University of Cincinnati

Date: 10/26/2011

I, Jianxun Liu, hereby submit this original work as part of the requirements for the degree of Doctor of Philosophy in Computer Science & Engineering.

It is entitled:

Pseudo-Exhaustive Built-in Self-Testing for Signal Integrity of High-Speed SoC Interconnects

Student's name: **Jianxun Liu**

This work and its defense approved by:

Committee chair: Wen Ben Jone, PhD

Committee member: Chien-In Henry Chen, PhD

Committee member: Harold Carter, PhD

Committee member: Carla Purdy, C, PhD

Committee member: Ranganadha Vemuri, PhD

2099

**Pseudo-Exhaustive Built-in Self-Testing for Signal Integrity
of High-Speed SoC Interconnects**

A dissertation submitted to the
Graduate School
of the University of Cincinnati
in partial fulfillment of the
requirements for the degree of

DOCTOR OF PHILOSOPHY

in the Department of
Electrical and Computer Engineering and Computer Science
of the College of Engineering

by

Jianxun Liu

M.S. University of Cincinnati

December, 2003

Committee Chair: Wen-Ben Jone, Ph.D.

Abstract

As technology approaches deep sub-micron and clock frequency approaches Giga Hertz, the signal integrity problem of high-speed interconnects is becoming a more and more serious issue. In this work, we propose a pseudo-exhaustive testing scheme for signal integrity faults of high-speed SoC interconnects.

We first validate the applicability of traditional pseudo-exhaustive testing methods to high-speed interconnect testing by validating the crosstalk locality. Base on the concept of crosstalk locality, a PE-BIST testing scheme for simple interconnect bus structures is proposed. The scheme uses a serial scan chain interface, and thus can be easily integrated with existing boundary scan architectures. Special boundary scan cells and instructions to support such integration are also discussed.

The proposed PE-BIST method is then extended to arbitrary interconnect structures. With the aid of a *Net Interference Graph (NIG)*, we can easily identify the PE-BIST test cone size and assign individual nets into PE-BIST channels. The test architecture for arbitrary interconnects is also very simple, largely reusing existing BIST components built on the chip. The hardware overhead can therefore be minimized.

In order to control the test cone size for PE-BIST, shield canbe inserted into the interconnect structure to control the test time. We also present a post global routing track placement method to reduce shielding overhead. Simulation results show that the interconnect

signal integrity problem can be dealt with by PE-BIST with minimum shielding overhead and reasonable test time.

Finally, PE-BIST uses a parallel testing scheme and excites many aggressor nets to do the transitions which may lead to excessive power dissipation during testing. Power limit is usually considered in current SoC design, and thus the power dissipation for PE-BIST cannot be negligible. We use an efficient high level power modeling scheme to partition a PE-BIST solution into small child PE-BIST solutions so that each child PE-BIST solution can be tested within a given test power limit.

Acknowledgments

First, I would like to thank my advisor, Dr. Wen-Ben Jone, for all the effort and support he has given me throughout my thesis work. Without his guidances and supports this thesis would not have been possible. All his technical insight and motivation through my work have been very helpful.

I would like to thank Dr. Harold Carter, Dr. Henry Chen, Dr. Carla Purdy and Dr. Ranga Vemuri for being on my committee and for their efforts to review the draft of the thesis and valuable suggestions.

Special thanks to all my friends in the University of Cincinnati for all the memorable moments they gave me.

Finally, I dedicate this work to my father, mother and brother who have been a tremendous source of encouragement and love.

Table Of Contents

Chapter 1	1
Introduction.....	1
1.1 Interconnect Scaling Trend Overview	1
1.2 Interconnect Signal Integrity Testing.....	5
1.3 Our Work	9
Chapter 2	11
Background	11
2.1 What is signal integrity	11
2.1.1 Crosstalk Glitch Effect (CTG)	12
2.1.2 Crosstalk Delay Effect (CDE)	13
2.2 Capacitance	14
2.3 Inductance	17
2.3.1 Self Inductance and Mutual Inductance.....	17
2.3.2 Loop and Partial Inductance Models	19
2.3.3 Inductance Effects.....	21
Summary	25
Chapter 3	27
PE-BIST Basics	27
3.1 Pseudo-Exhaustive Testing Overview	27
3.2 Pseudo-exhaustive Interconnect Testing.....	29
3.3 Noise Simulation for RLCK interconnect.....	38
3.3.1 Interconnect Model	39
3.3.2 Simulation Results	41
3.3.3 Process Variation	44
Summary	45
Chapter 4	46
PE-BIST for Simple Interconnect Bus.....	46
4.1 PE-BIST Test Cone Determination	46
4.2 Serial Chain Interface	48
4.3 PE-BIST Test Architecture	49

4.4 Test Pattern Generation.....	50
4.5 Boundary Scan Cell Design.....	51
4.6 Test Process	53
4.7 Results and Discussion	55
Summary	57
Chapter 5	59
PE-BIST for Complex Interconnect Structure	59
5.1 Net Interference Graph	59
5.2 PE-BIST Test Cone Determination	62
5.3 PE-BIST Test Pattern Generation and Delivery	64
5.4 PE-BIST Test Architecture	70
5.5 Cast Study	71
5.6 Experimental Results	75
Summary	77
Chapter 6	78
PE-BIST Test Cone Size Control	78
6.1 PE-BIST Test Cone Size Reduction	78
6.2 Experimental Results	85
6.2.1 Experimental Results for RC Interconnects.....	86
6.2.2 Experimental Results for RLCK Interconnects	91
Chapter 7	97
Post Global Routing for PE-BIST	97
7.1 Over-the-cell Routing Model.....	97
7.2 PE-BIST Track Placement Problem	100
7.3 PE-BIST Track Placement Algorithm	101
7.4 Experimental Results	109
7.4.1 Experimental Results for RC Interconnects.....	111
7.4.2 Experimental Results for RLCK interconnects.....	114
Summary	116
Chapter 8	117
PE-BIST Power Aware Testing.....	117
8.1 Background.....	117
8.2 PE-BIST Power Budgeting.....	122

8.2.1 Three Wire Power Estimation Model	123
8.2.2 PE-BIST Power Budgeting Algorithm	125
8.2.3 PE-BIST Test Scheduling.....	131
8.3 Experimental Results	132
Summary	135
Chapter 9	136
Conclusion and Future Research	136
9.1 Conclusion and Contribution	136
9.2 Future Work	138
Bibliography	140

List Of Figures

Figure 1.1 VLSI technology scaling, ITRS Roadmap, Intel Corp.	2
Figure 1.2 Delay for metal 1, ITRS roadmap, Intel Corp.	3
Figure 1.3 Coupling capacitance vs total interconnect capacitance	4
Figure 1.4 Maximal Aggressor Fault Model	6
Figure 1.5 MT model for 3-line interconnect	7
Figure 2.1 Crosstalk: the aggressor and the victim	11
Figure 2.2 Crosstalk Glitch Effect	13
Figure 2.3 Crosstalk Delay Effects	14
Figure 2.4 Normalized Capacitance vs Normalized Line Spacing	12
Figure 2.5 Line to Line Capacitance vs Metal Spacing	16
Figure 2.6 Capacitance vs Conductor Length	16
Figure 2.7 Calculation of Mutual Inductance	19
Figure 2.8 Loop Inductance	20
Figure 2.9 Partial Inductance	21
Figure 2.10 Regions where inductance effect is significant	23
Figure 2.11 Mutual Inductance Decreases Slowly with Distance	24
Figure 2.12 Comparison of RC/RLC noise effects	25
Figure 3.1 Circuit Partition is the core for pseudo-exhaustive testing	29
Figure 3.2 RC Interconnect Peak Noise Decreases with Aggressor Distance	31
Figure 3.3 Current Return in Low and High Frequency	33
Figure 3.4 Current Returns in Nearest Ground in High-Speed Interconnect	34
Figure 3.5 Interconnect Structure With Shield Insertion	35
Figure 3.6 Effective Mutual Inductance with and without shielding	36
Figure 3.7 Locality in RLCK interconnects	37
Figure 3.8 An interconnect PE-BIST cone	38
Figure 3.9 A RLCK interconnect model	40

Figure 3.10 Victims and aggressor test cones	42
Figure 3.11 Peak noise impact and aggressor group number	42
Figure 3.12 Signal delay impact and aggressor group number	43
Figure 4.1 Test cone size calculation for RC interconnect bus	47
Figure 4.2 Test cone size calculation for RLCK interconnect bus	47
Figure 4.3 PE-BIST serial chain interface	48
Figure 4.4 PE-BIST test architecture	50
Figure 4.5 Test pattern delivery boundary scan cell	53
Figure 4.6(a) Test pattern delivery BSC dataflow	54
Figure 4.6(b) Observation BSC data flow	56
Figure 5.1 Example interconnect structure in two routing regions	60
Figure 5.2 Net interference graph for example interconnects in Figure 5.1	60
Figure 5.3 Merge of two regional NIGs into a global NIG	61
Figure 5.4 Logic structure for PE-BIST test pattern generation.	65
Figure 5.5 Test cone covers victim net with smaller order in NIG	68
Figure 5.6 PE-BIST test pattern channel assignment.	69
Figure 5.7 PE-BIST test pattern delivery	69
Figure 5.8 PE-BIST distributed test pattern generation	71
Figure 5.9 Example routing solution	72
Figure 5.10 RNIG of Example Routing	72
Figure 5.11 GNIG of Example Routing	73
Figure 5.12 Channel Assignment of Example Routing	73
Figure 5.13 PE-BIST of Example Routing	74
Figure 5.14 PE-BIST test cone size VS cut-off locality (Circuit xerox).....	76
Figure 6.1 Sample PE-BIST GNIG).....	79
Figure 6.2 Sample PE-BIST GNIG After Shield Insertion	81
Figure 6.3 Sample PE-BIST GNIG After Shield Insertion (Channel Overflow 1).....	79

