der kapazitiven Kopplung auch die Kopplung durch die Gegeninduktivität von Bedeutung ist, wird der Signalverlauf komplexer. Möglichen Glitches oder Verzögerungen wird eine Schwingung überlagert. Diese Art der Verzerrung wird in Abbildung 3.2 c) für den Fall angedeutet, dass Leitung B einen konstanten Zustand behalten soll. Diese Schwingung wird wegen des Widerstandsbelags der Leitungen gedämpft. Für die Ableitung des Maximum Aggressor Fehlermodell soll davon ausgegangen werden, dass insbesondere bei langen Leitungen wie bei Inter-Core-Bussen die Dämpfung so groß ist, dass der dritte Fall nicht gesondert behandelt werden muss.

Die exakte Berechnung der qualitativ beschriebenen Signalverzerrungen, die sich aus den möglichen Fehlern und Abweichungen im Fertigungsprozess und ihren Kombinationen ergeben, ist unter wirtschaftlichen Aspekten nicht durchführbar.

Auf der Ebene des Fertigungsprozesses sind diese Fehler zunächst die Gesamtwirkung von kontinuierlich über alle Leitungen und Leitungstreiber verteilten Abweichungen von Prozessparametern. Die Anzahl der möglichen Kombinationen solcher Abweichungen und anderer Fertigungsdefekte wie z.B. Kurzschlüsse ist für ein Mehrleitersystem wie einen Bus so groß, dass nicht alle explizit betrachtet werden können.

Dadurch ergibt sich der Bedarf für ein einfaches Fehlermodell, das die Ableitung eines Tests bzw. die Bewertung eines Tests mit einer Schaltungssimulation ermöglicht. Die physikalischen Fehler können in mehreren Schritten immer weiter abstrahiert werden und zu einem solchen Fehlermodell, dem Maximum Aggressor Fehlermodell, entwickelt werden [CDBZ99], wie in den folgenden Absätzen beschrieben wird.

Physikalische Fehler können in ihrer Wirkung auf die Leitungseigenschaften auf Schaltungsebene zusammengefasst werden. Entsprechend Abbildung 3.1 kann man eine physikalische Leitung als ein lineares Netzwerk bestehend aus vielen Impedanzen beschreiben. Die Kopplung wird dabei durch Koppelkapazitäten und -induktivitäten beschrieben. Die physikalischen Fehler werden dann als Abweichung der Größe dieser Impedanzen modelliert.

Mit diesem Modell ist die Menge der möglichen Fehler noch zu groß. Erstens müssen für jede Impedanz dieses Modells eine Vielzahl von Größen betrachtet werden, zweitens muss jede Kombination von Impedanzen berücksichtigt werden. Daher muss das Fehlermodell auf Schaltungsebene weiter zu einem Verhaltensmodell vereinfacht werden, das alle möglichen Fehlereffekte erfasst. Dabei nehmen wir an, dass ein Fehler nur eine einzelne Leitung betrifft, die wir als Opfer bezeichnen. Eine Untermenge der übrigen Leitungen, die wir als Aggressoren bezeichnen, wirkt gemeinsam auf die Opferleitung und verursacht eine der vier folgenden Fehlerbedingungen: ein positiver Glitch, ein negativer Glitch, eine Verzögerung an einer steigenden Signalflanke oder eine Verzögerung an einer fallenden Signalflanke. Eine Fehlerbedingung ist erfüllt, wenn eine bestimmte minimale Amplitude oder eine bestimmte minimale Dauer der Verzögerung überschritten wird. Die Anzahl der möglichen Fehler eines N -Leiter-Systems unter diesem Fehlermodell beträgt $N \cdot 2^{N-1}$ (2^{N-1} mögliche Aggressorkombinationen für jede der N Leitungen). Im Einzelnen können die Fehlerbedingungen wie folgt definiert werden.

- Fehler „positiver Glitch“ (pG): Der Signalpegel auf der Opferleitung überschreitet am Lastgatter dessen Schaltschwelle $V_{th,p}$ für eine Zeitdauer größer als die notwendige Haltezeit t_{hold} des Lastgatters.
- Fehler „negativer Glitch“ (nG): Der Signalpegel auf der Opferleitung unterschreitet am Lastgatter dessen Schaltschwelle $V_{th,n}$ für eine Zeitdauer größer als die notwendige Haltezeit t_{hold} des Lastgatters.
- Fehler „Verzögerung bei steigender Flanke“ ($VstF$): Der Signalpegel auf der Opferleitung überschreitet am Lastgatter nach einer höchsten zulässigen Verzögerungszeit t_{slack} nicht die Schaltschwelle $V_{th,p}$ des Lastgatters.
- Fehler „Verzögerung bei fallender Flanke“ (VfF): Der Signalpegel auf der Opferleitung unterschreitet am Lastgatter nach einer höchsten zulässigen Verzögerungszeit t_{slack} nicht die Schaltschwelle $V_{th,n}$ des Lastgatters.

Dabei ist die höchste zulässige Verzögerungszeit t_{slack} als die zusätzliche Verzögerung definiert, die sich aus der Differenz der nominellen Signalankunftszeit und der spätest möglichen Signalankunftszeit, so dass sich gerade kein Fehler ergibt. Bei Überschreiten von t_{slack} nimmt das Signal am Eingang des auf das Lastgatter folgenden Flip-Flops noch keinen stabilen Zustand an, wenn es durch das auf den Signalwechsel folgende Taktsignal übernommen werden soll.

3.1 Fehlermodelle für Verbindungsnetze

	K_0	K_1	K_{i-1}	K_i	K_{i+1}	K_{N-2}	K_{N-1}	Zielfehler
t_0	0	0	0	0	0	0	0	init
t_1	1	1	1	0	1	1	1	pG

Tabelle 3.2: Testfolge für den Fehler „positiver Glitch“ (pG) auf Leitung K_i

	K_0	K_1	K_{i-1}	K_i	K_{i+1}	K_{N-2}	K_{N-1}	Zielfehler
t_0	1	1	1	1	1	1	1	init
t_1	0	0	0	1	0	0	0	nG

Tabelle 3.3: Testfolge für den Fehler „negativer Glitch“ (nG) auf Leitung K_i

	K_0	K_1	K_{i-1}	K_i	K_{i+1}	K_{N-2}	K_{N-1}	Zielfehler
t_0	1	1	1	0	1	1	1	init
t_1	0	0	0	1	0	0	0	$VstF$

Tabelle 3.4: Testfolge für den Fehler „Verzögerung bei steigender Flanke“ ($VstF$) auf Leitung K_i

	K_0	K_1	K_{i-1}	K_i	K_{i+1}	K_{N-2}	K_{N-1}	Zielfehler
t_0	0	0	0	1	0	0	0	init
t_1	1	1	1	0	1	1	1	VfF

Tabelle 3.5: Testfolge für den Fehler „Verzögerung bei fallender Flanke“ (VfF) auf Leitung K_i

Die Fehlermenge kann weiter reduziert werden, indem nur die Kombinationen von Koppelimpedanzen betrachtet werden, die zu einer maximalen Beeinflussung der Opferleitung führen. Bei Beschränkung auf kapazitive Kopplung ist das der Fall, wenn alle Aggressoren zeitgleich in die gleiche Richtung schalten. Dann überlagern sich erstens die Einflüsse aller Kapazitäten konstruktiv. Zweitens entfallen die kapazitiven Einflüsse der Aggressoren untereinander im Idealfall, weil die Spannung zwischen ihnen Null ist. Drittens ist die Gesamtimpedanz des Strompfades von der Spannungsquelle durch die Aggressoren zur Opferleitung minimal, da alle Aggressorimpedanzen parallel zueinander liegen.

Insgesamt kann für ein RC-Leitungsmodell gezeigt werden, dass ein maximaler Fehlereffekt erreicht wird, falls alle $N - 1$ Aggressorleitungen eines Busses zeitgleich in die gleiche Richtung schalten. In [CDBZ99] wird durch Simulationen bestätigt, dass ein entsprechender Test eine Fehlerabdeckung von 100 % für die möglichen physikalischen Fehler liefert.

Weil dieser Test auf der Maximierung der Gesamtwirkung der einzelnen Aggressorleitungen basiert, wird das zugehörige Modell als *Maximum Aggressor Fehler-*

modell (MAFM) bezeichnet. Die wesentlichen Eigenschaften des Modells sind, dass erstens nur ein einzelner Fehler auf der Opferleitung betrachtet wird, und dass zweitens genau ein Signalübergang der Aggressoren diesen Fehler stimulieren kann.

Für die vier obengenannten Fehlerbedingungen benötigt man daher je Opferleitung vier verschiedene Signalübergänge, die in den Tabellen 3.2 bis 3.5 dargestellt sind. Im ersten Schritt wird der Test initialisiert (*init*), indem ein bestimmter Zustand der Leitungen eingestellt wird. Im zweiten Schritt wird der Test ausgeführt, indem der notwendige Signalübergang stimuliert wird. Die Testlänge beträgt daher maximal $8N$ Schritte für einen N -Bit Bus, d.h. eine lineare Anzahl von Tests ist ausreichend, um eine exponentielle Anzahl von möglichen physikalischen Defekten zu testen.

Die für das Maximum Aggressor Fehlermodell entwickelten Tests sind außerdem geeignet, die in Abschnitt 3.1.1 beschriebenen Fehler zu erregen, weil die notwendigen Tests in jeder der Tabellen 3.2 bis 3.5 enthalten sind. Auch die vom statischen Fehlermodell nicht erfassten Fehler Unterbrechungsfehler und Weak-Short werden detektiert, weil diese sich in einer erhöhten Leitungsverzögerung äußern. Tests nach dem Maximum Aggressor Fehlermodell sind also eine Obermenge der Tests für das statische Fehlermodell.

3.2 On-Chip Testmustererzeugung

Inter-Core-Verbindungsnetze sind in der Regel in ein komplexes System eingebettet, so dass eine direkte Steuerung und Beobachtung durch einen Digital-Tester nicht möglich ist. Der Test muss daher mit Hilfe von Scan-Pfaden oder durch Integration eines Selbsttests implementiert werden. Im Abschnitt 3.1.1 wurden folgende Anforderungen an einen Test für Verbindungsnetze unter dem statischen Fehlermodell definiert:

- Jede Leitung eines Verbindungsnetzes muss mindestens einmal mit dem Pegel für den logischen Wert 1 und mindestens einmal mit dem Pegel für den logischen Wert 0 getrieben werden.
- Jedes Leitungspaar eines Verbindungsnetzes muss mindestens einmal mit dem Pegel für den logischen Wert 1 auf der einen und mit dem Pegel für den logischen Wert 0 auf der anderen Leitung oder umgekehrt getrieben werden.

Ein Test mit diesen Eigenschaften erregt alle Fehler der Typen AND-Short, OR-Short, Brückenfehler mit unterschiedlichem Treiber und Haftfehler. Für die Implementierung eines Tests als integrierter Selbsttest gilt die zusätzliche Anforderung, die

Anzahl bzw. die Fläche der für die Implementierung zusätzlich erforderlichen Schaltungskomponenten zu minimieren. Der in [Kau74] gezeigte optimale Leitungstest der Länge $k_{min} = \lceil \log_2(N + 2) \rceil$ für ein N -bit Leitungssystem ist nicht für eine Selbsttestimplementierung geeignet, weil die On-Chip-Generierung der vorab berechneten Testmuster zu großen Schaltungsaufwand erfordert.

Die genannten Bedingungen werden dagegen von einem *Walking-1-Test* erfüllt [CS93], der als Basis für alle in dieser Arbeit vorgestellten Leitungstests dient. Dieser Leitungstest besteht darin, genau eine Leitung mit dem Spannungspegel für die logische Eins zu treiben, während alle anderen Leitungen mit dem Pegel für die logische Null getrieben werden. Die logische Eins „wandert“ dabei mit jedem Takt zur nächsten benachbarten Leitung weiter.

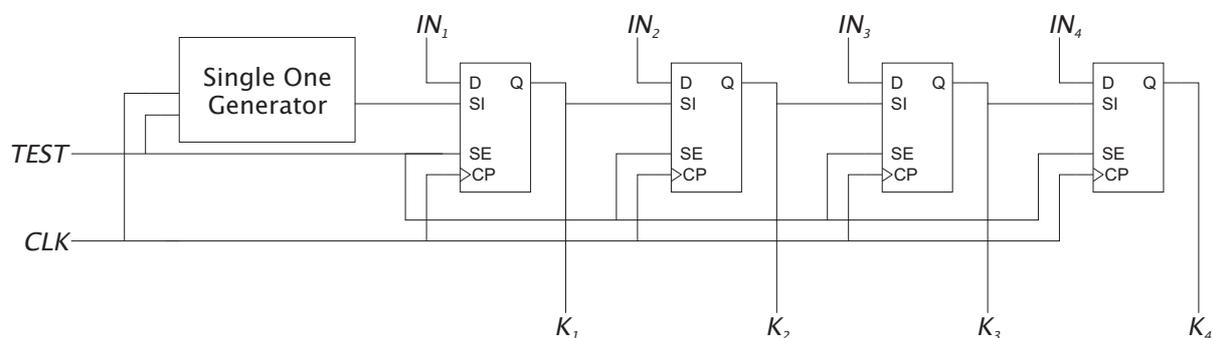


Abbildung 3.3: Testmustererzeuger für Walking-1-Test mit $N = 4$

Wegen der einfachen Struktur dieses Tests kann er sehr einfach durch ein Schieberegister generiert werden, wie in Abbildung 3.3 gezeigt wird. Dabei wird jede Leitung von einer Zelle des Schieberegisters getrieben. Alle Zellen des Schieberegisters werden zur Testdurchführung mit dem Wert Null initialisiert. Eine kleine zusätzliche Schaltung (*Single One Generator*) erzeugt eine Zahlenfolge von genau einer Eins und folgenden Nullen. Diese Folge wird von der ersten Zelle des Schieberegisters übernommen. Die Eins wird mit jedem Takt eine Zelle weitergeschoben. Zugleich werden in der ersten Zelle wieder Nullen übernommen. Insgesamt wird das in Tabelle 3.6 gezeigte Muster erzeugt.

Ein Walking-1-Test hat für ein N -bit Leitungssystem die Länge N und erfasst alle Leitungshaftfehler und Leitungskurzschlüsse des statischen Fehlermodells in Abschnitt 3.1.1. Die gegenüber dem optimalen Test größere Testlänge ist beim integrierten Selbsttest akzeptabel, weil eine Testarchitektur verwendet wird, bei der mit jedem Taktzyklus ein neues Testmuster angewendet wird (Ein-Test-pro-Takt-Architektur).

Eine mögliche Implementierung eines Testverfahrens für Inter-Core-Verbindungsnetze wird in der Arbeit von Grützner gezeigt [Grü88]. Als Testmuster werden Paare von logischen Einsen verwendet, die wie beim Walking-1-Test mit jedem Takt um eine Leitung verschoben werden. Die Besonderheit des Verfahrens besteht darin,

3 Test von Inter-Core Verbindungsnetzen

	K_0	K_1	\dots	K_{i-1}	K_i	K_{i+1}	\dots	K_{N-2}	K_{N-1}
t_0	1	0	\dots	0	0	0	\dots	0	0
t_1	0	1	\dots	0	0	0	\dots	0	0
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
t_{i-1}	0	0	\dots	1	0	0	\dots	0	0
t_i	0	0	\dots	0	1	0	\dots	0	0
t_{i+1}	0	0	\dots	0	0	1	\dots	0	0
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots
t_{N-2}	0	0	\dots	0	0	0	\dots	1	0
t_{N-1}	0	0	\dots	0	0	0	\dots	0	1

Tabelle 3.6: Walking-1-Test für ein N -bit Leitersystem

dass am Leitungsende eine so genannte Echo-Zelle implementiert wird, die die Testantwort auf das Leitungssystem zurückschreibt. Die Testauswertung erfolgt daher ebenso am Leitungsanfang wie die Testmuster-generierung. Das Verfahren erfasst alle Fehler des statischen Fehlermodells, sowie Mehrfachfehler und Fehler der Echo-Zellen.

Für einen Test auf Fehler nach dem Maximum Aggressor Fehlermodell sind jedoch zusätzliche Bedingungen zu erfüllen.

- Der Testmuster-generator muss die Signalübergänge nach den Tabellen 3.2 bis 3.5 erzeugen.
- Der Test muss mit der für den Betrieb spezifizierten Taktfrequenz ausgeführt werden.

Die Signalübergänge sind nicht unabhängig von einander, so dass die Gesamtzahl der Tests reduziert werden kann. Die Zeilen t_1 aus Tabelle 3.2 und t_0 aus Tabelle 3.3 sowie die Zeilen t_1 aus Tabelle 3.4 und t_0 aus Tabelle 3.5 sind jeweils identisch. Die Signalübergänge für eine Leitung können daher mit minimal 6 Tests erzeugt werden [BDR01]. Beide Bedingung können mit Selbsttestarchitekturen auf einfache Art erfüllt werden, weil bei diesen pro Takt ein Test ausgeführt werden kann.

Abbildung 3.4 stellt einen Testmuster-generator für einen vollständigen Test bezüglich des Maximum Aggressor Fehlermodells nach [BDR01] dar. Er besteht aus einem $\lceil \log_2 N \rceil$ -Bit-Zähler, einem endlichen Automaten mit 7 Zuständen (6 Testzustände und ein Zustand für den Normalbetrieb), einem $\lceil \log_2 N \rceil$ -zu- N -Decoder, einem 2-zu-1 Multiplexer für jede Leitung, sowie einem weiteren 2-zu-1-Multiplexer für jede Leitung zur Trennung von Testbetrieb und Systembetrieb³.

³ Der Zustand für den Systembetrieb und die zweite Gruppe von Multiplexern zur Umschaltung zwischen Test- und Systembetrieb sind in Abbildung 3.4 zur Vereinfachung nicht dargestellt.

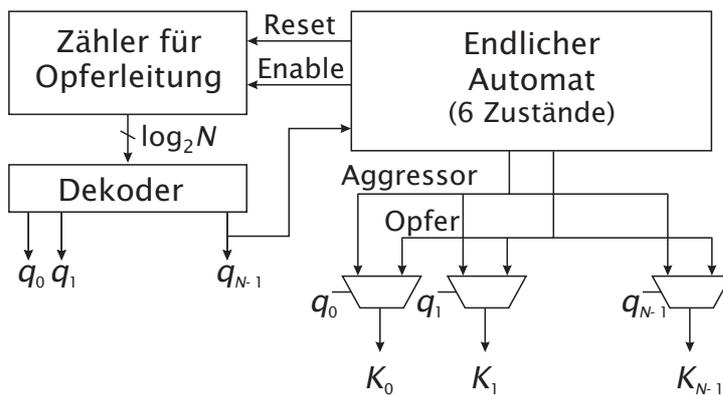


Abbildung 3.4: Testmustergenerator für einen vollständigen Test bezüglich des Maximum Aggressor Fehlermodells nach [BDR01]

Der endliche Automat erzeugt die sechs Muster zur Stimulierung der erforderliche Signalübergänge für den Test einer Leitung, die als Opferleitung bestimmt ist. Nach genau sechs Takten wird eine andere Leitung Opferleitung. Dazu wird der Zähler erhöht. Der Decoder liefert daraufhin andere Steuersignale q an die Multiplexer, die dadurch die Muster für Opferleitung und Aggressorleitungen für den Test der nächsten Leitung zuordnen. Insgesamt sind für den Test eines N -bit Leitungssystems $6N$ Takte erforderlich.

Der Testmustergenerator nach [BDR01] verursacht minimalen Schaltungsoverhead, weil insgesamt nur $\lceil \log_2 N \rceil + 7$ Flipflops notwendig sind, die Komplexität also weniger als linear mit der Busbreite N steigt. Man muss jedoch berücksichtigen, dass durch einen Core Test Wrapper nach dem Standard IEEE 1500 eine Speicherzelle, gemeinhin also ein Flipflop, an jedem Schaltungsausgang gegeben ist, zu denen die Flipflops des Testmustergenerators zusätzlich nötig sind. Eine günstigere Lösung ist die Wiederverwendung der Flipflops des Test-Wrappers. Ein derartiger Testmustergenerator wurde im Rahmen dieser Arbeit entwickelt und wird in Abschnitt 5.2 gezeigt.

3.3 On-Chip Testauswertung

Für die Detektion von Kopplungsfehlern sind in der Literatur unterschiedliche Methoden beschrieben, die sich drei Funktionsprinzipien zuordnen lassen. Das erste Prinzip beruht auf der Verwendung von speziellen Schaltkreisen, die die Signale auf Inter-Core-Verbindungsnetzen permanent überwachen und ein Verlassen von zulässigen Betriebsbereichen des Signals als Fehler melden. In [Laj01] werden fehlerkorrigierende Codes verwendet, bei denen ein Glitch- oder Verzögerungsfehler

zu einem ungültigen Codewort führt. Überwachungsschaltungen für Inter-Core-Verbindungsnetze können ungültige Codeworte während des Systembetriebs detektieren und korrigieren. In [MFR00] wird ein Signalübergang, der auftritt, während ein Data-Valid-Signal die Stabilität und damit die Gültigkeit des Signals anzeigt, als ein Glitch- oder Verzögerungsfehler betrachtet. Zugleich wird eine Zelle beschrieben, die Signalübergänge während der Data-Valid-Phase detektiert. Beide Verfahren sind geeignet, Glitch- und Verzögerungsfehler während des Systembetriebs anzuzeigen. Die Kosten aufgrund des großen Schaltungsoverheads machen sie jedoch ungeeignet für den Produktionstest von kostensensiblen Produkten, bei denen ein Online-Test nicht erforderlich ist.

In [NA01] wird das Über- bzw. Unterschreiten von kritischen Signalschwellen als ein Kopplungsfehler betrachtet. Dadurch wird kein zeitlicher Bezug zu einem Referenzsignal wie zum Beispiel das Data-Valid-Signal benötigt. Eine Fehlerdetektion ist mit gegengekoppelten Differenzverstärkern möglich, die mit geringem Schaltungsoverhead integriert werden können. Die Fehlerdetektion beruht jedoch auf der Hystereseeigenschaft des gegengekoppelten Differenzverstärkers, die durch die Dimensionierung der verwendeten Transistoren auf die kritischen Signalschwellen eingestellt werden. Diese Transistorparameter sind ihrerseits jedoch starken Fertigungsschwankungen unterworfen. In [NA01] wird keine Möglichkeit gezeigt, eine zuverlässige Detektion von Kopplungsfehlern trotz dieser Schwankungen zu erreichen.

Die genannten Verfahren sollen die Fehlerdetektion während des Betriebs ermöglichen. Ziel dieser Arbeit ist aber die Bereitstellung eines Verfahrens, mit dem während eines Produktionstest eine Schaltung auf Fertigungsfehler möglichst vollständig getestet wird. Dazu ist die Stimulierung der möglichen Fehler durch spezielle Testmuster wie in Abschnitt 3.1 beschrieben nötig. Dies kann von den in [MFR00, NA01] dargestellten Testverfahren nicht geleistet werden.

Dagegen wird in [BDR01] ein vollständiges Selbsttestverfahren für den Produktionstest von Leitungssystemen beschrieben. Die Testarchitektur ist in Abbildung 3.5 dargestellt. Ein Testmustergenerator erzeugt Testmuster nach dem Maximum Aggressor Fehlermodell (Abschnitt 3.1.2), die zum Leitungsende übertragen werden. Dort werden mit einem identischen Testmustergenerator dieselben Testmuster erzeugt und mit den übertragenen Signalen verglichen. Dazu genügt an jeder Leitung des Bussystems ein Exklusiv-ODER(XOR)-Gatter als Komparator. Bei einem bidirektionalen Bus ist an jedem Leitungsende ein Testmustergenerator notwendig, um in beide Betriebsrichtungen testen zu können. Der Testmustergenerator kann also für die Testauswertung wiederverwendet werden. Im Falle eines unidirektionalen Busses ist der Schaltungsoverhead für den zusätzlichen Testmustergenerator am Leitungsende jedoch beträchtlich.

In [CBD02] wird eine Methode beschrieben, wie der Schaltungsoverhead vollständig vermieden werden kann. Der Test beruht auf dem Transfer von Daten zwischen

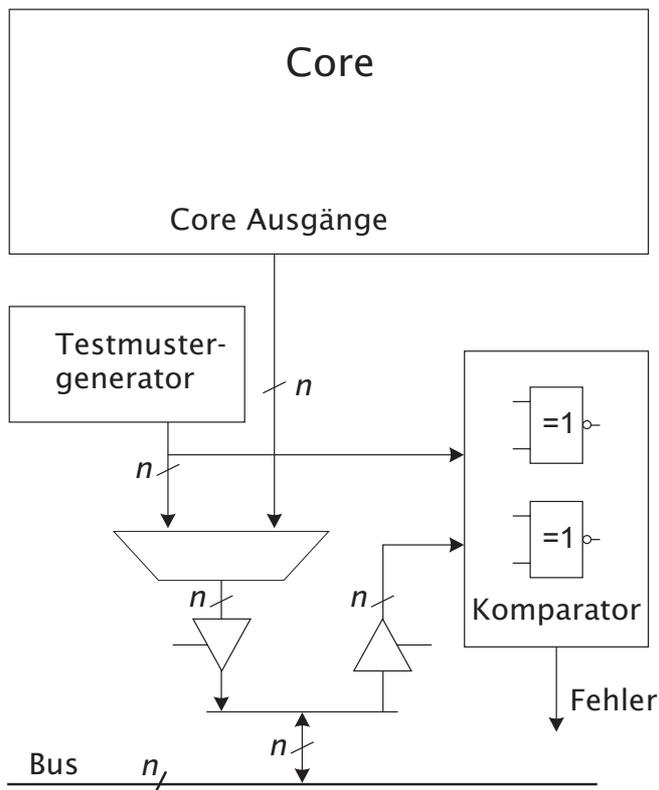


Abbildung 3.5: Testauswertung mit Komparator beim Selbsttest von Bussen

einem programmierbaren Mikroprozessor-Core und einem eingebetteten Speicher. Die Daten sind funktionale Testmuster für den Bustest, die von dem Mikroprozessor generiert werden, in den Speicher geschrieben und wieder ausgelesen werden. Dieses Verfahren ist jedoch nur für den Test eines Busses zwischen Mikroprozessor und Speicher, aber nicht zwischen allgemeinen Cores verwendbar.

Eine Erweiterung ist durch Hinzufügen von speziellen Core-Wrappern möglich [LHC01]. Dadurch wird der Gewinn an Schaltungsoverhead gegenüber Selbsttestverfahren entsprechend Abbildung 3.5 verringert. Der wesentliche Nachteil ist jedoch, dass ein fehlerfreier Mikroprozessor-Core vorausgesetzt werden muss. Hinsichtlich der Testdauer ist es aber günstiger, zunächst die Fehlerfreiheit der Busse zu prüfen, um diese dann für den breitbandigen Transfer von Testdaten zu den Cores zu nutzen.

Das Ziel, den Schaltungsoverhead zu verringern, wird auch in [YHS⁺01] verfolgt. Die Testauswertung erfolgt dabei nicht durch Vergleich mit einem lokalen Testmuster-generator, sondern durch einem Baum von XOR-Gattern. Dazu müssen die Testvektoren so gruppiert werden, dass mit jedem zweiten Testvektor ein kritischer Signalübergang stattfindet. Tabelle 3.7 zeigt diesen umgeordneten Test. Die Testauswertung muss mit jedem zweiten Takt erfolgen. Diese sind in Tabelle 3.7 hervorgehoben.

3 Test von Inter-Core Verbindungsnetzen

	K_0	K_1	\dots	K_{i-1}	K_i	K_{i+1}	\dots	K_{n-2}	K_{n-1}	Zielfehler
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
$t_{i,0}$	0	0	\dots	0	0	0	\dots	0	0	init
$t_{i,1}$	1	1	\dots	1	0	1	\dots	1	1	pG_i
$t_{i,2}$	1	1	\dots	1	1	1	\dots	1	1	init
$t_{i,3}$	0	0	\dots	0	1	0	\dots	0	0	nG_i
$t_{i,4}$	0	0	\dots	0	1	0	\dots	0	1	init
$t_{i,5}$	1	1	\dots	1	0	1	\dots	1	1	VfF_i
$t_{i,6}$	0	0	\dots	0	1	0	\dots	0	0	init
$t_{i,7}$	0	0	\dots	0	1	0	\dots	0	0	$VstF_i$
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots

Tabelle 3.7: Leitungstest mit kritischem Signalübergang bei jedem zweiten Testvektor

Für diese vier Testvektoren gilt, dass das Signal der Opferleitung den invertierten Wert der Aggressorleitungen hat. Sind alle Signale am Leitungsende gleich, liegt ein Kopplungsfehler vor. Zur Überprüfung dieser Fehlerbedingung werden für ein N -bit Leitungssystem $N - 1$ XOR-Gatter benötigt, die die Gewichte aller Signale auf den Leitungen aufsummieren.

$$\overline{\text{Fehler}} = \sum_i K_i \bmod 2 = \begin{cases} 0 & \text{für den fehlerfreien Fall} \\ 1 & \text{falls ein Kopplungsfehler vorliegt} \end{cases} \quad (3.1)$$

Obwohl der Schaltungsaufwand für die Testauswertung sehr gering ist, hat dieses Verfahren doch zwei Nachteile. Erstens hat der Baum von $N - 1$ XOR-Gattern eine Tiefe von $\log N$, so dass bei breiten Bussen die Gatterverzögerung durch diesen Baum nicht zu vernachlässigen ist. Zweitens können nicht alle Mehrfachfehler des statischen Fehlermodells nach Abschnitt 3.1.1 detektiert werden. Beispielsweise gilt für jeden Kurzschlussfehler mit einer ungeraden Anzahl von betroffenen Leitungen $\sum K_i \bmod 2 = 0$, obwohl ein Leitungsdefekt vorliegt.

Auch bei der Testauswertung ist es hinsichtlich des Schaltungsoverheads günstig, die bei IEEE 1500 Test-Wrappern vorhandene Flipflops wiederzuverwenden. Dies geschieht bei keinem der bekannten Verfahren zur Auswertung von Leitungstests. Insgesamt ist daher festzustellen, dass alle bisher beschriebenen Verfahren zur Auswertung der Testantworten beim Selbsttest von Bussystemen Nachteile aufweisen, die einer erfolgreichen Anwendung des Selbsttests von Bussystemen im Wege stehen.

4 Testimplementierung für ein monolithisch integriertes Multiprozessor-System

Diese Arbeit wurde im Wesentlichen motiviert durch die Testproblematik bei der Entwicklung eines Multiprozessor-Systems für die Videosignalverarbeitung am Laboratorium für Informationstechnologie, Universität Hannover. Dieser Video-Prozessor mit der Bezeichnung AxPe-GIS¹ ist wie aktuelle Systems-on-Chip busbasiert, und daher ein aufschlussreiches Beispiel für die Implementierung von Testverfahren für Cores und Inter-Core-Busse.

Das Multiprozessor-System verwirklicht ein Architekturkonzept, das richtungweisend für eine neue Klasse von Prozessorarchitekturen ist. Diese symmetrischen Multiprozessorsysteme (SMP) auf einem Chip werden beispielsweise vom führenden Prozessorhersteller Intel als Alternative zu hochgetakteten Ein-Prozessor-Chips für Server-Anwendungen entwickelt² [Bha04]. Bei der Entwicklung des Video-Prozessors standen für viele Probleme keine bewährten Lösungen zur Verfügung. Das betrifft die Realisierung als großflächig integrierte Schaltung ebenso wie Entwurf, Verifikation und Test der ungewöhnlich langen Inter-Core-Bussysteme.

4.1 Systemarchitektur

Die AxPe-GIS ist die Implementierung einer programmierbaren Multiprozessor-Architektur mit verteiltem Speicher für die Videosignalverarbeitung [HMHP00]. Zielanwendung ist die Kodierung von Videosignalen entsprechend ITU-T H.263 oder ISO MPEG-2 für maximal HDTV-Auflösung in Echtzeit. Die Architektur erlaubt es, eine veränderliche Anzahl von identischen Prozessorknoten (*Processing Node*, PN) einzusetzen, um verschiedene Videokodierungsalgorithmen auf Videodaten

¹ Großflächig integrierte Schaltung

² Ein wesentlicher Unterschied zwischen den Architekturen besteht allerdings darin, dass Intel für die SMP-Chips ein von den Prozessoren gemeinsam genutzten Speicher vorsieht, während bei dem Video-Prozessor jeder Prozessorknoten seinen eigenen Speicher besitzt.

4 Testimplementierung für ein Multiprozessorsystem

unterschiedlicher Bandbreite entsprechend der Auflösung des Videobildes auszuführen.

Dieses Architekturmerkmal wird durch ein spezielles Fertigungsverfahren ausgenutzt. Durch Fotokomposition können Systeme als großflächig integrierte Schaltung (GIS) mit einer nahezu beliebigen Anzahl von Prozessorknoten auf einem Chip gefertigt werden. Dabei wird die GIS aus identischen Subsystemen fotolithographisch auf einem Wafer zusammengesetzt, indem in den Lithographieschritten der oberen Metalllagen die Masken dieser Subsysteme mit Überlappungen belichtet werden [RRH⁺02].

Die implementierte AxPe-GIS besteht aus 16 Prozessorknoten, von denen jeder wiederum einen Videoprocessorkern vom Typ AxPe, 4 MBit Bildspeicher und anwendungsspezifische Schnittstellen enthält. Der AxPe-Videoprocessorkern besitzt eine Koprocessorarchitektur, die für hybride Video-Kodierungsverfahren entwickelt wurde [HHO⁺98]. Kontroll- und Videosignale werden den PNs parallel über einen Host-Bus bzw. über einen Videobus zur Verfügung gestellt. Ein Datenaustausch zwischen den Prozessorknoten findet über einen Kommunikationsbus statt (vergleiche Abbildung 4.1).