Figure 6.4 Sample PE-BIST GNIG After Shield Insertion.....	83
Figure 6.5 PE-BIST test cone size control algorithm by shield insertion.....	85
Figure 6.6 Test coverage VS cut-off locality for RC interconnects.....	87
Figure 6.7 Shield insertion VS cut-off locality for RC interconnects.....	88
Figure 6.8 Test coverage VS test cone size for RC interconnects.....	89
Figure 6.9 Shielding overhead VS test cone size for RC interconnect.....	89
Figure 6.10 Test coverage VS test cone size for RC/RLCK interconnects.....	92
Figure 6.11 Shielding overhead VS test cone size in RC/RLCK interconnect.....	93
Figure 7.1 Global Routing	99
Figure 7.2 Two Layer Routing	99
Figure 7.3 An example GNIG	102
Figure 7.4 Extracted NIG from GNIG	103
Figure 7.5 An example GNIG (cont.).....	105
Figure 7.6 Final GNIG and channel assignment	106
Figure 7.7 PE-BIST routing algorithm	109
Figure 7.9 Shielding overhead vs test cone size for RC interconnects	113
Figure 7.8 Shielding overhead vs cut-off locality for RC interconnects.....	111
Figure 7.10 Shielding overhead vs test cone size for RLCK interconnects	115
Figure 8.1 Crosstalk effects on energy dissipation in interconnects.....	119
Figure 8.2 Crosstalk power dissipation for coupling interconnects	120
Figure 8.3 The effect of different types of transitions on energy	121
Figure 8.4 Three Wire Power Estimation Model	124
Figure 8.5 Sample PE-BEST GNIG with test power required for each net	127
Figure 8.6 GNIG is partitioned into two child GNIGs	129
Figure 8.7 PE-BIST GNIG power budgeting algorithm.....	131
Figure 8.8 PE-BIST test pattern delivery with power scheduling	132
Figure 8.9 Number of child NIGs vs test power limit (circuit xerox).....	134

List Of Tables

Table 3.1 Impact of parasitics variation on the validation of RLCK locality	45
Table 4.1 Area overhead of PE-BIST BSC	56
Table 4.2 Interconnect overhead VS test patterns for 64-bit bus	57
Table 5.1 List of MCNC benchmark circuit results	75
Table 5.2 MCNC benchmark circuits PE-BIST result.....	77
Table 6.1 Shield Insertion Cost Calculation.....	81
Table 6.2 Shield Insertion Cost Calculation.....	82
Table 6.3 List of MCNC benchmark circuits.....	85
Table 6.4 Shielding overhead with different cut-off locality.....	87
Table 6.5 test coverage VS test cone size for RC interconnect.....	88
Table 6.6 MCNC benchmark results (cut-off locality= 0)	89
Table 6.7 MCNC benchmark results (cut-off locality= 1)	89
Table 6.8 test coverage VS test cone size for RLCK interconnect	91
Table 6.9 Effect of Pre-existing Shields.....	94
Table 6.10 Effect of Net Sensitivity on PE-BIST Test Coverage.....	95
Table 6.11 MCNC benchmark results (cut-off locality=0, test cone size=30)	95
Table 7.1 List of MCNC benchmark circuits.....	110
Table 7.2 Shielding overhead with different cut-off locality.....	110
Table 7.3 Shielding overhead vs test cone size.....	112
Table 7.4 MCNC benchmark results with cut-off locality equal to 0.....	113
Table 7.5 MCNC benchmark results with cut-off locality equal to 1.....	113
Table 7.6 Shielding overhead vs test cone size for RLCK interconnects.....	114
Table 7.7 MCNC benchmark results with cut-off locality	114
Table 8.1 Three Wire Power Lookup Table.....	133
Table 8.2 List of MCNC benchmark circuits	133
Table 8.3 List of MCNC benchmark circuit results with test power limit.....	134

Chapter 1

Introduction

In 1965, Gordon Moore predicted that the number of transistors which can be integrated on a single die would double every two years [1]. This is held true since then and expected to continue until 2015 or 2020. The concept of CMOS scaling is to systematically downsize MOS transistors such that the new smaller devices are faster, more power-efficient, and reliable [2]. Advances in the complementary metal–oxide–semiconductor (CMOS) technology over the past 30 years have led to an explosion in the performance of integrated circuits (ICs). With circuit devices become smaller and circuit operation frequency goes higher, the signal integrity problem has become more and more important.

1.1 Interconnect Scaling Trend Overview

Figure 1.1 shows the technology roadmap [88] for interconnect technologies. As the frequency goes higher, the device size and switching delay have shrunk accordingly. At present, 45nm technologies are already in production and microprocessor clock frequency has exceeded 3 or 4 GHz. The speed of a signal in integrated circuits is governed by two factors: the circuit (transistor devices) delay and the interconnect delay. For many generations, the circuit delay has

been limiting the circuit performance. With fast advancement in technology, the interconnect delay has become dominant in current IC design.

Figure 1.1 VLSI technology scaling, ITRS Roadmap [88]

Figure 1.2 shows the ITRS roadmap for interconnect delay. This figure shows that as technology scales, the global interconnect delay will increase substantially. In quasi-ideal scaling, the vertical dimensions are scaled more slowly than the horizontal dimensions. This will lead to tall and narrow wires. With wires becoming taller and narrower, combined with the shrinking line widths and spacing in modern processes, the wires become not only taller but close to each other. As a natural consequence, it will dramatically increase the coupling capacitance in the DSM regime. Figure 1.3 shows the rise in coupling capacitance for scaled processes vs the total wire capacitance. The rise of coupling capacitance leads to more signal interference between wires.

The enhanced noise effects can degrade signal integrity and increase delay predictability. For this reason, the interconnect wire noise problem has become a more and more serious problem [3]. In fact, the ITRS roadmap has identified signal interferences as one of the most challenging design issues, in parallel with design productivity, power management, design for manufacturability, and reliable and resilient computing.

Figure 1.2 Delay for metal 1, ITRS roadmap [88]

As technology moves into the Giga Hertz era, higher operational frequencies and lower resistivity copper interconnects make the inductive impedance of an on-chip wire become comparable to or larger than the resistive impedance. At low speeds, current follows the path of the least resistance. At high speeds, the inductance of a given return current path may become more significant than its resistance. High-speed return currents follow the path of least inductance, not the path of least resistance. Therefore, inductance (L) can no longer be neglected as in the

low frequency interconnect design. Inductive coupling can occur over long distance, whereas capacitive coupling is limited to adjacent interconnects. As a result, the signal integrity problem becomes more complicated when inductive coupling is involved.

Figure 1.3 Coupling capacitance vs total interconnect capacitance [57]

Signal integrity testing deals with the issues of integrity loss, test pattern generation, and test sensing techniques. For interconnects, the fault model generally includes crosstalk, overshoot and electro-magnetic interference [4].

Signal integrity issues arise from the parasitic elements of long on-chip interconnects, especially the long interconnect trend in current SoC design products. Coupling capacitance, ground capacitance, resistance, self-inductance, mutual inductance all contribute to noise issues in long interconnects. A major noise source is crosstalk which causes signal glitch spikes or delay, and thus may kill a design. Several design techniques, including physical design and analysis

tools, have been developed to help design for margin and minimize crosstalk problems [58][59]. However, it is hard to anticipate in advance the impact of a full range of all possible process variations and manufacturing defects. Due to the complexity of the signal integrity problem, it is very hard to fix it in the design phase. Hence, there is a critical need to develop testing techniques for manufacturing defects that may produce crosstalk effects. We will focus more on crosstalk review in the next chapter.

1.2 Interconnect Signal Integrity Testing

In dealing with the signal integrity testing problem for RC interconnects, one of the most famous fault model is the maximum aggressor (MA) model [5]. In the MA model, the signal transition is generalized by a generic coupling component and the effect could lead to any of the four crosstalk errors on the victim line: positive glitch (g_p), negative glitch (g_n), rising delay (d_r) and falling delay (d_f). The basic idea of the MA fault model is to apply identical transitions to all wires except the victim line to create the maximal integrity loss in the victim line. Figure 1.4 shows the patterns for the four different crosstalk faults.

For example, by assigning rising (falling) transitions to all aggressor lines with the victim line remaining at logic 0 (1), the worst positive (negative) pulse on the victim line may be triggered. This fault model is powerful in dealing with interconnects at lower frequencies where inductance is negligible. Based on this fault model, a method has been developed to estimate the fault coverage of any general test set [6]; further, a self-test methodology has been developed to

enable on-chip at-speed testing of crosstalk defects in SoC interconnects [7]. The self-test methodology involves the insertion of test generators (to generate test patterns based on the MA fault model), error detectors, and a test controller in the SoC. To avoid excessive hardware overhead, a software-based self-test methodology based on embedded processors has been proposed in [8]. The basic idea is to enable an on-chip embedded processor core to test for crosstalk in system-level interconnects by executing a self-test program in the normal operational mode of the SoC. In [9], a low-cost self-test scheme called LI-BIST for SoC logic cores and interconnects has been proposed. LI-BIST reuses the existing LFSR structure for logic circuits, but generates high-quality tests for interconnect crosstalk defects. This is achieved by adding an extension circuit that modifies the LFSR vectors such that they have high interconnect defect coverage. Again, this work is developed based on the MA fault model.

Figure 1.4 Maximal Aggressor Fault Model [5]

In the MA model, only coupling capacitance is taken into account. However, the MA fault model fails to deal with long range, complex inductive coupling which is significant in current Giga Hertz designs. In [10], it has been found that there exist test patterns creating worse

delay and/or noise and causing more integrity loss compared to those generated by the MA model.

Quiescent at 0 x0x x0x	Transition 0 $\rightarrow$ 1 x0x x1x	Quiescent at 1 x1x x1x	Transition 1 $\rightarrow$ 0 x1x x0x
000 101	000 111	010 111	010 101
001 100	001 110	011 110	011 100
100 001	100 011	110 011	110 001
101 000	101 010	111 010	111 000

Figure 1.5 MT model for 3-line interconnect [46]

To take inductive coupling into account, multiple transition model (MT) has been proposed [46]. MT essentially is a superset of MA patterns. It covers all possible transitions on the interconnects to stimulate integrity losses. The problem of MT model is that it only takes very limited aggressors into account. For example, Figure 1.5 shows the test patterns for a three-line interconnect structure. It fails to fully consider the long range effects of inductive coupling. . In [60][61], the authors further reported that a device failed when the nearest aggressor lines change in one direction and the other aggressors change in the opposite direction. Due to the complexity of inductive coupling in high-speed RLCK interconnects, finding test patterns guaranteed to create the worst-case scenarios for integrity loss is almost impractical. It is even concluded in [11] that random test patterns are more qualified than those based on conjectured models to create the worst-case integrity test.

There are few test pattern generation methods targeting RLCK interconnects with full consideration of long range inductive coupling effects. One interesting attempt is to use an efficient simulation method to do test pattern generation [12]. To enhance the performance, model order reduction is applied to alleviate the computation complexity with slight loss of accuracy. Due to the complexity of a real circuit, it might be hard to apply this method to a large interconnect structure in the real circuit design. In [71], a high-level fault model based on interconnect topology is proposed to avoid complex Spice simulations. Boundary scan has also been adapted for signal integrity testing of SoC interconnects. New instructions have been developed and new noise detection cells were designed. Special receiver cells are designed to monitor the disturbed signals. These special cells are then extended to take skew noise detection into consideration [12][39][72][73]. Many test generation approaches [80-81][87] were developed assuming boundary scan DFT is available. These algorithms generate small test sets to detect all open and short faults on interconnect. BIST [82] [85] and oscillation ring [83] provide other alternatives for interconnect testing, but none of which can provide the worst case test patterns. In [84], interconnect test scheme is proposed to exploit circuit characteristics, inherent test resources in design, and test patterns of embedded cores to test interconnect. Recent work on deterministic test pattern generation and built-in self-test (using pseudo-random test patterns) for crosstalk fault detection of RC interconnects without considering the long range inductive coupling effect can be found in [62-65]. Recently, a couple of interconnect testing schemes are developed for MCM and Network-On-Chip (NoC) [111-115]. The above research works provide cornerstones for test pattern delivery and fault detection. However, due to the lack of an effective test pattern

generation method and a corresponding fault coverage analysis technique, the test quality cannot be guaranteed.

1.3 Our Work

It is very hard to generate test patterns for signal integrity faults of Giga Hertz RLCK interconnect. While random test pattern generation cannot guarantee the fault coverage, the proposed PE-BIST for interconnect testing can avoid deterministic test pattern generation. It is observed that not all signal lines around a victim interconnect are effective aggressors for the victim under testing. For example, for capacitive-dominated interconnects, only a few nearby signal lines are effective aggressors for a victim line. Those far away from the victim line have little or no noise effect on the victim signal. Any signal line has (and should have) a limited set of aggressors. Thus, we can reduce our test set to include only those effective aggressors. Pseudo-exhaustive testing fits into this scenario perfectly.

The aim of this research is to apply pseudo-exhaustive testing method to high-speed interconnects, to achieve both high test speed and quality, without worrying about test pattern generation. The PE-BIST test structure is simple, and can be integrated into existing boundary scan techniques in the SoC chip as well as a separate structure built with existing BIST components. There is no need to partition the circuit for the proposed pseudo-exhaustive interconnect testing. Hardware overhead is minimized by integrating PE-BIST into existing BIST circuits on chip or boundary scan testing. Test time can be controlled by shield insertion to existing interconnect structure. The shielding overhead can be further reduced by an efficient post global

routing track placement method. Simulation results demonstrate that the proposed method can achieve high fault coverage, small hardware overhead and reasonable test application time.

The rest of the thesis is organized as follows.

Chapter 2 reviews signal integrity issues and high-speed interconnects

Chapter 3 introduces the basis of PE-BIST and validates its application to interconnect testing.

Chapter 4 applies PE-BIST to simple interconnect bus structures.

Chapter 5 extends PE-BIST to complex interconnect structure.

Chapter 6 discusses shield insertion method to control PE-BIST test cone size

Chapter 7 provides a post global routing track placement solution to reduce the shielding overhead as well as control the PE-BIST test time.

Chapter 8 deals with how to limit power dissipation during PE-BIST testing. An efficient algorithm is proposed to partition one PE-BIST solution into a few child PE-BIST solutions to meet the test power requirement.

Chapter 9 concludes the thesis and provides insights for future research

Chapter 2

Background

As technology approaches deep sub-micron and clock frequency approaches Giga Hertz, the signal integrity problem of high-speed interconnects is becoming a more and more serious issue. In this chapter, we will discuss the origin of interconnect signal integrity issues and how they are being dealt with.

2.1 What is signal integrity

Signal integrity is the ability of a signal to generate correct responses in a circuit. The signal integrity problem causes a circuit to malfunction with the distortion of the signal waveform. There are two general forms for signal waveform distortion: (1) voltage level is out of normally defined range, and (2) signal transition time is out of designed timing range. Signal integrity problem in long interconnects largely arises from neighboring signal crosstalk with each other.

Figure 2.1 Crosstalk: the aggressor and the victim

Crosstalk is a noise coupling effect provoked by the parasitic capacitance among transmission lines. Let us consider two signals A and V driven by two CMOS inverter. As shown in Figure 2.1, there are both capacitive coupling and inductive coupling between these two wires. When the signal A makes a transition, a noise is injected on the signal V . If V is in a steady state, the noise has the form of a spike and is absorbed by the V 's driver after some delay. On the contrary, if V is making its own transition at the same time, the crosstalk noise leads to a shorter or longer transition delay. The signal A is called the *aggressor* and V the *victim*. Crosstalk degrades signal waveform. There are basically two kinds of crosstalk noise effects: crosstalk glitch and crosstalk delay.

2.1.1 Crosstalk Glitch Effect (CTG)

The CTG effect refers to the fact that one signal line's voltage level can be affected by another signal line (aggressor) which is doing transition. The aggressive line switches with transient current, which induces current to flow through the coupling capacitor between the victim and aggressor lines. This current then makes a noise peak at the victim line. Depending on the switching direction, it is either an overshoot peak or undershoot peak. This peak noise subsequently affects the transistors which the victim line connects to, potentially make unnecessary transitions. Only the overshoot peak on low logic value and the undershoot peak on high logic value can cause logic errors at the next logic gates of the victim line.

Figure 2.2 Crosstalk Glitch Effect

Figure 2.2 shows two coupled signals that crosstalk with each other. When A is making a transition, signal B exhibits noise glitch pattern. A data glitch is generally defined as an undesired transition or bounce in a signal. Glitches destroy the integrity of signals on a line. Data glitches can cause corruption of data if the glitch occurs at the sampling time at the receiver, therefore corrupting future data.

2.1.2 Crosstalk Delay Effect (CDE)

Due to the interactions among the transmission lines, the propagation of the signals may be delayed significantly. The delay due to the cross talk effect is referred as crosstalk delay effect. The crosstalk delay is caused by the coupling interference among the lines. Figure 2.3 shows an example of the crosstalk delay effect. When there is coupling capacitance, the output waveform is distorted and delayed in time as illustrated in Figure 2.3.

In digital systems, digital data is transmitted by continuous analog signal waveforms. On the receiving end, signal sampling is applied to obtain the encoded digital codes. Usually, the

sampling point is at the rising edge or the falling edge of a clock signal. The data must arrive and settle down within limited time period in the sampling window to reach a non-ambiguous logic state. Long signal delay will cause the wrong logic value on the receiving end thus cause signal integrity problems.

Figure 2.3 Crosstalk Delay Effects

One of the major causes of crosstalk is coupling parasitics. In the following discussions, we will focus on parasitics like capacitance and inductance and discuss how they produce signal crosstalk.

2.2 Capacitance

Coupling capacitance is defined as the capacitance between two wires and is affected by several factors. Previously, researchers have developed equations to relate the magnitude of coupling capacitance to geometry parameters such as metal thickness (T), width (W), spacing(S), and dielectric layer thickness (H) using existing models [13]. In this section, we will discuss

several important aspects of coupling capacitance which dominates the crosstalk in RC interconnects.

One of the most important factors for coupling capacitance is the wire spacing. As wire spaces are scaled smaller, the line-to-line capacitance and the line-to-ground capacitance will be affected, but differently. Figure 2.4 shows the change of both ground capacitance and coupling capacitance with respect to line spacing [57]. Although line-to-ground capacitance decreases slightly, line-to-line capacitance increases rapidly when line space decreases.

Figure 2.4 Normalized Capacitance vs Normalized Line Spacing [57]

Figure 2.5 Line to Line Capacitance vs Metal Spacing [57]

Figure 2.6 Capacitance vs Conductor Length [57]

While the metal spacing increases, the capacitance between lines will decrease (See Figure 2.5). This also indicates that the crosstalk is usually more serious in adjacent wires since adjacent lines are much closer to each other [91].

Not only does the line spacing affect the capacitance, but also the conductor length affects. As the conductor length increases, the capacitance between two coupled lines will increase. This complies with the equations for calculating the capacitance: the capacitance is proportional to the area of the conductor. Figure 2.6 show the relationship between coupling capacitance and conductor length [92][93].

In summary, the line to ground capacitance is not affected much by the line spacing, but the line to line capacitance is greatly influenced by the line to line spacing and line length. As capacitance increases, the crosstalk noise will increase accordingly. Long parallel wires close to each other are more likely to create serious crosstalk noise. This results in greater risk for crosstalk in global interconnects.

2.3 Inductance

2.3.1 Self Inductance and Mutual Inductance

A wire with electric current or changing electric field creates magnetic fields. This relationship is represented by Ampere's law [14] in the following equation:

$$\nabla \times B = \mu J + \mu \varepsilon \frac{\partial E}{\partial t}$$

where μ is the magnetic permeability and ϵ is the electric permittivity of the material. The first term on the right-hand represents a magnetic field generated from electric current. The second term is the magnetic field generated from the changing electric field.

The integral form of Ampere's law based on Stokes' law is as follows:

$$\oint_C B \cdot dl = \mu \int_S \left(J + \epsilon \frac{\partial E}{\partial t} \right) \cdot ds$$

where C is the contour that encloses the surface S.