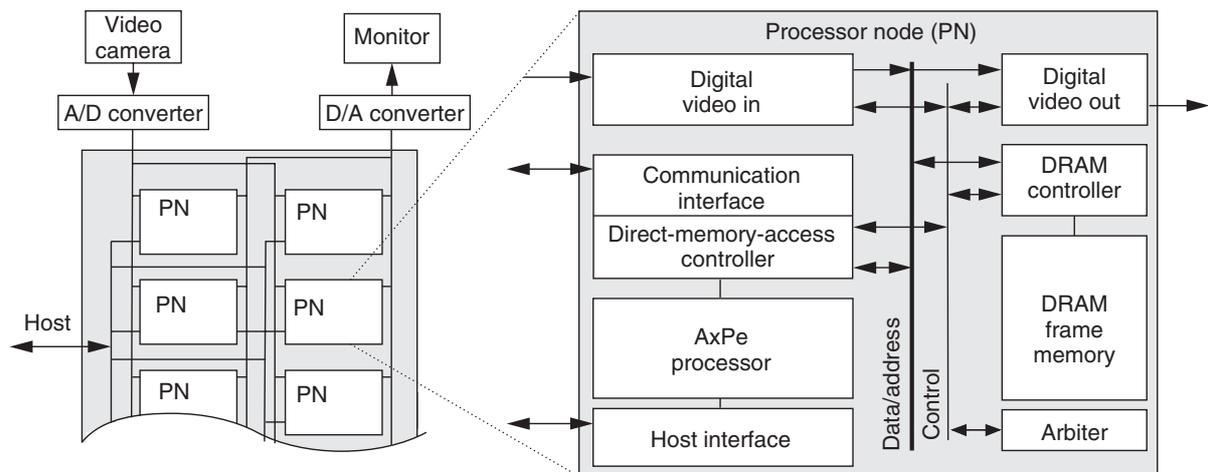


Abbildung 4.1: Vereinfachtes Blockdiagramm für ein AxPe-Multiprozessorsystem nach [RRH⁺02]

Dabei bearbeitet jeder PN (nahezu) unabhängig einen anderen Teilbereich des Bildes. Dazu wird er während des Boot-Vorgangs durch einen Satz von Parametern initialisiert, der den aufzunehmenden Bereich der Bilddaten spezifiziert. Die Größe des Bildspeichers³ ist so ausgelegt, dass eine AxPe-GIS alle benötigten Bilddaten lokal speichern kann, so dass die Speicherbandbreite nicht durch die Verwendung externer Speicher begrenzt ist.

³ insgesamt 64 MBit integrierter Speicher

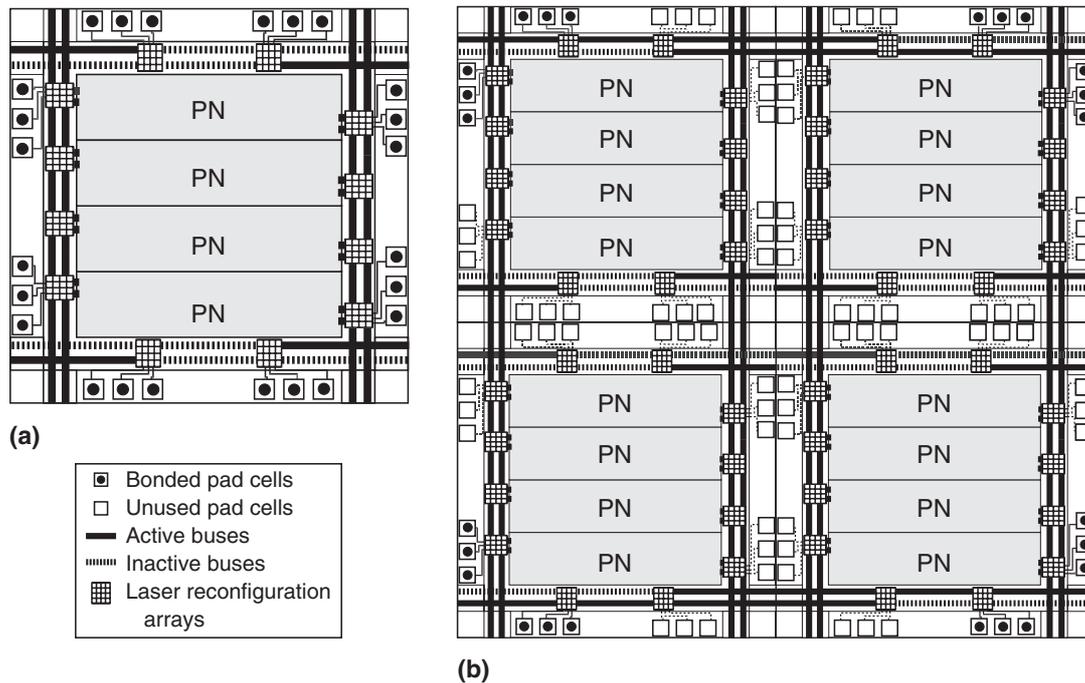


Abbildung 4.2: Vereinfachter Floorplan von AxPe-Subsystem (links) und AxPe-GIS (rechts, verkleinert) nach [RRH⁺02]

Die AxPe-Prozessorknoten und der DRAM-Bildspeicher wurden als IP-Cores übernommen und wiederverwendet. Dagegen mussten die Schnittstellen neu entwickelt werden.

Vier PN's bilden zusammen mit selbstkonfigurierenden Leitungssystemen und Padzellen ein so genanntes AxPe-Subsystem. Dieses AxPe-Subsystem füllt den möglichen *Design Frame*⁴ des verwendeten Herstellungsprozesses vollständig. Eine AxPe-GIS mit 16 PN's besteht also aus vier AxPe-Subsystemen und hat eine Gesamtfläche von 16,89 cm². Abbildung 4.2 zeigt vereinfacht den Floorplan eines AxPe-Subsystems mit vier PN's und einer AxPe-GIS, die 4 AxPe-Subsysteme monolithisch integriert. Zu erkennen ist, dass die Inter-Core-Busse, d.h. der Video-Bus, der Host-Bus und der Kommunikationsbus, in separaten Verdrahtungskanälen ausgeführt werden. Die PN's werden in Feldern von Laserschaltern (*Laser reconfiguration array*) mit den Inter-Core-Bussen verbunden.

Die Signalintegrität auf diesen Bussen stellt neben deren Anfälligkeit für Fertigungsdefekte ein Hauptproblem dar. Kopplungseffekte, wie sie im Abschnitt 3.1 beschrieben werden, sind messtechnisch nur schwer zu erfassen und nachzuweisen, weil

⁴ Die Größe des *Design Frame* wird durch die maximale in einem Schritt belichtbare Fläche bei der Fotolithographie bestimmt.

die potentiell betroffenen Leitungen nach außen nicht zugänglich sind. Der Entwurf der Busse muss daher auf Simulationsergebnissen beruhen, die eine Bewertung der Signalintegrität für verschiedene Leiteranordnungen und -längen ermöglichen. Dazu ist hier der Schaltungssimulator Eldo und das Leitungsmodell Lossywire [GGRW95] verwendet worden. Die notwendigen Modellparameter werden mit dem Programm Lipacific [GDZ94] extrahiert. Als Grundlage dienen die nominellen Materialparameter und die Mindestabstände der AxPe-GIS-Fertigungstechnologie.

Weil für dieses simulationsbasierte Entwurfsverfahren die genaue Modellierung des Leitungssystems besonders wichtig ist, sind die Modelle messtechnisch verifiziert worden. Dazu wurden mit einem Netzwerkanalysator Zweitor-Messungen an speziellen Messstrukturen im Frequenzbereich bis 20 GHz durchgeführt, die in der AxPe-GIS-Fertigungstechnologie gefertigt wurden. Dabei konnte die genaue Übereinstimmung der gemessenen s-Parameter der Teststrukturen mit den aus der Modellierung gewonnenen Parametern für diese Strukturen gezeigt werden [AWW⁺00].

4.1.1 Übersicht über die Architektur der Prozessorknoten

Der Ansatz, großflächig integrierte Schaltungen aus identischen Subsystemen zusammenzusetzen, ist neuartig und stellt besondere Anforderungen an den Entwurf der Subsysteme [RRT⁺00]. Besonders augenfällig sind die Auswirkungen auf den Floorplan. Die Prozessorknoten sind als schmale Rechtecke geformt. Eine Gruppe von Prozessorknoten wird auf allen vier Seiten von Verdrahtungskanälen für die Inter-Processor-Verbindungsleitungen umgeben. Diese Verdrahtungskanäle gehen über den außen liegenden Ring von Padzellen hinaus. Die Leitbahnen der globalen Verbindungen werden in diesen Kanälen über den Rand eines Subsystems hinausgeführt. Bei der Fertigung entsteht so eine Überlappung von nebeneinander liegenden Subsystemen in den Metalllagen fünf und sechs.

Insgesamt entsteht so auf einem Wafer ein System von Subsystemen, die in einer Netzstruktur mit globalen Leitbahnen untereinander verbunden sind. Allein durch das Sägen des Wafers wird schließlich bestimmt, welche Anzahl von Subsystemen eine großflächig integrierte Schaltung bildet, im Fall der AxPe-GIS sind es vier AxPe-Subsysteme. Die Netzstruktur der gefertigten Verbindungen muss durch Deaktivierung von Teilen der Verbindungen in die von der Architektur her erforderliche Busstruktur umgewandelt werden. Die dazu deaktivierten Verbindungen sind in Abbildung 4.2 gestrichelt gezeichnet. Die aktiven Bussysteme sind durchgezogen gezeichnet. Dabei ist in Abbildung 4.2 (rechts) zu erkennen, dass die Busse über Subsystemgrenzen hinweg reichen.

Eine AxPe-GIS hat eine zweistufige Hierarchie. Auf der oberen Ebene verbindet ein gemeinsames Input/Output Bussystem die Prozessorknoten. Die Prozessorknoten

selbst bestehen aus allen Modulen, die für paralleles Kodieren oder Dekodieren von Videodaten erforderlich sind. Ein Prozessorknoten kann daher beide Aufgaben übernehmen.

Das Blockschaltbild eines AxPe-Prozessorknotens (Abbildung 4.1) soll anhand der Kodierung von Videodaten kurz erläutert werden. Die Video-Eingangsschnittstelle (Digital video in) übernimmt Video-Eingangsdaten vom Video Interface Input Bus und wandelt diese in das für die Encodierung benötigte Datenformat um. Nach der Umwandlung werden die Video-Daten im lokalen DRAM-Bildspeicher (DRAM frame memory) abgelegt.

Speicher-Anfragen werden von einem Arbiter behandelt, der entweder dem Prozessormodul (AxPe processor) oder einem Refresh Controller den Zugriff auf den Speicher gewährt. Die Kommunikationsschnittstelle (Communication interface) behandelt den Datentransfer zwischen Prozessorknoten. Über den bidirektionalen Communications Interface I/O Bus können dadurch Prozessorknoten auf Daten im Bildspeicher eines benachbarten Prozessorknotens zugreifen. Die Bilddaten werden zur Kodierung über einen DMA Controller in den AxPe-Prozessorkern übertragen. Die FIFO-basierte Host-Ausgangsschnittstelle (Host interface out) liefert schließlich die kodierten Daten über den Host Interface Output Bus an einen externen Host-Prozessor.

Die Dekodierung von Bilddaten verläuft entsprechend unter Verwendung der komplementären Schnittstellen Video-Ausgangsschnittstelle (Digital video out) und Host-Eingangsschnittstelle (Host interface in).

4.1.2 Fehlertoleranzkonzept

Bei der Realisierung der AxPe-GIS müssen sowohl Ausbeuteverluste durch Fertigungsdefekte als auch intermittierende Fehler, die im Embedded-DRAM auftreten können, berücksichtigt werden. Ohne den Einsatz von redundanten Schaltungselementen ist es nicht möglich, einen Schaltkreis der angestrebten Komplexität und Größe funktionsfähig zu fertigen. Die verwendeten Testverfahren erlauben daher eine Lokalisierung der Fertigungsdefekte, damit Redundanzen zur Maskierung dieser Defekte eingesetzt werden können.

Defektbehaftete Prozessorknoten werden mittels des in Abschnitt 4.1.3 beschriebenen integrierten Selbsttests erkannt. Erkannte Defekte können maskiert werden, indem der betroffene Prozessorknoten abgeschaltet und vom System getrennt wird. Die Abschaltung eines Prozessorknoten erfolgt durch Programmierung eines Laserschalters oder durch die Programmierung eines Konfigurationsregisters beim Booten. Die Eingänge von abgeschalteten Prozessorknoten werden auf

einen festen logischen Pegel gesetzt, die Ausgänge in einen hochohmigen Zustand gebracht.

Da die Spannungsversorgungen für die Prozessorknoten eines Subsystems separat ausgeführt sind, können einzelne Prozessorknoten, z.B. durch Aktivieren von Lasertrennern, von der Spannungsversorgung getrennt werden. Dadurch nehmen defekte Prozessorknoten keine Leistung mehr auf. Durch diese Maßnahmen bleibt die Funktionsfähigkeit der AxPe-GIS erhalten, eine Einbuße bei der Leistungsfähigkeit muss jedoch in Kauf genommen werden, und die Bildbereiche müssen durch modifizierte Parametersätze - entsprechend den abgeschalteten Prozessorknoten - neu verteilt werden.

Defekte, die die Bussysteme betreffen, wirken sich jedoch nicht auf einzelne Prozessorknoten, sondern auf die gesamte GIS aus und müssen daher repariert werden. Leitungsabschnitte, die von Defekten betroffen sind, können mit Lasertrennern heraus getrennt und durch redundante Leitungen ersetzt werden [MRRT00].

Um intermittierende Fehler im Embedded-DRAM zu maskieren, wurde ein (39, 32)-SEC-DED Hamming-Code [LC83] implementiert. Die physikalisch vorhandene Speichergröße beträgt deswegen 5 MBit pro Prozessorknoten und ergibt eine Vergrößerung der DRAM-Fläche um 37 % gegenüber einer Implementierung ohne fehlerkorrigierenden Code. In [RN99] wurde jedoch gezeigt, dass sich der Flächenzuwachs aufgrund einer Maskierung von Fertigungsdefekten durch den fehlerkorrigierenden Code nicht negativ auf die Ausbeute der AxPe-GIS auswirkt.

4.1.3 Testkonzept

Vorraussetzung für den erfolgreichen Einsatz von Redundanzen zur Ausbeutesteigerung sind eine hohe Testqualität und zuverlässige Fehlerlokalisierung. Diese Anforderungen müssen trotz einer Testdatenbandbreite erreicht werden, die durch den zur Verfügung stehenden Digital-Tester HP82000 beschränkt wird.

Für einige der für das AxPe-Subsystem bzw. die AxPe-GIS verwendeten Cores bestehen bewährte Testprogramme. Diese wurden z.B. im Falle des AxPe-Prozessorkerns für frühere nicht parallele Systeme [HHO⁺98] entwickelt oder wurden im Falle der DRAM-Cores vom Core-Anbieter bereitgestellt. Der Entwurfsaufwand sollte durch Wiederverwendung dieser verifizierten Tests minimiert werden.

Der Test des AxPe-Prozessorkerns beruht auf einem integrierten Selbsttest (BIST). Dieser wird mit Hilfe von BILBO-Registern durchgeführt, die alle herkömmlichen Register bzw. Speicherzellen ersetzen. BILBO-Register werden sowohl zur Erzeu-

gung von pseudo-zufälligen Testmustern als auch zur Bildung von Testsignaturen verwendet. Diese Signaturen werden lokal in jedem Prozessorknoten in einem Testregister zu einer einzigen Signatur komprimiert. Diese Signatur wird nicht mit einer vorausgerechneten Gutsignatur verglichen, sondern es wird folgende Eigenschaft zur Testauswertung ausgenutzt. Für jede mögliche Signatur gibt es eine eindeutige Zahl von Taktzyklen, die benötigt werden, um die Signatur in dem Testregister zu invertieren. Das *Pass/Fail*-Signal wird durch einen Vergleich der invertierten mit der ursprünglichen Signatur nach einer vorausbestimmten Anzahl von Taktzyklen erzeugt.

Die Steuerung des Selbsttests erfolgt durch einen fehlertoleranten programmierbaren Selbsttest-Controller, der außerdem die Auswertung der Signaturen lokal vornimmt. Die Programmierbarkeit des Selbsttest-Controllers erlaubt die Modifikation des Testablaufs und damit die Anpassung des Tests an Erweiterungen des Prozessorkerns. Ein Prozessorknoten des AxPe-Subsystems ist gegenüber dem AxPe-Prozessorkern um die anwendungsspezifischen Schaltungsteile Video-Schnittstelle, Host-Schnittstelle und Kommunikationsschnittstelle sowie den lokalen DRAM-Bildspeicher mit zugehörigem DRAM-Controller erweitert.

Auf eine Einbindung des DRAM-Controllers in den Selbsttest wurde dagegen verzichtet. Der DRAM-Controller und das eingebettete DRAM (eDRAM) wurden als verifizierte *Cores* von STMicroelectronics bereitgestellt. Daher sollten die verifizierten Test von STMicroelectronics für diese Komponenten genutzt werden: ein Selbsttest für das eDRAM und ein *Scan*-Test für den DRAM-Controller.

Ein wesentlicher Schritt des gesamten Testablaufes für AxPe-Subsystem und AxPe-GIS ist der Test und die anschließende Reparatur der globalen Bussysteme, da deren korrektes Funktionieren Voraussetzung für einen fehlerfreien Zugriff auf die Prozessorknoten ist. Kurzschlüsse und Unterbrechungsfehler, die die Bussysteme betreffen, werden mittels eines fehlertoleranten Scan-Pfades lokalisiert, vergleichbar mit dem EXTEST-Modus des *Boundary Scan*-Standards [IEE01].

Die beschriebenen Tests sind für den Prozessorknoten implementiert und auf den Ebenen von Prozessorknoten und AxPe-Subsystem funktional verifiziert worden. Sie sind für AxPe-Subsystem und AxPe-GIS gleichermaßen verwendbar.

4.2 Exemplarische Implementierung eines Selbsttests

4.2.1 Videoprozessorkern

Der Videoprozessorkern vom Typ AxPe besteht aus den Hauptteilen RISC-Prozessor und Low-Level-Co-Prozessor, die jeweils getrennt getestet werden. Der Low-Level-Co-Prozessor ist gekennzeichnet durch einen hochgradig manuellen Entwurf zur Optimierung der Leistungsfähigkeit mit asynchronen Komponenten. In beiden Teilen ersetzen BILBO-Register alle herkömmlichen Register. Die Ersetzung erfolgte durch Editieren der Verilog-Beschreibung des Prozessorkerns auf Register-Transfer-Ebene, indem explizit Instanzen der BILBO-Register angegeben wurden. Die Verknüpfung der BILBO-Register zu BILBO-Ketten sowie die Implementierung der Rückkopplungspolynome durch EXOR-Gatter wurde manuell im *Schematic* des Prozessorkerns mit Hilfe des Composer von Cadence [Cad99b] durchgeführt.

Die Steuerung des Selbsttests erfolgt durch einen programmierbaren Selbsttestcontroller (BIST-Controller). Dieser enthält eine programmierbare *Look-Up-Table*, in der der Ablauf des Testprogramms festgelegt wird, und einen Dekoder, der abhängig von der Testphase die Testsignale zur Steuerung von z.B. BILBO-Registern erzeugt. Der Dekoder ist ebenfalls programmierbar, die Anzahl der dekodierten Testsignale ist jedoch durch die Schaltung festgelegt. Für den RISC-Prozessor wird der Testablauf, d.h. welches BILBO-Register in welcher Testphase als Testmuster-generator bzw. als Signatur-Register konfiguriert wird, anhand von manuell aus der Register-Transfer-Beschreibung abgeleiteten Signalflussdiagrammen entwickelt. Ein Werkzeug, um diese Aufgabe automatisiert auszuführen, stand nicht zur Verfügung.

Der Koprozessor wird durch spezielle von extern zu ladende Testprogramme getestet. Diese Programme sind in Assembler-Code geschrieben und umfassen Operationen auf den Komponenten des Koprozessors im Testmodus und im Betriebsmodus. Die Testauswertung basiert auf Signaturbildung und wird ebenfalls vom BIST-Controller ausgeführt.

Der Entwurf dieser Tests verlangte vom Testentwickler detaillierte Kenntnisse der Schaltungsstruktur und war daher nur durch die direkte Einbindung in die Schaltungsentwicklung möglich. Die beschriebene Testimplementierung wurde für den AxPe-Prozessorkern entwickelt und wird hier vollständig übernommen.

4.2.2 Anwendungsspezifische Schnittstellen im Prozessorknoten

Der Test der anwendungsspezifischen Schnittstellen eines Prozessorknotens erfolgt durch die Erweiterung des in Abschnitt 4.2.1 beschriebenen RISC-Prozessor-Selbsttests. Entsprechend dem oben beschriebenen Verfahren werden auch hier alle Speicherelemente durch BILBO-Register ersetzt. Diese BILBO-Register werden an die vorhandenen BILBO-Ketten angehängt, so dass sich deren Länge um etwa den Faktor vier vergrößert.

Eine besondere Schwierigkeit besteht darin, *X-Violations*, d.h. die Invalidierung der Testsignaturen durch undefinierte Zustände, während des Selbsttests zu vermeiden. Daher werden insbesondere die eingebetteten Zwei-Port-Speicher mit einem speziellen Testzustand versehen, der eine Initialisierung vor einem pseudo-zufälligen lesenden Zugriff auf ihre Inhalte ermöglicht, damit das System im Selbsttest nicht in einen unbestimmten Zustand gerät. Die Initialisierung findet in der ersten von vier Testphasen statt. Die Testabdeckung dieser Phase ist für die betroffenen Module Video-Interface, Kommunikationsinterface und Host-Interface zwangsläufig gering. Ebenfalls um X-Violations zu vermeiden, muss der Koprozessor und das DRAM während des Selbsttests isoliert werden, weil eine einfache Initialisierung, beispielsweise ein Reset, für diese Elemente nicht vorgesehen ist.

4.2.3 DRAM und Controller

Das Embedded-DRAM und sein Controller sind als IP-Cores zusammen mit einer Testbeschreibung von ST Microelectronics zur Verfügung gestellt worden. Diese Beschreibung sieht vor, zunächst einen Wafer-Test der DRAMs für jeden Prozessorknoten getrennt durchzuführen. Für den Wafertest müssen einzelne Prozessorknoten mit einer Nadelkarte auf 126 dedizierten Wafer-Test-Pads kontaktiert werden, die auf jedem Subsystem hinzugefügt werden müssen. Die Ergebnisse dieser Tests sind die Grundlage für die Aktivierung von redundanten Spalten von Speicherzellen zur Ausbeutesteigerung durch den Halbleiterhersteller.

Nach dem Packaging der AxPe-Subsysteme bzw. der großintegrierten Schaltungen AxPe-GIS wird ein finaler Test des DRAM-Controllers und ein Selbsttest der DRAMs, der von einem integrierten Speicher-BIST-Controller gesteuert wird, am Laboratorium für Informationstechnologie (Lfi) ausgeführt. Der Selbsttest des DRAM-Controller-Cores zeigte aber bereits in der Verifikation Fehler. Daher ist am Lfi zusätzlich ein *March-Test*-ähnlicher Pseudozufallstest [van98] für das DRAM entwickelt und implementiert worden, der den ursprünglich geplanten DRAM-Test ersetzt. Der Test des DRAM-Controller wiederum erfolgt im Gegensatz zur allgemei-

nen Teststrategie extern über einen in den Controller integrierten Scan-Pfad. Die Testmuster-generierung ist mit Synopsys Test Compiler ATPG [Syn00] durchgeführt worden.

4.2.4 Leitungssysteme

Der Entwurf der globalen Inter-Core-Busse ist vollständig manuell erfolgt (inkl. Entwicklung des Layouts). Für den Test dieser Leitungssysteme wird ein Ad-Hoc-Ansatz verwendet, der vom EXTEST-Modus des *Boundary-Scan*-Standards IEEE 1149.1 [IEE01] abgeleitet worden ist. Alle Ein- und Ausgänge von Prozessorknoten werden mit vereinfachten *Boundary-Scan*-Zellen versehen, die zu insgesamt zwei Ketten auf jedem AxPe-Subsystem verbunden sind. Damit wird einerseits jedes Leitungsende für den Test von außen zugänglich gemacht, andererseits können die Prozessorknoten für den Leitungstest isoliert werden. Der Test basiert auf Walking-1-Testmustern und erlaubt daher die Lokalisierung aller Leitungskurzschlüsse, Stuck-At-Fehler und Unterbrechungsfehler. Ein Test auf dynamische Fehler durch Leitungskopplung ist nicht möglich.

Der Test erfolgt seriell, d.h. ein Testmuster wird in die *Boundary-Scan*-Kette hinein geschoben, wird dann an die Leitungen angelegt bzw. an den Leitungsenden⁵ abgefragt, und schließlich werden die Testergebnisse seriell heraus geschoben und extern analysiert. Die Analyseergebnisse liefern die notwendige Information zur Reparatur der Leitungssysteme. Nach erfolgter Reparatur ist ein erneuter Testdurchlauf erforderlich.

Die Testrealisierung für die AxPe-GIS behandelt das Problem der Signalintegrität der globalen Busse mit einem „Correct-by-Construction“-Ansatz. Die maximale Länge der globalen Verbindungsnetze beträgt auf der AxPe-GIS von Bond-Pads an der linken oberen Kante bis zu einem Prozessorknoten rechts unten etwa 6 cm. Daher sind Leitungsverstärker erforderlich, um die Signalintegrität zu erhalten. Zur Dimensionierung und Platzierung der Leitungsverstärker wurden RLC-Leitungsparameter für verschiedenste Buskonfigurationen und -längen extrahiert und mit dem Schaltungssimulator ELDO simuliert. Das Ergebnis der Simulationen ist, dass zur Gewährleistung der Signalintegrität und zur Minimierung der Signallaufzeiten auf den Leitungssystemen im Abstand von 2 cm Leitungsverstärker erforderlich sind. Ein entsprechender Test ist jedoch nicht implementiert. Die Frage, wie ein geeigneter Leitungstest realisiert werden kann, wird in Kapitel 5 behandelt.

⁵ Das sind die Eingänge der Prozessorknoten.

Modul	Anzahl der <i>Stuck-at</i> -Fehler		Testabdeckung	Simulationsdauer
	total	gefunden		
adm	6974	6530	93,6%	11 min 57 s
alu	12358	1401	11,3%	15 h 33 min 43 s
dc	2748	2671	97,2%	3 min 14 s
dm	4160	3805	95,9%	11 min 39 s
nau	5994	5431	90,6%	9 h 35 min 32 s
pgm	4054	3251	80,4%	3 h 1 min 59 s
rio	12268	11325	92,3%	15 min 5 s
ros	2984	2863	95,9%	3 min 6 s
slm	30334	28903	95,2%	6 h 1 min 7 s

Tabelle 4.1: Ergebnisse der Fehlersimulation für den AxPe-Prozessorkern

4.2.5 Fehlerabdeckung

Die erreichte Testqualität muss mit Hilfe von Fehlersimulationen mit dem Programm Verifault-XL [Cad99c] überprüft werden. Dabei wird gemäß der allgemeinen Praxis nur das Einzelhaftfehlermodell verwendet. Zur Reduzierung der Simulationsdauer⁶ sind diese Simulationen modular durchgeführt worden, d.h. nicht für einen vollständigen Prozessorknoten, sondern einzeln für seine Elemente und Unterelemente. Die Ergebnisse sind in Tabelle 4.1 exemplarisch für die Elemente des AxPe-Prozessorkerns dargestellt.

Die Testabdeckung liegt bei den meisten Elementen oberhalb von 90%. Das Element alu ist dagegen mit pseudo-zufälligen Tests nur unzureichend zu testen. Die Simulationsdauer für die betrachteten Elemente zeigt, dass die Fehlersimulation ein besonders zeitaufwändiger Schritt bei der Testentwicklung ist.

Diese Ergebnisse zeigen, dass mit der beschriebenen Selbsttestarchitektur eine ausreichende Fehlerabdeckung bei minimaler Testdatenbandbreite zu einem (externen) Digital-Tester erreicht werden kann. Die Testabdeckung kann durch Verwendung einer größeren Zahl von pseudo-zufälligen Testmustern erhöht werden. Bei den Simulationen wird eine Beschränkung der maximalen Testlänge auf 64.000 Taktzyklen berücksichtigt, die durch den Digital-Tester HP82000 gegeben ist. Die erzielten Testabdeckungen lassen sich auch mit anderen Testarchitekturen erreichen – unter anderen Randbedingungen für die verfügbaren Testressourcen. Kommerzielle DFT- und ATPG-Werkzeugen sind aber zum Beispiel wegen asynchroner Schaltungsteile nicht anwendbar, so dass Vergleichswerte für die Testabdeckung nicht ermittelt werden können.

⁶ Die Fehlersimulationen sind auf einer Workstation Sun 420R (4 × UltraSparcIII, 450 MHz, 4 GB RAM) durchgeführt worden.

4.3 Bewertung der Schaltungsentwicklung für die AxPe-GIS

Mit der Implementierung des beschriebenen Selbsttestverfahrens und den Konzepten zur Realisierung des Wafer-Step-Verfahrens sind im Rahmen dieser Arbeit wesentliche Beiträge zur Entwicklung des Multiprozessor-Systems AxPe-GIS geleistet worden. Die technologische Entwicklung erlaubte durch Bereitstellung einer eDRAM-Technologie mit minimalen Strukturgrößen von $0,25\ \mu\text{m}$ erstmalig die vollständige Integration der benötigten Bildspeicher – ein Weg, der auch bei der weit verbreiteten PlayStation2 von Sony Computer Entertainment erfolgreich beschritten wurde [Hol02].

Die Fertigung von maskenübergreifenden, großflächig integrierten Schaltungen hat in diesem Projekt eine technologische Hürde dar, die durch die Entwicklung eines neuen Wafer-Step-Verfahrens überwunden werden kann [RRH⁺02]. Damit können sehr kompakte programmierbare Systeme für die Echtzeit-Verarbeitung von Videodaten realisiert werden.

Allerdings entstehen durch die Nutzung dieser technologisch sehr anspruchsvollen Möglichkeiten neue Risiken für die Entwicklung und Fertigung solcher Systeme. Das Wafer-Step-Verfahren verlangt vom Halbleiterhersteller eine Veränderung des Herstellungsablaufs, vor allem bei der Maskenbelichtung für die oberste Metalllage. Dabei ist es schwierig, einen industriellen Partner zu finden, der diese Änderungen zulässt.

Dies, sowie die Nutzung neuartiger Technologieoptionen wie das eDRAM, machen eine Festlegung auf eine bestimmte Fertigungstechnologie bei einem Halbleiterhersteller erforderlich. In diesem Vorhaben sind durch diese Festlegung zeitliche Verzögerungen bei der Entwicklung der Systeme dadurch verursacht worden, dass der für den Schaltungsentwurf nötige DRAM-Core erst sehr spät zur Verfügung gestellt wurde und zuletzt den ursprünglichen Spezifikationen nur ungenügend entsprach. Daher sind kurzfristig umfangreiche Änderungen und Ergänzungen am Design notwendig geworden.

Zuletzt hat jedoch der Halbleiterhersteller für die Embedded-DRAMs keine Produktionsreife erzielen können, so dass der für die Fertigung der Schaltung AxPe-GIS vorgesehene Herstellungsprozess kurzfristig abgekündigt wurde. Auf die Fertigung der Option eDRAM für den $0,25\ \mu\text{m}$ -Prozess ist vermutlich verzichtet worden, weil entgegen den ursprünglichen Annahmen das eDRAM gegen Ende des Projekts für den $0,25\ \mu\text{m}$ -Prozess an wirtschaftlicher Bedeutung verloren hat. Daher ist die Fertigung der vollständig entwickelten und verifizierten Schaltung AxPe-GIS nicht ausgeführt worden.

Eine kurzfristige Portierung auf einen angebotenen alternativen Herstellungsprozess in 0,18 μm -Technologie ist für verwendeten Cores in der bestehenden Form nicht möglich. Die Gründe liegen in sehr technologiespezifischen Schaltungsteilen, wie z.B. Laserschaltern [MRRT00] und eDRAMs, sowie in den Architekturbesonderheiten der AxPe-GIS. Diese Besonderheiten sind asynchrone Schaltungsteile im Prozessorkern und die Anpassungen an die maskenübergreifende großflächige Integration. Der Entwurf dieser Komponenten wird nur unzureichend durch automatische Werkzeuge unterstützt, so dass Änderungen unter großem Zeitaufwand manuell durchgeführt werden müssten. Es bestände daher das Risiko, dass bereits der nächste Wechsel der Technologiegeneration stattfindet, bevor die Änderungen abgeschlossen sein können.

Aufgrund dieser Erfahrungen ergeben sich folgende Konsequenzen für die Realisierung eines programmierbaren, monolithisch integrierten Systems zur Echtzeit-Bildverarbeitung.

- Der Entwurf muss so modifiziert werden, dass eine vollständige Unterstützung durch automatische Entwurfswerkzeuge erreicht wird, damit die Entwurfszeit nicht dem Technologiefortschritt nachläuft.
- Die Herstellung von großen eDRAMs wird bisher von nur wenigen Herstellern beherrscht und birgt daher ein Risiko. Andererseits hat sich die Zugriffszeit auf externe Speicher und die erreichbare Datenrate mit dem Fortschritt der Technologie verbessert, z.B. durch Speichertypen wie DDR SDRAM [HP03], so dass auf eDRAMs verzichtet werden kann, ohne dass die Leistungsfähigkeit beeinträchtigt wird.
- Durch die Verwendung externer Speicher und einer 0,18 μm -Technologie verringert sich die benötigte Chipfläche soweit, dass die maskenübergreifende Vernetzung der Subsysteme der GIS nicht mehr erforderlich ist und auf das spezielle Wafer-Step-Verfahren verzichtet werden kann.
- Als Folge müssen auch die Subsysteme einer GIS nicht mehr identisch sein, d. h. es kann ein Multi-Core-Prozessor realisiert werden, der bezüglich der zukünftig zu verarbeitenden Algorithmen eine größere Flexibilität und Leistungsfähigkeit bietet.

Die Entwicklung der AxPe-GIS ist von der Wiederverwendung des Videoprozessor-kerns und die auf identischen Prozessorknoten basierende parallele Architektur geprägt. Typisch für den Core-basierten Entwurf ist die Wiederverwendung der für die Cores entwickelten Tests. Die Multiprozessor-Architektur führt zur Implementierung von Inter-Core-Bussen. Diese sind wegen der System-Realisierung als großflächig integrierte Schaltung ausgesprochen lang und damit anfällig für Fertigungsfehler. Der in Abschnitt 4.2.4 beschriebene Leitungstests für diese Busse

4 Testimplementierung für ein Multiprozessorsystem

bedeutet ein Risiko, weil Fehler durch Leitungskopplung nicht erfasst werden. Geeignete Testmethoden werden im folgenden Kapitel 5 präsentiert. Diese haben aber während der Planung der AxPe-GIS noch nicht zur Verfügung gestanden und sind daher nicht implementiert worden.

5 Schieberegister-basierter Selbsttest von Inter-Core Verbindungsnetzen mit induktiver Kopplung

Die bekannten Verfahren für den Test von Verbindungsnetzen, wie sie in Kapitel 3 beschrieben sind, weisen im Wesentlichen zwei Lücken auf. Erstens basieren sie auf einem einfachen Leitungsmodell, das induktive Leitungsbeläge vernachlässigt, und zweitens nutzen sie nicht die durch standardisierte Testarchitekturen gegebenen schaltungstechnischen Randbedingungen und führen daher zu vermeidbarem Schaltungsoverhead.

Daher werden in dieser Arbeit Bedingungen untersucht, unter denen bei Berücksichtigung induktiver Kopplung auf einer Leitung eine maximale Signalverzögerung oder eine maximale Amplitude eines Glitch auftritt. Ferner wird eine Realisierung eines Testmuster-generators und eines Testauswerters für den integrierten Selbsttest dargestellt, die durch Erweiterung eines standardisierten Core-Wrappers entsteht.

In der Regel beziehen sich die Untersuchungen zum Verhalten von Inter-Core-Verbindungsnetzen auf ein System von mehreren parallelen Leitungen mit jeweils genau einem Leitungstreiber und einem Empfänger. Reale Bussysteme können aber mehr als zwei Cores verbinden, die sowohl schreibenden als auch lesenden Zugriff auf den Bus haben. Für die Implementierung eines Leitungstests ergeben sich daraus zusätzliche Randbedingungen, die hier durch einen Kopplungsgraphen beschrieben werden.

5.1 Testverfahren für Verbindungsnetze mit induktiver Kopplung

Verschiedene Arbeiten auf dem Gebiet der Modellierung langer, verlustbehafteter Leitungen haben gezeigt, dass die Vernachlässigung von Induktivitäten bei Leitungsgeometrien, wie sie für aktuelle und zukünftige Technologiegenerationen typisch

sind, zu signifikanten Fehlern führt [Gra91, DKR⁺97, MRR98, IF00]. Das in Abschnitt 3.1.2 beschriebene Maximum Aggressor Fehlermodell beruht auf der Modellierung gekoppelter Leitungen durch ein Netzwerk von Kapazitäten und Widerständen. Eine einfache Fehlerbedingung kann dadurch gefunden werden, dass sich die Wirkung der kapazitiven Kopplung auf eine so genannte Opferleitung von alle übrigen Leitungen, den Aggressorleitungen, konstruktiv überlagert. Diese Eigenschaft gilt unabhängig von der Geometrie der Leitungen. Dadurch können Tests abgeleitet werden, bei denen auf allen Aggressorleitungen ein Signalübergang in dieselbe Richtung stattfindet. Diese Tests decken das Maximum Aggressor Fehlermodell vollständig ab.

Ein Signalverlauf, bei dem alle Aggressorleitungen in dieselbe Richtung schalten, während die Opferleitung in die entgegengesetzte Richtung schaltet, wird als *Odd Mode* bezeichnet. In den Arbeiten von Grabinski und anderen [Gra91, NTO⁺98] wird dieser Signalverlauf verwendet, um für die Analyse des Signalübertragungsverhaltens von Leitungen maximale Signalverzögerung auf der Opferleitung zu erzeugen. Er entspricht den Tests für das Maximum Aggressor Fehlermodell nach den Tabellen 3.4 und 3.5.

Ein Signalverlauf, bei dem alle Aggressorleitungen und auch die Opferleitung in dieselbe Richtung schalten, wird als *Even Mode* bezeichnet. Dieser Signalverlauf wird verwendet, um minimale Signalverzögerung auf der Opferleitung zu erzeugen. Ein derartiger Test gehört nicht zum Maximum Aggressor Fehlermodell. Anschaulich gesehen kann aber auch eine geringere als die spezifizierte Signalverzögerung zu einem Fehler führen, wenn dadurch ein Signalwechsel am Eingang einer Speicherzelle auftritt, bevor diese den aktuellen Wert sicher übernommen hat (Halte-Zeit-Verletzung). Ein solcher Verzögerungsfehler gehört zum Fehlertyp *Hazard*.

Die maximale Signalverzögerung auf einer Leitung kann weiter vergrößert werden, indem die Schaltzeitpunkte der Aggressorleitungen gegenüber der Opferleitung verschoben werden [NTO⁺98, GARP98, CMS01]. Für den Test von Leitungssystemen ist eine solche Verschiebung aber unrealistisch, weil eine exakte Einstellung schaltungstechnisch sehr aufwändig wäre. Das Verhältnis der Schaltzeitpunkte ist in einer integrierten Schaltung durch das Netzwerk für die Taktverteilung festgelegt. Davon abweichende Schaltzeiten führten zu Fehlerbedingungen, die im realen Betrieb nicht auftreten können, so dass getestete Schaltungen fälschlich als fehlerhaft deklariert werden können. Weil aber die Schaltzeitpunkte einen großen Einfluss auf den Signalverlauf haben können, ist es notwendig, während des Tests dieselben Schaltzeitpunkte wie im Systembetrieb zu verwenden. Das kann erreicht werden, wenn für den Test dieselben Signalpfade, d.h. Speicherzellen und Treiber, zu den Leitungen hin genutzt werden wie im Systembetrieb.

Die Auswirkung eines Schaltvorgangs auf eine Opferleitung ist jedoch bei kapazitiver und induktiver Kopplung unterschiedlich. Bei kapazitiver Kopplung führt eine steigende Signalflanke der Aggressorleitung auf der Opferleitung zu einem Span-

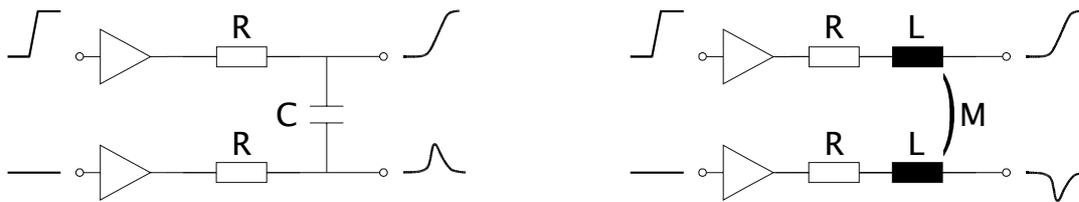


Abbildung 5.1: Qualitative Darstellung von kapazitiver Kopplung (links) und induktiver Kopplung (rechts)

nungspuls mit positivem Spitzenwert. Bei induktiver Kopplung ist dagegen ein Spannungspuls mit negativem Spitzenwert zu beobachten. Abbildung 5.1 stellt diesen Zusammenhang schematisch dar. Dadurch ist das Zusammenwirken von induktiver und kapazitiver Kopplung wesentlich komplexer als die konstruktive Überlagerung bei einem Netzwerk von Kapazitäten und Widerständen.

In [NA01] wird durch Simulation eines Leitungsmodells mit kapazitiven und induktiven Leitungsbelägen an einem Beispiel gezeigt, dass es Signalverläufe auf gekoppelten Leitungssystemen gibt, die eine größere Signalverzögerung als der *Odd Mode* liefern. Anschaulich lässt sich das dadurch erklären, dass bei dicht nebeneinander liegenden Leitungen die kapazitive Kopplung dominiert und für eine Zunahme der Signalverzögerung sorgt. Zu den äußeren Leitungen dominiert jedoch die induktive Kopplung, die die Signalverzögerung wieder verringert. Die maximale Signalverzögerung auf der Opferleitung ist daher zu erwarten, wenn einige der benachbarten Aggressorleitungen in entgegengesetzter Richtung schalten und einige der entfernteren Aggressorleitung in dieselbe Richtung wie die Opferleitung schalten. Dabei ist das Signalverhalten abhängig von der Leitungsgeometrie und damit von den Leitungsparametern.

Zur Bestimmung der maximalen Signalverzögerung wird in [AN02] ein Verfahren dargestellt, das die vollständige Suche über alle möglichen Signalverläufe der Aggressorleitungen ermöglicht. Für Bussysteme mit praxisrelevanter Breite ist die vollständige Suche mit dem zugrunde liegenden Berechnungsverfahren nicht durchführbar. Die vollständige Suche umfasst $N \cdot 4^N$ Berechnungen für ein N -bit Leitungssystem. Darin steht die Vier in der Basis für die vier möglichen Zustände einer Leitung (steigendes Signal, fallendes Signal, konstante logische 1 oder konstante logische 0). Die Autoren schlagen daher die Verwendung von pseudo-zufälligen Leitungstests vor. Sie können in [NA02] exemplarisch zeigen, dass mit diesen mehr Verzögerungs- und Glitchfehler gefunden werden können als mit den Tests nach den Tabellen 3.2 bis 3.5. Die Autoren können jedoch keine allgemeinen Bedingungen für die Leitungsgeometrien angeben, unter denen das Maximum Aggressor Fehlermodell ungültig wird.

In dieser Arbeit wird in Anlehnung an [NTO⁺98] davon ausgegangen, dass der *Odd Mode* eine hinreichende Fehlerbedingung für alle dynamischen Fehler ist, d.h. auch

unter Berücksichtigung induktiver Kopplung. Weitere Untersuchungen deuten darauf hin, dass für Technologiegenerationen jenseits der $0,10\ \mu\text{m}$ die Dämpfung durch ohmsche Leitungsverluste so groß wird, dass die Auswirkungen induktiver Kopplung kaum noch beobachtbar sind [TRGK02]. Eine nach dem Stand der Technik beste Abdeckung aller Kopplungsfehler wird erreicht, wenn der Test im *Odd Mode* durch einen pseudo-zufälligen Leitungstests ergänzt wird. Beide Tests können im Gegensatz zu den nach [AN02] deterministisch berechneten Tests mit Schieberegistern erzeugt werden, wie in den folgenden Abschnitten gezeigt wird.

5.2 On-Chip Testmustererzeugung mit Schieberegistern

Mit den in den Tabellen 3.2 bis 3.5 gezeigten Testmusterfolgen, die für ein Leitungsmodell ohne induktive Kopplung hergeleitet wurden, können (mit den oben genannten Einschränkungen) Bedingungen für maximale Verzögerung und maximale Glitches auch unter Berücksichtigung induktiver Kopplung gegeben werden. Diese Testmusterfolgen werden vom Walking-1-Test abgeleitet, so dass ihre Implementierung basierend auf Schieberegistern möglich ist. In den folgenden Absätzen wird diese Erweiterung in zwei Schritten gezeigt. Im ersten Schritt sollen nur die Tests für Verzögerungsfehler laut Tabelle 3.4 und Tabelle 3.5 hinzugefügt werden.

Tabelle 5.1 zeigt für ein N -bit Leitungssystem einen Leitungstest der Länge $4N$, der von einem Walking-1-Test abgeleitet wird, indem zunächst nach jedem Testmuster das invertierte Testmuster eingefügt wird, ohne dass die Eins zur nächsten Schieberegisterzelle verschoben wird. Die Zweiersequenz von Testmuster und invertiertem Testmuster wird ausserdem wiederholt, so dass aus jedem Testmuster des Walking-1-Tests vier Testmuster des erweiterten Tests werden. Um dies zu verdeutlichen, werden die einzelnen Schritte des Test $t_{k,l}$ in Vierergruppen zusammengefasst, bei denen nicht der Zählindex k , sondern ein Gruppenindex l inkrementiert wird.

Dieser erweiterte Test enthält bereits die notwendigen Signalübergänge entsprechend den Tabellen 3.4 und 3.5. Diese sind in Tabelle 5.1 hervorgehoben. Die mit Zielfehler bezeichnete Spalte gibt an, welchen kritischen Signalübergang der Test stimuliert. Beispielsweise initialisiert der Test $t_{i,1}$ den Fehler „Verzögerung bei steigender Flanke“ auf Leitung i ($VstF_i$). Der Test $t_{i,2}$ stimuliert diesen Fehler und initialisiert zugleich den Fehler „Verzögerung bei fallender Flanke“ auf Leitung i (VfF_i). Der Test $t_{i,3}$ schließlich stimuliert den zweiten Verzögerungsfehler. Tests, die mit - gekennzeichnet sind, werden nur zum Auffüllen des Viererschemas benötigt, das sich wegen der Realisierung des Testmustergenerators als Schieberegister ergibt.

5.2 On-Chip Testmustererzeugung mit Schieberegistern

	K_0	K_1	\dots	K_{i-1}	K_i	K_{i+1}	\dots	K_{N-2}	K_{N-1}	Zielfehler
$t_{0,0}$	1	0	\dots	0	0	0	\dots	0	0	-
$t_{0,1}$	0	1	\dots	1	1	1	\dots	1	1	init
$t_{0,2}$	1	0	\dots	0	0	0	\dots	0	0	$VstF_0$;init
$t_{0,3}$	0	1	\dots	1	1	1	\dots	1	1	VfF_0
$t_{1,0}$	0	1	\dots	0	0	0	\dots	0	0	-
$t_{1,1}$	1	0	\dots	1	1	1	\dots	1	1	init
$t_{1,2}$	0	1	\dots	0	0	0	\dots	0	0	$VstF_1$;init
$t_{1,3}$	1	0	\dots	1	1	1	\dots	1	1	VfF_1
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
$t_{i-1,0}$	0	0	\dots	1	0	0	\dots	0	0	-
$t_{i-1,1}$	1	1	\dots	0	1	1	\dots	1	1	init
$t_{i-1,2}$	0	0	\dots	1	0	0	\dots	0	0	$VstF_{i-1}$;init
$t_{i-1,3}$	1	1	\dots	0	1	1	\dots	1	1	VfF_{i-1}
$t_{i,0}$	0	0	\dots	0	1	0	\dots	0	0	-
$t_{i,1}$	1	1	\dots	1	0	1	\dots	1	1	init
$t_{i,2}$	0	0	\dots	0	1	0	\dots	0	0	$VstF_i$;init
$t_{i,3}$	1	1	\dots	1	0	1	\dots	1	1	VfF_i
$t_{i+1,0}$	0	0	\dots	0	0	1	\dots	0	0	-
$t_{i+1,1}$	1	1	\dots	1	1	0	\dots	1	1	init
$t_{i+1,2}$	0	0	\dots	0	0	1	\dots	0	0	$VstF_{i+1}$;init
$t_{i+1,3}$	1	1	\dots	1	1	0	\dots	1	1	VfF_{i+1}
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
$t_{N-2,0}$	0	0	\dots	0	0	0	\dots	1	0	-
$t_{N-2,1}$	1	1	\dots	1	1	1	\dots	0	1	init
$t_{N-2,2}$	0	0	\dots	0	0	0	\dots	1	0	$VstF_{N-2}$;init
$t_{N-2,3}$	1	1	\dots	1	1	1	\dots	0	1	VfF_{N-2}
$t_{N-1,0}$	0	0	\dots	0	0	0	\dots	0	1	-
$t_{N-1,1}$	1	1	\dots	1	1	1	\dots	1	0	init
$t_{N-1,2}$	0	0	\dots	0	0	0	\dots	0	1	$VstF_{N-1}$;init
$t_{N-1,3}$	1	1	\dots	1	1	1	\dots	1	0	VfF_{N-1}

Tabelle 5.1: Erweiterter Walking-1-Test für Verzögerungsfehler bei einem N -bit Leistersystem

5 Schieberegister-basierter Selbsttest von Inter-Core Verbindungsnetzen

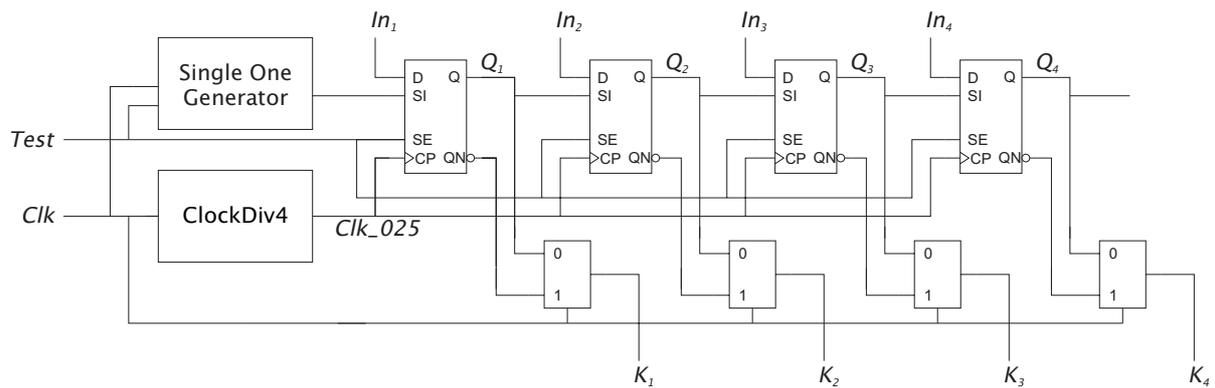


Abbildung 5.2: Erweiterter Testmustergenerator für Tests auf Verzögerungsfehler

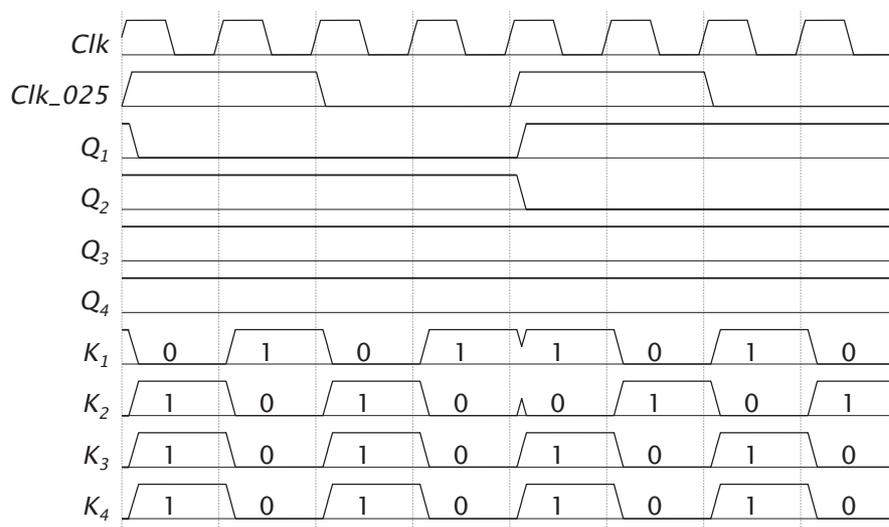


Abbildung 5.3: Steuersignale des erweiterten Testmustergenerators

Die Erzeugung dieser Testmuster erfolgt durch eine Erweiterung des in Abbildung 3.3 dargestellten Schieberegisters für den Walking-1-Tests. Für jedes Flipflop des Schieberegisters wird nur ein zusätzlicher 2-zu-1-Multiplexer benötigt. Dieser verbindet entweder den Ausgang Q eines Flipflops oder den invertierten Ausgang QN mit der zu testenden Busleitung. Außerdem ist ein Takteiler pro Testmustergenerator nötig, der das Taktsignal der Schaltung für den Takteingang der Flipflops heruntersetzt, so dass der Registerinhalt nur alle vier Systemtakte um einen Schritt weiter geschoben wird. Diese Schaltung ist in Abbildung 5.2 für eine Breite von 4 bit dargestellt.

Der Testmustergenerator ist also ein Schieberegister, bei dem abwechselnd der Registerinhalt bzw. der invertierte Registerinhalt auf das Leitungssystem geschrieben wird. Der Wechsel von positivem und negativem Ausgang wird durch das Taktsignal Clk gesteuert. Das Schieberegister wird zunächst vollständig mit logischen Nullen initialisiert, bevor es von links nach rechts von einer einzelnen logischen Eins

durchlaufen wird. Diese Eins wird nach jeweils vier Taktzyklen durch eine positive Flanke des Signal Clk_{025} weitergeschoben. Dadurch können die notwendigen Vierergruppen von Testmustern generiert werden.

Der Schaltungsoverhead ist minimal. Die Signalverzögerung im Systembetrieb wird gegenüber einem normalen Register durch die Verwendung Scan-fähiger Flipflops sowie durch den Multiplexer im Signalpfad vergrößert. Es sind andere Schieberegister-basierte Testmustererzeuger möglich, die keinen zusätzlichen Multiplexer im Signalweg erfordern, aber dafür eine aufwändigere Steuerung mit einem endlichen Automaten benötigen. Abbildung 5.3 zeigt den zeitlichen Verlauf des Taktsignals Clk , des abgeleiteten Taktsignals Clk_{025} , der Zustände der vier Flipflops Q_1 bis Q_4 , sowie der Eingangssignale der zu testenden Busleitungen K_1 bis K_4 .

Für einen vollständigen Test bezüglich dynamischer Fehler sind zusätzlich zu den in Tabelle 5.1 dargestellten Testmustern für jede Opferleitung die Signalübergänge nach Tabelle 3.2 und Tabelle 3.3 notwendig. Tabelle 5.2 zeigt für ein N -bit Leitungssystem einen Leitungstest der Länge $8N$. Die notwendigen Signalübergänge sind wieder hervorgehoben.

Die mit Zielfehler bezeichnete Spalte gibt wieder an, welchen kritischen Signalübergang der Test stimuliert. Beispielsweise initialisiert der Test $t_{i,1}$ den Fehler „positiver Glitch“ auf Leitung i (pG_i). Der Test $t_{i,2}$ stimuliert diesen Fehler und initialisiert zugleich den Fehler „Verzögerung bei steigender Flanke“ auf Leitung i ($VstF_i$). Der Test $t_{i,3}$ schließlich stimuliert diesen Fehler. Tests, die mit - gekennzeichnet sind, werden nur zum Auffüllen des Achterschemas benötigt, das sich wegen der Realisierung des Testmustererzeugers als Schieberegister ergibt.

Ein vollständiger Testmustererzeuger ist in Abbildung 5.4 für einen 4-Bit-Bus gezeigt. Neben dem Schieberegister wird ein 4-zu-1-Multiplexer für jede Leitung sowie ein endlicher Automat mit 8 Zuständen für die Steuerung der Multiplexer benötigt. Die Schaltungskomplexität ist daher unabhängig von der Breite des Bussystems. Die Multiplexer verbinden die Ausgänge der Flipflops und deren invertierte Ausgänge, sowie ein konstantes Signal Low für die logische 0 oder ein Signal High für die logische 1 mit den einzelnen Leitungen.

Dabei durchläuft wiederum eine Walking-1 das Schieberegister. Der Registerinhalt wird mit dem Signal Shift wegen der geraden Taktteilung alle acht Perioden des Taktsignals Clk um eine Registerstufe weiter geschoben. Daher besteht eine Testsequenz für eine Opferleitung K_i aus 8 Tests $t_{i,0..7}$ anstelle der minimal notwendigen 6 Tests, so dass der endliche Automat 8 statt 6 Zustände benötigt.

Abbildung 5.5 zeigt die Steuersignale des Testmustererzeugers für den Test auf dynamische Fehler. Die Signale zur Ansteuerung der Multiplexer S_0 und S_1 sind ebenfalls (verschobene) Teiler des Taktsignals Clk . Daher könnten sie auch direkt

	K_0	K_1	\dots	K_{i-1}	K_i	K_{i+1}	\dots	K_{N-2}	K_{N-1}	Zielfehler
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots
$t_{i-1,0}$	0	0	\dots	0	0	0	\dots	0	0	init
$t_{i-1,1}$	1	1	\dots	0	1	1	\dots	1	1	$pG_{i-1};\text{init}$
$t_{i-1,2}$	0	0	\dots	1	0	0	\dots	0	0	$VstF_{i-1}$
$t_{i-1,3}$	1	1	\dots	1	1	1	\dots	1	1	-
$t_{i-1,4}$	1	1	\dots	1	1	1	\dots	1	1	init
$t_{i-1,5}$	0	0	\dots	1	0	0	\dots	0	0	$nG_{i-1};\text{init}$
$t_{i-1,6}$	1	1	\dots	0	1	1	\dots	1	1	VfF_{i-1}
$t_{i-1,7}$	0	0	\dots	0	0	0	\dots	0	0	-
$t_{i,0}$	0	0	\dots	0	0	0	\dots	0	0	init
$t_{i,1}$	1	1	\dots	1	0	1	\dots	1	1	$pG_i;\text{init}$
$t_{i,2}$	0	0	\dots	0	1	0	\dots	0	0	$VstF_i$
$t_{i,3}$	1	1	\dots	1	1	1	\dots	1	1	-
$t_{i,4}$	1	1	\dots	1	1	1	\dots	1	1	init
$t_{i,5}$	0	0	\dots	0	1	0	\dots	0	0	$nG_i;\text{init}$
$t_{i,6}$	1	1	\dots	1	0	1	\dots	1	1	VfF_i
$t_{i,7}$	0	0	\dots	0	0	0	\dots	0	0	-
$t_{i+1,0}$	0	0	\dots	0	0	0	\dots	0	0	init
$t_{i+1,1}$	1	1	\dots	1	1	0	\dots	1	1	$pG_{i+1};\text{init}$
$t_{i+1,2}$	0	0	\dots	0	0	1	\dots	0	0	$VstF_{i+1}$
$t_{i+1,3}$	1	1	\dots	1	1	1	\dots	1	1	-
$t_{i+1,4}$	1	1	\dots	1	1	1	\dots	1	1	init
$t_{i+1,5}$	0	0	\dots	0	0	1	\dots	0	0	$nG_{i+1};\text{init}$
$t_{i+1,6}$	1	1	\dots	1	1	0	\dots	1	1	VfF_{i+1}
$t_{i+1,7}$	0	0	\dots	0	0	0	\dots	0	0	-
\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots	\ddots	\vdots	\vdots	\vdots

 Tabelle 5.2: Walking-1-basierter Test für dynamische Leitungsfehler bei einem N -bit Leitersystem

5.2 On-Chip Testmustererzeugung mit Schieberegistern

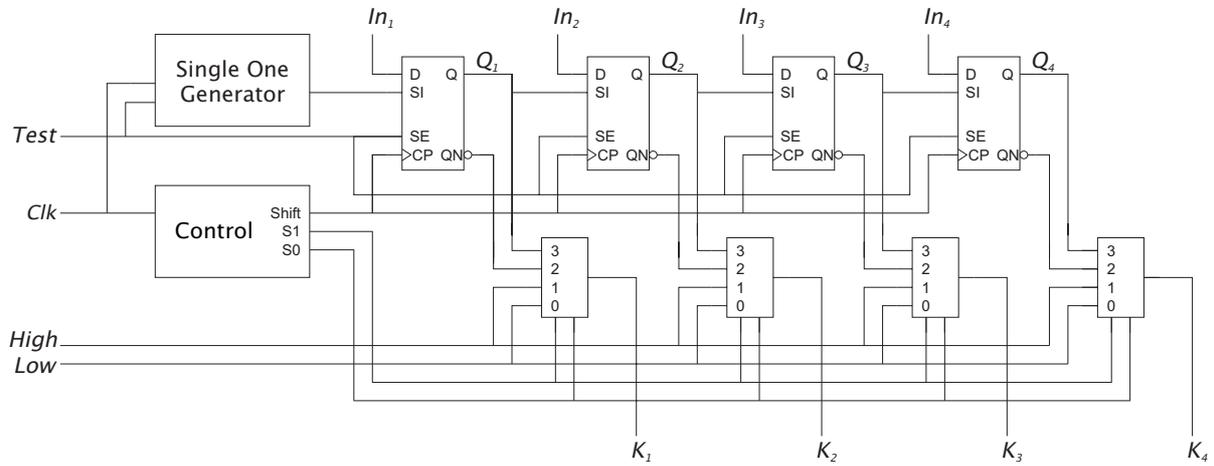


Abbildung 5.4: Testmustergenerator für einen vollständigen Test auf dynamische Leitungsfehler

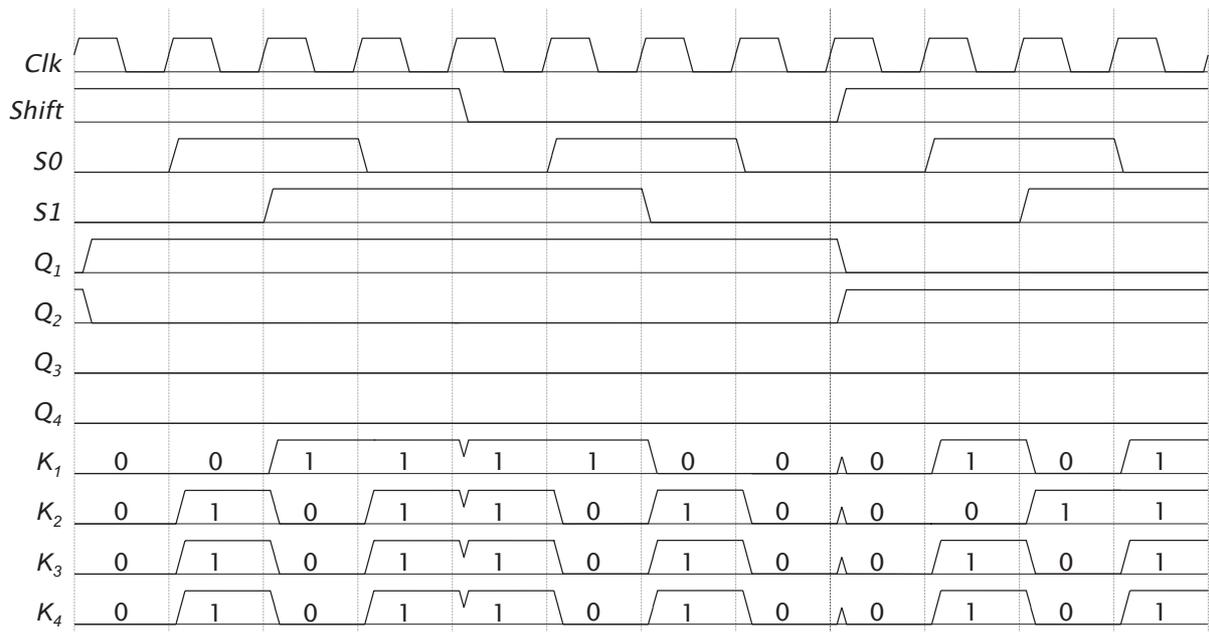


Abbildung 5.5: Steuersignale des Testmustergenerators für dynamische Leitungsfehler

von diesem abgeleitet werden. Die Realisierung mit einem endlichen Automaten ist jedoch leichter zu testen.

Geht man davon aus, dass die Ausgangsports eines Schaltungsmoduls mit Registern versehen sind und diese Ausgangsregister für den Testmuster-generator wiederverwendet werden, besteht der Schaltungsoverhead aus einem 4-zu-1-Multiplexer pro Busleitung, einem endlichen Automaten mit einem 3-bit Zustandsregister, sowie einem Taktteiler. Die Signalverzögerung im Systembetrieb wird gegenüber einem normalen Register durch die Verwendung Scan-fähiger Flipflops sowie durch den Multiplexer im Signalpfad vergrößert. Im Gegensatz zu den bekannten Testmuster-generatoren kann der Testmuster-generator nach Abbildung 5.4 durch Erweiterung eines IEEE 1500 Core-Test-Wrappers integriert werden, wie Abschnitt 5.5 zeigt.

5.3 On-Chip Testauswertung mit MISR

In dieser Arbeit wird davon ausgegangen, dass die Ein- und Ausgänge der Schaltungs-Cores in der Regel mit Registern versehen sind. Deshalb liegt es nahe, diese Register auch für die Testauswertung zu nutzen. Für den Selbsttest digitaler Logik finden rückgekoppelte Schieberegister mit mehreren Eingängen (Multiple Input Shift Register, MISR) häufig Anwendung. Ihre Vorteile sind eine effektive Kompaktierung von großen Testdatenmengen mit sehr geringen Einbußen bei der Testabdeckung durch prinzipbedingten Informationsverlust, ein geringer Schaltungsoverhead und eine einfache Implementierbarkeit.

In diesem Abschnitt soll die Eignung von MISRs für die Testauswertung beim Selbsttest von Bussystemen untersucht werden. Die Eigenschaften geringer Schaltungsoverhead und einfache Implementierbarkeit gelten uneingeschränkt auch für den Bustest. Für alle relevanten Busbreiten existieren Rückkopplungspolynome, die mit wenigen XOR-Gattern realisierbar sind [BMS87]. Ein Maß für den Informationsverlust durch die Kompaktierung ist die Aliasingwahrscheinlichkeit. Die Aliasingwahrscheinlichkeit ist die Wahrscheinlichkeit, dass die Folge von Testantworten bei Auftreten eines Leitungsfehlers zum gleichen Zustand des MISR wie die Folge von fehlerfreien Testantworten führt, so dass der Fehler nicht festgestellt werden kann. Für alle Fehler eines Fehlermodells wird die Aliasingwahrscheinlichkeit a wie folgt berechnet.

$$\text{Aliasingwahrscheinlichkeit } a = \frac{\text{Anzahl der nicht detektierten Fehler } N_{nd_F}}{\text{Anzahl der simulierten Fehler } N_F} \quad (5.1)$$

5.3.1 Aliasing durch die Testauswertung mit rückgekoppelten Schieberegistern

Beim Test digitaler Logik strebt die Aliasingwahrscheinlichkeit für die Kompaktierung der Testantworten durch MISRs gegen 2^{-N} , wenn N die Anzahl der Registerzellen des MISR ist. In [KPI90] werden dafür folgende Bedingungen hergeleitet:

1. Das MISR muss ein Rückkopplungsnetzwerk besitzen, das durch ein primitives Polynom beschrieben wird.
2. Es existieren wenigstens zwei verschiedene Testantworten der fehlerbehafteten Schaltung mit einer Wahrscheinlichkeit ungleich Null.
3. Das Verhalten des MISR kann durch einen zeitlich homogenen Markov-Prozess beschrieben werden. Dafür müssen die Testantworten in aufeinander folgenden Taktzyklen statistisch unabhängig sein.

Bedingung 1 kann durch entsprechende Konstruktion des MISR erfüllt werden.

Bedingung 2 ist in der Regel ebenfalls erfüllt. Eine Ausnahme liegt vor, wenn ein Fehler immer zur fehlerfreien Testantwort führt, so dass die Wahrscheinlichkeit für diese Antwort gleich Eins ist. Die Wahrscheinlichkeit für alle anderen möglichen Testantworten ist dann Null. Der Fehler ist nicht detektierbar. Beim Leitungstest lässt sich jedoch für jeden Fehler der beschriebenen Fehlermodelle ein Test finden, der diesen Fehler detektiert. Ebenso ist theoretisch denkbar, dass alle Leitungen gleichzeitig von einem Fehler nach dem Stuck-At-Fehlermodell betroffen sind. Dann hat die zugehörige Testantwort die Wahrscheinlichkeit Eins, während alle anderen Testantworten inklusive der fehlerfreien Testantwort die Wahrscheinlichkeit Null haben. Dieser Fall ist sehr unwahrscheinlich und könnte mit einem funktionalen Test einfach detektiert werden.