Faraday's law states that a time-varying magnetic field creates an induced electric field, which is represented as follows:

$$\nabla \times E = -\frac{\partial B}{\partial t}$$

The integral formulation of Faraday's law is presented in the following equation [89]:

$$\oint_C E \cdot dl = -\frac{\partial \int_S B \cdot ds}{\partial t}$$

With those two equations, we can derive the equations for inductance. Consider the circuit in

Figure 2.7, the flux linkage enclosed in contour S2 is the following:

$$\Psi_{12} = \int_{S_2} B_1 \cdot ds_2$$

The *self inductance*, L11, is defined from the magnetic flux produced by I1 enclosed by the

contour S1 (Figure 2.7) as:

$$L_{11} \equiv \frac{N_1 \Lambda_{11}}{I_1} = \frac{N_1^2 \Psi_{11}}{I_1}$$

The *mutual inductance* M12 between two loops is defined as:

$$M_{12} \equiv \frac{N_2 \Lambda_{12}}{I_1} = \frac{N_1 N_2 \Psi_{12}}{I_1}$$

where N1 (N2) is the number of turns in each circuit. We can see that the mutual inductance is a measure of conductor interaction to link magnetic flux.

Figure 2.7 Calculation of Mutual Inductance

2.3.2 Loop and Partial Inductance Models

Inductance is a physical property of a closed current loop. Loop inductance is defined as

the induced magnetic flux in the loop by the unit current in the other loop (Figure 2.8). The loop model includes the current return path information in a single inductor and resistor.

Figure 2.8 Loop Inductance

In the loop model, the current return path needs to be determined. However, in a real chip environment, there are multiple return paths available, so it is complex to define a closed current loop. Since the current return path is frequency-dependent due to the proximity effect, the loop inductance and resistance model are also frequency-dependent.

When the current return path is not known a priori, the Partial Element Equivalent Circuit (PEEC) model [15] can be used. In the PEEC model, the return path is assumed to be infinite for each conductor segment. The partial self-inductance for each wire segment and mutual inductance between segments are determined. Figure 2.9 shows the concept of infinite return path. Each conductor segment is marked as green and forms a loop with an infinite return path.

This fully coupled partial RLC model is then put into a circuit simulator like HSPICE to determine the current return path. The problem with PEEC model is that it may result in a huge number of circuit elements, a dense inductance matrix which in turn leads to excessive simulation

time [16].

Both loop inductance model and partial inductance model have their merits. In this work, we use both whenever it comes to our convenience. Mostly, loop inductance is used to analyze the inductive effects and used to simulate circuits when return current is determined as priori. Partial inductance is used to simulate circuits without the return path information. When we use HSPICE to simulate circuits, we use the partial inductance model.

Figure 2.9 Partial Inductance

2.3.3 Inductance Effects

Impedance is defined by the following equation:

$$Z = R + j\omega L$$

Where ω is determined by clock edge $\omega \sim 1/(\text{rising time})$. The real part R is resistance while the imaginary part is reactance. Reactance arises from the presence of inductance and capacitance. As the clock frequency grows fast, the reactance becomes larger for on-chip

interconnections.

Inductance is directly related to the number of magnetic field lines around a conductor per amp of current. Inductance has the effect of opposing an instantaneous change in current [17].

Ismail [18] defined the rules to characterize the importance of on-chip inductance based on the signal rise time and the line damping ratio. The significance of inductance is specified by the following equation:

$$\frac{t_r}{2\sqrt{LC}} < l < \frac{2}{R}\sqrt{\frac{L}{C}}$$

They derived a region graph to show in which region inductance will become significant (Figure 2.10). In summary, two factors determine if the inductance effects are significant. The first is high frequency, so that the inductive part of the line impedance becomes comparable to the resistive part. A good measure of this frequency, f , is known to be $0.34/t_r$, where t_r is the input signal rise time [19]. The second condition is low damping, or low signal attenuation [20]. The damping ratio can also be interpreted as the ratio of RC delay over LC delay. In today's interconnect technology, inductance effects occur predominantly on global interconnects. The number of global interconnects is much less than that of local interconnects [47]. Even among the global interconnects, only a small number of them show inductance effects.

An important property of coupling inductance is that it is a long range effect. Unlike capacitance, it decreases slowly with distance. Figure 2.11 shows the loop inductance obtained from a three dimensional electromagnetic solver, called FastHenry [21]. The plot is obtained based on a co-planar interconnect structure where all wires are 0.8 μ m wide, 2 μ m tall and 2000 μ m

long, with the wire space equal 0.8 μ m. The inductance of wire number 1 is the self-inductance, and other wire numbers are the mutual inductance between the left most wire and other wires. It is clearly shown that the mutual inductance decreases slowly with wire number. It cannot be ignored when compared to the self inductance.

Figure 2.10 Regions where inductance effect is significant [47]

Another important property of inductive coupling is that the noise effect of inductance is far more complicated than capacitance. The signal waveform effect with inductance present includes ringing, reflections and simultaneous switching noise. Figure 2.12 shows a comparison of RC and RLC difference in noise effects. The figure is derived from simulation of a circuit where a CMOS inverter ($W_p/W_n=250/100$) drives a RLC interconnect segment ($R=30\Omega$, $L=5nH$, $C=1.4pF$) with a signal rising/falling time equal to 10ps. Clearly, the inductance effects cause a signal waveform with both ringing and overshoot.

Figure 2.11 Mutual Inductance Decreases Slowly with Distance [90]

In summary, inductance is a physical property of a closed current loop. Inductive coupling is essentially a conductor interaction to link magnetic flux. Without nearby current return, the inductive coupling is a long range effect and decreases slowly with distance. The inductive coupling effect is far more complicated than the capacitive coupling effect in circuit analysis. Noise effects of inductive coupling are far more unpredictable. Inductive effects become significant when the circuit frequency goes higher. In modern high-speed interconnects, inductive effects are no longer negligible in current high-speed interconnects.

Figure 2.12 Comparison of RC/RLC noise effects

Summary

In this chapter, we first defined the signal integrity problem in modern high-speed interconnects. Two basic signal integrity effects are explained: crosstalk glitch effect and crosstalk delay effect. Then, we discussed two important parasitics which introduce signal integrity: capacitance and inductance.

We explored the importance of coupling capacitance in inducing the crosstalk noise effect. Various factors affecting capacitance are discussed. Two parallel long wires which are close to each other will have a large line-to-line coupling capacitance, and may result in a serious

crosstalk. Inductance becomes important when circuit speed becomes higher. Inductive coupling is a long-range effect. It exhibits complex noise behavior without nearby current return path.

Both capacitive coupling and inductive coupling pose challenges to current signal integrity. The background about both coupling effects established a solid background for our PE-BIST in the latter chapters.

Chapter 3

PE-BIST Basics

3.1 Pseudo-Exhaustive Testing Overview

Circuit testing schemes can be divided into the following four categories in regard to the test pattern generation process.

Deterministic Testing depends on definitive test patterns. In circuit testing, it relies on software programs to analyze the circuit behavior, and then generate test patterns based on a certain fault model. Deterministic test pattern generation often has good fault coverage with an appropriate fault model. But, the test generation process is often time-consuming and the test pattern storage is space-consuming.

Random Testing is to apply random test patterns. The occurrence probability of each test pattern may not be distributed equally. If all test patterns are equally probable, it is said uniform random test patterns, otherwise, it is biased. In reality, all random test patterns are replaced with pseudo-random test pattern because they are much easier to generate, and have the equal quality in testing as those true random test patterns. The pseudo-random test patterns are often repeatable. The problem with random testing is that the test quality cannot be guaranteed.

Exhaustive testing is to apply all combinations of input test patterns to the circuit [22]. Exhaustive testing guarantees that all detectable faults that do not produce sequential behavior

will be detected. But, it may not be always feasible to use exhaustive testing. The reason comes from the fact that, depending on the clock rate, the number (called n) of inputs can be too large to be exhaustively exercised. For example, if the value of n equals 64 for a circuit, then it is very time-consuming to apply all 2^{64} test patterns for the circuit. The test virtually becomes impossible when the number of inputs of the circuit under test is large.

Pseudo-exhaustive testing is to apply pseudo-exhaustive test patterns to circuit inputs instead of exhaustive test patterns [23]. The rationale for pseudo-exhaustive testing is that often a circuit can be partitioned in a way that not all inputs are related to each other. This gives space to exploit and reduce exhaustive test patterns into pseudo-exhaustive test patterns. In Figure 3.1, Circuit core C1 and C2 have n_1, n_2 inputs respectively. Their outputs go through an AND gate to final output F. To exhaustively test C1, 2^{n_1} test patterns are applied to A while B maintain some value such than D=1. C2 is tested in a similar manner. The total test patterns are $2^{n_1} + 2^{n_2} + 1$ (AND gate is also tested). An exhaustive testing would require $2^{n_1 + n_2}$ test patterns. Such a testing is called pseudo-exhaustive testing. It depends on circuit logic partition, sometimes. The advantage of pseudo-exhaustive testing is that it needs only to exercise a (often very small) subset of otherwise exhaustive test patterns, but achieve the same fault coverage of exhaustive testing. Pseudo-exhaustive testing achieves many benefits of exhaustive testing, but usually requires far fewer test patterns.

Figure 3.1 Circuit Partition is the core for pseudo-exhaustive testing

3.2 Pseudo-exhaustive Interconnect Testing

Test patterns applied by conventional interconnect testing methods have been fallen into two categories we discussed in the last chapter. (1) deterministic test patterns [24], (2) random test patterns [25][79], precharacterized test patterns [25].

Unfortunately, as discussed, generating deterministic test patterns for high-speed RLCKinterconnects is generally very hard [12]. On the contrary, random testing for interconnects does not require any fault model, and test patterns are selected arbitrarily. As can be expected, random test patterns might not sensitize the worst case for interconnect coupling, and many faults will remain undetected. The disadvantage of precharacterized test patterns is that it might be very difficult to detect the worst-case noise using the precharacterized test patterns.

The basic idea of pseudo-exhaustive testing is to first partition the logic circuit such that the value of n (called cone size) can be controlled to a threshold, e.g., 20. Then, exhaustive test

patterns are applied to each output cone generated after circuit partitioning. If we consider interconnect structure as a circuit core and examine the coupling effects within different wires, we discover that the interconnect structure is partitioned naturally for pseudo-exhaustive testing.