Bedingung 3 ist nicht exakt erfüllt, weil die Testmuster deterministisch erzeugt werden und damit die Testantworten nicht unabhängig voneinander sein können. Dennoch weisen die Testmuster und damit die Testantworten charakteristische Eigenschaften von Zufallsmustern auf, wie im folgenden gezeigt wird.

In [Gol82] werden drei notwendige Eigenschaften postuliert, die charakteristisch für eine zufällige Zahlenfolge sind. Eine Folge von Testmustern $(a_n) = (a_0, a_1, a_2, \dots)$ mit $a_n \in \{1, -1\}$ verhält sich wie eine zufällige Zahlenfolge, wenn sie diese drei Eigenschaften besitzt. Dabei repräsentiert der Wert 1 den logischen Wert WAHR und -1 den logischen Wert FALSCH. Die Ergebnisse der Überlegungen für $a_n \in \{1, -1\}$ sind ohne weiteres auf Folgen $(b_n) = (b_0, b_1, b_2, \dots)$ mit $b_n \in \{1, 0\}$ übertragbar, die

die Leitungstests bilden, weil die Werte 1 und 0 nur eine andere Repräsentation für die logischen Werte WAHR und FALSCH darstellen. Die Testmuster werden als periodische Folge mit der Periode $p = 8N$ betrachtet. Dabei ist N die Bussbreite. Diese Periode entsteht, wenn man die $8N$ Testmuster des dynamischen Leitungstests hintereinander hängt.

- Eigenschaft 1:
Die Anzahl von Nullen und Einsen unterscheidet sich maximal um eins:

$$\left| \sum_{n=1}^p a_n \right| \leq 1. \quad (5.2)$$

- Eigenschaft 2:
In jeder Periode beträgt die Anzahl der Sequenzen der Länge k bezogen auf die Gesamtzahl der Sequenzen 2^{-k} . Eine Sequenz der Länge k ist eine Folge von k aufeinander folgenden identischen Zeichen. Dabei ist außerdem die Anzahl der Sequenzen der Länge k mit dem Wert 1 gleich der Anzahl der Sequenzen der Länge k mit dem Wert -1.

- Eigenschaft 3:
Die Autokorrelationsfunktion $C(\tau)$ ist zweiwertig:

$$pC(\tau) = \sum_{n=1}^p a_n a_{n+\tau} = \begin{cases} p & \text{für } \tau = 0 \\ K & \text{für } 0 < \tau < p \end{cases} \quad (5.3)$$

Diese Eigenschaften sollen nun für die Testfolge untersucht werden. Die Anzahl der Elemente mit dem Wert 1 beträgt $4N$ für $p = 8N$ (vergleiche Tabelle 5.2). Dasselbe gilt für Elemente mit dem Wert -1. Damit ist Eigenschaft 1 erfüllt.

Durch Abzählen von Sequenzen der Länge $k = 1$ bis 4 in Tabelle 5.2 erhält man Tabelle 5.3. Sie zeigt die Verteilung von Sequenzen der Länge k in der Testfolge mit $p = 8N$ für eine Leitung eines N -Bit Bussystems.

k	1	2	3	4
Anzahl	$4N - 4$	$2N - 3$	2	1
p_k^*	≈ 0.66	≈ 0.33	≈ 0	≈ 0
p_k	0.5	0.25	0.125	0.0625

Tabelle 5.3: Verteilung von Sequenzen der Länge k beim Leitungselbsttest

Darin ist p_k^* das Verhältnis der Anzahl der Sequenzen der Länge k zur Gesamtzahl der Sequenzen $6N - 4$ für große N . Die Verteilung p_k^* entspricht nicht den nach Eigenschaft 2 zu erfüllenden Werten p_k .

Schließlich kann berechnet werden, dass die Autokorrelationsfunktion der Testfolge für eine Leitung nicht zweiwertig ist. Beispielsweise erhält man für $N = 8$ eine Menge von Autokorrelationswerten $\{pC(\tau)\}$ mit $0 \leq \tau < p - 1$, für die gilt

$$\begin{aligned} \{pC(\tau)\} = \{ & 64, -24, 0, 20, -56, 20, 0, -16, \\ & 48, -16, 0, 16, -48, 16, 0, -16, \\ & 48, -16, 0, 16, -48, 16, 0, -16, \dots \}. \end{aligned}$$

Die Werte der Autokorrelationsfunktion können mit der Konstruktion der Testfolge aus zwei verschiedenen Folgen von je 8 Elementen erklärt werden. Die Berechnung der Autokorrelationsfunktion kann als inneres Produkt einer um τ verschobenen Folge mit der nicht verschobenen Folge betrachtet werden. Bei der Testfolge gibt es 16 verschiedene Verschiebungen τ , so dass die Autokorrelationsfunktion maximal 16 unterschiedliche Werte besitzt. Eigenschaft 3 ist ebenfalls nicht erfüllt.

Insgesamt ist damit die Bedingung 3 nach [KPI90] durch die Testmuster für den Leitungsselbsttest nicht erfüllt, so dass mathematisch nicht gezeigt werden kann, dass die Aliasingwahrscheinlichkeit gegen 2^{-N} strebt. In Abschnitt 5.3.2 kann aber durch eine Fehlersimulation gezeigt werden, dass ein MISR mit primitiven Rückkopplungspolynomen beim Test von N -Bit Leitungssystemen zur Kompaktierung von Testantworten geeignet ist. Die Testauswertung selbst erfolgt durch Auslesen der berechneten Testsignatur über einen Scan-Pfad und anschließendem Vergleich mit der Gutsignatur auf einem Digital-Tester. Alternativ kann die Testauswertung auch durch einen On-Chip-Testcontroller wie in Abschnitt 4.2.1 beschrieben durchgeführt werden.

5.3.2 Fehlersimulation für Inter-Core Verbindungsnetze

Um die Eignung von MISRs für die Testauswertung von Leitungsselbsttests experimentell zu zeigen, sind in dieser Arbeit Fehlersimulationen für Inter-Core Verbindungsnetze durchgeführt worden. Dazu ist in MATLAB ein einfaches, taktgenaues Verhaltensmodell eines MISR implementiert worden. Dabei wird ein N -bit MISR durch die Zustandsübergangsmatrix \mathbf{A} der Dimension $N \times N$ und durch seinen Zustandsvektor \underline{s}_i der Dimension N beschrieben. Die Testantwort zum Zeitpunkt t_i wird durch den Vektor \underline{r}_i gegeben. Der Zustand des MISR im nächsten Zeitschritt t_{i+1} kann beschrieben werden durch das rekursive Gleichungssystem

$$\underline{s}_{i+1} = \mathbf{A} \cdot \underline{s}_i + \underline{r}_i \text{ in GF}(2). \quad (5.4)$$

Der Folgezustandsvektor \underline{s}_{i+1} des MISR ist also das Produkt der Zustandsübergangsmatrix \mathbf{A} des MISR und des aktuellen Zustandsvektors \underline{s}_i , addiert zum aktuellem

Testantwortvektor \underline{r}_i . Die Zustandsübergangsmatrix \mathbf{A} wird konstruiert aus einer oberen rechten Einheitsmatrix der Dimension $(N - 1) \times (N - 1)$ und den Koeffizienten des charakteristischen Polynoms¹ des MISR h_1 bis h_{N-1} in der N -ten Zeile von \mathbf{A} . Die Signatur \underline{s} ist der Zustandsvektor des MISR im $8N$ -ten Zeitschritt \underline{s}_{8N} .

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & 0 & \cdots & 0 & 0 \\ 0 & 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & 0 & \cdots & 0 & 1 \\ 1 & h_1 & h_2 & \cdots & h_{N-2} & h_{N-1} \end{pmatrix} \quad (5.5)$$

Die Fehlersimulation für den Test nach Tabelle 5.2 erfolgt durch Berechnung der Testsignatur \underline{s}_f für den Fehlerfall und Vergleich mit der Gutsignatur \underline{s}_{gut} . Eine Differenz von Gutsignatur \underline{s}_{gut} und Testsignatur \underline{s}_f zeigt, dass der Fehler detektiert wurde.

Der Test nach Tabelle 5.2 läßt sich als eine Folge von Testvektoren

$$(\underline{v}_n) = (\underline{v}_0, \underline{v}_1, \underline{v}_2, \dots) \quad (5.6)$$

der Länge $8N$ darstellen. Bei einem fehlerfreien Leitungssystem ist die Folge der Testantworten $(\underline{r}_n)_{gut}$ gleich der Folge von Testvektoren (\underline{v}_n) . Die Berechnung der Gutsignatur \underline{s}_{gut} wird durchgeführt, indem Gleichung 5.4 für alle Zeitschritte des Leitungstests, d.h. für alle Glieder von (\underline{v}_n) , ausgewertet wird.

$$\underline{s}_{i+1, gut} = \mathbf{A} \cdot \underline{s}_{i, gut} + \underline{v}_i \quad (5.7)$$

Bei einem fehlerhaften Leitungssystem ist die Folge der Testantworten $(\underline{r}_n)_f$ ungleich der Folge von Testvektoren (\underline{v}_n) . Man erhält beispielsweise die Folge der Testantworten $(\underline{r}_n)_{L_k s-a-1}$, falls die k -te Leitung L_k von einem Stuck-at-1 Fehler betroffen ist, indem man das k -te Element jeden Testvektors \underline{v}_i auf Eins setzt. Man erhält die Folge der Testantworten $(\underline{r}_n)_{L_m VstF}$, falls die m -te Leitung L_m von einem Verzögerungsfehler bei steigender Flanke (*VstF*) betroffen ist, indem man das m -te Element des Testvektors $\underline{v}_{8(m-1)+3}$ auf Null setzt. Die Berechnung der Testsignatur \underline{s}_f wird jeweils durchgeführt, indem Gleichung 5.4 für alle Zeitschritte des Leitungstests, d.h. für alle Glieder von $(\underline{r}_n)_f$, ausgewertet wird.

$$\underline{s}_{i+1, f} = \mathbf{A} \cdot \underline{s}_{i, f} + \underline{r}_i \quad (5.8)$$

Tabelle 5.4 zeigt die simulierten Fehlermodelle, die Anzahl der simulierten Fehler pro Fehlermodell N_F , sowie die Aliasingwahrscheinlichkeit a der Fehlermodelle für eine Breite des Leitungssystems von 8, 16, 32 und 64 Bit.

¹ Das charakteristische Polynom $P(x) = 1 + h_1 \cdot x + \dots + h_{N-2} \cdot x^{N-2} + h_{N-1} \cdot x^{N-1} + x^N$ repräsentiert die Rückkopplung des MISR

Fehlertyp	8 Bit		16 Bit		32 Bit		64 Bit	
	N_F	a	N_F	a	N_F	a	N_F	a
Einzelhaftfehler	16	0	32	0	64	0	128	0
2-Leiter Haftfehler	112	0	480	0	1984	0	8064	0
3-Leiter Haftfehler	448	0,0022	4480	0	39680	0	333312	0
2-Leiter Brücke	56	0,0179	240	0	992	0	4032	0
3-Leiter Brücke	112	0	1120	0	9920	0	83328	0
Einzel- $VstF$	8	0	16	0	32	0	64	0
2-Leiter $VstF$	28	0	120	0	496	0	2016	0
3-Leiter $VstF$	56	0	560	0	4960	0	41664	0
Einzel- VfF	8	0	16	0	32	0	64	0
2-Leiter VfF	28	0	120	0	496	0	2016	0
3-Leiter VfF	56	0	560	0	4960	0	41664	0
Einzel- nG	8	0	16	0	32	0	64	0
2-Leiter nG	28	0	120	0	496	0	2016	0
3-Leiter nG	56	0	560	0	4960	0	41664	0
Einzel- pG	8	0	16	0	32	0	64	0
2-Leiter pG	28	0	120	0	496	0	2016	0
3-Leiter pG	56	0	560	0	4960	0	41664	0

Tabelle 5.4: Aliasingwahrscheinlichkeit a für Busbreiten von 8, 16, 32, und 64 bit

Die simulierte Aliasingwahrscheinlichkeit ist für fast alle Fehlertypen und Busbreiten N Null. Nur für die Busbreite $N = 8$ gibt es für die Fehlertypen 2-Leiter Brücke und 3-Leiter Haftfehler eine Aliasingwahrscheinlichkeit ungleich Null. In beiden Fällen existiert genau ein Fehler, der auf die gleiche Testsignatur wie die Gutsignatur führt. Die Fehlersimulation zeigt also, dass eine Registerbreite des MISR von 8 bit für die Kompaktierung der Testantworten zu gering ist. Für größere Registerbreiten ist ein MISR ein effektives, leicht zu implementierendes Werkzeug für die Auswertung der Testdaten beim Leitungsselbsttest mit minimalem Schaltungsoverhead.

Neben den Untersuchungen zum Aliasing ist das beschriebene Simulationsverfahren auch geeignet, um die Diagnosefähigkeit des Leitungsselbsttests mit Kompaktierung

der Testantworten durch ein MISR zu untersuchen. Dazu werden die Testsignaturen s_f für jeden Fehler gespeichert. Nach der Simulation kann damit festgestellt werden, ob jeder Fehler zu einer eindeutigen Testsignatur führt. Bei den laut Tabelle 5.4 untersuchten Fehlertypen ist das der Fall. Daher kann durch Fehlersimulation eine eindeutige Zuordnung von Testsignatur zu Fehler gefunden werden, so dass eine vollständige Diagnose auf Basis von beschriebenem Test und Testsignatur möglich ist.

5.4 Ablaufplan für den Selbsttest der Verbindungsnetze

Die Erstellung eines Testablaufplans, des so genannten Test-Schedulings, ist ein Optimierungsproblem mit dem Ziel, einen möglichst günstigen Ablauf für den Test aller Komponenten eines Core-basierten Systems zu finden. Günstig im Sinne des Test-Scheduling bedeutet dabei in der Regel die Minimierung der Gesamtdauer aller Tests. Die Testdauer kann im Wesentlichen durch gleichzeitiges Ausführen von Tests für unterschiedliche Schaltungskomponenten minimiert werden. Dies wird durch Auswahl einer idealen Testreihenfolge und optimaler Zuordnung der zur Verfügung stehenden Testressourcen wie Testmusterspeicher bzw. Testmustergeneratoren, Testauswerter und Testschnittstellen erreicht.

Dabei sind Randbedingungen einzuhalten, die sich z.B. funktional ergeben, weil bestimmte Tests um die gleiche Testressource konkurrieren oder weil die Bandbreite der Testschnittstelle nicht ausreichend ist. Andere Randbedingungen ergeben sich schaltungstechnisch, z.B. zur Einhaltung einer maximalen Verlustleistung. Die Leitungssysteme müssen dabei als eine der zu testenden Schaltungskomponenten berücksichtigt werden. Bussystemen, auf die mehrere Cores schreibend zugreifen, sind wiederum auch als eine Ressource für den Testzugriff zu betrachten. Ein exklusives Schreiben eines Cores auf den Bus kann durch Aktivierung bzw. Deaktivierung von Leitungstreibern mit einem dritten, hochohmigen Zustand realisiert werden. Eine Teststeuerung muss daher konkurrierende Leitungstreiber kontrollieren [ST01].

Zur Lösung dieses Optimierungsproblems existieren zahlreiche Untersuchungen wie zum Beispiel [BSCB02], die Verfahren wie Graphenfärbung oder lineare Programmierung verwenden. In dieser Arbeit sollen daher Lösungsverfahren nicht weiter behandelt werden. Es werden jedoch Randbedingungen dargestellt, die sich speziell für den Test von Inter-Core Bussystemen ergeben.

Bei der Untersuchung der Leitungstest wurde bisher nicht auf die Topologie des zu testenden Bussystems eingegangen, sondern die einfachste mögliche Topolo-

gie zugrunde gelegt. Dies ist ein uni-direktionales System von N parallelen Leitungen ohne Abzweigungen, d.h. jede Leitung besitzt genau einen Leitungstreiber am Leitungsanfang und einen Empfänger am Leitungsende, und alle Leitungen haben die gleiche Länge und verlaufen parallel. In diesem Abschnitt wird dargestellt, welche Modifikationen des Selbsttests bei allgemeineren Topologien notwendig sind.

Die strukturellen Unterschiede von Bussystemen führen zu speziellen Anforderungen bei der Erstellung des Testablaufplans. Dabei ist zu beachten, dass die Steuerung des Selbsttests lokal durch die einzelnen Cores durchgeführt werden soll, damit das Selbsttestverfahren konform zum Standard IEEE 1500 ist. Die Steuerung des Ablaufs kann über den Test Acces Port (TAP) der Cores erfolgen.

5.4.1 Verbindungsgraph von Leitungssystemen

Bussysteme unterscheiden sich in der Anzahl der Cores, die sie verbinden, und in der Art, wie die Cores mit Schreib- und Leseschnittstellen, die hier als Ports bezeichnet werden, auf sie zugreifen. Man unterscheidet Leseports oder Empfänger, die nur lesend auf einen Bus zugreifen, Schreibports oder Sender, die nur schreibend auf einen Bus zugreifen, und bidirektionale Ports, die sowohl schreibend als auch lesend auf einen Bus zugreifen. Bidirektionale Ports können in einen Schreib- und einen Leseport aufgeteilt werden, die aber nicht zugleich aktiv sein können.

Leitungen verbinden als Teil eines Busses Lese- und Schreibports von mindestens zwei Cores. Betrachtet man eine Leitung als einen ungerichteten Graphen, dessen Knoten die Ports und dessen Zweige die von der Leitung gebildeten Verbindungen zwischen den Ports repräsentieren, dann erhält man einen vollständigen Graphen. Für den Test auf Fehler nach dem statischen Fehlermodell nach Abschnitt 3.1.1 ist es nicht notwendig, jeden Zweig des Graphen zu testen, um alle Fehler der Leitung erkennen zu können, sondern es genügt jene Verbindungen zu testen, die von einem Baumzweig des Graphen repräsentiert werden. Diese Verbindungen sollen als unabhängige Pfade einer Leitung bezeichnet werden. Abbildung 5.6 zeigt eine einfache Leitung mit zwei Sendern A und D und zwei Empfängern B und C und den zugehörigen Graphen. Dabei sind die Zweige AB, AC und BD, die einen Baum des Graphen bilden, hervorgehoben. Der Test dieser Zweige genügt für einen vollständigen Test des Bussystems nach dem statischen Fehlermodell.

Damit ein Baumzweig testbar ist, müssen die zugehörigen Knoten jeweils ein Sender und ein Empfänger sein. Da der Graph vollständig ist, lässt sich immer ein Baum finden, der diese Bedingung für alle Baumzweige erfüllt, wenn der Graph mindestens einen Sender und einen Empfänger besitzt.

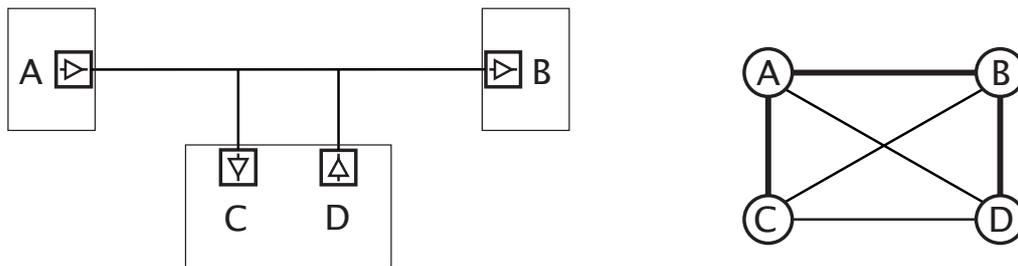


Abbildung 5.6: Leitung mit vier Ports und zugehöriger ungerichteter Graph

Zweckmäßigerweise wandelt man daher den Graphen in einen gerichteten Graphen, den Verbindungsgraphen. Dabei sind Kanten immer vom Sender zum Empfänger gerichtet. Kanten zwischen gleichartigen Knoten (zwischen Sender und Sender bzw. zwischen Empfänger und Empfänger) entfallen im Vergleich zum ungerichteten Graphen. Außerdem entfallen Kanten zwischen dem Schreibe- und dem Leseteil eines bidirektionalen Ports (Knoten C und Knoten D in Abbildung 5.6). Der gerichtete Verbindungsgraph ist daher nicht vollständig aber zusammenhängend.

Allgemein wird eine Leitung mit S Sendern und R Empfängern durch einen Graphen mit $K = S + R$ Knoten und E Kanten repräsentiert. Daher müssen $Z = S + R - 1$ Baumzweige getestet werden. Die Testlänge beträgt daher für einen allgemeinen N -Bit Bus maximal $8N \cdot Z$. Dabei kann die Testdauer durch Parallelisierung reduziert werden, weil Baumzweige, die einen Knoten gemeinsam haben, der einen Sender repräsentiert, gleichzeitig getestet werden können.

Für den Test von dynamischen Fehlern genügt es nicht, einen Baum des Graphen zu testen. Die Last- und Kopplungsbedingungen sind auf den verschiedenen Kanten, die zu demselben Knoten führen, unterschiedlich und verursachen abweichende Signalverzögerungen und abweichendes Übersprechen. Daher müssen bei diesen Fehlern alle Kanten des Graphs getestet werden. Dabei gilt für die Anzahl der Kanten $E = S \cdot R - B$. Darin ist B die Anzahl der bidirektionalen Ports. Die Testlänge für einen allgemeinen N -Bit Bus beträgt daher maximal $8N \cdot E$. Auch hier ist eine Reduzierung durch Parallelisierung von Tests mit gleichem Sender möglich. Das Leitungsnetzwerk repräsentiert man zweckmäßigerweise durch einen gerichteten Graphen.

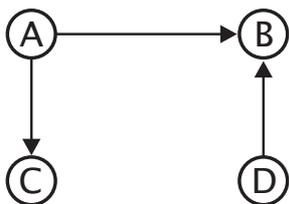


Abbildung 5.7: Verbindungsgraph der Leitung mit vier Ports

Die Erstellung eines Ablaufplans für den Bustest anhand des Verbindungsgraphen soll am Beispiel der in Abschnitt 4.1 beschriebenen Schaltung AxPe-Subsystem untersucht werden. Dieses Multiprozessorsystem besteht aus 4 Prozessor-Cores (PN), deren Ports von drei Bussen verbunden werden, einem Host-Bus, einem Kommunikationsbus und einem Video-Bus. Der Host-Bus und der Video-Bus verbinden einen externen Host-Prozessor mit den Prozessor-Cores und sind systemtechnisch in jeweils zwei unabhängige Teilbusse für die Dateneingabe und die Datenausgabe aufgeteilt. Der Kommunikationsbus ist bidirektional und verbindet die Prozessor-Cores und einen externen Host-Prozessor untereinander. Damit müssen für die Ablaufsteuerung des Selbsttests fünf funktional unabhängige Busse berücksichtigt werden:

- Host Interface Output Bus (HIOut, 32 bit),
- Host Interface Input Bus (HIIn, 18 bit),
- Video Interface Input Bus (VIIn, 19 bit),
- Video Interface Output Bus (VIOut, 19 bit),
- Communications Interface Input/Output Bus (CIIO, 18 bit).

Abbildung 5.8 zeigt den Graphen für das Bussystem eines AxPe-Subsystems. Die Knoten des Graphen sind die funktional unabhängigen Ports des Systems, die Kanten sind die zugehörigen Teilbusse. Weil es für jeden Prozessor-Core fünf funktional unabhängige Ports gibt, besteht der Verbindungsgraph aus fünf disjunkten Teilgraphen. Der Knoten Host stellt dabei jeweils die Schnittstelle zum externen Host-Prozessor dar² und die Knoten PN1, PN2, PN3 und PN4 die Prozessorknoten. Die Indizes geben an, zu welchem Bus der jeweilige Port gehört. Der bidirektionale Bus wird durch Paare von Pfeilen repräsentiert.

Der Verbindungsgraph verdeutlicht folgenden Randbedingungen für die Reihenfolge der Tests.

- Alle Kanten des Host Interface Input Bus (HIIn) können parallel getestet werden, weil die Ressource Testmustergenerator des Knotens Host gemeinsam genutzt werden kann. Dabei werden diese Busse über die externe Schnittstelle mit dem Leitungstest stimuliert, während die Testantworten an allen Prozessorknoten zeitgleich ausgewertet werden. Es ist also genau eine Testphase notwendig, in der insgesamt vier Tests ausgeführt werden.

² Die Testmustergenerierung und die Testauswertung werden für Knoten Host von einem externen Digital-Tester durchgeführt. Soll der Selbsttest autonom durchführbar sein, müssen die Padzellen durch Testlogik erweitert werden.

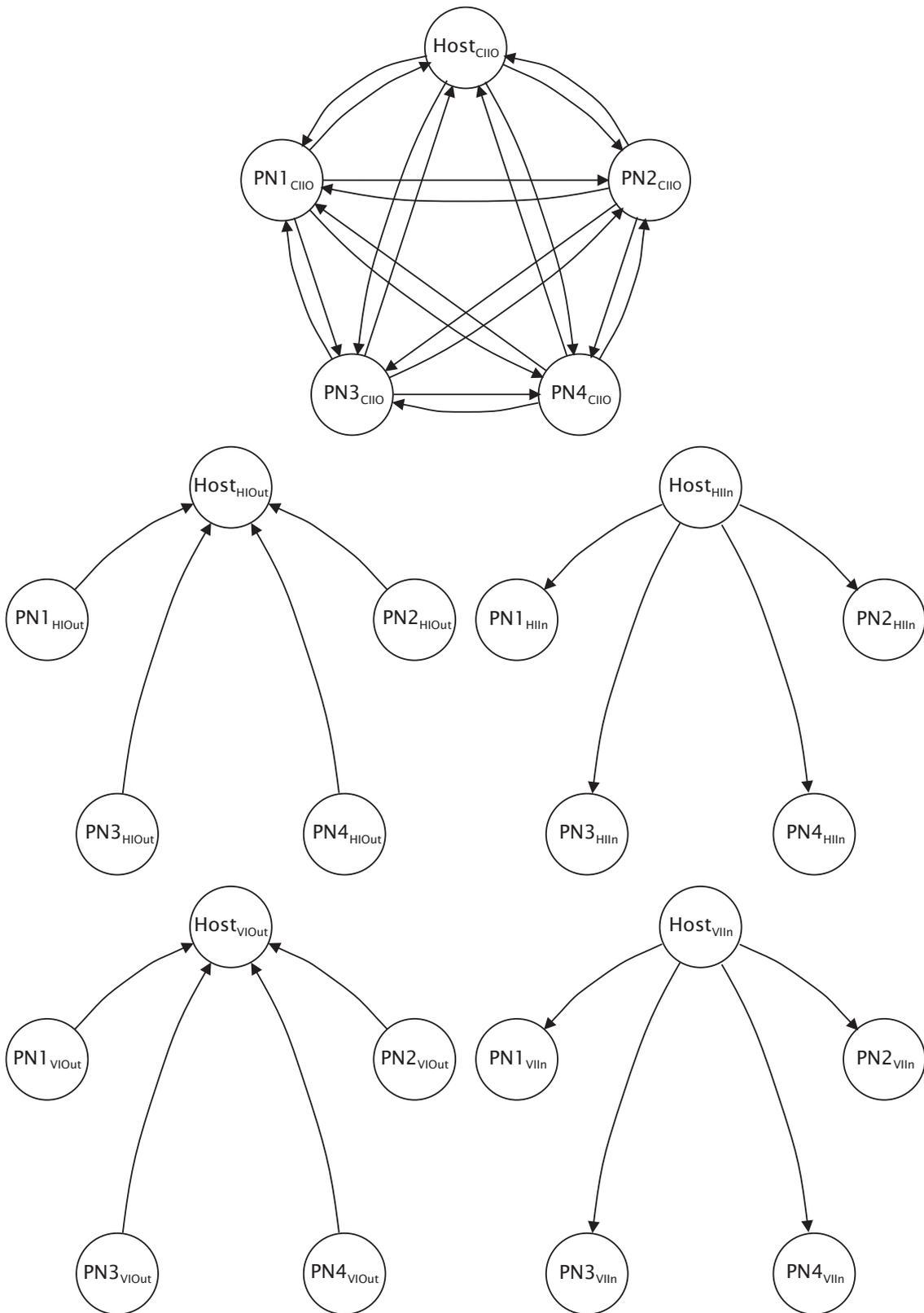


Abbildung 5.8: Verbindungsgraph für das Bussystem eines AxPe-Subsystems

- Alle Kanten des Video Interface Input Bus (VIIn) können parallel getestet werden, weil die Ressource Testmustergenerator des Knotens Host gemeinsam genutzt werden kann. Dabei werden diese Busse über die externe Schnittstelle mit dem Leitungstest stimuliert, während die Testantworten an allen Prozessorknoten zeitgleich ausgewertet werden. Es ist also genau eine Testphase notwendig, in der insgesamt vier Tests ausgeführt werden.
- Die Kanten des Video Interface Output Bus (VIOut) können nicht zeitgleich getestet werden, weil die Ressource Testauswertung am Knoten Host exklusiv für den Test genau einer Kante genutzt werden muss. Es sind also vier Testphasen notwendig, in denen jeweils ein Test ausgeführt wird.
- Die Kanten des Host Interface Output Bus (HIOut) können nicht zeitgleich getestet werden, weil die Ressource Testauswertung am Knoten Host exklusiv für den Test genau einer Kante genutzt werden muss. Es sind also vier Testphasen notwendig, in denen jeweils ein Test ausgeführt wird.
- Der bidirektionale Communications Interface Input/Output Bus hat 20 Kanten. Davon können jeweils 4 Kanten parallel getestet werden, die von einem Knoten ausgehen und zu den anderen vier Knoten führen. Diese Parallelität entspricht dem Test der Input-Busse. Es sind also 5 Testphasen notwendig, in denen jeweils 4 Tests ausgeführt werden.

Da keine weiteren Abhängigkeiten zwischen den Bussen bestehen, genügen für den Test der Bussysteme fünf Testphasen. Ein möglicher Testablauf ist in Tabelle 5.5 beschrieben. Dabei bedeutet $X \rightarrow Y$ dass ein Test der Kante von Knoten X nach Knoten Y durchgeführt wird. Eine Testphase besteht aus drei Teilen, einer Initialisierung der Testschaltung, der Testausführung und der Ergebnisauswertung. Die Testdauer der einzelnen Testphasen ist unterschiedlich und richtet sich nach der Anzahl der Leitungen des getesteten Busses.

Die Initialisierung erfolgt durch ein synchrones Reset der Testschaltung und benötigt daher 1 Taktzyklus. Die Länge des Tests für einen Bus der Breite N bit beträgt $8N$ Taktzyklen. Die Testauswertung erfolgt durch Auslesen der Testsignaturen über eine Schieberegisterkette, die hier als Bustest-Scanpfad bezeichnet werden soll. Für die Schaltung AxPe-Subsystem soll angenommen werden, dass alle Ports in einem Bustest-Scanpfad zusammengefasst sind. Das Auslesen der Testsignatur erfordert daher immer das Auslesen bzw. Shiften des vollständigen Bustest-Scanpfad und benötigt 424 Taktzyklen (4 PNs à 104 Ports). Ein erneutes Reset für die folgende Testphase ist nicht nötig.

Außerdem müssen für die Steuerung des Testablaufs zusätzliche Taktzyklen eingeplant werden. Die Steuerung hängt jedoch eng mit der Implementierung des Testzugriffsmechanismus für die Prozessor-Cores zusammen und soll hier nicht weiter behandelt werden. Für das AxPe-Subsystem ist eine Steuerung über den integrierten

5 Schieberegister-basierter Selbsttest von Inter-Core Verbindungsnetzen

	Host Interface Input Bus	Host Interface Output Bus	Video Interface Input Bus	Video Interface Output Bus	Communications Interface I/O Bus
Reset					
Phase 1	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>	<i>PN1 → Host</i>	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>	<i>PN1 → Host</i>	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>
Shift					
Phase 2		<i>PN2 → Host</i>		<i>PN2 → Host</i>	<i>PN1 → Host, PN1 → PN2, PN1 → PN3, PN1 → PN4</i>
Shift					
Phase 3		<i>PN3 → Host</i>		<i>PN3 → Host</i>	<i>PN2 → Host, PN2 → PN1, PN2 → PN3, PN2 → PN4</i>
Shift					
Phase 4		<i>PN4 → Host</i>		<i>PN4 → Host</i>	<i>PN3 → Host, PN3 → PN1, PN3 → PN2, PN3 → PN4</i>
Shift					
Phase 5					<i>PN4 → Host, PN4 → PN1, PN4 → PN2, PN4 → PN3</i>
Shift					

Tabelle 5.5: Ablaufplan für den Bustest eines AxPe-Subsystems

Selbsttestcontroller möglich, so dass für die Ablaufsteuerung keine weiteren Zyklen erforderlich sind. Die Länge des vollständigen Leitungsselbsttests beträgt daher insgesamt 2977 Taktzyklen.

5.4.2 Kopplungsgraph von Leitungssystemen

Der Floorplan der Schaltung AxPe-Subsystems in Abbildung 4.2 (links) deutet aber an, dass die Verdrahtungskanäle um die Prozessorknoten jeweils von wenigstens zwei Bussen gemeinsam genutzt werden. Daher besteht Kopplung zwischen den Bussen innerhalb eines Verdrahtungskanals, die insbesondere zu einer Beeinflussung des Leitungstestergebnisses führen kann, wenn diese Busse zeitgleich getestet werden. Der Fall größter Leitungsverzögerung oder stärksten Übersprechens auf die

5.4 Ablaufplan für den Selbsttest der Verbindungsnetze

	Verdrahtungskanäle			
	links	oben	rechts	unten
Busse	Videofc InBus HostIfc OutBus	HostIfc InBus Videofc InBus	HostIfc InBus Videofc OutBus ComIfc Bus	Videofc OutBus ComIfc Bus HostIfc OutBus

Tabelle 5.6: Zuordnung von Bussen zu Verdrahtungskanälen beim AxPe-Subsystem

Opferleitung des einen Busses tritt dann auf, wenn nicht nur die Aggressorleitungen dieses Busses einen Signalwechsel entgegen dem Opferbus aufweisen, sondern auch alle Leitungen des zweiten Busses.