The key point here is that the interconnect coupling effect generally must be localized to a block of neighboring aggressor lines. Basically, the capacitive coupling effect between interconnects is local, i.e., an interconnect suffers from the capacitive crosstalk effect only by its (immediately) adjacent interconnects. On the contrary, the value for coupling inductance between non-adjacent wires cannot be ignored, unless the aggressor lines are far away from the victim line. Fortunately, techniques such as shielding [27] and the interdigitated layout technique [28] can greatly reduce the inductive coupling effect for non-adjacent lines. Thus, the inductive coupling effect can be controlled local (e.g., the neighboring lines in both left and right sides of the victim line) as will be verified in the next section.

For signal integrity testing, it has been observed that not all signal lines around a victim interconnect are effective aggressors for the victim under testing [29]. For example, for a capacitive-dominated interconnect structure, only a few nearby signal lines are effective aggressors. Those far away from the victim line have little or no noise effect on the victim signal. Thus, we can reduce our test set from exhaustive test patterns to include only those effective aggressors. To gauge which aggressors are effective ones, we introduce the concept of *locality* as a measurement of an aggressor's significance. Due to different noise behaviors in RC and RLCK interconnects, we define locality differently from these two kinds of interconnects.

In low- and mid-range frequencies, capacitive coupling has been the major noise source for signal integrity problems. It is well known that capacitive coupling has local effects. The capacitive coupling effect between interconnects decreases substantially for non-adjacent lines. In a typical RC interconnect bus simulation, results show that we can test the victim line by exhaustively exercising only several nearest aggressors instead of all aggressors, without loss of the desired test accuracy. For RC-like interconnects, capacitive coupling decreases greatly with distance.

Figure 3.2 RC Interconnect Peak Noise Decreases with Aggressor Distance

Figure 3.2 shows the victim peak noise versus the aggressor number (e.g., the immediately adjacent line of the victim is numbered 1) in a typical RC interconnect. We use only

one aggressor in the simulation while keeping all other lines quiet. As we can see, the noise peak depends greatly on several nearest aggressors. The capacitive coupling noise is local, and it drops dramatically for the first three aggressors. In this case, we can test the victim line by exhaustively exercising only several nearest aggressors instead of all aggressors.

*Definition: **Locality in RC interconnects*** is the distance between an aggressor line and the victim line, expressed by the number of lines between the aggressor and the victim. For aggressors immediately adjacent to the victim line, the locality is zero.

As an aggressor is far away from the victim line, it becomes an ineffective aggressor and thus can be ignored in the test set for the victim line. Depending on the desired test accuracy, there is a milestone locality to decide which aggressors are effective ones. Beyond this locality, the aggressors can be considered insignificant for the test case. This locality is defined as the *cut-off locality*.

While at high frequencies, inductive coupling becomes no longer negligible. Research shows that inductive coupling can contribute more than 50% of the total noise at high frequencies [30]. Worst of all, the inductive coupling is no longer a short range effect. In fact, the inductive coupling decreases slowly in distance which results in a long range effect [27]. Crosstalk generally involves multiple coupled RLCK interconnects. There are so many aggressors involved and complex circuit equations become extremely expensive to solve in a given context.

Deterministic test pattern generation for coupled RLCK interconnects thus becomes very difficult.

Shield insertion is known as an efficient technique to reduce the inductive coupling between signal wires [75]. A shield is basically a metal line directly connected to ground. A shield can reduce inductive noise because it supplies a current return path for the signal wire, thus reducing the coupling between signal wires. A dedicated shield is generally considered as a good current return path in high-speed interconnect designs.

Figure 3.3 Current Return in Low and High Frequency

At high frequencies, it is well known that the return current of a signal line concentrates on only a few return grounds next to the signal wire to reduce the total impedance [31]. If multiple ground lines are present near a signal line, the return current distribution is frequency dependent. At low frequency, the return current is widely spread over multiple return paths to minimize the resistance. While at high frequency, the return current flows only through the closest return paths to minimize the inductance. This frequency dependent current return behavior

is called proximity effect. [32]. It has been suggested that no more than four nearest returns are sufficient to estimate the signal delay. [33]

When inductive effective is significant, the return current can be calculated as [34]:

$$[P]^T[L][P][I]=[V]$$

where $[P]$ is the mesh matrix, $[I]$ is the matrix of branch currents flowing through return paths, and $[L]$ is the partial inductance matrix. One calculated current return distribution is shown in Figure 3. 4. It is shown that most current returns of a signal line flow through its most adjacent four paths.

Figure 3.4 Current Returns in Nearest Ground in High-Speed Interconnect

The effective coupling inductance between two signals can be calculated, if the return currents of both signals are known. Consider a system with two signals where one is an aggressor

with current I_a and the other is the victim with current I_v . There is a set of current return paths, and the current distribution of this system can be represented by matrix $[I]$ ($[I]=[I_1, I_2, \dots, I_a, \dots, I_v, \dots, I_n]$), and $[Lp]$ is the partial inductance matrix. The loop inductance matrix can be derived by using the equivalence of magnetic energy stored in the system [24]:

$$\frac{1}{2}[I]^T[Lp][I] = \frac{1}{2}[I_a \ I_v][Leff][I_a \ I_v]^T$$

where

$$[Leff] = \begin{bmatrix} La & Meff \\ Meff & Lv \end{bmatrix}$$

Based on the above equation, the effective self inductance La (Lv) of aggressor (victim) is determined solely by its own current return paths, while the effective mutual inductance ($Meff$) is determined by both the aggressor's and victim's return paths.

Figure 3.5 Interconnect Structure With Shield Insertion

We calculated the mutual inductance between two signal wires using the above equation. Figure 3.5 shows the interconnect structure used in our analysis. In this work, each signal (shielding) line width is assigned 0.8 (1.6) μm , the space between signal and signal (shielding) is assigned 0.8 (1.2) μm , and all wires are 1mm in length, 2 μm in thickness. Figure 3.5 shows the

case of shielding pitch equal to 2, i.e., there are two signals enclosed by a pair of shielding lines.

Note that different pitches might be used for the interconnect structures discussed later.

Figure 3.6 Effective Mutual Inductance with and without shielding

Figure 3.6 shows the normalized effective mutual inductance between a pair of victim and aggressor for current returns through four nearest shields with shielding pitch equal to 4 in the interconnect structure. The mutual inductance of aggressor number one is set to 1, and all other groups are normalized according to aggressor number one. The return current distribution for the four return paths is assigned as 0.4-0.1 (two sides). To compare, we also show the partial mutual inductance without assuming any current return path in the interconnect structure. As we can see, the mutual inductance between a pair of victim and aggressor decreases very slowly without shields. But if shields are present in the interconnect structure, the effective mutual inductance decreases rapidly in the first block. Here, a *block* is defined as a group of wires in

between two shields. The effective mutual inductance shows a general trend to decrease along with distance (Distance is represented by line number counted from the victim in Figure 3.6).

In the next section, we are going to use HSPICE simulation to verify that the noise impact of a group of aggressor generally decrease with distance. Here, we can introduce a similar locality concept in RLCK interconnects.

Definition: Locality in RLCK interconnects is the group distance between an aggressor group and the victim line, expressed by the number of aggressor groups between the aggressor group and the victim. For aggressors in the same group as the victim line, the locality is zero.

Figure 3.7 Locality in RLCK interconnects

Figure 3.7 illustrates a typical RLCK interconnect structure with shield insertion. The aggressors are partitioned by shields into different groups. We mark two aggressor groups with the same group number if they are symmetrical to the group which the victim line resides in. The group of aggressors to which the victim belongs has a locality of zero.

Locality serves as a measurement of the influence on the victim line by an aggressor line in both the RC and RLCK cases. By choosing a cut-off locality, the entire exhaustive testing

space can be substituted by a set of effective aggressor test spaces without loss of the desired accuracy. That is, we can just apply exhaustive patterns of those effective aggressor lines to the victim line. Such a test set is called a *pseudo-exhaustive test set* of the victim line.

Figure 3.8 An interconnect PE-BIST cone

Figure 3.8 shows an interconnect PE-BIST cone, and the victim line is “V”. If each aggressor has two states (rising and falling transitions), all 2^{20} aggressor patterns are applied to both the left and right ten effective aggressor lines (each is marked by “A” in Figure 3.7) adjacent to victim line V. Instead of exercising full combinations of test patterns at all interconnects (except V), we apply exhaustive test patterns only to the cone of interconnects (except V) as shown in Figure 3.8. This is the reason why victim line V is called pseudo-exhaustively tested. The whole test setting (one victim line and 20 aggressor lines in Figure 3.8) is called a *test cone*. Thus, the test cone of a victim line contains the victim itself and its left and right effective aggressors.

3.3 Noise Simulation for RLCK interconnect

3.3.1 Interconnect Model

A real wire in an integrated circuit not only serves as a conductor but also introduces capacitive, resistive and inductive parasitic that can have a dominant influence on the circuit signal integrity. The situation is aggravated by the fact that the improvements in technology result in higher clock rates and longer interconnect wire length with worse parasitic effects. In low- and mid-range frequencies, the crosstalk capacitance of interconnect wires is the most important parasitic. As frequency of operation increases, inductance effects are becoming increasingly important in affecting the signal integrity.

Many distributed interconnect models have been proposed previously [35][36]. A full segmented RLCK interconnect model comprises resistance (R), ground capacitance (C_g) and partial self inductance (L) for each wire segment, mutual inductance (M) and capacitance coupling (C_c) between wire segments. The model comprises resistance (R), partial self inductance (L) and ground capacitance (C) for each wire segment. There is a coupling capacitance between any two parallel adjacent wire segments, while a coupling inductance exists between any two wire segments. This is also called a full RLCK model [37].

To avoid the loss of precision in simulation, the number of segments for extraction is determined as [38]:

$$number\ of\ segments \geq \frac{10 \times l}{Tr \times v}$$

where l represents the interconnect length, T_r denotes the signal rising time, and v is the velocity of the EM wave traveling along the metal line that carries the signal.

Figure 3.9 A RLCK interconnect model

In our simulation, we consider parallel coplanar interconnect structures where all wires have the same length. We use a PEEC-based extraction tool to extract the inductive parasitics for each wire segment [39]. The length of each segment strictly satisfies the precision of simulation (250 μ m segment is used in our simulation, if not otherwise stated). The capacitance parasitic is extracted only between two adjacent lines, while the mutual inductance is extracted for every two segments (refer to Figure 3.9). Each signal (shielding) line width is assigned 0.8 (1.6) μ m, the space between signal and signal (shielding) is assigned 0.8 (1.2) μ m, and all wires are 1mm in length, 2 μ m in thickness. The shielding pitch is set to 4. We use a PEEC-based extraction tool to extract the inductive parasitics for each wire segment. Each segment is 250 μ m in length, strictly satisfying the precision of simulation. The rising time is set to 100ps in HSPICE simulations. We also used TSMC 0.18 μ m technology with a 30x driver size and 10x receiver size [94].

3.3.2 Simulation Results

Two kinds of simulations are conducted in this section. For signal glitch testing, we keep the victim line quiet and detect the maximum peak noise induced on the victim line. For signal delay testing, we keep the victim line switching and detect the maximum signal delay induced by aggressors.

We conducted several HSPICE simulations to test the noise impact of each aggressor group. We assign all aggressors in the same block as the victim with group number one. The groups adjacent to group one in both sides are group number two, and so on. (Figure 3.10 shows an example of aggressor group number assigned according to victim 1.) The *noise impact* by a group of aggressors is defined as the difference between the maximum and the minimum signal distortions (either glitch peak or signal delay) induced by the group of aggressors under study. For example, if we want to find the noise impact of group two on the victim with a group size of four, we will first fix a switching pattern for all other wires except the group under study. Then, we exhaustively apply all switching patterns for group two aggressors (which is $2^4=16$ transition patterns). For each fixed pattern combination of other group aggressors, we can get one noise impact of group two aggressors on the victim line. The same process is repeated by changing the fixed (switching) pattern combination of other aggressor groups exhaustively. Finally, the noise impact of group two can be obtained. The noise impact of a group is an upper bound of possible noise contributed by the group of aggressors.

Figure 3.10 Victims and aggressor test cones

Figure 3.11 Peak noise impact and aggressor group number

Figure 3.11 shows the normalized noise impact in peak noise to the victim by each group of aggressors, while Figure 3.12 shows that for the signal delay. To compare, we also show the case without shield insertion. We assign every four aggressors as a group. The noise impact of group one is set to 1 and all other groups are normalized according to group one.

Figure 3.12 Signal delay impact and aggressor group number

Group one has the largest noise impact because all aggressors in group one reside in the same block as the victim. There are both strong inductive and capacitive coupling effects between the victim and the first group of aggressors. Without shield insertion, the noise impact decreases slowly with respect to aggressor group number. The noise impact of the sixth aggressor group is still significant. On the other hand, with shield insertion, the noise impact decreases rapidly after the first group. The aggressor group far away from the victim has much smaller effect on the noise of the victim line. The aggressor group noise impact exhibits a characteristic of locality. This will enable us to test the victim line by considering only a number of nearest aggressor groups instead of all aggressors.

3.3.3 Process Variation

The key point of PE-BIST is locality. In RLCK interconnect, if a current return path (e.g. shields) is provided, it can be used to isolate aggressors from victims, that is, to divide aggressors into different groups. By providing dedicated current return paths, the coupling effect (especially inductive coupling) by an aggressor group far from the victim will become negligible. Intuition tells that such a principle should stand even when there are certain variations in interconnect parasitics. In real interconnects, the parasitics may be variant due to process variations. In order to further validate the concept of PE-BIST, we randomly disturb all extracted (RLC) parasitics with $\pm 10\%$ variation to simulate the real process. To reduce the simulation time, we use a 12-bit bus. We run HSPICE simulations to find the noise impact in peak noise with and without shield insertion. The first line of the bus is set as the victim with a testing cone size $n=6$, and the shielding pitch is set to 2. Exhaustive transition patterns are exercised to test the worst victim peak noise. Then, we determine the difference between the worst case noise peak of the pseudo-exhaustive patterns and that of the exhaustive patterns. Totally, we run 100 simulations. In each simulation, every R, L, C component in the interconnect structure is varied between $+10\%$ and -10% with uniform distribution as described above. Table 3.1 shows the simulation results. As we can see, if we run without shield insertion, in 64% of the testing cases the pseudo-exhaustive switching patterns can be in the range of more than $1\%V_{dd}$ difference to the worst case (exhaustive switching pattern). If there are shield insertions, the situation is much more favorable. As shown in the second row, there are 100% test cases falling into the $0.1\%V_{dd}$

difference category. This indicates that even with process variation, proper shield insertion can still localize the RLCK noise impact on the victim by dividing aggressor into groups.

Difference Bus	<0.01% Vdd	<0.1% Vdd	<1% Vdd
No Shield	0%	6%	36%
Shielding Pitch=2	64%	100%	100%

Table 3.1 Impact of parasitics variation on the validation of RLCK locality

Summary

In this chapter, we first introduced popular test pattern generation methods for interconnect signal integrity testing. Exhaustive testing is impossible due to large interconnect bus size. Random testing can not guarantee the test quality. While inductive coupling is significant, deterministic test pattern generation becomes virtually impossible. Pseudo-exhaustive is a natural choice for interconnect due to the fact that the interconnect noise impact is localized. We discussed noise behavior of both RC and RLCK interconnects. RC interconnect exhibits locality naturally. RLCK interconnect exhibits locality when there is nearby current return path. The discussion of locality forms the foundation of PE-BIST testing scheme.

Chapter 4

PE-BIST for Simple Interconnect Bus

In the last chapter, we covered all basic PE-BIST principles and showed how the crosstalk noise in a RC or RLCK interconnect exhibits *locality* in terms of distance between aggressor and victim. In this chapter, we will apply the proposed PE-BIST method to simple interconnect bus structure and integrate it with existing boundary scan techniques.

4.1 PE-BIST Test Cone Determination

As we have discussed in the last chapter, for a RC interconnect, the noise impact of an aggressor x to the victim y drops substantially with the distance between x and y . While for a RLCK interconnect, the noise impact of an aggressor group g to the victim y drops substantially with the distance between g and y . An aggressor group is defined as a group of aggressor enclosed by a current return structure (e.g. shielding). There is a cut-off locality by which we draw a line between effective aggressors and non-effective aggressors.

Theorem 4.1: For a RC interconnect bus, if the cut-off locality is k , the minimum PE-BIST test cone size is $2k+3$

Proof: As we can see from Figure 4.1, a test cone must include all aggressors within cut-off locality. That is, the test cone must include left $k+1$ aggressors, right $k+1$ aggressors and victim net itself. So the minimum test cone size is $2k+3$.

Figure 4.1 Test cone size calculation for RC interconnect bus

Figure 4.2 Test cone size calculation for RLCK interconnect bus

Theorem 4.1 For a RLCK interconnect bus with a regular current return pitch p , if the cut-off locality is k , the minimum PE-BIST test cone size is $(2k+1)p$

Proof: As we can see from Figure 4.2, the test cone must contain all aggressor groups within cut-off locality. For cut-off locality k , there are k adjacent aggressor groups involved plus the aggressor group the victim net resides in. So the total test cone size is $(2k+1)p$.

4.2 Serial Chain Interface

Interconnect bus is a regular structure. The test pattern has symmetry for the individual victim line across the whole bus. That is, a test pattern for line 1 can be reused to line 2. This naturally leads to a serial chain test pattern delivery. The test patterns are shifted in the interconnect bus. Each test pattern shift operation will test next victim in the bus.

Figure 4.3 PE-BIST serial chain interface

When a test pattern is being generated, simultaneously, it is shifted into the serial scan chain bit by bit. Figure 4.3 shows the serial interface when pseudo-exhaustive patterns are shifted in and out through a segment of the bus. For the illustration purpose, the pseudo-exhaustive cone size is set to 8 in this figure. We omit possible shields which stay grounded all the time, and do

not influence the cone size. As we can see, when a test pattern is shifted through the interconnects, all lines will be tested by that test pattern. The distance between two victims is exactly the test cone size n . As pseudo-exhaustive patterns are generated and shifted into the serial interface, each victim (e.g., victim 1) will be exhaustively tested with all combinations of transitions in the cone of aggressor lines. In this figure, two lines (driven by black cells) are tested simultaneously by patterns a and b. In the next testing cycle, both patterns will be shifted one step right, and two new victim lines will be tested by test patterns a and b. As can be imagined, after one test pattern has been shifted from the scan chain input to the scan chain output, all wires driven by the scan chain will be tested by this test pattern. Best of all, many wires are being tested simultaneously, which will greatly reduce the test time. In fact, the test time is in the order of pseudo-exhaustive testing cone size, orderly independent of the numbers of wires in the system as discussed later in this chapter.

4.3 PE-BIST Test Architecture

Figure 4.4 illustrates the test architecture of PE-BIST. The test architecture consists of self-test structures like a Test Pattern Generator (TPG), a Test Controller (TC).

The TPG generates test vectors for the serial scan chain, while the TC controls the entire test process. It gives control signals to all the test structures and is also responsible for seeding the TPG. The entire PE-BIST architecture can be integrated with existing boundary scan systems with little overhead. The pseudo-exhaustive patterns are shifted into the serial chain, and then applied to the interconnects. The responses are collected into the receiver cells.

Figure 4.4 PE-BIST test architecture

4.4 Test Pattern Generation

After determining the PE-BIST cone size, a LFSR can be used to generate the pseudo-exhaustive patterns under that cone size. As shown in Figure 4.4, the TPG circuit for interconnect PE-BIST contains a LFSR, an aggressor signal generator, a victim signal generator,

and a PE-BIST controller. The LFSR is used to generate 0 and 1 logic values except the all-zero test pattern to satisfy the pseudo-exhaustive criteria. The all-zero pattern will be generated by the PE-BIST controller as a special case. The LFSR works under half of the testing clock frequency. For every two test clock cycles, the LFSR generates one bit. Upon receiving one bit from the LFSR, the aggressor signal generator (ASG) will issue either a rising or falling transition (a two-pattern pair in two test clock cycles), which is instilled into the serial scan chain. The victim signal generator (VSG) is used to generate signals for the victim interconnect, and the signals include Q1, Q0, rising and falling. Note that Q1 (Q0) signal can be generated by giving two consecutive logic 1's (0's).