Dieser zweite Bus muss daher während des gesamten Tests in einem Modus betrieben werden, in dem alle Signale synchron mit den Aggressorleitungen des getesteten Busses schalten. Auf allen Leitungen dieses Busses finden also identische Signalübergänge statt. Dieser Modus wird als *Even Mode* bezeichnet. Der in Abbildung 5.4 dargestellte Testmustergenerator kann durch eine einfache Modifikation für den Test gekoppelter Busse in den *Even Mode* gebracht werden. Dieser Testmustergenerator beruht auf der Implementierung einer Walking-1-Folge in der Schieberegisterkette. Durch die Deaktivierung des Single-One-Generators wird die Schieberegisterkette vollständig mit Null initialisiert, so dass auf allen Leitungen identische Signalwechsel stattfinden.

Für den Test auf Inter-Bus-Kopplung müssen die zu testenden Busse mit einem synchronen Takt betrieben werden. Diese Bedingung ist im Allgemeinen schwer zu erfüllen, da Busse oft verschiedenen Takt-Domänen zugeordnet sind. Die Implementierung eines Core-Test-Wrappers ist eine mögliche Lösung, da dieser einen dedizierten Takteingang WRCK für den Testbetrieb hat, so dass alle Cores und Busse synchron betrieben werden können.

Die aufgrund von Kopplung benachbarter Busse entstehenden Abhängigkeiten für die Erstellung des Testablaufplans können nicht mit dem Verbindungsgraph beschrieben werden. Dieser ist eine rein topologische Beschreibung der Schaltung und ihrer Verbindungsnetze bzw. Busse. Er ist daher unabhängig vom Floorplan oder dem Layout der Schaltung. Wenn der Floorplan feststeht, können die Nachbarschaften der Busse jedoch in einem Kopplungsgraphen beschrieben werden, der den Verbindungsgraphen ergänzt. Dabei sind die Busse die Knoten des Kopplungsgraphen. Zwischen zwei Knoten besteht eine Kante, wenn die Busse benachbart sind, also wenn Kopplung auftreten kann. Falls die Länge, auf der zwei Busse in einem Verdrahtungskanal parallel verlaufen, eine bestimmte Mindestlänge unterschreitet, kann die Kopplung vernachlässigt werden, so dass der Kopplungsgraph keine Kante zwischen den entsprechenden Knoten erhält.

Die Zuordnung der Busse zu den verschiedenen Verdrahtungskanälen für die Schaltung AxPe-Subsystems ist in Tabelle 5.6 zusammengefasst. Daraus ergibt sich der in

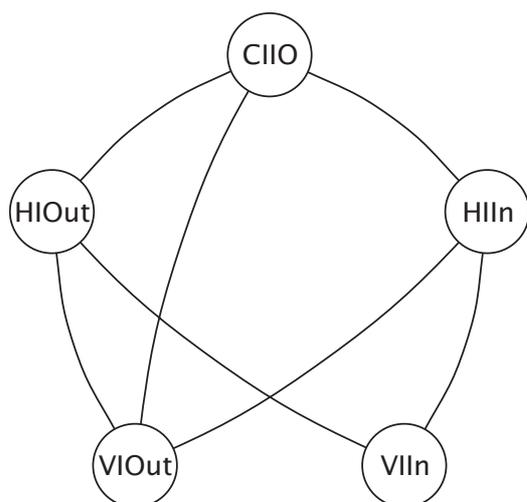


Abbildung 5.9: Kopplungsgraph für das Bussystem eines AxPe-Subsystems

Abbildung 5.9 dargestellte Kopplungsgraph für diese Schaltung. Für den Ablaufplan des Bustests erhält man beispielsweise die Nebenbedingung, dass während des Tests des Host Interface Output Bus (HIOut) die Busse Video Interface Input Bus (VIIn), Video Interface Output Bus (VIOut) und Communications Interface Input/Output Bus (CIIO) im oben beschriebenen *Even Mode* betrieben werden müssen. Daher ist beispielsweise Phase 2 des in Tabelle 5.5 gegebenen Ablaufplans nicht erlaubt. Ein modifizierter Ablaufplan unter Berücksichtigung der Kopplung zwischen verschiedenen Bussen (Inter-Bus-Kopplung) ist in Tabelle 5.7 gegeben. Der Eintrag *even* bedeutet, dass ein Bus in dieser Testphase im *Even Mode* betrieben werden muss. Der Eintrag *idle* bedeutet, dass der entsprechende Bus in dieser Testphase nicht verwendet wird. Er steht für andere Tests zur Verfügung. Die Reset- und Shiftphasen sind der Übersichtlichkeit wegen nicht dargestellt.

Aufgrund der Berücksichtigung von Inter-Bus-Kopplung werden insgesamt 13 Testphasen benötigt. Die Testlänge beträgt 7529 Taktzyklen. Durch die zusätzlichen Randbedingungen, die durch den Kopplungsgraphen beschrieben werden, können nur wenige Testphasen zur parallelen Durchführung von Tests genutzt werden. Eine Optimierung der Testlänge durch geeignete Wahl von Testreihenfolge und Parallelisierung ist damit praktisch nicht mehr möglich. Die Beschreibung der Inter-Bus-Kopplung durch den Kopplungsgraphen liefert jedoch die Möglichkeit, die Optimierung der Testlänge anders zu behandeln. Ziel der Optimierung ist demnach nicht mehr, eine Reihenfolge für den Testablauf mit minimaler Anzahl von Testphasen bzw. minimaler Testlänge zu finden, sondern eine Zuordnung von Bussen zu Verdrahtungskanälen zu finden, so dass eine Reihenfolge mit minimaler Anzahl von Testphasen möglich wird. Das Problem, die Testlänge zu minimieren, muss also bereits beim Floorplanning berücksichtigt werden.

5.4 Ablaufplan für den Selbsttest der Verbindungsnetze

	Host Interface Input Bus	Host Interface Output Bus	Video Interface Input Bus	Video Interface Output Bus	Communications Interface I/O Bus
Phase 1	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>	<i>PN1 → Host</i>	even	even	even
Phase 2	idle	<i>PN2 → Host</i>	even	even	even
Phase 3	idle	<i>PN3 → Host</i>	even	even	even
Phase 4	idle	<i>PN4 → Host</i>	even	even	even
Phase 5	even	even	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>	<i>PN1 → Host</i>	even
Phase 6	even	even	idle	<i>PN2 → Host</i>	even
Phase 7	even	even	idle	<i>PN2 → Host</i>	even
Phase 8	even	even	idle	<i>PN2 → Host</i>	even
Phase 9	even	even	idle	even	<i>Host → PN1, Host → PN2, Host → PN3, Host → PN4</i>
Phase 10	even	even	idle	even	<i>PN1 → Host, PN1 → PN2, PN1 → PN3, PN1 → PN4</i>
Phase 11	even	even	idle	even	<i>PN2 → Host, PN2 → PN1, PN2 → PN3, PN2 → PN4</i>
Phase 12	even	even	idle	even	<i>PN3 → Host, PN3 → PN1, PN3 → PN2, PN3 → PN4</i>
Phase 13	even	even	idle	even	<i>PN4 → Host, PN4 → PN1, PN4 → PN2, PN4 → PN3</i>

Tabelle 5.7: Ablaufplan für den Bustest eines AxPe-Subsystems mit Inter-Bus-Kopplung (ohne Reset- und Shift-Phasen)

5.5 IEEE 1500 Core-Test-Wrapper für den Bustest

Der in Abschnitt 2.3 beschriebene Standard IEEE 1500 bietet über den Testmodus für externe Schaltungen die prinzipielle Möglichkeit, Inter-Core-Bussysteme zu testen. Diese Möglichkeit ist jedoch dadurch eingeschränkt, dass für den Leitungstest nach dem dynamischen Fehlermodell mindestens zwei Tests direkt aufeinander folgend mit dem spezifizierten Systemtakt ausgeführt werden müssen. Die dafür erforderliche Testdatenbandbreite kann nur über die Wrapper-Testdaten-Ports (WSI und WPI) und mit hohem Aufwand zur Verfügung gestellt werden. Beträgt beispielsweise die spezifizierte Betriebsfrequenz f eines Inter-Core-Busses der Breite N bit und die des Wrapper-Ports WPI f_w , so beträgt die Mindestbreite des WPI $\left\lceil \frac{f}{f_w} \right\rceil \cdot N$ bit. In diesem Abschnitt wird gezeigt, wie die Architektur eines IEEE 1500 Core-Test-Wrappers durch minimale Erweiterungen vollständig bustestfähig wird. Dazu werden folgende Eigenschaften der bestehenden Architektur genutzt:

- Das Prinzip, die Funktionalität und nicht die Hardware zu standardisieren. Dadurch können Erweiterungen vorgenommen werden, ohne die Konformität zum Standard zu verlieren.
- Die Steuerung des Core-Test-Wrappers durch Befehle im Wrapper Interface Register. Eine Architektur-erweiterung entspricht dann einer Erweiterung des Wrapper-Befehlssatzes.
- Der Einschluss aller Ein- und Ausgänge eines Cores mit Wrapper Boundary Registern.

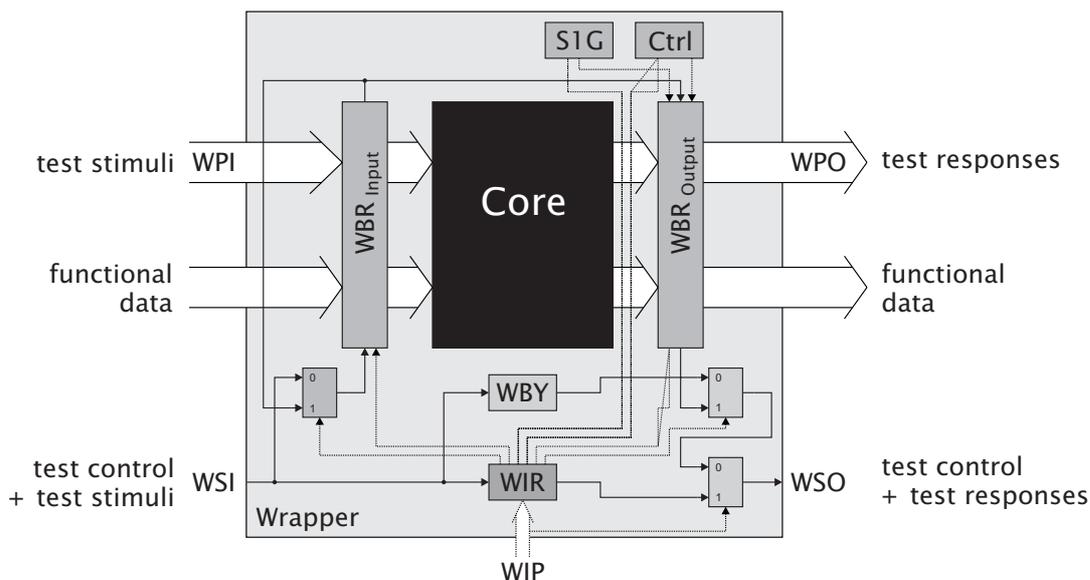


Abbildung 5.10: Architektur eines Core Test Wrappers für den Leitungselbsttest

Die Architektur des IEEE 1500 Core-Test-Wrappers bietet die Möglichkeit, einen Selbsttest externer Schaltungsteile zu implementieren. Die dazu notwendigen Erweiterungen zeigt Abbildung 5.10. Dieser Core-Test-Wrapper für den Leitungselbsttest ist wie der IEEE 1500 Core-Test-Wrapper nach Abbildung 2.9 aufgebaut. Die modifizierten oder hinzugefügten Blöcke sind dunkel schattiert. Die Wrapper Boundary Register auf der Eingangsseite werden modifiziert, so dass sie als MISR zur Testdatenauswertung nach Abschnitt 5.3 verwendet werden können. Die notwendige Rückkopplung über einen 2:1-Multiplexer ist in Abbildung 5.10 dargestellt. Auf der Ausgangsseite werden die Wrapper Boundary Register zu Testmustergeneratoren nach Abbildung 5.4 modifiziert. Dazu werden für jedes Register ein Controller (Ctrl) zur Erzeugung der Steuersignale und ein Single-One-Generator (S1G) zur Initialisierung des Tests benötigt.

Die Wrapper Boundary Register werden aus den entsprechenden Wrapper-Zellen nach Abbildung 5.11 aufgebaut. Auch hier sind die zusätzlichen Elemente gegenüber einer Standard-Wrapper-Zelle [MKL⁺02] dunkel schattiert. Eingangsseitig werden pro Wrapper-Zelle zwei Gatter benötigt, die die XOR-Verknüpfung der Eingangssignale

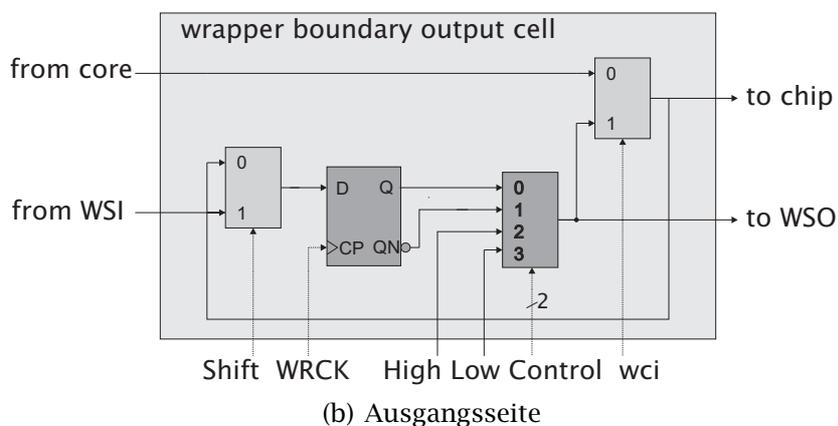
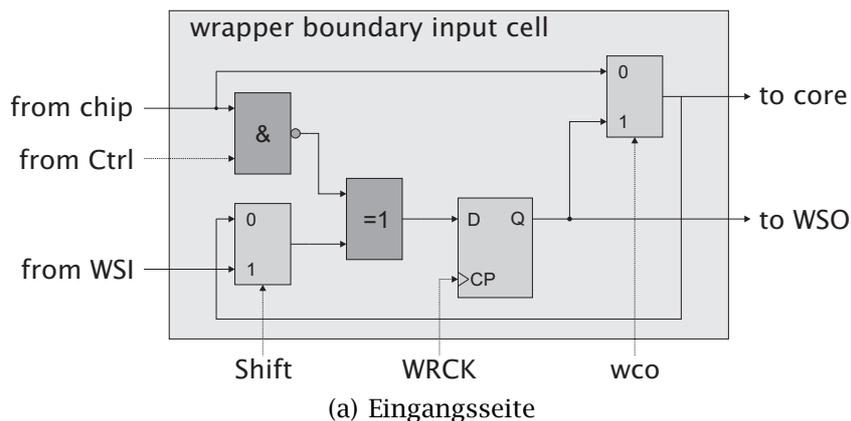


Abbildung 5.11: Wrapperzellen für den Leitungselbsttest

mit dem Registerzustand durchführen. Ausgangsseitig werden lediglich ein zusätzlicher 4-zu-1-Multiplexer und der negierte Ausgang der Registerzelle benötigt. Die Erweiterungen haben jeweils einen Umfang von wenigen Standardzellen, die Kosten für die Erweiterung des Selbsttestwrappers sind daher gering.

Für die Durchführung des Leitungselbsttests muss der Core-Test-Wrapper drei Operationen ausführen, die durch zwei zusätzliche Wrapperbefehle eingeleitet werden:

- **wBusBISTodd**: Die ausgangsseitigen Wrapper Boundary Register operieren als Testmustergenerator und erzeugen Testmuster nach Tabelle 5.2. Zugleich operieren die eingangsseitigen Wrapper Boundary Register als MISR zur Testdatenauswertung.
- **wBusBISTeven**: Die ausgangsseitigen Wrapper Boundary Register operieren als Testmustergenerator und erzeugen Even-Mode-Testmuster. Zugleich operieren die eingangsseitigen Wrapper Boundary Register als MISR zur Testdatenauswertung.

Falls aber ein Core zwei oder mehr Ausgangsports hat, die im Kopplungsgraphen nach Abschnitt 5.4.2 eine gemeinsame Kante besitzen, ist die oben beschriebene Befehlssatzerweiterung nicht ausreichend, weil die den Ausgangsports zugeordneten Wrapper Boundary Register unterschiedliche Operationen ausführen müssen. Dazu sind zwei weitere Befehle nötig. Es muss einen Wrapperbefehl `wSetBISTModeReg<No.>` geben, der jedes Wrapper Boundary Register explizit adressieren und diesem entweder den Selbsttestzustand `even` oder den Selbsttestzustand `odd` zuweisen kann. Mit Hilfe dieses Befehls können dann nacheinander alle Wrapper Boundary Register in den benötigten Testmodus gebracht werden. Schließlich wird noch der Befehl `wStartBusBIST` benötigt, der die Ausführung des Selbsttests aller Wrapper Boundary Register synchron startet.

Der Core-Test-Wrapper hat aber einen Nachteil gegenüber einer Core-internen BIST-Implementierung, bei der die letzten Systemregister im Signalpfad vor den Core-Ausgängen zu Testmustergeneratoren nach Abbildung 5.4 und die ersten Systemregister im Signalpfad nach den Core-Eingängen zu MISRs umgewandelt werden. Bei der Verwendung des Core-Test-Wrappers werden die Testsignale nicht von Systemregister erzeugt und empfangen, sondern von den Wrapper-Zellen. Auch die Leitungsgeometrie zu den Bustreibern ist daher abweichend. Dadurch können Abweichungen bei der zeitlichen Lage der Signalflanken im Testbetrieb gegenüber dem Systembetrieb auftreten. Die relative Lage der Signalflanken ist jedoch wichtig für die Größe der entstehenden Kopplungseffekte [CMS01]. Prinzipiell können diese Abweichungen zu fehlerhaften Testergebnissen führen. Diese Fehler können aber minimiert werden, indem bei der Schaltungsentwicklung versucht wird, einen

möglichst geringen *Clock Skew*³ von einer Systemregisterzelle zur entsprechenden Wrapper-Zelle zu erreichen. Ideal wäre außerdem eine Platzierung der Zellen in enger räumlicher Nachbarschaft.

³ *Clock Skew* ist die durch unterschiedliche Signallaufzeiten entstehende relative zeitliche Verschiebung der Flanken des Taktsignals bei einem Flipflop gegenüber einem anderen.

6 Messung von Leitungskopplung mit einer digitalen Testschaltung

Die im vorangehenden Kapitel beschriebene Methode für den Selbsttest von Inter-Core-Verbindungsnetzen sollte durch die Entwicklung, Fertigung und Validierung eines Testchips in der Praxis geprüft werden. Der konventionelle Weg wäre dazu die Implementierung der Methode in der dargestellten Form für ein Multiprozessor-SOC wie die AxPe-GIS gewesen. Dagegen haben zwei Gründe gesprochen. Einerseits sollte bei der AxPe-GIS jedes zusätzliche Entwurfsrisiko vermieden werden, andererseits wäre bei der geplanten geringen Stückzahl und der spezifizierten Taktrate für den Inter-Core-Bus die Wahrscheinlichkeit sehr gering gewesen, dass charakteristische Kopplungsfehler z.B. durch Parameterschwankungen tatsächlich vorliegen, so dass eine Aussage über die Wirksamkeit der Selbsttestmethode nicht möglich gewesen wäre.

Für die Untersuchungen im Rahmen dieser Arbeit war es daher erfolgversprechender, eine Testschaltung gezielt zu entwickeln, die die Auswirkung von Leitungskopplung auch ohne das Auftreten von Fertigungsfehlern einfach beobachtbar macht. Die Messungen sollten dabei mit dem Digital-Tester HP 82000 durchführbar sein.

Diese Testschaltung soll mit dem Namen ACSelfTest bezeichnet werden. Der Namensbestandteil AC weist darauf hin, dass der implementierte Test im Gegensatz zu statischen Tests (DC-Fall) dynamische Fehler durch Leitungskopplung berücksichtigt. Die Testschaltung ACSelfTest beruht auf der Generierung und Auswertung von Testmustern nach Kapitel 5 und realisiert ein neues On-Chip Messverfahren für Kopplung auf Leitungssystemen mit zwei Zielen:

- einfache Bestimmung von Kopplungseffekten durch Messung von Leitungsverzögerung,
- Verzicht auf komplexe Hochfrequenzmesstechnik durch Testauswertung mit einem Digital-Tester.

Damit sollen die Einschränkungen der in Abschnitt 1.2.3 beschriebenen On-Chip Messverfahren nach [AWW⁺00, CDBS01, EEJK00, KQM⁺02, SSK⁺03] für die Anwen-

derung in der Schaltungsentwicklung überwunden werden. Diese Einschränkungen sind insbesondere das Voraussetzen von spezieller Messgeräteausstattung und Kenntnisse der Hochfrequenzmesstechnik und die Beschränkung auf Leitungssysteme mit maximal fünf Leitungen.

Wenn bei einem System-on-Chip intermittierende oder datenabhängige Fehler trotz sorgfältiger Entwicklung auftreten, kann es für eine Diagnose¹ schwierig sein, den Fehler einem begrenzten Teil des Systems zuzuordnen, z.B. einem bestimmten Core oder einem zugehörigen Inter-Core-Bus. Die in Kapitel 3 beschriebenen Tests sind aber nicht geeignet, Fehler zu diagnostizieren, die beispielsweise auf ein Unterschreiten von im Entwurf berücksichtigten Leitungsverzögerungen zurückzuführen sind. Der Testmustergenerator nach Abbildung 5.4 kann aber durch eine einfache Modifikation auch diese Fehler stimulieren und so eine Diagnose ermöglichen, wie in den folgenden Abschnitten beschrieben wird.

6.1 Testschaltung ACSelfTest

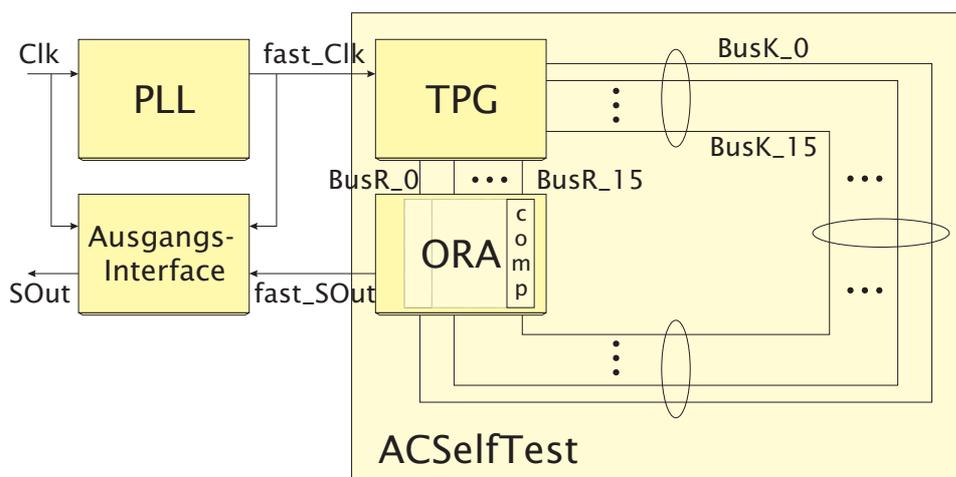


Abbildung 6.1: Blockschaltbild der Testschaltung ACSelfTest

Die Testschaltung ACSelfTest zur Validierung eines digitalen Tests für dynamische Fehler auf einem Bussystem in einer sequentiellen Schaltung ist auf einem Testchip implementiert worden. Der Testchip ist im Verlauf der Entwicklung der AxPe-GIS in einem 0,25 μm -Prozess mit 6 Metalllagen gefertigt worden, in dem auch die Fertigung der Schaltung AxPe-GIS geplant war. Abbildung 6.1 stellt die Schaltung schematisch dar. Sie besteht aus den drei Hauptkomponenten Testmustergenerator TPG, Testauswerter ORA und Testbus bestehend aus den Leitungen BusK_0 bis BusK_15.

¹ hier: Bestimmung und Lokalisierung der Fehlerursache

Bei Messungen mit dem automatischen Testsystem für digitale Schaltungen (Digital-Tester) HP 82000 besteht die Schwierigkeit, dass keine Hilfsmittel zur Verfügung stehen, die es erlauben, die Ein- und Ausgänge der Testschaltung direkt zu kontaktieren. Vielmehr musste die integrierte Testschaltung in einem Gehäuse montiert und über eine Adapterplatine mit dem Digital-Tester verbunden werden. Bei diesem Aufbau ist eine Stimulierung und Messung der Testschaltung auf dem Digital-Tester HP 82000 nur bis zu einer Taktrate von etwa 100 MHz möglich. Damit Kopplungseffekte durch einen digitalen Test beobachtet werden können, ist aber eine weit höhere interne Taktrate der Testschaltung erforderlich. Daher wird der interne Takt `fast_CLK` von einer PLL generiert, die Taktraten bis etwa 400 MHz liefern kann. Außerdem wird ein Ausgangsinterface benötigt, das die hohe Taktrate, mit der die Testergebnisse intern zur Verfügung gestellt werden, auf eine niedrige äußere Taktrate herabsetzt, die über die Gehäusekontakte transferiert werden kann.

Die Testschaltung ACSelfTest ist aus Standardzellen aufgebaut, deren Platzierung und Verdrahtung manuell erfolgt ist, um mittels eines möglichst regulären Aufbaus sekundäre Störeffekte durch die Logikkomponenten der Testschaltung zu minimieren. Das Layout von Testmustergenerator und Testauswerter ist in Abbildung 6.2 dargestellt. Die Zellen des Testmustergenerators TPG sind in der oberen Hälfte symmetrisch zum horizontal verlaufenden Bussystem angeordnet, die Zellen des Testauswerterers ORA befinden sich darunter. Außerdem sind die Zellen des Ausgangsinterfaces zu erkennen.

Der Testmustergenerator befindet sich in direkter räumlicher Nachbarschaft zum Testauswerter. Diese Anordnung weicht von der in Abschnitt 3.3 vorgestellten Anordnung ab und wird in integrierten Schaltungen praktisch nicht auftreten. Dort sind Testmustergenerator und Testauswertung räumlich getrennt. Sie minimiert aber die Einflüsse von Laufzeitunterschieden des Taktsignals zu Testmustergenerator und Testauswerter, um die Messgenauigkeit der Versuchsanordnung zu verbessern. Im Folgenden werden die Komponenten der Testschaltung ACSelfTest und ihre Funktion beschrieben.

6.1.1 Testmustergenerator TPG

Der Schaltungsteil Testmustergenerator TPG entspricht der in Abbildung 5.2 auf Seite 62 dargestellten Schaltung. Der von diesem Testmustergenerator erzeugte Test deckt damit sowohl klassische Fehler nach Abschnitt 3.1.1 als auch eine Teilmenge der Verzögerungsfehler ab, nämlich „Verzögerung bei steigender Flanke“ ($VstF$) und „Verzögerung bei fallender Flanke“ (VfF). Ein Test auf Glitches ist in der Testschaltung nicht implementiert, weil Glitches nicht zur Bestimmung von Leitungsverzögerungen genutzt werden können. Stattdessen ermöglicht der

Testmustergenerator die Erzeugung von Even-Mode-Testsignalen. Eine Beschreibung der Schaltungsfunktion des Testmustergenerators TPG und seiner Steuersignale findet man in Abschnitt 5.2.

Dieser Testmustergenerator erzeugt die Testmuster, die in den Arbeiten [GTA⁺99, NTO⁺98, NGT⁺98] mit *Even Mode* und mit *Odd Mode* bezeichnet werden. Im *Even Mode* erfolgt ein zeitgleicher Signalwechsel auf allen Leitungen in die gleiche Richtung, während im *Odd Mode* auf genau einer Opferleitung ein Signalwechsel in entgegengesetzter Richtung zu den übrigen Leitungen, den Aggressorleitungen, erfolgt. Damit werden auf dem Testbus die Signale erzeugt, die auf der Opferleitung die schnellste Signalausbreitung *Even Mode* bzw. die langsamste Signalausbreitung *Odd Mode* erwarten lassen.

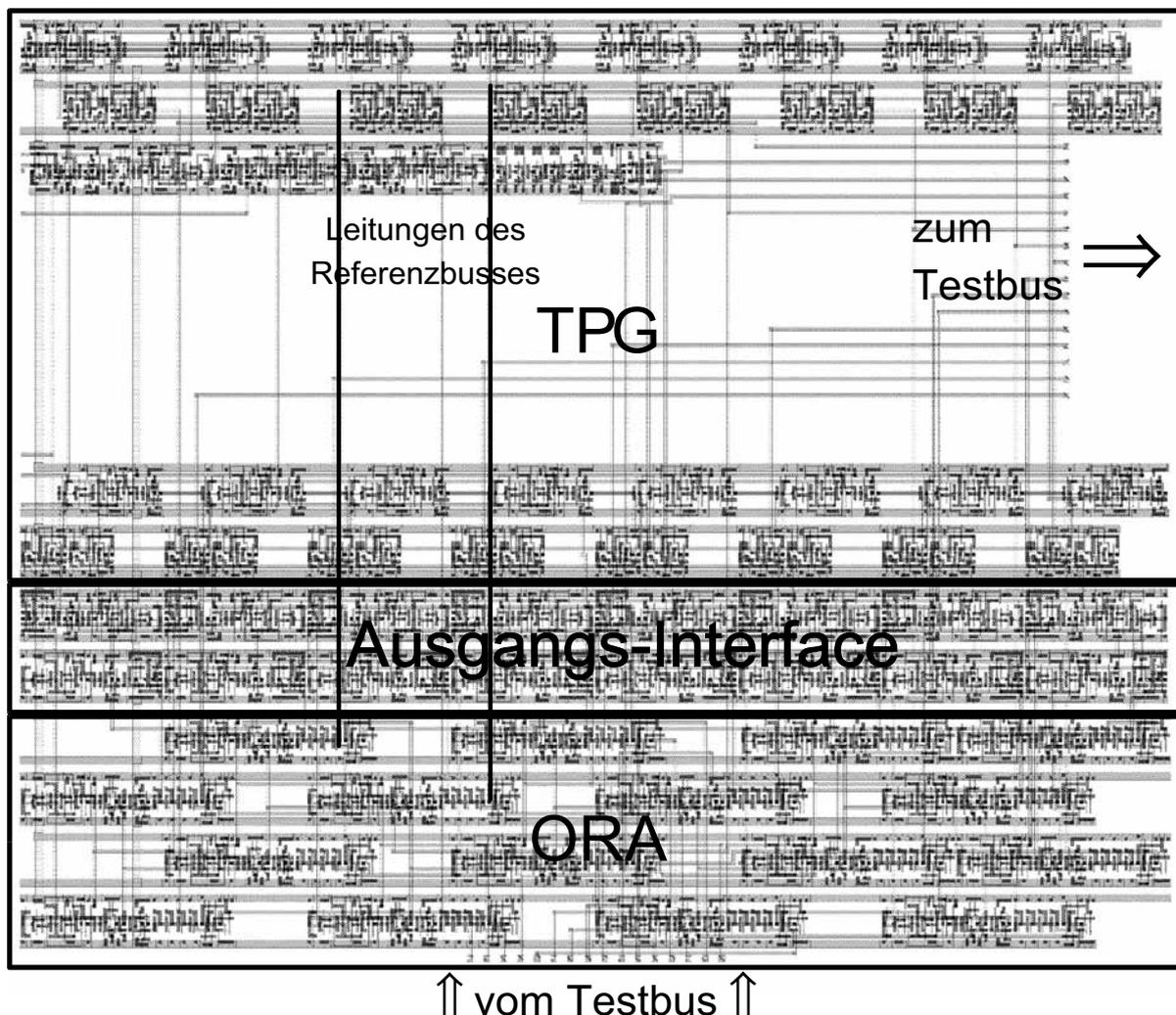


Abbildung 6.2: Layout von Testmustergenerator und Testauswerter.

6.1.2 Testauswerter ORA

Der Schaltungsteil Testauswerter ORA basiert auf dem Vergleich der Signale am Ende des Testbusses BusK mit entsprechenden Referenzsignalen. Als solche werden hier ebenfalls die Signale des Testmustergenerators TPG verwendet, die jedoch über den Referenzbus BusR übertragen werden, der sehr kurze Leitbahnlängen besitzt und mit erweiterten Leitbahnabständen entworfen ist. Störungen des Referenzbusses BusR durch Kopplungseffekte sind daher vernachlässigbar.

Die Signale des Testbus BusK werden mit den Referenzsignalen bitweise einem Komparator zugeführt. Der Wert des Komparatorausgangs wird mit jeder positiven Flanke des Taktsignals fast_CLK in ein Register übernommen. Ein Kopplungsfehler wird daher nur dann detektiert, wenn zum Zeitpunkt der Wertübernahme durch das Register eine Differenz zwischen den verzögerten Testbussignalen und den Referenzsignalen auftritt. Der Verlauf von Test- und Referenzsignalen ist in Abbildung 6.3 schematisch dargestellt.

Die erste Zeile zeigt den Verlauf eines Signals auf der Leitung i des Referenzbusses BusR (BusR< i >). Sie stellt einen Signalübergang am Leitungsende mit einer Verzögerung τ_{BusR} nach der steigenden Flanke des Taktsignals Clk (letzte Zeile) dar. Die zweite Zeile zeigt den entsprechenden Signalübergang auf der Leitung i des Testbusses BusK (BusK< i >) mit einer Verzögerung τ_{BusK} gegenüber dem Taktsignal Clk. Die dritte Zeile zeigt das Ausgangssignal des Komparators, der mit einer Verzögerung τ_{comp} eine Änderung der Eingangssignale BusK< i > und BusR< i > weitergibt. Dieser Ausgangswert wird mit der zweiten steigenden Flanke des Taktsignals Clk in das Register übernommen. Diese vereinfachte Darstellung vernachlässigt die Signalverzögerung des Testmustergenerators, die eine weitere Verschiebung der Signale auf BusK< i > und BusR< i > nach recht verursacht, sowie die Abtast- und Haltezeit des Registers im Testauswerter.

Ein Verzögerungsfehler wird detektiert, falls die zweite steigende Flanke des Taktsignals Clk zu einem Zeitpunkt auftritt, wenn das Ausgangssignal des Komparators den logischen Pegel Null aufweist. Dieser Fall ist in Abbildung 6.3 dargestellt. Das Ausgangssignal des Komparators wird mit der positiven Flanke des Taktsignals in ein Ergebnis-Flipflop übernommen und gespeichert. Wenn das Ausgangssignal des Komparators den logischen Pegel Eins aufweist, wird kein Kopplungsfehler detektiert. Die Summe aus τ_{BusK} und τ_{comp} ist dann geringer als eine Periode $T_{Periode}$ des Taktsignals Clk.