The PE-BIST controller coordinates the entire test pattern generation process. It will shift in the first half cone of aggressor patterns to the scan chain. Then, it will trigger the VSG to shift in the victim patterns into the scan chain. Finally, it will shift in the other half cone of aggressor patterns into the scan chain. The victim pattern is sandwiched by these two groups of aggressor patterns.

4.5 Boundary Scan Cell Design

For signal integrity test, a valid test pattern is a test pair that can exercise the circuit switching activity. Each interconnect line must exercise one transition for each test pattern, so two bits are involved. The conventional boundary scan design can be used to accommodate the signal integrity testing pairs. Basically, the first and the second bits are scanned into a

conventional BSC and stored in FF2 and FF1, respectively. By using UpdateDR, a transition can be applied to the interconnect. While scanning and applying test bits are quite straightforward in conventional boundary scan techniques, it requires a large number of clocks which increases test time.

We designed a special BSC which is suitable for our proposed PE-BIST as well as for general signal integrity testing. The test pattern delivery BSC architecture is shown in Figure 4.5 . One extra control signal SI_TEST is added. The BSC can work in two modes:

- 1) **Normal BSC mode:** when SI_TEST=0, the test pattern delivery BSC becomes a standard BSC.
- 2) **Signal Integrity Testing mode:** when SI_TEST=1, the clock of FF2 is driven by ClockDR and FF2 output is selected for SOUT

The PE-BIST architecture will work when the **Signal Integrity Testing mode** is activated, and ShiftDR=1 and ClockDR jointly shift the serial test bits into each boundary scan cell. After a test bit enters FF1, it will continue to be loaded into FF2 in the next clock cycle because FF2 is now clocked by ClockDR instead of UpdateDR. This way, we can apply an effective pair of test bits for signal integrity testing of each interconnect.

To test a signal integrity fault, we also need to customize the conventional BSC for test observation. Many works have been published in this field [39]. Once the detector cell detects a signal integrity fault, it will set the fault bit in the following FF. After test patterns are applied

(either partial or all), the fault bit stored in the FF can be scanned out just like normal boundary scan testing. Note that our pseudo-exhaustive test method can easily adapt to different fault detection cell by simply integrating the detection cell into our test architecture.

Figure 4.5 Test pattern delivery boundary scan cell

4.6 Test Process

In the **Signal Integrity Testing mode**, the testing data flow is shown in Figure 4.6. In Figure 4.6(a), a test pattern delivery BSC receives pseudo-exhaustive patterns from either its previous BSC or the TDI. The data flows into two different directions after FF2. Not only is each test pattern asserted to the interconnect, but also forwarded to the next BSC. The data asserted to the interconnect will be captured by the observation BSC (Figure 4.6(b)). The observation BSC can also transparently forward the PE-BIST test data to the next BSC. There is no interference

between shifting the test patterns and detecting the signal integrity loss. As shown in Figure 4.6(b), these are two different routes. There is no need to change boundary scan instruction or state during the testing pattern shifting process. The detected signal integrity fault is flagged in the FF, and will be scanned out after test patterns are applied.

Figure 4.6(a) Test pattern delivery BSC dataflow

To integrate PE-BIST with existing boundary scan methods, we add a new instruction: EX-PEBIST. This instruction is very similar to the existing EXTEST instruction with only one added control signal SI_TEST. When the tap controller IR is loaded with EX_PEBIST, the PE-BIST controller will generate pseudo-exhaustive test patterns and shift them into the scan chain (by ShiftDR=1, ClockDR, and SI-TEST=1). The signal integrity loss detection cell in the

observation BSC will capture the signal integrity fault at the other end of each interconnect. After the testing process is completed, the results stored in each FF will be scanned out to see whether the interconnects are defective.

4.7 Results and Discussion

The PE-BIST boundary scan cell is evaluated by Synopsys design analyzer. The area overhead of a BSC used in PE-BIST is listed in Table 4.1. As we can see, the average area overhead is about 34%. Table 4.2 shows the interconnect overhead due to shield insertion versus the number of required pseudo-exhaustive test patterns. The bus width is set to 64 bits. We can see that test patterns can be greatly reduced by introducing more shields in the interconnect structure. But if shields are already present in the interconnect structure, the overhead induced by PE-BIST can be further reduced.

Due to the shift-and-shoot testing architecture, PE-BIST testing is finished once the last test pattern is shifted out of the serial chain. If the serial chain length is m and the pseudo-exhaustive cone size is set to n , then the test time is about $(4(n+1)2^n + m) \times 2$. The reasons are: (1) the number of exhaustive (aggressor) test patterns with cone size n is 2^n ; (2) the number of victim test patterns is four (i.e., Q0, Q1, low-to-high, and high-to-low); (3) for each test pattern, the number of transitions generated by ASG and VAG is $(n+1)$ where ASG (VSG) generates n (1) bits; and (4) each transition requires two clock cycles. Note that m in the above test time equation is the number of shifts for the last test pattern to go through the entire serial test scan

chain. In fact, it should be $(m-n)$ but we simplify it to m since $m \gg n$ in most cases. When $(n+1)2^n \gg m$, the test time is only related to pseudo-exhaustive test cone size. Thus, the test time of PE-BIST is almost independent of the serial chain length (in other words, the bus width), and it can be applied to very wide buses that contain many interconnects.

Figure 4.6(b) Observation BSC data flow

Table 4.1 Area overhead of PE-BIST BSC

Cell \ Test	STD (nand)	PE-BIST (nand)	Area Overhead
Delivery BSC	26	32	23%
Observation BSC	26	38	46%
Average	26	35	34%

Table 4.2 Interconnect overhead VS test patterns for 64-bit bus

Shielding Pitch	Test Patterns	Interconnect Overhead
2	128	48%
4	8192	23%
6	524288	15%
8	33554432	11%

Compared to exhaustive testing, PE-BIST uses much fewer test patterns while achieving similar results. When compared with random testing, the fault coverage of PE-BIST is guaranteed with proper shield insertion. The patterns generated by the popular MA fault model cannot apply well to signal integrity faults of RLCK interconnects [40]. On the other hand, considering only a few aggressors in the vicinity of victims without proper shield insertion structure will generate a big loss in testing accuracy, if the inductive effect is strong in the interconnects. For most testing schemes, the test time is dependent on the bus size. PE-BIST achieves good parallelism with its innovative architecture. In fact, the test time is orderly independent of the bus size, and can be arbitrarily controlled by a powerful shield insertion scheme.

Summary

The complexity of interconnect RLCK models makes it very difficult to generate deterministic test patterns for noise defects. Random patterns cannot guarantee the fault coverage and the worst case signal integrity defects. Upon observing that the interconnect coupling effect

can be controlled local with a proper shield insertion scheme, PE-BIST technique is quite efficient to detect signal integrity faults.

In this chapter, we applied PE-BIST to simple interconnect bus structure. The PE-BIST test architecture is very simple and can be shared with existing BIST testing structure. The test pattern can be easily through serial scan chain which can be integrated with boundary scan testing technique. The test application time can be well controlled by test cone size which can be controlled by crosstalk localization technique (e.g. shield insertion). The hardware cost of PE-BIST is very low because it can utilize existing testing structures. In this way, PE-BIST is very suitable for testing buses of modern SoC designs.

Chapter 5

PE-BIST for Complex Interconnect Structure

The concept of pseudo-exhaustive built-in self-testing (PE-BIST) for crosstalk noises of high-speed interconnects has been discussed in Chapter 3. PE-BIST is a natural choice for interconnect noise testing due to the local property of capacitive and inductive noises. This local property results in the idea of *pseudo-exhaustive test cone* without circuit partitioning. The greatest benefit of pseudo-exhaustive testing is to achieve the highest test quality without the requirement of a fault model to generate deterministic test patterns. In Chapter 4, we apply PE-BIST to simple interconnect bus structure. In this chapter, we will extend PE-BIST to arbitrary interconnect structure without the assumption of a simple interconnect bus structure.

5.1 Net Interference Graph

For an arbitrary interconnect structure, a long signal line may span different routing regions, and the aggressors to this line are distributed. To record the relationship between a victim net and its aggressor, we introduce the *Net Interference Graph (NIG)*.

Each node of a NIG represents a signal net. If two nets constitute a potential aggressor-victim pair, there is an edge connecting them in the NIG.

To account for aggressors in a region, we use a *Regional Net Interference Graph* (RNIG) to record the effective aggressors in the routing region. After we get all the RNIGs for different regions, we then combine them to form the *Global Net Interference Graph* (GNIG). The test cone size can then be deduced from the GNIG.

Only when two nets are within the cut-off locality, are they considered as an aggressor-victim pair. Accordingly, there is an edge connecting these two aggressor-victim pair nets in the corresponding NIG.

Figure 5.1 Example interconnect structure in two routing regions

Figure 5.2 Net interference graph for example interconnects in Figure 5.1

Figure 5.3 Merge of two regional NIGs into a global NIG

Figure 5.1 (a) and Figure 5.1 (b) show two interconnect structures. Figure 5.2 (a) and Figure 5.2 (b) show their corresponding (regional) NIGs. For simplicity, we do not define any cut-off locality. Instead, we assume any two nets form an aggressor-victim pair if there is no shield between them. Based on this criterion, in Figure 5.1(a), all signal lines are potential aggressor-victim pairs. So, Figure 5.2 (a) shows that each node is connected to all others to represent their aggressor-victim relationships. While in the case of Figure 5.1 (b), signal nets are segregated by shields, and nets in different shielding blocks are not aggressor-victim pairs by our

criterion. This results in that nets 4, 5 constitute an aggressor-victim pair, while nets 4, 3 do not. The corresponding NIG can be given in Figure 5.2 (b). Note that in Figure 5.2, each NIG is constructed from a specific routing region, and is called a *regional* NIG (RNIG).