Abbildung 6.3 verdeutlicht, wie die Detektion eines Kopplungsfehlers forciert werden kann. Die Verzögerungswerte τ_{BusR} , τ_{BusK} und τ_{comp} sind unabhängig von der durch den Digital-Tester eingestellten Periode $T_{Periode}$ des Taktsignals Clk. Das bedeutet, dass durch Verkürzung der Taktperiode immer der Fall eingestellt werden kann, bei dem die positive Flanke des Taktsignals Clk auf den logischen Pegel Null am

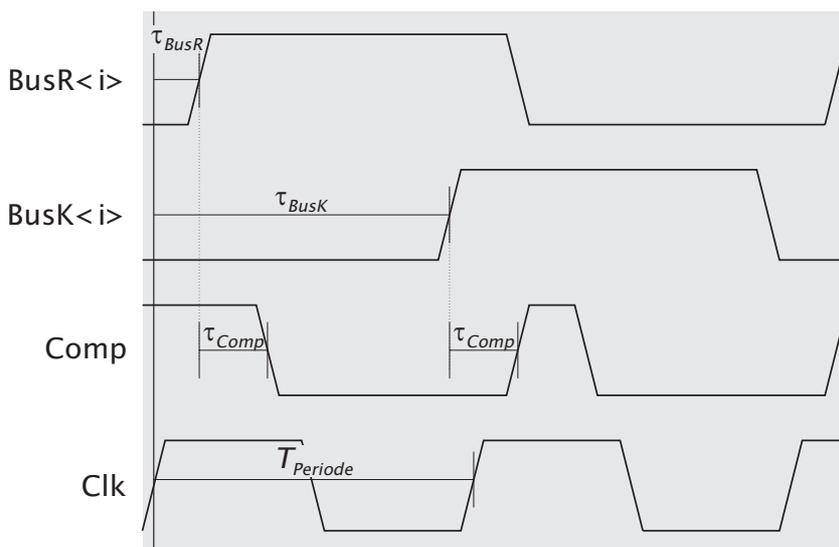


Abbildung 6.3: Signalverläufe bei der Testauswertung

Komparator trifft. Durch sukzessives Verringern der Taktperiode, bis die Schaltung einen Kopplungsfehler anzeigt, kann also ein Maß für τ_{BusK} bestimmt werden. Dieses Verfahren entspricht der Durchführung eines Maximalfrequenztests für eine integrierte Schaltung, wie sie beispielsweise zum *Speed Binning* oder zur Schaltungscharakterisierung durchgeführt werden. Solche Messungen lassen sich daher mit dem Digital-Tester HP 82000 automatisiert durchführen.

Die Komparatorsignale werden für jede Leitung einzeln in einem Register von Ergebnis-Flipflops gespeichert. Dabei dominiert eine gespeicherte logische Null, d.h. eine ankommende logische Eins kann eine logische Null nicht überschreiben. Dadurch bleibt der Zustand für die Detektion eines Fehlers bis zum Testende erhalten. Dieser Zustand muss durch ein Reset der Schaltung nach Ende des Tests zurückgesetzt werden. Der Inhalt des Ergebnisregisters wird am Ende des Tests über das Ausgangsinterface nach Abbildung 6.1 seriell ausgelesen. Dadurch ist eine eindeutige Zuordnung von Fehlersignal und betroffener Leitung möglich.

6.1.3 Testbus BusK

Der Testbus BusK zur Demonstration von Kopplungseffekten ist ein System von 16 parallelen Leitungen mit einer Gesamtlänge von $10590 \mu\text{m}$ für den äußeren Leiter und $10375 \mu\text{m}$ für den inneren Leiter. Abbildung 6.4 zeigt eine Übersicht des Testbus-Layouts. Der Testbus beginnt in der Abbildung unten rechts an der Komponente ACSelfTest und verläuft gegen den Uhrzeigersinn um eine Fläche, die mit

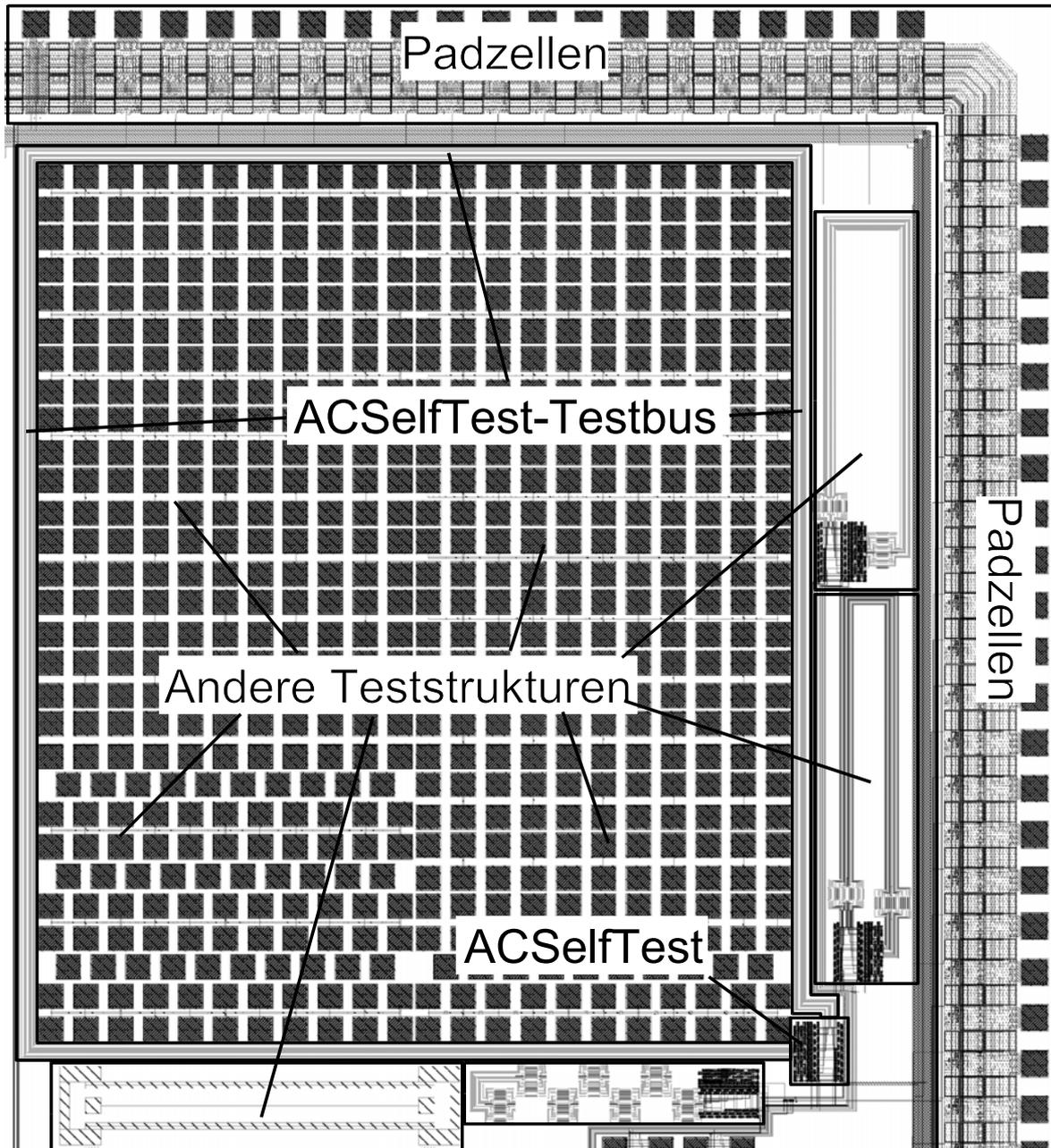


Abbildung 6.4: Layout des Testbus, Übersicht

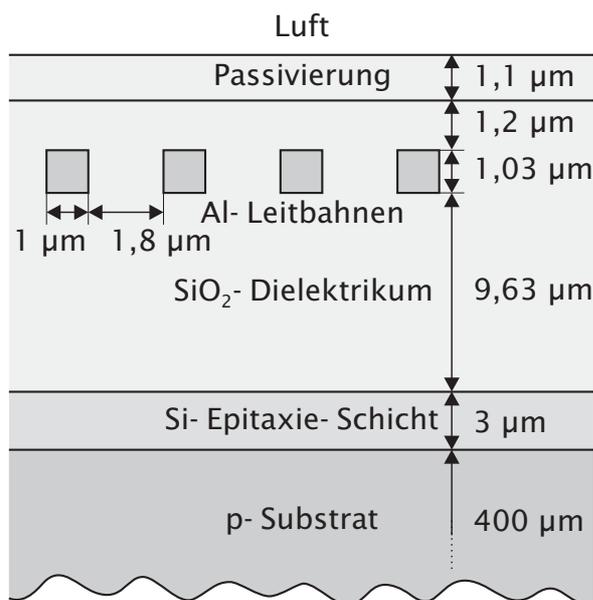


Abbildung 6.5: Schematischer Querschnitt des Testbus BusK

anderen Messstrukturen² belegt ist, bis zurück zur Komponente ACSelfTest. Außerhalb des vom Testbus BusK gebildeten Rings sind auf dem Testchip noch weitere Teststrukturen und die Padzellen des Testchips an der rechten und oberen Kante zu erkennen. Insgesamt zeigt dieser Layoutausschnitt etwa ein Viertel der Gesamtfläche des Testchips der Größe 7,18 mm × 8,22 mm. Die Länge der Testbusschleife ist vergleichbar mit der Länge der kritischen Inter-Core-Bussysteme der Schaltung AxPe-GIS.

Der Querschnitt des Testbus BusK ist in Abbildung 6.5 schematisch dargestellt. Der Chip basiert auf einem etwa 400 µm starkem Substrat aus gut leitendem p-dotiertem Silizium. Darauf befindet sich eine 3 µm dicke Schicht aus schwach leitendem p-dotiertem Silizium, die epitaktisch abgeschieden wird. In dieser Epitaxieschicht werden die aktiven Bauelemente hergestellt. Die Inter-Core-Busse der Schaltung AxPe-GIS werden in separaten Verdrahtungskanälen verdrahtet, die nicht oberhalb von Gebieten verlaufen, in denen logische Schaltungen platziert sind. Daher befinden sich auch in der Epitaxieschicht unterhalb des Testbus BusK keinerlei aktiven Bauelemente.

Die Aluminium-Leitbahnen des Testbus wurden in Metallage 6 der Fertigungsprozesses hergestellt und haben daher einen Querschnitt von 1,0 µm × 1,03 µm. Der Leitbahnabstand beträgt 1,8 µm, ihr Abstand zur Epitaxieschicht 9,63 µm. Die Leitbahnen sind von einem Dielektrikum aus Siliziumdioxid (SiO₂) umgeben. Die Schaltung wird schließlich durch eine Passivierungsschicht aus Siliziumnitrid und Polysilikatglas

² Es handelt sich um Strukturen zur Messung der Lebensdauer von Laserumschaltern. Im Wesentlichen sind die quadratischen Kontaktflächen zum Aufsetzen der Messspitzen zu erkennen.

mit einer Gesamtdicke von $1,1 \mu\text{m}$ gegen die Umgebungsluft versiegelt. Die gezeigten Abmessungen dieses Querschnitts entsprechen den für diese Fertigungstechnologie gegebenen Nominalabmessungen. Die tatsächlichen Maße des gefertigten Chips können davon um mehrere Prozent abweichen.

6.1.4 Fehlerdiagnose mit dem Leitungsselbsttest

Grundlage einer Diagnose von intermittierenden oder datenabhängigen Fehlern ist die Messung eines Maximalwerts für datenabhängige Laufzeitunterschiede auf einem Inter-Core-Bussystem. Der Minimalwert der Signallaufzeit wird durch einen *Even Mode* Test verursacht, also das Auftreten eines gleichen Signalübergangs auf allen Leitungen. Dagegen wird der Maximalwert durch einen *Odd Mode* Test verursacht, also wenn eine Leitung gegen alle anderen schaltet. Für beide Testmodi wird ein Maximalfrequenztest durchgeführt. Die Differenz dieser Messwerte ist auf Leitungskopplung zurückzuführen und damit ein Maß für die Signalverzögerung auf dem untersuchten Bus.

Sind die *Cores* eines System-on-Chip mit dem in Abschnitt 5.5 beschriebenen Core-Test-Wrapper ausgestattet, kann die Messung dieses Wertes für jeden einzelnen Inter-Core-Bus durchgeführt werden, wenn die Testmustergeneratoren um den *Even Mode* Test erweitert sind. Durch Vergleich der an dem ausfallenden Chip gemessenen Laufzeitunterschiede zwischen *Even Mode* und *Odd Mode* mit den beim Entwurf vorgesehenen Toleranzen, kann bestätigt werden, ob ein Inter-Core-Bus der Verursacher des untersuchten Fehlers ist.

6.2 Modellbasierte Verifikation der verwendeten Tests

Das Selbsttestverfahren der Schaltung ACSelfTest verwendet als Test die Signalübertragung im *Even Mode* und im *Odd Mode*. In Abschnitt 5.1 wird aber erläutert, dass diese Signalverläufe nicht zwangsläufig zu maximalen Verzögerungszeiten auf der Opferleitung führen. Durch exemplarische Simulationen unterschiedlicher Signalverläufe wird daher untersucht, ob mit dem *Odd Mode* eine maximale Signalverzögerung auf der Opferleitung erzeugt wird, oder ob die induktive Kopplung dazu führt, dass ein anderer Signalübergang eine größere Signalverzögerung verursacht.

Abschnitt	Lage	Länge	Segmente	Abstand zum Rückleiter
I	rechts	2880 μm	25	355,5 μm
II	oben	2430 μm	25	14,8 μm
III - IV	links und unten	je 2880 μm	je 25	5000 μm

Tabelle 6.1: Teilabschnitte des Modells für BusK

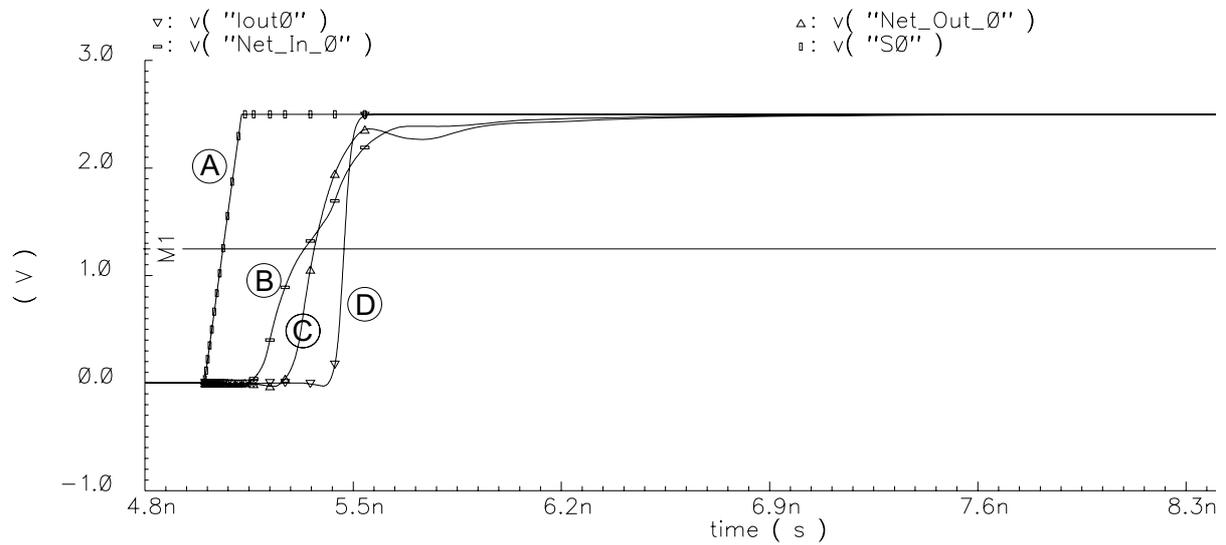
wegen der Verwendung von Spectre notwendig. Weil Leitungsmodelle, die auch Wellenausbreitung auf verlustbehafteten Leitungssystemen beschreiben können, wie z.B. LOSSYWIRE [GGRW95], nicht in Spectre implementiert sind, werden die gekoppelten Leitungen des Testbus BusK als Kette von RLCM-Segmenten modelliert.

Zur Bestimmung der Leitungsparameter wird BusK in vier Abschnitte aufgeteilt, die jeweils in Längsrichtung näherungsweise homogen sind. Die Abschnitte unterscheiden sich entsprechend Tabelle 6.1 in ihrer Länge, der Anzahl der RLCM-Segmente und in der Lage des Rückleiters. Die Querschnittsdimensionen können Abbildung 6.5 entnommen werden. Die Leitungsbeläge werden für die einzelnen Abschnitte mit dem Programm lipacific für eine Frequenz von 1 GHz berechnet. Mit der gewählten Extraktionsfrequenz wird ein Kompromiss für die Genauigkeit von hohen und niedrigen Frequenzanteilen der trapezförmigen Eingangssignale gewählt. Die in der Simulation gewählte Anstiegszeit der Signale ist mit 125 ps moderat. Wie man am Vergleich des Signals am Knoten A und am Knoten D sieht, entspricht sie aber näherungsweise den realen Anstiegszeiten der Testschaltung ACSelfTest. Hintereinander geschaltet ergeben die Modelle für Abschnitt I bis Abschnitt IV ein RLCM-Modell für BusK mit insgesamt 100 Segmenten.

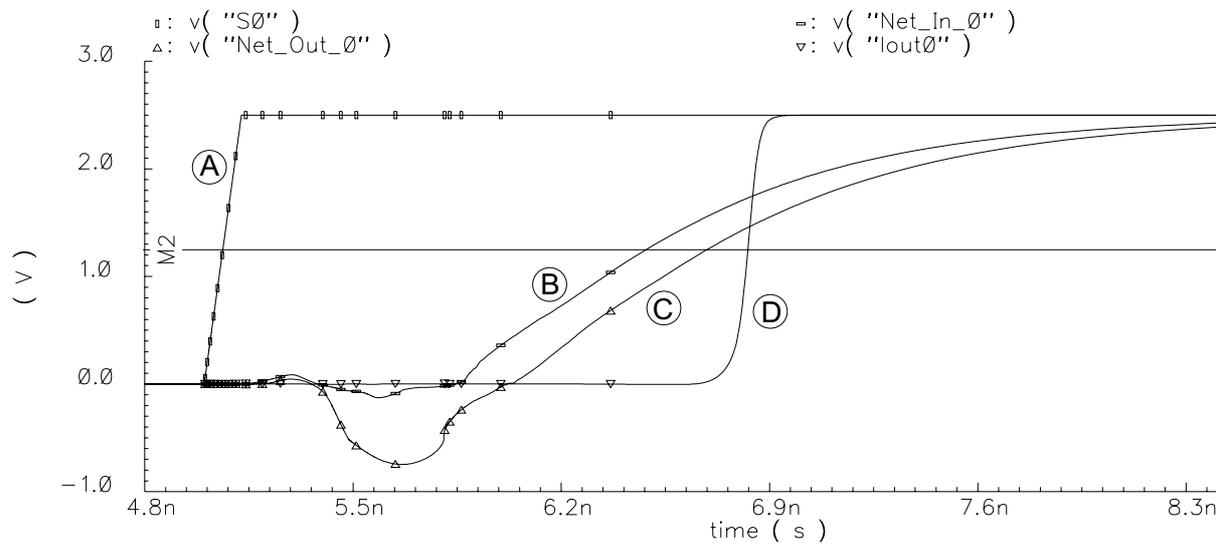
Die Kopplung von benachbarten Leitungen beeinflusst nicht nur die Signallaufzeit auf den Leitungen sondern auch die Gatterverzögerungen der angeschlossenen Treiber- und Lastgatter. In einem einfachen Modell stellt die Leitung eine Last für den Treiber dar, die über den (nichtlinearen) Innenwiderstand des Treibers umgeladen werden muss. Dabei ist die Last wegen der Kopplung abhängig von den Signalen auf benachbarten Leitungen — und damit ebenso die Zeit, die das Signal am Leitungsanfang zum Überschreiten der Schaltschwelle benötigt. Diese Zeitverzögerung wird dem Leitungstreiber zugeordnet.

Darüber hinaus hängt die Signalverzögerung eines Gatters von der Anstiegsrate des Signals am Gattereingang ab. Die am Leitungsende auftretende Anstiegsrate hängt wegen der Kopplung wieder von den Signalen auf benachbarten Leitungen ab. Dieser Zusammenhang betrifft hier den Komparator, der die Last für BusK darstellt.

6 Messung von Leitungskopplung mit einer digitalen Testschaltung



(a) Even Mode



(b) Odd Mode

Abbildung 6.7: Signalverlauf im *Even Mode* (oben) und im *Odd Mode* (unten)

Test Modus	Verzögerung in ns		
	Treiber	Leitung	Last
	τ_{AB}	τ_{BC}	τ_{CD}
<i>Even Mode</i>	0,274	0,033	0,100
<i>Odd Mode</i>	1,423	0,204	0,138
Differenz	$\Delta\tau_{AB}$	$\Delta\tau_{BC}$	$\Delta\tau_{CD}$
	1,149	0,171	0,038

Tabelle 6.2: Kopplungseinfluss auf Gatterverzögerungen

Abbildung 6.7 verdeutlicht exemplarisch für eine Randleitung des Busses die Abhängigkeit der Verzögerungszeiten von der Leitungskopplung. Sie zeigt den simulierten Zeitverlauf im *Even Mode* und im *Odd Mode* an den vier Knoten auf der Leitung 0 von *BusK*, die in Abbildung 6.6 mit A, B, C und D bezeichnet sind. Der Schaltzeitpunkt wird hier als das Überschreiten eines Signalpegels von 1,25 V festgelegt. Die so ermittelten Verzögerungen für das Treibergatter τ_{AB} (von Knoten A bis Knoten B), die Leitung τ_{BC} (von Knoten B bis Knoten C) und das Lastgatter τ_{CD} (von Knoten C bis Knoten D) sind in Tabelle 6.2 zusammengefasst. Die Differenzen $\Delta\tau_{AB}$, $\Delta\tau_{BC}$ und $\Delta\tau_{CD}$ der jeweiligen Verzögerungen im *Even Mode* und im *Odd Mode* sind ein Maß für den Einfluss der Leitungskopplung.

Dabei ist zunächst zu erkennen, dass die Gatterverzögerungen tatsächlich stark mit dem Testmodus variieren. Dies gilt insbesondere für das Treibergatter. Die Unterschiede können nur auf die Leitungskopplung zurückgeführt werden. Als wesentliches Ergebnis ist aber festzuhalten, dass die Zunahme der reinen Signallaufzeit auf der Leitung $\Delta\tau_{BC}$ wesentlich geringer ausfällt als die Zunahme der Gatterverzögerung des Treibers $\Delta\tau_{AB}$. Bei der Simulation mit einem linearen Treiber- und Lastmodell wie in den anderen Arbeiten kann nur die Signallaufzeit von Knoten B nach Knoten C ermittelt werden. Vergleichssimulationen, bei denen die Transistormodelle für Leitungstreiber und Last durch lineare Treiber- und Lastmodelle (Treiberinnenwiderstand 50 Ω , Lastkapazität 40 fF) ersetzt werden, ergeben Leitungsverzögerungen von 83,4 ps im *Even Mode* und 305,3 ps im *Odd Mode*, also ein $\Delta\tau_{BC}$ von 221,9 ps.

Die Verwendung linearer Treiber- und Lastmodelle kann, wie hier exemplarisch gezeigt, zu einer deutlichen Unterschätzung der durch Leitungskopplung verursachten gesamten Signalverzögerung führen. Eine korrekte Abschätzung erfordert daher die Modellierung der Treiber- und Lastgatter auf Transistorebene.

6.2.2 Simulationsergebnisse

Mit Hilfe des in Abschnitt 6.2.1 beschriebenen Leitungsmodells wird untersucht, ob die Testsignale *Even Mode* und *Odd Mode* für den Testbus zu minimaler bzw. maximaler Signalverzögerung und Größe von Glitches auf einer Opferleitung führen. Um einen vollständigen Nachweis zu führen, wäre die Simulation aller Kombinationen der möglichen Signalübergänge auf den Leitungen notwendig. Bei einem Bus der Breite N bit beträgt die Anzahl der möglichen Kombinationen 4^N . Bei einer Simulationsdauer zwischen 10 und 15 Minuten pro Kombination auf einer Workstation SUN 420R³ ist dieser Nachweis für interessierende Busbreiten N nicht zu führen.

Daher wurde exemplarisch für eine Randleitung des Busses (Leitung 0) und für eine mittlere Leitung des Busses (Leitung 7) untersucht, ob eine additive Überlagerung der Kopplungseinflüsse zu beobachten ist. Dazu wird für beide Leitungen zunächst die intrinsische Signalverzögerung bestimmt. Dabei findet auf der betrachteten Leitung ein Signalübergang statt, während alle übrigen Leitungen einen konstanten Pegel haben. Als Schaltschwelle wird wieder der Spannungspegel 1,25 V festgelegt.

Anschließend werden Simulationen durchgeführt, bei denen zunächst der direkte Nachbar (Leitung 0) bzw. beide direkten Nachbarn (Leitung 7) in entgegengesetzter Richtung schalten, und die Signalverzögerung bestimmt. Die Anzahl der direkten Nachbarn, die als Aggressorleitungen in entgegengesetzter Richtung schalten, wird immer weiter erhöht, bis schließlich alle Leitungen entgegen der Opferleitung schalten. Dies entspricht dem Testmodus *Odd Mode*.

Abbildung 6.8 zeigt die simulierten Verzögerungen für Leitung 0 und Abbildung 6.9 für Leitung 7. Die Signalverzögerung von Knoten A bis Knoten D entspricht der Gesamthöhe der Balken. In beiden Fällen nimmt die Verzögerung mit steigender Anzahl von Aggressorleitungen zunächst schnell zu. Bei Simulationen mit mehr als vier Aggressorleitungen nimmt die Signalverzögerung nur noch sehr schwach zu. Der Einfluss dieser zusätzliche Aggressorleitungen ist daher praktisch zu vernachlässigen. Die maximale Verzögerung wird mit dem Testmodus *Odd Mode* erreicht. Damit kann hier exemplarisch gezeigt werden, dass der Testmodus *Odd Mode* die Fehlerbedingungen für Verzögerungsfehler bereit stellt.

Schließlich wurde versucht, eine größere Signalverzögerung auf der Leitung 0 zu erreichen, indem zunächst Leitung 15, und anschließend auch Leitung 14 und Leitung 13 in dieselbe Richtung wie Leitung 0 schalten. In Übereinstimmung mit den übrigen Simulationsergebnissen nimmt die Gesamtverzögerung je stärker ab, desto mehr Leitungen in dieselbe Richtung wie die Opferleitung schalten. Tabelle 6.3 zeigt dieses Ergebnis.

³ 4 × UltraSparcIII, 450 MHz, 4 GB RAM

6.2 Modellbasierte Verifikation der verwendeten Tests

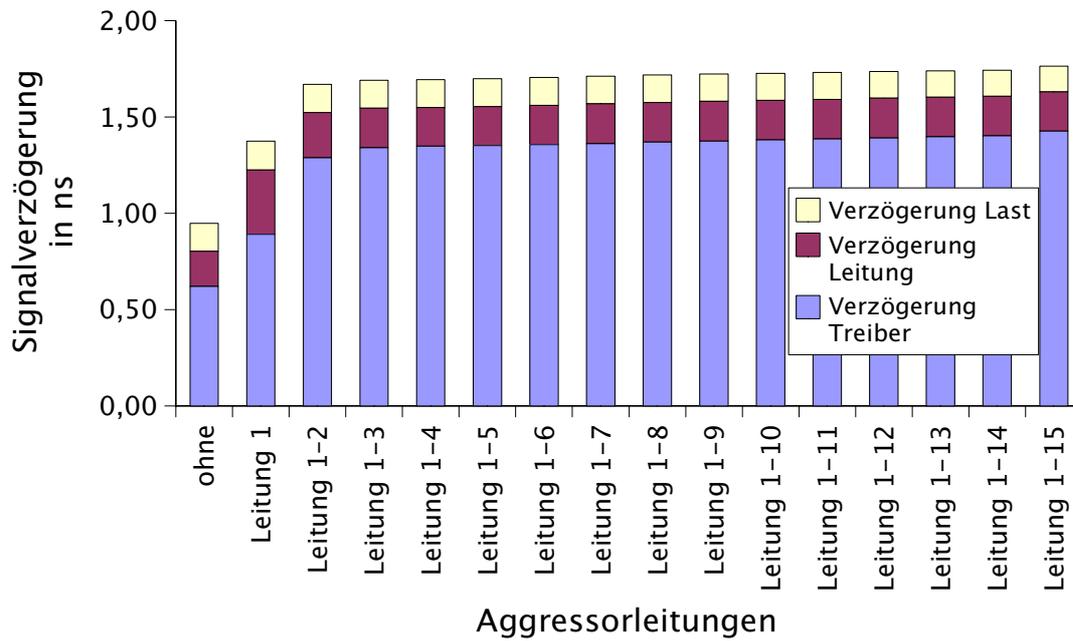


Abbildung 6.8: Zusätzliche Signalverzögerung auf der Leitung 0 mit zunehmender Anzahl von Aggressorleitungen

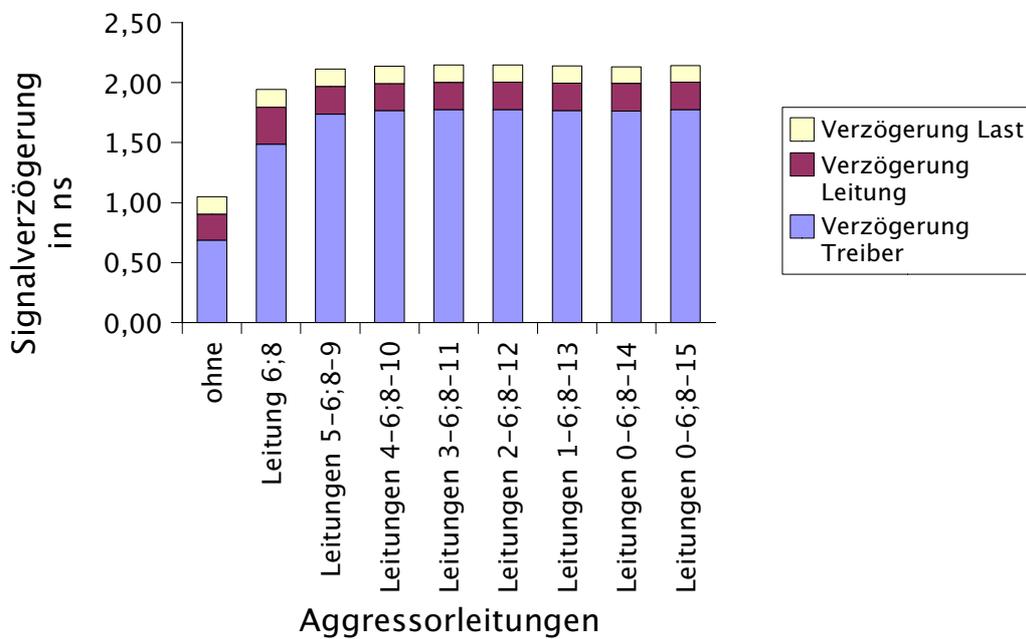


Abbildung 6.9: Zusätzliche Signalverzögerung auf der Leitung 7 mit zunehmender Anzahl von Aggressorleitungen

Schalten in Richtung der Opferleitung	Verzögerung
keine (<i>Odd Mode</i>)	1,765 ns
Leitung 15	1,744 ns
Leitungen 14-15	1,731 ns
Leitungen 13-15	1,723 ns

Tabelle 6.3: Signalverzögerung auf Leitung 0 bei Abweichung vom *Odd Mode*

Die Einwände von [NA01, AN02] hätten dagegen erwarten lassen, dass die Signalverzögerung wieder abnimmt, wenn die entfernteren Aggressorleitungen in entgegengesetzter Richtung zur Opferleitung schalten. Dieser Effekt müsste bei Kopplung zwischen entfernteren Leitungen am ehesten zu bemerken sein, weil die kapazitive Kopplung mit der Entfernung stark abnimmt, so dass die induktive Kopplung dominieren kann. Allerdings wird bei diesen Untersuchungen die reine Leitungsverzögerung betrachtet. Beschränkt man die exemplarischen Simulationsergebnisse nach Abbildung 6.8 auf den Anteil der reinen Leitungsverzögerung (dunkelgraue Balken), findet man die Erwartung von [NA01, AN02] bestätigt. Die größte reine Leitungsverzögerung tritt auf, wenn nur die direkte Nachbarleitung der Opferleitung schaltet. Die Gatterverzögerung ist jedoch, wie gezeigt, deutlich größer als die Leitungsverzögerung und nimmt mit zunehmender Anzahl der Aggressorleitungen stetig zu. Dadurch wird der Effekt der abnehmenden Leitungsverzögerung vollständig verdeckt.

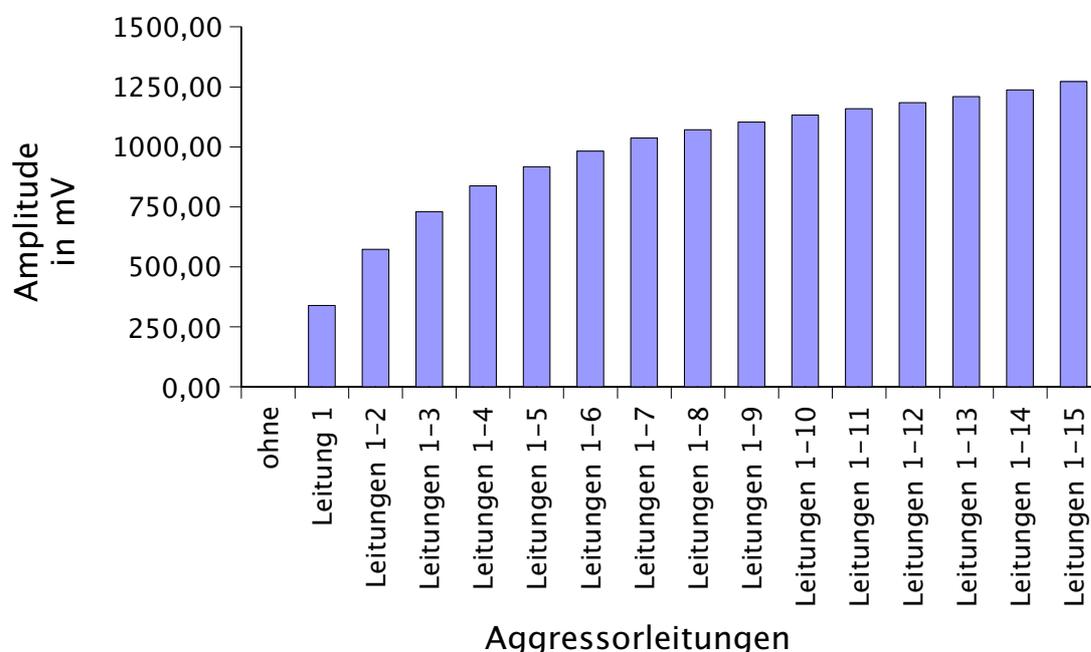


Abbildung 6.10: Glitch auf der Leitung 0 mit zunehmender Anzahl von Aggressorleitungen

Für das Verhalten eines Leitungssystems und für den Test auf dynamische Fehler ist aber die Gesamtverzögerung des Leitungssystems von Interesse. Für den Fall der Schaltung ACSelfTest mit dem Leitungssystem BusK kann hier exemplarisch gezeigt werden, dass mit dem Testsignal *Odd Mode* maximale Verzögerungen generiert werden.

Ein entsprechendes Ergebnis konnte auch bei der Bestimmung der Höhe von Glitches erzielt werden. Für die Leitung 0 mit konstanten Signalpegel 0 V wurde die Anzahl der direkt benachbarten Aggressorleitungen sukzessive erhöht. Abbildung 6.10 zeigt die Simulationsergebnisse. Die Glitches am Ende der Opferleitung nehmen mit steigender Anzahl der Aggressorleitungen immer weiter zu, bis für den *Odd Mode* das Maximum erreicht ist. Für den Fall der Schaltung ACSelfTest mit dem Leitungssystem BusK kann also exemplarisch ebenso gezeigt werden, dass mit Testsignal *Odd Mode* maximale Glitches generiert werden.

Insgesamt kann daher im Modell die Gültigkeit des implementierten Testverfahrens auch bei Berücksichtigung von induktiver Kopplung bestätigt werden.

6.3 Ergebnisse des Hardwaretest

Das wesentliche Hilfsmittel zur Auswertung und Charakterisierung der Testschaltung ACSelfTest sind Maximalfrequenztests, die mit dem Digital-Tester Hewlett-Packard 82000 D400 an einer Stichprobe von 46 der produzierten und in ein PQFP208-Gehäuse montierten Testchips durchgeführt werden.

Abbildung 6.11 zeigt den genutzten Digital-Tester an seinem Standort im Laboratorium für Informationstechnologie, Universität Hannover. Der zu untersuchende Chip wird von einem Testadapter aufgenommen, der auf dem *Loadboard* des Digital-Testers montiert ist. Beide sind in einer Detailansicht des Digital-Testers in Abbildung 6.12 gezeigt. Im Testsockel befindet sich ein Chip mit dem Aufdruck TC9 SYSTEM, auf dem die Testschaltung ACSelfTest integriert ist. *Loadboard* und Testadapter verbinden die Gehäusekontakte des Testchips mit der Elektronik des Digital-Testers und erlauben ein einfaches Austauschen der Testchips. Während der Durchführung einer Messung ist der Testadapter geschlossen.

Vor Beginn der Maximalfrequenztests wird für jedes Muster ein *Continuity*-Test durchgeführt. Mit diesem Test kann nachgewiesen werden, dass alle Ein- und Ausgänge eines Chips ausreichenden Kontakt zum Digital-Tester haben. Bei zwei der 46 geprüften Chips ist ein vollständiger Kontakt nicht erreicht worden, so dass diese für die weiteren Untersuchungen nicht zur Verfügung stehen.



Abbildung 6.11: Digital-Tester HP 82000

Der Maximalfrequenztest wird durch schrittweises Erhöhen der Betriebsfrequenz des Digital-Testers durchgeführt. Bei jeder eingestellten Betriebsfrequenz stimuliert der Digital-Tester die Durchführung des Leitungselbsttests und wertet die von der Testschaltung gelieferten Testantworten aus. Die höchste eingestellte Betriebsfrequenz, bei der die Testauswertung ein fehlerfreies Verhalten ergibt, ist die Maximalfrequenz des Chips.

Die Einstellung der Betriebsfrequenz kann beim Digital-Tester Hewlett-Packard 82000 D400 in Schritten von 50 ps mit einer absoluten Genauigkeit von 500 ps erfolgen. Die minimale einstellbare Taktzykluszeit beträgt 4,5 ns. Das entspricht einer maximalen Betriebstaktrate von 111 MHz⁴. Bei der Testschaltung ACSelfTest vervielfacht eine

⁴ Eine Taktperiode der Testschaltung beträgt zweimal die Taktzykluszeit des Digital-Testers, weil der Digital-Tester in der verwendeten Konfiguration pro Taktzyklus nur einen Signalwechsel durchführen kann. Daher werden zwei Testerzyklen benötigt um eine Periode des Taktsignals zu erzeugen.

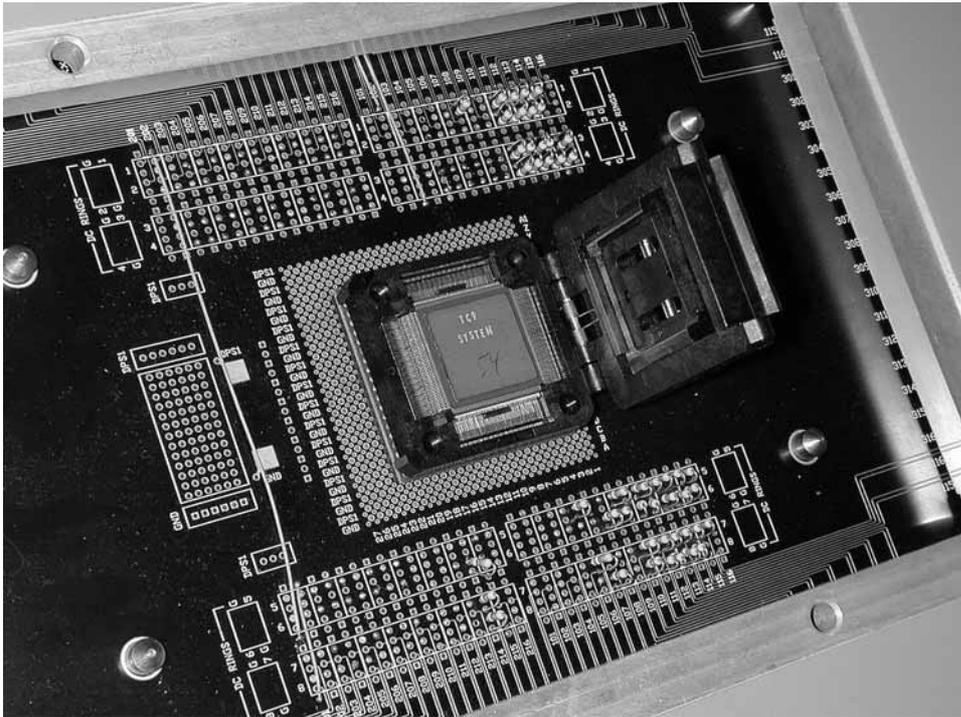


Abbildung 6.12: Detailansicht des Digital-Tester: *Loadboard* mit offenem Testadapter

PLL wie in Abschnitt 6.1 beschrieben den externen Betriebstakt um den Faktor 4, um eine höhere interne Taktrate zu erzeugen. Bei der maximalen Betriebstaktrate führt dies zu einer internen Taktrate von 444 MHz, die bereits außerhalb der Spezifikation der PLL liegt. Eine weitere Erhöhung der internen Taktrate wäre daher auch seitens der Testschaltung ACSelfTest nicht sinnvoll.

6.3.1 Messergebnisse

Der Maximalfrequenztest ist für jeden Testchip aus der Stichprobe jeweils in den Testmodi *Odd Mode* und *Even Mode* durchgeführt worden. Die ermittelten minimalen Taktperioden sind für den *Odd Mode* in Abbildung 6.13 und für den *Even Mode* in Abbildung 6.14 dargestellt. Sie liegen zwischen 17,2 ns und 19,2 ns bzw. zwischen 9,2 ns und 10,6 ns. Die interne Maximalfrequenz der Testschaltung beträgt dabei 220,0 MHz bzw. 408,6 MHz. Tabelle 6.4 fasst diese wesentlichen Messergebnisse zusammen.

Die minimale Taktperiode ist im *Odd Mode* näherungsweise doppelt so hoch wie im *Even Mode*. Diese Unterschiede sind allein durch die elektromagnetische Kopplung der Leitungen des Testbus BusR zu erklären. Im ungünstigsten Fall einer Opferleitung, auf der ein Signalwechsel in entgegengesetzter Richtung zu allen anderen Leitungen

6 Messung von Leitungskopplung mit einer digitalen Testschaltung

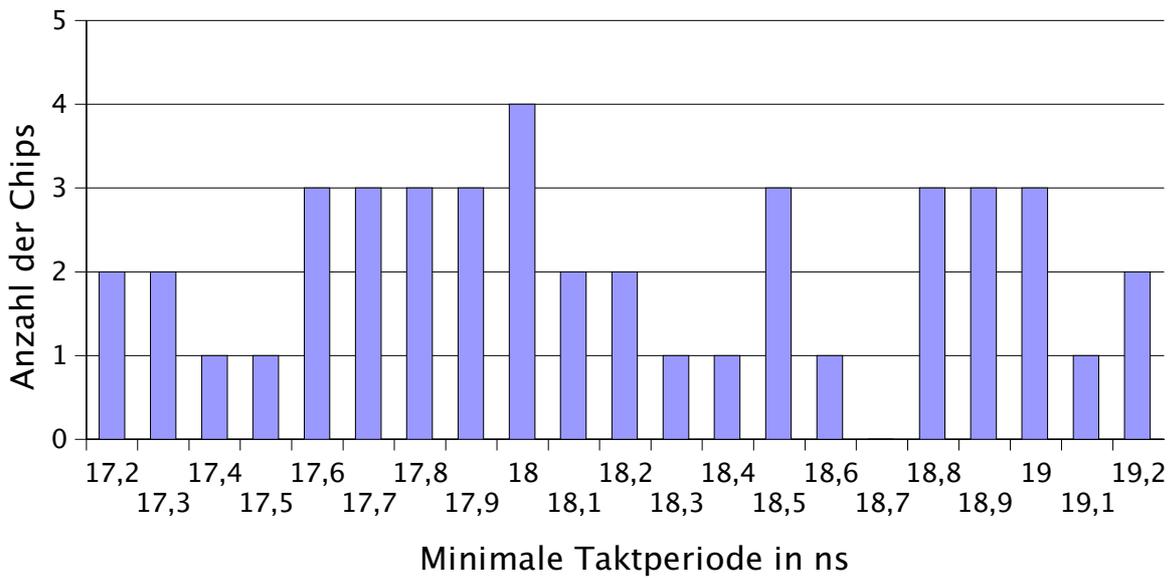


Abbildung 6.13: Verteilung der minimalen Taktperiode im Testmodus *Odd Mode*

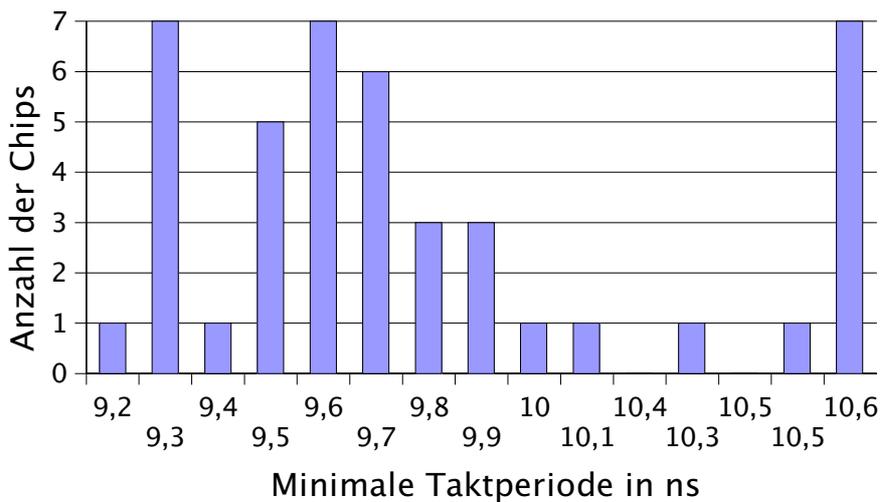


Abbildung 6.14: Verteilung der minimalen Taktperiode im Testmodus *Even Mode*

	Testmodus	
	<i>Odd Mode</i>	<i>Even Mode</i>
Maximum	19,2 ns	10,6 ns
Minimum	17,2 ns	9,2 ns
Mittel	18,18 ns	9,79 ns

Tabelle 6.4: Minimale Taktperiode der Testschaltung ACSelfTest

stattfindet, kann die Testschaltung nur mit halber Betriebsfrequenz gegenüber dem günstigsten Fall betrieben werden, bei der alle Signalwechsel in die gleiche Richtung stattfinden. Eine genauere quantitative Auswertung der Signalverzögerungen wird in Abschnitt 6.3.2 präsentiert.

Erwartungsgemäß liefert die Testschaltung im *Odd Mode* mit sinkender Taktperiode zunächst Fehler auf inneren Leitungen 7 und 8 des Testbusses. Die Summe der Einflüsse von Nachbarleitungen, die die Signalverzögerung auf einer inneren Leitung vergrößern, ist in der Mitte des Testbusses am größten. Dagegen liefert die Testschaltung im *Even Mode* zunächst Fehler auf einer der beiden äußeren Leitungen 0 oder 15. Auch dies deckt sich mit den Erwartungen, weil der *Even Mode* zu einer Verringerung der Leitungsverzögerung gegenüber der intrinsischen Leitungsverzögerung führt. Maximale Signalverzögerung, d.h. minimale Taktperiode, tritt daher im *Even Mode* am Rand des Busses auf, weil dort die Summe der Einflüsse von Nachbarleitungen am geringsten ist.

Für beide Testmodi schwanken die Werte der minimalen Taktperiode deutlich. Die Verteilungen laut Abbildung 6.13 und Abbildung 6.14 verlaufen nach keiner erkennbaren statistischen Verteilungsfunktion. Die Schwankungen sind im Wesentlichen auf Prozessschwankungen zurückzuführen.

Diese führen bei der Chipfertigung zu Abweichungen bei den elektrischen Parametern der gefertigten Bauelemente, die sich auf die maximale erreichbare Taktrate auswirken. Bei Kleinserien für die Prototypenfertigung wie bei der Testschaltung ACSelfTest fallen die elektrischen Parameter im Vergleich zu anderen in demselben Prozess gefertigten Schaltungen generell eher ungünstig aus, weil eine optimale Prozessjustierung auf die Schaltung nicht durchgeführt werden kann. Ebenso sind die Schwankungen innerhalb der Serie tendenziell größer.

Eine Quantifizierung dieser Schwankungen ist für die Testschaltung ACSelfTest nicht möglich, weil aus der Fertigung keine Daten über die Auswertung von Prozesskontrollstrukturen vorliegen. Vergleichsmessungen an anderen auf dem Testchip integrierten Schaltungen liefern eine auf den arithmetischen Mittelwert bezogene Standardabweichung im Bereich von 5 % bis 16%. Auch dabei ist die Größe der Stichproben mit minimal 25 und maximal 39 sehr gering. Die Standardabweichung für die Messwerte des Leitungsselbsttests liegt bei 3,2% im *Odd Mode* und bei 4,5% im *Even Mode*.

Außerdem trägt die Messgenauigkeit zur Streuung der Messergebnisse bei. Die absolute Messgenauigkeit des Digital-Testers beträgt 500 ps und damit mehr als 10% der minimalen einstellbaren Taktrate von 4,5 ns. Die Messungen an einem Chip in den Testmodi *Even Mode* und *Odd Mode* sind direkt nacheinander ausgeführt worden. Daher kann davon ausgegangen werden, dass der absolute Messfehler bei Messungen an einem Chip sehr ähnlich ist. Dagegen können die Abweichungen zwischen unterschiedlichen Chips signifikant sein, weil die Messbedingungen nicht

immer konstant gehalten werden konnten. Die minimale Schrittweite der Taktperiode von 50 ps, also die Ablesegenauigkeit, muss außerdem als Fehlerquelle berücksichtigt werden.

6.3.2 Berechnung des Kopplungseinflusses aus Messdaten

Wie in Abschnitt 6.1.2 dargestellt liefert die Testauswertung für eine Leitung des Testbusses dann eine logische Null, also eine Fehlermeldung, wenn die Gesamtheit der Signalverzögerungen von Schaltungselementen und des der Leitungskopplung unterworfenen Testbusses größer ist als die Taktperiode.

$$T_{\text{Periode}} \leq \tau_{\text{gesamt}} \quad (6.1)$$

In diesem Fall kann eine korrekte Übertragung des Signals vom Testdatengenerator zur Testauswertung nicht mehr stattfinden.

Diesen Signalpfad stellt Abbildung 6.15 schematisch dar. Dem Testmustergenerator TPG wird über den Eingang fast_Clk (Knoten A) das schnelle interne Taktsignal zugeführt, während der Dateneingang In konstant ist. Das Flip-Flop des Testmustergenerators hat während einer Testphase daher einen konstanten Zustand. Ein Multiplexer verbindet abwechselnd den Ausgang Q oder den invertierten Ausgang QN des Flip-Flops mit dem Eingang des Leitungstreibers (Knoten B), so dass abwechselnd Nullen und Einsen auf die i -te Leitung des Testbusses BusK und des Referenzbusses BusR (Knoten C) geschrieben werden.

Am Leitungsende werden beide Signale (Knoten D und E) einem Komparator zugeführt. Der Komparatorausgang (Knoten J) wird über eine Verzögerungskette in

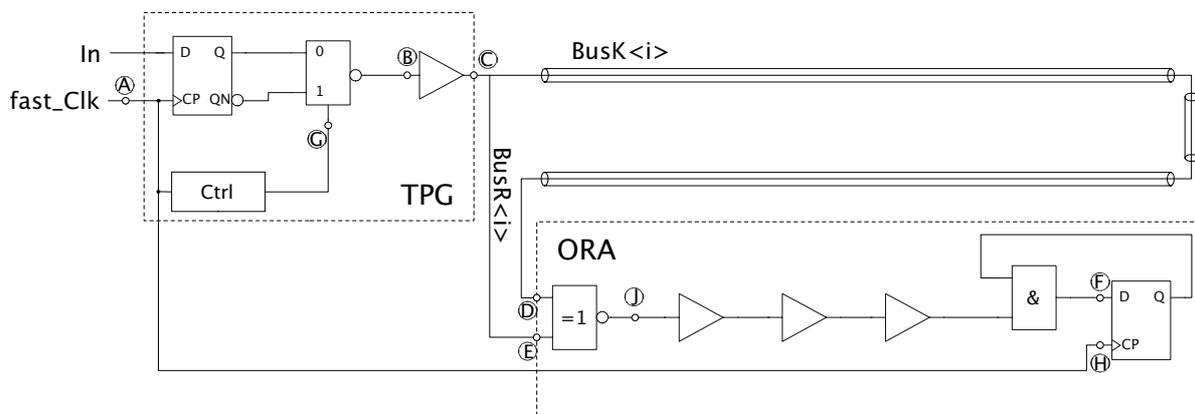


Abbildung 6.15: Schematische Darstellung des Signalpfades für den Test einer Leitung

einem Ergebnisregister gespeichert (Knoten F). Die Rückkopplung über eine UND-Verknüpfung gewährleistet, dass der dem Fehlerfall zugeordnete Zustand 0 des Ergebnisregisters⁵ bis zum Testende erhalten bleibt.

Insgesamt setzt sich die Gesamtverzögerung des Signalpfades von Knoten A bis Knoten F in Abbildung 6.15 also wie folgt zusammen.

$$\tau_{gesamt} = \tau_{TPG} + \tau_{BusK} + \tau_{comp} - (T_{skew,ORA} - T_{skew,TPG}) \quad (6.2)$$

Darin ist

- τ_{TPG} die Signalverzögerung von einer steigenden Signalflanke des Taktsignals am Eingang des Testcontrollers Ctrl bis zum korrespondierenden Signalwechsel am Ausgang des zugehörigen Leitungstreiber, d.h. von Knoten A bis Knoten C,
- τ_{BusK} die Signalverzögerung auf der Leitung des Testbusses, d.h. von Knoten C bis Knoten D,
- τ_{comp} die Signalverzögerung des Komparators und der Verzögerungskette im Testauswerter, d.h. von Knoten D bzw. Knoten E bis Knoten F,
- $T_{skew,ORA}$ die Verschiebung der Flanke des Taktsignals am Register des Testauswerters bezogen auf die korrespondierende Flanke des Taktsignals am Ausgang des Taktgenerators, d.h. von Knoten A bis Knoten H,
- $T_{skew,TPG}$ die Verschiebung der Flanke des Taktsignals am Register des Testmuster-generators bezogen auf die korrespondierende Flanke des Taktsignals am Ausgang des Taktgenerators, d.h. von Knoten A bis Knoten G.

Durch die Terme $T_{skew,ORA}$ und $T_{skew,TPG}$ wird berücksichtigt, dass das Taktsignal in einer integrierten Schaltung nicht zu allen Knoten gleiche Signalverzögerungen aufweist.

Die Bedingung 6.1 kann nur durch Verringerung von $T_{Periode}$ oder durch Vergrößern der Differenz $\tau_{BusK} - \tau_{BusR}$ erfüllt werden. Ersteres geschieht bei der Durchführung des Maximalfrequenztests, letzteres durch Wechsel vom Testmodus *Even Mode* zum Testmodus *Odd Mode*. Für die mit Hilfe des Maximalfrequenztests gefundene minimale Taktperiode T_{min} gilt wegen der Taktvervielfachung durch die PLL im Rahmen der Messgenauigkeit

$$T_{min} = 4 \cdot \tau_{gesamt}. \quad (6.3)$$

⁵ Der Startzustand des Ergebnisregisters muss 1 sein.

Damit lässt sich grundsätzlich die Signalverzögerung τ_{BusK} auf der Leitung des Testbusses aus der gemessenen minimalen Taktperiode T_{min} berechnen.

$$\tau_{BusK} = \frac{1}{4}T_{min} - \tau_{TPG} - \tau_{comp} + (T_{skew,ORA} - T_{skew,TPG}) \quad (6.4)$$

Dabei sind die Terme $T_{skew,ORA}$ und $T_{skew,TPG}$ jeweils unabhängig vom Testmodus. Dagegen ist die Signallaufzeit auf einer Busleitung τ_{BusK} abhängig vom Testmodus, ebenso wie die Signalverzögerungen τ_{TPG} des Testmustergenerators und τ_{comp} des Komparators.

Die Ermittlung von genauen Werten für die benötigten Terme in Gleichung 6.4 erweist sich jedoch als schwierig, weil diese ebensowenig einer direkten Messung zugänglich sind wie τ_{BusK} . Eine Bestimmung dieser Werte kann nur näherungsweise durch Simulation eines möglichst exakten Schaltungsmodells gewonnen werden. Eine Simulation kann aber nur eingeschränkt die Schwankungen von Schaltungsparametern durch Prozessabweichungen erfassen, so dass für die gesuchten Daten nur Bereichsgrenzen angegeben werden können.

Es soll aber eine Größe für die durch Kopplung der Busleitungen verursachte Signalverzögerung gefunden werden, die auf die Messwerte zurückführbar ist. Die nicht messbaren Werte müssen daher aus der Berechnung eliminiert werden. Dies gelingt durch Differenzenbildung der Messwerte T_{min} einer Schaltung im *Even Mode* und im *Odd Mode*. Außerdem werden dadurch die korrelierten Messfehler von $T_{min,odd}$ und $T_{min,even}$ eliminiert.

$$\Delta T_{min} = T_{min,odd} - T_{min,even} \quad (6.5)$$

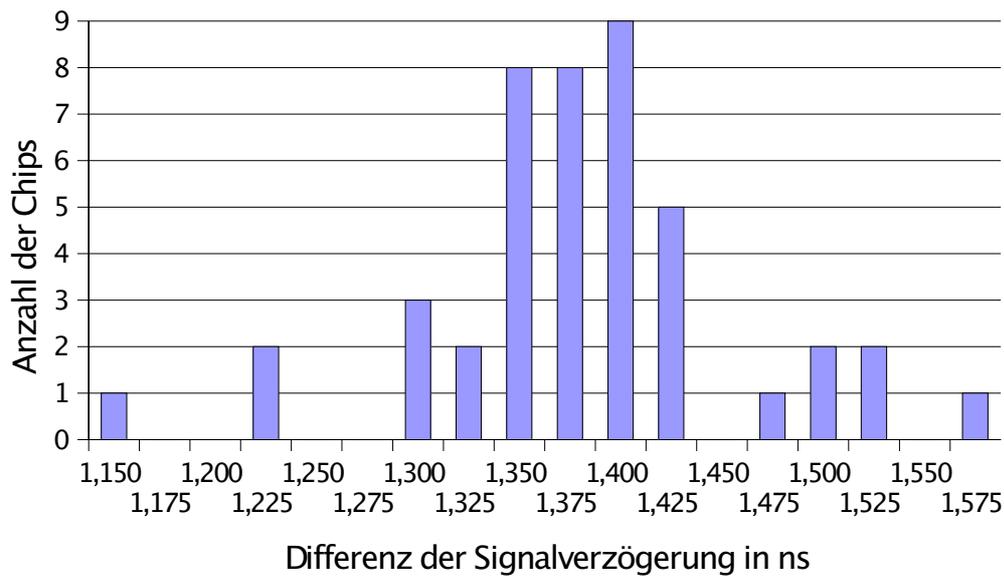
$$= 4\tau_{gesamt,odd} - 4\tau_{gesamt,even} \quad (6.6)$$

In Gleichung 6.6 kann man jetzt die Summe 6.2 jeweils für den *Odd Mode* und den *Even Mode* einsetzen.

$$\begin{aligned} \frac{1}{4}\Delta T_{min} &= \tau_{TPG,odd} + \tau_{BusK,odd} + \tau_{comp,odd} \\ &\quad - (\tau_{TPG,even} + \tau_{BusK,even} + \tau_{comp,even}) \end{aligned} \quad (6.7)$$

Die Verzögerungszeit τ_{TPG} lässt sich ferner in die von der Last abhängige Signalverzögerung durch den Leitungstreiber τ_{Driver} und die übrige Signalverzögerung des Testmustergenerators $\tau_{TMG,0}$ aufteilen. Die Bauelemente des Testmustergenerators werden durch den Leitungstreiber von der Leitungslast entkoppelt und sind daher ebenfalls unabhängig von dieser Last bzw. vom Testmodus. Gleichung 6.7 geht über in

$$\begin{aligned} \frac{1}{4}\Delta T_{min} &= \tau_{Driver,odd} - \tau_{Driver,even} \\ &\quad + \tau_{BusK,odd} - \tau_{BusK,even} + \tau_{comp,odd} - \tau_{comp,even} \\ &= \Delta\tau_{Driver} + \Delta\tau_{BusK} + \Delta\tau_{comp} =: \Delta\tau_{Kopplung} \end{aligned} \quad (6.8)$$

Abbildung 6.16: Verteilung $\Delta\tau_{Kopplung}$

Der Gesamteinfluss $\Delta\tau_{Kopplung}$ lässt sich nach Gleichung 6.8 und 6.5 direkt aus den Messwerten berechnen. Abbildung 6.16 zeigt die Verteilung von $\Delta\tau_{Kopplung}$ für die untersuchten Testschaltungen ACSelfTest. Der Minimalwert liegt bei 1,15 ns, der Maximalwert bei 1,575 ns und der Mittelwert bei 1,382 ns. Trotz der geringen Anzahl der untersuchten Chips entspricht die Verteilung näherungsweise einer Gauß-Verteilung mit dem oben genannten Mittelwert und einer Standardabweichung von 79 ps. Im Gegensatz zu den Messwerten $T_{min,odd}$ und $T_{min,even}$ sind hier die unkorrelierten Schwankungen bei der Einstellung der Taktperiode des Digital-Testers eliminiert, so dass die korrelierten Fertigungsschwankungen zwischen den einzelnen Testchips die Verteilungsfunktion von $\Delta\tau_{Kopplung}$ dominieren.

Das in der Schaltung ACSelfTest implementierte Verfahren liefert mittels zweier Maximalfrequenztests einen Maximalwert für mögliche Laufzeitunterschiede verschiedener Daten auf einem Inter-Core-Bussystem. Diese Unterschiede müssen bei der Schaltungsentwicklung bekannt sein, damit das Timing für die Datenübertragung auf diesem Bus fehlersicher ausgelegt werden kann. Die notwendigen Daten können für ein spezifisches Bussystem messtechnisch unter Verwendung eines Digital-Testers ermittelt werden.

6.4 Bewertung der Messergebnisse

Um die Plausibilität der in Abschnitt 6.3 erzielten Messergebnisse zu prüfen, können sie mit den Simulationsergebnissen aus Abschnitt 6.2 verglichen werden. Tabelle 6.2 zeigt die Signalverzögerungen in einem Schaltungsmodell aus Treibergattern,

gekoppeltem Leitungssystem und Lastgattern nach Abbildung 6.6. Die Knoten A, B, C und D entsprechen dabei den Knoten B, C, D und J des tatsächlichen Signalpfads nach Abbildung 6.15. Daher gilt

$$\begin{aligned}\Delta\tau_{Kopplung} &= \Delta\tau_{Driver} + \Delta\tau_{BusK} + \Delta\tau_{comp} \\ &= \Delta\tau_{AB} + \Delta\tau_{BC} + \Delta\tau_{CD}.\end{aligned}\tag{6.9}$$

Das Einsetzen der Werte aus Tabelle 6.2 in Gleichung 6.9 liefert nach Tabelle 6.5 ein simuliertes $\Delta\tau_{Kopplung}$ von 1,358 ns. Dieser Wert stimmt mit großer Genauigkeit mit dem Mittelwert der gemessenen Verzögerungszunahme von 1,382 ns überein. Das mit einem linearen Treiber- und Lastmodell ermittelte Ergebnis $\Delta\tau_{linear} = 221,9\text{ps}$ weicht dagegen um mehr als einen Faktor 6 von dem Messergebnis ab.

Test Modus	simulierte Verzögerung in ns				Messung
	Treiber	Leitung	Last	gesamt	Mittelwert
	τ_{AB}	τ_{BC}	τ_{CD}	τ_{gesamt}	
<i>Even Mode</i>	0,274	0,033	0,100	0,407	
<i>Odd Mode</i>	1,423	0,204	0,138	1,765	
Differenz	$\Delta\tau_{AB}$	$\Delta\tau_{BC}$	$\Delta\tau_{CD}$	$\Delta\tau_{Kopplung}$	$\Delta\tau_{Kopplung}$
	1,149	0,171	0,038	1,358	1,382

Tabelle 6.5: Vergleich des simulierten und gemessenen Kopplungseinfluss

Durch die Übereinstimmung mit dem Transistor-basierten Treiber- und Lastmodell kann gezeigt werden, dass mit dem implementierten Testverfahren die Zunahme der Signalverzögerung als ein praxisrelevantes Maß für den Einfluss von Leitungskopplung gemessen werden kann. Eine Ableitung der Leitungsbeläge R' , L' und C' auf Grundlage dieser Messungen ist nicht möglich. Das Verfahren kann daher nicht zur Modellbildung, sondern nur zur Modellverifikation verwendet werden.

Diese Übereinstimmung belegt ferner, dass der Testmodus *Odd Mode* wie im Modell ermittelt, maximale Signalverzögerung auf der Opferleitung verursacht und daher die möglichen dynamische Fehler auf dem untersuchten Bus vollständig abdeckt. Die Untersuchungen werden an einem Bussystem durchgeführt, das hinsichtlich der Anzahl der Leitungen, seiner Querschnittsdimensionen und seiner Länge praxisrelevant ist. Obwohl die Ergebnisse dieser Testschaltung nicht ohne weiteres verallgemeinert werden können, liefern sie doch einen wesentlichen Hinweis darauf, dass der in Tabelle 5.2 beschriebene Test auch bei anderen Leitungsgeometrien eine hohe Testabdeckung für dynamische Fehler liefert.

Der Mittelwert von $\Delta\tau_{Kopplung}$ liegt für die Testschaltung ACSelfTest bei 1,382 ns. Das entspricht 56% der maximalen internen Taktrate. Damit liegt der Kopplungseinfluss

schon bei der untersuchten $0,25\ \mu\text{m}$ -Technologiegeneration in einer Größenordnung, die eine fehlerfreie Funktion eines Inter-Core-Busses mit einer Länge von weniger als 10 mm gefährdet. Bei späteren Technologiegenerationen nimmt die Dominanz der Leitungsverzögerung weiter zu. Dies unterstreicht die Notwendigkeit, Inter-Core-Bussysteme durch einen geeigneten Test wie in Kapitel 5 beschrieben auf mögliche Kopplungsfehler zu prüfen.

Die Untersuchungen in Abschnitt 6.2 weisen außerdem auf den Einfluss von Treiber- und Lastmodell bei der Leitungssimulation hin. Die Verwendung eines einfachen linearen Modells aus idealer Spannungsquelle und linearem Treiberwiderstand führt zu einer deutlichen Fehleinschätzung der Gesamtverzögerung eines Leitungssystems. Sie kann daher zu Entwurfsfehlern bei Leitungssystemen führen, die die Funktionalität einer Bus-basierten Schaltung gefährden.

7 Zusammenfassung

Die Fortschritte der Halbleitertechnologie mit ihrer fortschreitenden Strukturverkleinerung ermöglichen die Realisierung von leistungsfähigen und kompakten elektronischen Systemen auf einem Chip. Diese Systems-on-Chip sind in der Regel Bus-basierte Architekturen. Dafür gibt es im Wesentlichen zwei Gründe. Erstens ist die Fertigung von Systemen derartiger Komplexität möglich, dass sie nur durch die Wiederverwendung von komplexen Teilsystemen, so genannten IP-Cores, entwickelt, verifiziert und getestet werden können. Die IP-Cores kommunizieren über integrierte Busse mit standardisierten Busprotokollen.

Zweitens ermöglicht die Strukturverkleinerung immer kürzere Transistorschaltzeiten, während Verbindungsleitungen mit zunehmenden Signalfrequenzen von störenden dynamischen Effekten durch Leitungskopplung wie Signalverzögerung und Übersprechen zwischen benachbarten Leitungen betroffen sind. Daher sind hierarchische Leitungsarchitekturen die Regel. Sie realisieren mit einer Menge von unkritischen kurzen Leitungen die enge strukturelle Verknüpfung der Bauelemente innerhalb von Teilsystemen, während die Menge der zeitkritischen, globalen Leitungen mit groß dimensionierter Leitungsgeometrie entworfen wird, um die störenden Effekte für die Signalübertragung zwischen den Teilsystemen zu minimieren.

Die globalen Leitungen bilden die Inter-Core-Busse. Sie sind ein zentraler Bestandteil von Systems-on-Chip und als solche kritisch für Leistungsfähigkeit und Zuverlässigkeit der Systems-on-Chip. Für die IP-Cores existieren bereits verschiedene in der Industrie etablierte Testverfahren. Die Integration der IP-Core-Tests soll durch den neuen Standard IEEE 1500 vereinfacht behandelt werden. Hinsichtlich der Inter-Core-Busse gibt es weder etablierte Testmethoden noch Standardisierungsbemühungen.

Die vorliegende Arbeit beruht auf Beiträgen zur Entwicklung eines großflächig integrierten Multiprozessorsystems für die Videosignalverarbeitung. Dieses System zeichnet sich durch die Verwendung von IP-Cores, die Chipfläche von 16,89 cm² mit einem auf Fotokomposition basierenden Fertigungsverfahren und die dafür notwendigen Fehlertoleranz- und Selbsttestkonzepte aus. Die Signalintegrität auf den Inter-Core-Bussen war ein Schwerpunkt bei der Entwicklung dieses Systems.

Als Grundlage für einen deterministischen Test von Bussystemen dient das Maximum Aggressor Fehlermodell. Dieses Fehlermodell wird aus der Beschreibung eines

gekoppelten Leitungssystem abgeleitet, das als Kettenschaltung von RC-Gliedern modelliert wird. Dadurch kann eine Fehlerbedingung für eine Opferleitung stets gefunden werden, indem auf allen anderen Leitungen ein Signalwechsel entgegen der Opferleitung stattfindet. Für das Maximum Aggressor Fehlermodell werden Tests abgeleitet, die durch Erweiterung des bekannten Walking-1 Tests entstehen. Ein Test auf alle Störsignale und Signalverzögerungen eines N -bit Leitungssystems benötigt $6N$ Takte.

Theoretische und messtechnische Untersuchungen von Leitungssystemen integrierter Schaltungen haben nachgewiesen, dass die Beschreibung eines gekoppelten Leitungssystems als Kettenschaltung von RC-Gliedern bei heutigen und zukünftigen Leitungsdimensionen und Signalfrequenzen nicht mehr ausreichend ist. Vielmehr muss das Modell die Wellenausbreitung auf dem Leitungssystem beschreiben. Dies gelingt zumindest näherungsweise durch die Modellierung als Kette von Gliedern aus Widerstand, Kapazität, Induktivität und Gegeninduktivität (RLCM-Modell). Für dieses Leitungsmodell kann in dieser Arbeit jedoch keine allgemeingültige Fehlerbedingung angegeben werden¹, so dass Leitungstests durch Simulationen geprüft werden müssen. Für den Leitungstest wird der so genannte Odd Mode als der ungünstigste Fall für Störsignale und Leitungsverzögerung verwendet. Der Odd Mode ist das Schalten einer Opferleitung gegen alle anderen Leitungen und entspricht der Fehlerbedingung des Maximum Aggressor Fehlermodells.

Es wird ein Testmustergenerator und ein Testauswerter vorgestellt, die auf einem Schieberegister basieren und daher mit minimalem Overhead in einen so genannten Core Test Wrapper nach dem Standard IEEE 1500 integriert werden können. Die Testauswertung erfolgt durch ein linear rückgekoppeltes Schieberegister mit mehreren Eingängen (MISR). Mit einem taktgenauen Verhaltensmodell für das fehlerbehaftete Leitungssystem und den Testauswerter kann gezeigt werden, dass für alle praxisrelevanten Busbreiten $N > 8$ kein Aliasing auftritt. Jeder Fehler führt vielmehr zu einer eindeutigen Testsignatur, die zu Diagnosezwecken verwendet werden kann.

Bei der Testentwicklung von Systemen mit mehreren unabhängigen Bussen muss Inter-Bus-Kopplung berücksichtigt werden. Maximale Störungen einer Opferleitung resultieren, wenn alle Leitungen eines benachbarten Busses gemeinsam als Aggressoren fungieren. Diese Randbedingung für den Testablaufplan kann durch einen Verbindungsgraphen und einen Kopplungsgraphen beschrieben werden. Um Inter-Bus-Kopplung zu minimieren, sollte der Kopplungsgraph schon beim Floorplanning berücksichtigt werden.

Das Leitungstestverfahren ist zusammen mit einem 16-bit Bus auf einem Testchip integriert worden, der in einem $0,25 \mu\text{m}$ -Fertigungsprozess mit 6 Metallagen gefertigt worden ist. Die Testschaltung ermöglicht die Erzeugung von Testsignalen im Odd

¹ Dieses Problem bleibt als wichtige Frage für weitere theoretische Untersuchungen offen.

Mode und im Even Mode, bei dem alle Leitungen in die gleiche Richtung schalten, sowie die Erkennung von Fehlern, die durch erhöhte Signallaufzeiten aufgrund von Leitungskopplung entstehen. Durch Messung und Vergleich von maximalen Taktfrequenzen der Testschaltung im Even Mode und im Odd Mode kann der Einfluss der Leitungskopplung demonstriert und quantitativ bestimmt werden.

Das beschriebene Leitungstestverfahren ist prinzipbedingt von der Messgröße und der Messgenauigkeit her nicht mit On-Wafer-Messverfahren für gekoppelte Doppelleitungen vergleichbar. Die Zielsetzung liegt daher nicht in der Charakterisierung von Leitungen, sondern im praktischen Nachweis der Leitungskopplung in Inter-Core-Bussen sowie in der Verwendung als Demonstrator für den Leitungselbsttest.

Die Messungen sind mit dem Digital-Tester HP 82000 durchgeführt worden. Die gemessene minimale Taktperiode für die Testschaltung ist im Odd Mode mit 18,18 ns nahezu doppelt so groß wie im Even Mode mit 9,79 ns. Die interne Maximalfrequenz beträgt dabei 220,0 MHz bzw. 408,6 MHz. Diese Differenz ist allein auf die durch Kopplung verursachte zusätzliche Signalverzögerung der jeweiligen Opferleitung zurückzuführen. Sie beträgt rechnerisch im Mittel 1,38 ns.

Für die Testschaltung kann durch Schaltungssimulationen anhand eines Leitungsmodells mit verketteten RLCM-Gliedern und genauen, nicht linearen Treiber- und Lastmodellen nachgewiesen werden, dass Testsignale im Odd Mode maximale Leitungsverzögerung und maximale Störungen auf der Opferleitung verursachen. Die simulierte zusätzliche Signalverzögerung von 1,36 ns stimmt praktisch exakt mit der Messung überein.

Die Schaltungssimulationen zeigen ferner, dass weniger als 15 % der Signalverzögerung auf die reine Signallaufzeit der Leitung entfallen. Der Hauptanteil liegt vielmehr in der Gatterverzögerung des Leitungstreibers. Dieser Anteil kann nicht durch ein lineares Treibermodell bei der Leitungssimulation beschrieben werden. Für verschiedene Bereiche des Schaltungsentwurfs wie zum Beispiel die Timing-Analyse, die physikalische Synthese oder die Post-Layout-Optimierung wäre daher die Entwicklung von effizienten integrierten Leitungs- und Treibermodellen sinnvoll, die den Kopplungseinfluss auf die Gatterverzögerung quantifizieren.

Mit dieser Arbeit wird ein Verfahren für den Selbsttest von Inter-Core-Verbindungsnetzen präsentiert, das mit geringem Overhead Fehler durch Leitungskopplung vollständig erfasst und kompatibel zum neuen Standard IEEE 1500 für den Test von Systems-on-Chip ist. Die Funktionsfähigkeit des Verfahrens ist durch die Fertigung und Analyse eines Testchips demonstriert worden. Die Messergebnisse unterstreichen die Notwendigkeit, einen Leitungstest auf Kopplungsfehler für Inter-Core-Busse in Systems-on-Chip zu implementieren.

Literaturverzeichnis

- [ABF95] ABRAMOVICI, Miron ; BREUER, Melvin A. ; FRIEDMAN, Arthur D.: *Digital Systems Testing and Testable Design*. Revised Printing. Piscataway, USA : IEEE Press, 1995
- [AC89] AMERASEKERA, E. Ajith ; CAMPBELL, David S.: *Failure Mechanisms in Semiconductor Devices*. Chichester, Großbritannien : John Wiley & Sons, 1989
- [AN01] ACHAR, Ramachandra ; NAKHLA, Michel S.: Simulation of High-Speed Interconnects. In: *Proceedings of the IEEE* 89 (2001), Mai, Nr. 5, S. 693 – 728
- [AN02] ATTARHA, Amir ; NOURANI, Mehrdad: Test Pattern Generation for Signal Integrity Faults on Long Interconnects. In: *Proceedings of the 20th IEEE VLSI Test Symposium (VTS 02)*, 2002, S. 336 – 341
- [Arm99] Arm Limited: *AMBA™ Specification (Rev. 2.0)*. 1999
- [AWW⁺00] ARZ, Uwe ; WILLIAMS, Dylan F. ; WALKER, David K. ; ROGERS, Janet E. ; RUDACK, Markus ; TREYTNAR, Dieter ; GRABINSKI, Hartmut: Characterization of Asymmetric Coupled CMOS Lines. In: *IEEE MTT-S International Microwave Symposium Digest*, 2000, S. 609 – 612
- [BA00] BUSHNELL, Michael L. ; AGRAWAL, Vishwani D.: *Essentials of Electronic Testing for Digital, Memory & Mixed-Signal VLSI Circuits*. Boston, USA : Kluwer Academic Publishers, 2000
- [BBD⁺01] BARNHART, Carl ; BRUNKHORST, Vanessa ; DISTLER, Frank ; FARNSWORTH, Owen ; KELLER, Brion ; KOENEMANN, Bernd: OPMISR: The Foundation for Compressed ATPG Vectors. In: *Proceedings of the International Test Conference*, 2001, S. 748 – 757
- [BDR01] BAI, Xiaoliang ; DEY, Sujit ; RAJSKI, Janusz: Self-Test Methodology for At-Speed Test of Crosstalk in Chip Interconnects. In: *Proceedings of the 38th Design Automation Conference*, 2001, S. 619 – 624
- [Bha04] BHANDARKAR, Dileep. *Billion Transistor Microprocessor Chips in Mainstream Enterprise Platforms of the Future*. Vortrag am Laboratorium für Informationstechnologie, Universität Hannover, 5. Mai 2004. Mai 2004
- [BM82] BARDELL, Paul H. ; MCANNEY, William H.: Self-Testing of Multichip Logic Modules. In: *Proceedings of the International Test Conference*, 1982, S. 200 – 204
- [BM02] BANERJEE, Kaustav ; MEHROTRA, Amit: Analysis of On-Chip Inductance Effects for Distributed RLC Interconnects. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 21 (2002), August, Nr. 8, S. 904 – 915
- [BMS87] BARDELL, Paul H. ; MCANNEY, William H. ; SAVIR, Jacob: *Built-In Test for VLSI: Pseudo-random Techniques*. Somerset, USA : John Wiley & Sons, 1987
- [Boh95] BOHR, Mark T.: Interconnect Scaling – The Real Limiter to High Performance ULSI. In: *Proceedings of the International Electron Devices Meeting*, 1995, S. 241 – 244

- [BR01] BURNS, Mark ; ROBERTS, Gordon W.: *An Introduction to Mixed-Signal IC Test and Measurement*. New York, USA : Oxford University Press, 2001
- [BRK093] BOTHRA, Subhas ; ROGERS, Boyd ; KELLAM, Mark ; OSBURN, Carlton M.: Analysis of the Effects of Scaling on Interconnect Delay in ULSI Circuits. In: *IEEE Transactions on Electron Devices* 40 (1993), März, Nr. 3, S. 591 - 597
- [BSCB02] BASU, Subhayu ; SENGUPTA, Indranil ; CHOWDHURY, Dipanwita R. ; BHAWMIK, Sudipta: An Integrated Approach to Testing Embedded Cores and Interconnects Using Test Access Mechanism (TAM) Switch. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 475 - 485
- [Cad99a] Cadence Design Systems, Inc., San Jose, USA: *Affirma Spectre Circuit Simulator Reference*. 1999
- [Cad99b] Cadence Design Systems, Inc., San Jose, USA: *Composer: Design Entry User Guide*. 1999
- [Cad99c] Cadence Design Systems, Inc., San Jose, USA: *Verifault-XL Reference*. 1999
- [CBD02] CHEN, Li ; BAI, Xiaoliang ; DEJ, Sujit: Testing for Interconnect Crosstalk Defects Using On-Chip Embedded Processor Cores. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 529 - 538
- [CC01] CHANDRA, Anshuman ; CHAKRABARTY, Krishnendu: System-on-a-Chip Test-Data Compression and Decompression Architectures Based on Golomb Codes. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20 (2001), März, Nr. 3, S. 355 - 368
- [CDBS01] CAIGNET, Fabrice ; DELMAS-BENDHIA, Sonia ; SICARD, Etienne: The Challenge of Signal Integrity in Deep-Submicrometer CMOS Technology. In: *Proceedings of the IEEE* 89 (2001), April, Nr. 4, S. 556 - 573
- [CDBZ99] CUVIELLO, Michael ; DEY, Sujit ; BAI, Xiaoliang ; ZHAO, Yi: Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects. In: *Proceedings of the International Conference on Computer-Aided Design*, 1999, S. 297 - 303
- [CGB97] CHEN, Wei-Yu ; GUPTA, Sandeep K. ; BREUER, Melvin A.: Analytical Models for Crosstalk Delay And Pulse Analysis Under Non-Ideal Inputs. In: *Proceedings of the International Test Conference*, 1997, S. 809 - 818
- [CMS01] CHEN, Lauren H. ; MAREK-SADOWSKA, Malgorzata: Aggressor Alignment for Worst-Case Crosstalk Noise. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 20 (2001), may, Nr. 5, S. 612 - 621
- [Cro99] CROUCH, Alfred L.: *Design for Test for Digital ICs and Embedded Core Systems*. Upper Saddle River, USA : Prentice Hall, 1999
- [CS93] CHANG, Chiyuan ; SU, Chauchin: A Universal BIST Methodology for Interconnects. In: *Proceedings of the International Symposium on Circuits and Systems*, 1993, S. 1615 - 1618
- [Dae82] DAEHN, Wilfried: *Testmuster-generierung für den Selbstest integrierter Digitalschaltungen*, Universität Hannover, Dissertation, 1982
- [DCK⁺01] DEUTSCH, Alina ; COTUES, Paul W. ; KOPCSAY, Gerard V. ; SMITH, Howard H. ; SUROVIC, Christopher W. ; KRAUTER, Byron L. ; EDELSTEIN, Daniel C. ; RESTLE, Phillip J.: On-Chip Wiring Design Challenges for Gigahertz Operation. In: *Proceedings of the IEEE* 89 (2001), April, Nr. 4, S. 529 - 555

- [DKR⁺97] DEUTSCH, Alina ; KOPCSAY, Gerard V. ; RESTLE, Phillip J. ; KATOPIS, George ; BECKER, W.D. ; SMITH, Howard H. ; COTUES, Paul W. ; SUROVIC, Christopher W. ; RUBIN, B.J. ; DUNNE, R.P. ; GALLO, T.A. ; JENKINS, K.A. ; TERMAN, L.M. ; DENNARD, R.H. ; SAI-HALASZ, G.A. ; KNEBEL, D.R.: When Are Transmission-Line Effects Important for On-Chip Interconnections. In: *Proceedings of the Electronic Components and Technology Conference*, 1997, S. 704 – 712
- [DM81] DAEHN, Wilfried ; MUCHA, Joachim: Hardware Test Pattern Generation for Built-In Testing. In: *International Test Conference, Digest of Papers*, 1981, S. 110 – 113
- [DM98] DAVIS, Jeffrey A. ; MEINDL, James D.: Is Interconnect The Weak Line? In: *IEEE Circuits & Devices Magazine* 14 (1998), März, Nr. 2, S. 30 – 36
- [EEJK00] EO, Yungseon ; EISENSTADT, William R. ; JEONG, Ju Y. ; KWON, Oh-Kyong: A New On-Chip Interconnect Crosstalk Model and Experimental Verification for CMOS VLSI Circuit Design. In: *IEEE Transactions on Electron Devices* 47 (2000), Januar, Nr. 1, S. 129 – 140
- [EPRB03] ENGELKE, Piet ; POLIAN, Ilia ; RENOVELL, Michel ; BECKER, Bernd: Simulating Resistive Bridging Faults and Stuck-At Faults. In: *Proceedings of the International Test Conference*, 2003, S. 1051 – 1059
- [GARP98] GROSS, Paul D. ; ARUNACHALAM, Ravishankar ; RAJAGOPAL, Karthik ; PILEGGI, Lawrence T.: Determination of Worst-Case Aggressor Alignment for Delay Calculation. In: *Proceedings of the International Conference on Computer-Aided Design*, 1998, S. 212 – 219
- [GDZ94] GROTELÜSCHEN, Enno ; DUTTA, Lohit S. ; ZAAGE, Stefan: Quasianalytical Analysis of the Broadband Properties of Multiconductor Transmission Lines on Semiconducting Substrates. In: *IEEE Transactions on Components, Packaging, and Manufacturing Technology – Part B* 17 (1994), August, Nr. 3, S. 376 – 382
- [GGRW95] GROTELÜSCHEN, Enno ; GRABINSKI, Hartmut ; ROCHEL, Steffen ; WINKEL, Thomas M.: LOSSYWIRE - A Model Implementation for Transient and AC Analysis of Lossy Coupled Transmission Lines in the Circuit Simulator EldoTM. In: *Archiv für Elektronik und Übertragungstechnik (AE)* 49 (1995), Januar, Nr. 1, S. 37 – 43
- [Gol82] GOLOMB, Solomon W.: *Shift Register Sequences*. 2nd Edition. Laguna Hills, USA : Aegan Park Press, 1982
- [Grü88] GRÜTZNER, Matthias: *Zum Test von großflächig integrierten Schaltungen unter besonderer Berücksichtigung der Netze*, Universität Hannover, Dissertation, 1988
- [Gra91] GRABINSKI, Hartmut: *Theorie und Simulation von Leitbahnen*. Berlin : Springer Verlag, 1991
- [GTA⁺99] GRABINSKI, Hartmut ; TREYTNAR, Dieter ; ARZ, Uwe ; KTATA, Faiez ; NORDHOLZ, Petra: Influence of Frequency-Dependent Characteristics on Deep Submicron Crosstalk Simulations. In: *Proceedings of the European Conference on Circuit Theory and Design*, 1999, S. 543 – 546
- [HH00] HAVEMANN, Robert H. ; HUTCHBY, James A.: High-Performance Interconnects: An Integration Overview. In: *Proceedings of the IEEE* 89 (2000), Mai, Nr. 5, S. 586 – 601
- [HHO⁺98] HILGENSTOCK, Jörg ; HERRMANN, Klaus ; OTTERSTEDT, Jan ; NIGGEMEYER, Dirk ; PIRSCH, Peter: A Video Signal Processor for MIMD Multi-processing. In: *Proceedings of the 35th Design Automation Conference*, 1998, S. 50 – 55

- [HMH01] HO, Ron ; MAI, Kenneth W. ; HOROWITZ, Mark A.: The Future Of Wires. In: *Proceedings of the IEEE* 89 (2001), April, Nr. 4, S. 490 - 504
- [HMHP00] HERRMANN, Klaus ; MOCH, Sören ; HILGENSTOCK, Jörg ; PIRSCH, Peter: Implementation of a Multiprocessor System with Distributed Embedded DRAM on a Large Area Integrated Circuit. In: *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2000, S. 105 - 113
- [Hol02] HOLMAN, Paul: *The Technology Behind PlayStation 2*. IEEE United Kingdom and Republic of Ireland Section Lecture. 2002. - <http://www.ieee.org.uk/docs/sony.pdf>
- [HP03] HENNESSY, John L. ; PATTERSON, David A.: *Computer Architecture - A Quantitative Approach*. 3rd Edition. Amsterdam, Niederlande (u.a.) : Morgan Kaufman Publishers, 2003
- [ICM02] IYENGAR, Vikram ; CHAKRABARTY, Krishnendu ; MARINISSEN, Erik J.: Test Wrapper and Test Access Mechanism Co-Optimization for System-on-Chip. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 213 - 230
- [IEE99] IEEE Computer Society, Test Technology Standards Committee: *IEEE Standard Test Interface Language (STIL) for Digital Test Vector Data (IEEE Std 1450-1999)*. 1999
- [IEE01] IEEE Computer Society, Test Technology Standards Committee: *IEEE Standard Test Access Port and Boundary-Scan Architecture (IEEE Std 1149.1-2001)*. 2001
- [IEE05] IEEE Computer Society, Test Technology Standards Committee: *IEEE Standard Testability Method for Embedded Core-based Integrated Circuits (IEEE Std 1500-2005)*. 2005
- [IF00] ISMAIL, Yehea I. ; FRIEDMAN, Eby G.: Effects of Inductance on the Propagation Delay and Repeater Insertion in VLSI Circuits. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 8 (2000), April, Nr. 2, S. 195 - 206
- [Int04] Intel Corp., Santa Clara, USA: *Intel Microprocessor Quick Reference Guide*. 2004. - <http://www.intel.com/pressroom/kits/quickreffam.htm>
- [ITR03] *International Technology Roadmap for Semiconductors, 2003 Edition*. European Semiconductor Industry Association, Japan Electronics and Information Technology Industries Association, Korea Semiconductor Industry Association, Taiwan Semiconductor Industry Association, Semiconductor Industry Association. 2003. - <http://public.itrs.net>
- [Kau74] KAUTZ, W. H.: Testing for Faults in Wiring Networks. In: *IEEE Transactions on Computers* 23 (1974), April, Nr. 4, S. 358 - 363
- [KMSS98] KAHNG, Andrew B. ; MUDDU, Sudhakar ; SARTO, Eginio ; SHARMA, Rahul: Interconnect Tuning Strategies for High-Performance ICs. In: *Proceedings of the 1998 Design, Automation and Test in Europe Conference and Exhibition (DATE 98)*, 1998, S. 471 - 478
- [KMOV01] KRISHNASWAMY, V. ; MA, A. B. ; VISHAKANTAIAH, P.: A Study of Bridging Defect Probabilities on a Pentium^TM 4 CPU. In: *Proceedings of the International Test Conference*, 2001, S. 688 - 695
- [KMZ79] KÖNEMANN, Bernd ; MUCHA, Joachim ; ZWIEHOFF, Günther: Built-In Logic Block Observation Techniques. In: *Digest of Papers 1979 Test Conference*, 1979, S. 37 - 41
- [KMZ80] KÖNEMANN, Bernd ; MUCHA, Joachim ; ZWIEHOFF, Günther: Built-In Test for Complex Digital Integrated Circuits. In: *IEEE Journal of Solid-State Circuits* 15 (1980), Juni, Nr. 3, S. 315 - 318

- [Kon04] Koninklijke Philips Electronics N.V., Niederlande: *MOS Model, level 902*. 2004. – <http://www.semiconductors.philips.com/acrobat/other/philipsmodels/m902.pdf>
- [Kor02] KORANNE, Sandeep: A Novel Reconfigurable Wrapper for Testing of Embedded Core-Based SOCs and its Associated Scheduling Algorithm. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 415 – 434
- [KPI90] KAMEDA, Tiko ; PILARSKI, Slawomir ; IVANOV, Andre: Notes on Multiple Input Signature Analysis / Centre for Systems Science/Laboratory for Computer and Communications Research (CSS/LCCR), Simon Fraser University, Canada. 1990 (TR90-16). – Forschungsbericht
- [KQM+02] KLEVELAND, Bendik ; QI, Xiaoning ; MADDEN, Liam ; FURUSAWA, Takeshi ; DUTTON, Robert W. ; HOROWITZ, Mark A. ; WONG, S. S.: High-Frequency Characterization of On-Chip Digital Interconnects. In: *IEEE Journal of Solid-State Circuits* 37 (2002), Juni, Nr. 6, S. 716 – 725
- [KVMW01] KIEFER, Gundolf ; VRANKEN, Harald ; MARINISSEN, Erik J. ; WUNDERLICH, Hans-Joachim: Application of Deterministic Logic BIST on Industrial Circuits. In: *Journal of Electronic Testing: Theory and Applications* 17 (2001), März, Nr. 3-4, S. 351 – 362
- [KW99] KAPUR, Rohit ; WILLIAMS, Thomas W.: Tough Challenges as Design and Test Go Nanometer. In: *Computer* 32 (1999), November, Nr. 11, S. 42 – 45
- [Laj01] LAJOLO, Marcello: Bus Guardians: An Effective Solution for Online Detection and Correction of Faults Affecting System-On-Chip Buses. In: *IEEE Transactions on Very Large Scale Integration (VLSI) Systems* 9 (2001), Dezember, Nr. 6, S. 974 – 982
- [LC83] LIN, Shu ; COSTELLO, Daniel J.: *Error Control Coding – Fundamentals and Applications*. Englewood Cliffs, USA : Prentice Hall, 1983
- [LCL96] LILLIS, John ; CHENG, Chung-Kuan ; LIN, Ting-Ting Y.: Optimal Wire Sizing and Buffer Insertion for Low Power and a Generalized Delay Model. In: *IEEE Journal of Solid-State Circuits* 31 (1996), März, Nr. 3, S. 437 – 447
- [LHC01] LAI, Wei-Cheng ; HUANG, Jing-Reng ; CHENG, Kwang-Ting (Tim): Embedded-Software-Based Approach To Testing Crosstalk-Induced Faults at On-Chip Buses. In: *Proceedings of the 19th IEEE VLSI Test Symposium (VTS 01)*, 2001, S. 204 – 209
- [LP02] LARSSON, Erik ; PENG, Zebo: An Integrated Framework for the Design and Optimization of SOC Test Solutions. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 385 – 400
- [Mei03] MEINDL, James D.: Interconnect Opportunities For Gigascale Integration. In: *IEEE Micro* 23 (2003), Mai, Nr. 3, S. 28 – 35
- [MFR00] METRA, Cecilia ; FAVALLI, Michele ; RICCO, Bruno: Self-Checking Detection and Diagnosis of Transient, Delay, and Crosstalk Faults Affecting Bus Lines. In: *IEEE Transactions on Computers* 49 (2000), Juni, Nr. 6, S. 560 – 574
- [MKL+02] MARINISSEN, Erik J. ; KAPUR, Rohit ; LOUSBERG, Maurice ; LAURIN, Teresa M. ; RICCHETTI, Mike ; ZORIAN, Yervant: On IEEE P1500's Standard for Embedded Core Test. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 365 – 383
- [Moo65] MOORE, Gordon E.: Cramming More Components Onto Integrated Circuits. In: *Electronics* 38 (1965), April, Nr. 8

- [MRR98] MOLL, Francesco ; ROCA, Miquel ; RUBIO, Antonio: Inductance in VLSI Interconnection Modelling. In: *IEE Proceedings - Circuits, Devices and Systems* 145 (1998), Juni, Nr. 3, S. 175 - 179
- [MRRT00] MENDE, Ole ; REDEKER, Michael ; RUDACK, Markus ; TREYTNAR, Dieter: A Multifunctional Laser Linking and Cutting Structure for Standard 0.25 μm CMOS-Technology. In: *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2000, S. 114 - 122
- [NA01] NOURANI, Mehrdad ; ATTARHA, Amir: Built-In Self-Test for Signal Integrity. In: *Proceedings of the 38th Design Automation Conference*, 2001, S. 792 - 797
- [NA02] NOURANI, Mehrdad ; ATTARHA, Amir: Signal Integrity: Fault Modeling and Testing in High-Speed SoCs. In: *Journal of Electronic Testing: Theory and Applications* 18 (2002), August, Nr. 4, S. 539 - 554
- [NGT⁺98] NORDHOLZ, Petra ; GRABINSKI, Hartmut ; TREYTNAR, Dieter ; OTTERSTEDT, Jan ; NIGGEMEYER, Dirk ; ARZ, Uwe ; WILLIAMS, T.W.: Core Interconnects Testing Hazards. In: *Proceedings of the 1998 Design, Automation and Test in Europe Conference and Exhibition (DATE 98)*, 1998, S. 953 - 954
- [Nig03] NIGGEMEYER, Dirk: *Generation of diagnostic memory tests and on-chip test application with special consideration of embedded memories in systems-on-chips*, Universität Hannover, Dissertation, 2003
- [NTO⁺98] NORDHOLZ, Petra ; TREYTNAR, Dieter ; OTTERSTEDT, Jan ; GRABINSKI, Hartmut ; NIGGEMEYER, Dirk ; WILLIAMS, T.W.: Signal Integrity Problems in Deep Submicron Arising From Interconnects Between Cores. In: *Proceedings of the 16th IEEE VLSI Test Symposium (VTS 98)*, 1998, S. 28 - 33
- [RMBF96] RODRIGUES-MONTANES, R. ; BRULS, Eric ; FIGUERAS, Joan: Bridging Defects Resistance In The Metal Layer Of A CMOS Process. In: *Journal of Electronic Testing: Theory and Applications* 8 (1996), Februar, Nr. 1, S. 35 - 46
- [RN99] RUDACK, Markus ; NIGGEMEYER, Dirk: Yield Enhancement Considerations for a Single-Chip Multiprocessor System with Embedded DRAM. In: *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 1999, S. 31 - 39
- [Rot66] ROTH, J. Paul: Diagnosis of Automata Failures: A Calculus and a Method. In: *IBM Journal of Research and Development* 10 (1966), Juli, Nr. 4, S. 278 - 291
- [RRH⁺02] RUDACK, Markus ; REDEKER, Michael ; HILGENSTOCK, Jörg ; MOCH, Sören ; CASTAGNE, Jens: A Large-Area Integrated Multiprocessor System for Video Applications. In: *IEEE Design & Test of Computers* 19 (2002), Januar, Nr. 1, S. 6 - 17
- [RRT⁺00] RUDACK, Markus ; REDEKER, Michael ; TREYTNAR, Dieter ; MENDE, Ole ; HERRMANN, Klaus: Self-Configuration of a Large Area Integrated Multiprocessor System for Video Applications. In: *Proceedings of the International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2000, S. 78 - 86
- [RT97] RAJSKI, Janusz ; TYSZER, Jerzy: *Arithmetic Built-In Self-Test for Embedded Systems*. Upper Saddle River, USA : Prentice Hall PTR, 1997
- [RTK⁺02] RAJSKI, Janusz ; TYSZER, Jerzy ; KASSAB, Mark ; MUKHERJEE, Milanjan ; THOMPSON, Rob ; TSAI, Kun-Han ; HERTWIG, Andre ; TAMARAPALLI, Nagesh ; MRUGALSKI, Grzegorz ; EIDER, Geir ; QIAN, Jun: Embedded deterministic test for low cost manufacturing test. In: *Proceedings of the International Test Conference*, 2002, S. 301 - 310

- [SSK⁺03] SASAKI, Yasuhiko ; SATO, Mitsumasa ; KURAMOTO, Masaru ; KIKUCHI, Fujio ; KAWASHIMA, Tsutomu ; MASUDA, Hiroo ; YANO, Kazuo: Crosstalk Delay Analysis of a 0.13- μ m Node Test Chip and Precise Gate-Level Simulation Technology. In: *IEEE Journal of Solid-State Circuits* 38 (2003), Mai, Nr. 5, S. 702 - 708
- [ST01] SU, Chauchin ; TSENG, Wenliang: Configuration Free SoC Interconnect BIST Methodology. In: *Proceedings of the International Test Conference*, 2001, S. 1033 - 1038
- [Syn00] Synopsys, Inc., Mountain View, USA: *Synopsys Test Compiler ATPG Reference Manual*, v2000.11. 2000
- [The00] THEIS, Thomas N.: The Future of Interconnection Technology. In: *IBM Journal of Research and Development* 44 (2000), Mai, Nr. 3, S. 379 - 390
- [TRGK02] TREYTNAR, Dieter ; REDEKER, Michael ; GRABINSKI, Hartmut ; KTATA, Faiez: LFSR Test Pattern Crosstalk in Nanometer Technologies. In: *Proceedings of the 6th International Workshop on Signal Propagation on Interconnects (SPI)*, 2002, S. 115 - 118
- [van98] VAN DE GOOR, Ad J.: *Testing Semiconductor Memories - Theory and Practice*. Gouda, Niederlande : ComTex Publishing, 1998
- [WDGS88] WILLIAMS, Thomas W. ; DAEHN, Wilfried ; GRÜTZNER, Matthias ; STARKE, Cord W.: Bounds and Analysis of Aliasing Errors in Linear-Feedback Shift-Registers. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 7 (1988), Januar, Nr. 1, S. 75 - 83
- [YHS⁺01] YANG, Jun ; HU, Chen ; SHI, Youhua ; ZHANG, Zhe ; SHI, Longxing: A New Self-Test Structure for At-Speed Test of Crosstalk in SoC Busses. In: *Proceedings of the 4th International Conference on ASIC (ASICON2001)*, 2001, S. 633 - 636

Wissenschaftlicher Werdegang

von Markus Rudack
geboren am 3. August 1971 in Soest

Ausbildung

- 1991 Abitur am Städt. Aldegrever-Gymnasium, Soest
- 1991 - 1997 Studium der Elektrotechnik an der Universität Hannover,
Studienschwerpunkt Mikroelektronik
Abschluß am 22. Dezember 1997 und Erlangung des
Hochschulgrades Diplom-Ingenieur
Thema der Diplomarbeit: „Methoden zur integrierten
Temperaturmessung auf CMOS Schaltungen“

Berufliche Tätigkeit

- 1998 - 2003 Wissenschaftlicher Mitarbeiter bei Prof. Dr.-Ing.
Joachim Mucha und Prof. Dr.-Ing. Wolfgang Mathis
am Institut für Theoretische Elektrotechnik und
Hochfrequenztechnik und am Laboratorium für
Informationstechnologie, Abteilung Design und Test,
Universität Hannover
- 2003 - 2004 Wissenschaftlicher Mitarbeiter bei Prof. Dr.-Ing.
Erich Barke am Institut für Mikroelektronische Systeme,
Abteilung Entwurfsautomatisierung,
Universität Hannover
- Seit 2005 Entwicklungsingenieur im Bereich Digital Test Methodology
bei Infineon Technologies AG, München