While a net may span multiple routing regions, to identify all its potential aggressors, all RNIGs must be combined to form the GNIG. This combination is a simple addition of all RNIGs. Figure 5.3 shows an example of the GNIG formed from two RNIGs. The weight of an edge in the GNIG is the total number of times the edge appearing in all regional NIGs. For example, nets 3, 4, 5 span both routing regions 1 and 2. In the GNIG, each of edges (3, 4), (4, 5), and (3, 5) has weight equal to 2 (Figure 5.3), since those edges appear in both RNIGs.

5.2 PE-BIST Test Cone Determination

As we discussed in the last chapter, the effective aggressors of a victim line can be identified by setting its cut-off locality. Any aggressor with its locality (with respect to the victim line) less than the cut-off locality is considered as a significant aggressor, and thus should be included in the pseudo-exhaustive test cone of the victim. For a simple interconnect structure like a data bus, when the cut-off locality is set, the test cone size can be easily determined. For a RC interconnect bus structure, if the cut-off locality is set to n , the test cone size is $2n+3$ ($2n+2$ aggressors and one victim).

We have modeled the aggressor-victim relationship in last section by Net Interference Graph. For arbitrary interconnect structure, we define a net can be tested by pseudo-exhaustive patterns as long as the pseudo-exhaustive patterns cover all of its effective aggressors.

Theorem 5.1 The maximum vertex degree in the NIG is a lower bound of pseudo-exhaustive test cone size.

Proof: Any edge in the NIG implies an aggressor-victim relationship between two nets. To exhaustive test a victim, all its aggressors must be exhaustively tested. To ensure all victims are exhaustively tested within the test cone size, the minimum cone size must be able to test the net with maximum vertex degree. Thus, the maximum vertex degree in the NIG is a lower bound of pseudo-exhaustive test cone size

Once we have the GNIG formed, we can determine the effective aggressors for each victim net which in turn indicates the test cone size requirement. For example, to test net 1 in Figure 5.3, we need to exhaustively test its aggressor nets 3, 4 and 5. The number of aggressor nets for a victim equals to the degree of the victim vertex in the GNIG. To test the victim net, we must cover all its aggressors in the test set. For a node with a small degree, its test set is small. The PE-BIST test cone size must be large enough to cover all the aggressors of the node which has the maximum degree (called m). By doing so, the test set will also be able to cover all those nodes with degree less than m . Consequently, the maximum degree of all vertices in the GNIG constitutes the minimum test cone size required by the PE-BIST scheme.

5.3 PE-BIST Test Pattern Generation and Delivery

As shown in Figure 5.4, the test pattern generation circuit for interconnect PE-BIST contains an LFSR-driven aggressor signal generator, a victim signal generator, a test pattern mixer, and a PE-BIST controller. The LFSR is used to generate all combinations of 0 and 1 logic values except the all-zero test pattern. The *aggressor signal generator* (ASG) will issue either a rising or falling transition upon excitation from the LFSR. For example, an LFSR bit with logic 0 (1) will generate a rising (falling) transition in the corresponding aggressive line. Since an LFSR can generate all combinations of 0 and 1 for all LFSR stages, this design guarantees all falling and rising transitions in all corresponding aggressive lines. The *victim signal generator* (VSG) is used to generate signals for a victim interconnect, and the signals include Q1 (constant logic 1), Q0 (constant logic 0), rising and falling. The *PE-BIST controller* is used to coordinate the operation of the entire test pattern generation process. The output of the test pattern mixer is abstracted as “Channels”. The *test pattern generation controller* works in such a way that it will select one channel as the victim channel and connect it to the victim signal generator. All other channels are connected to the aggressor signal generator.

For each victim excitation pattern, the aggressor signal generator will excite all aggressors to generate a pseudo-exhaustive pattern set for this victim channel. Then, it will set the next channel as the victim channel, and generate pseudo-exhaustive patterns for that channel. This process is repeated until each channel has been selected as a victim channel.

Figure 5.4 Logic structure for PE-BIST test pattern generation.

Based on the above discussion, if all aggressors of a victim net and the victim net are connected to different channels of the test pattern mixer, the victim net will be pseudo-exhaustively tested. Furthermore, if the number of available channels in the test pattern mixer is n , then the test pattern generator cannot be used to test a victim net with test cone size greater than n . That is, if the available number of channels driven by the test pattern mixer is n , then the test pattern generator can be used to test any victim net with test cone size less than or equal to n . Therefore, if we can design a test pattern generator for a victim net with the maximum required test cone size, then the same test pattern generator can be used to test all other victim nets as well.

For a test cone size $n+1$, there are n aggressor channels and one victim channel. The total number of test patterns is $4(n+1) \times 2^n$. The reasons are: (1) there are $n+1$ different victims within this cone, (2) each victim will be tested by 2^n different test patterns from n aggressor

lines, and (3) each victim can have four different values (Q1, Q0, rising and falling) for pseudo-exhaustive test patterns from its aggressors.

Theorem 5.2: If all aggressors of a victim net and the victim net are connected to different channels of the test pattern generation controller, the victim net will be pseudo-exhaustively tested.

Proof: Since the test pattern generation controller alternates victim patterns (Q1, Q0, rising, falling) in each channel, each victim net will receive victim patterns at some period during the entire test generation process. According to the test pattern generation configuration, for a channel which receives victim patterns, all other channels will receive exhaustive aggressor patterns. So, as long as all of the victim net's aggressors are connected to other different channels, the victim net will receive exhaustive aggressor test patterns, i.e. the victim net will be pseudo-exhaustively tested.

Corollary 5.3: If the number of available channels in the test pattern generation controller is n , then the test pattern generator cannot be used to test a victim net with test cone size greater than n .

Proof: For a test cone size greater than n , there are at least n aggressors to be connected the test pattern generation controller. Since we only have n channels available, at least one aggressor must be connected to the same channel as another aggressor or the victim; that is, at

least two signal nets will not be independent during the test generation process. This simply means it will not form an exhaustive test set within the test cone for the victim.

Corollary 5.4: If the available number of channels driven by the test pattern generation controller is n , the test pattern generator can be used to test any victim net with test cone size less than or equal to n .

Proof: Trivial and thus omitted.

Corollary 5.4 is just a simple extension of Corollary 5.3. This corollary indicates that if we can create a test pattern generator for a victim net with the maximum required test cone size, the same test pattern generator can be used to test all other victim nets as well.

Corollary 5.4 gives us much freedom in constructing the PE-BISTable interconnect structure. As in Figure 5.5, net 2 has two aggressors: net 1 and net 3. Since net 1, 2, 3 are all connected to different channels in the test pattern generator, net 2 can be exhaustively tested in one complete test pattern generation. Note that the test pattern generator is 6 bit. So for every victim pattern, the aggressor channels will issue 32 test patterns. Only 4 out of 32 patterns are needed to test net 2. Other test patterns are redundant for the case of testing net 2.

Corollary 5.4 also indicates that there are more than one way to connect signal nets to test pattern generator. In Figure 5.5, we connect net 1-3 to channel 1-3. According to Corollary 5.4, we can also connect net 1-3 to channel 4-6. As long as net 1-3 are all connected to different channels, they will be exhaustively tested.

C(1)	C(2)	C(3)	C(4)	C(5)	C(6)
0	V	0	1	1	1
0	V	1	1	1	1
1	V	0	1	1	1
1	V	1	1	1	1
0	V	0	0	1	1
..	V	..	..	..	..

Figure 5.5 Test cone covers victim net with smaller order in NIG

To assign nets to each channel, we begin with the node with the maximum degree of connections. Then, we assign different channel numbers to the node and all its connected nodes. For the example interconnect structure in Figure 5.6, nets 3, 4, 2, 7, 8 are assigned to C1, C2, C3, C4, C5, respectively. For nets 1 and 5, they are assigned C5 while net 6 is assigned to C3. There are multiple ways to assign nets to channels. As long as all connected nets are assigned different channels, they will be pseudo-exhaustively tested during the test pattern generation process. For example, when channel 3 is assigned victim test patterns (nodes 2 and 6 are victim nets), channels C1, C2, C4, C5 will apply exhaustive aggressor patterns to aggressor nets 1, 3, 4, 5, 7, 8. Figure 5.7 shows the connection of channels to the interconnect signal lines. Note that nets 2 and 6 (1, 5, and 8) are tested simultaneously.

The test pattern generator can be distributed to different circuit cores. There is no need for one central test pattern generator. All test pattern generators are synchronized such that any channel with the same label will generate the same test pattern at any given time. Each signal net

driving an output cell from any circuit core will be connected to the corresponding channel determined by the labeling process discussed above.

Figure 5.6 PE-BIST test pattern channel assignment.

Figure 5.7 PE-BIST test pattern delivery

Figure 5.8 shows how the test pattern generators and response analyzers are distributed in different circuit cores. The figure shows how test patterns generated from different cores can come together to form a pseudo-exhaustive test cone. For the test cone shown in Figure 5.8, the test cone is formed by signal nets from different circuit cores. Specifically, circuit 1 supplies channels C4 and C5 while circuit 2 supplies channels C1, C2 and C3. Since those channels are independent of each other in the test pattern generation process, they form a pseudo-exhaustive test set.

5.4 PE-BIST Test Architecture

In order to coordinate the entire test process, there is a centralized test controller. The controller controls the entire test process: setting the victim channel, triggering the distributed test pattern generators to start, and collecting the circuit responses when pseudo-exhaustive test patterns are applied. All test generators are synchronized, and generate the same test pattern in each clock cycle. For each victim channel under testing, the corresponding response analyzer (including noise sensors [39][86]) will collect the channel responses for signature analysis. Once the victim channel is pseudo-exhaustively tested, the PE-BIST controller can then collect the responses from all distributed response analyzers and perform the final fault analysis. Note that a victim channel may drive several victim lines. That is the reason why several test response analyzers may be involved.

The whole test process is quite straightforward. It is easy to reuse some existing BIST circuits, e.g., pseudo-random (or exhaustive) test pattern generators and signature analyzers, which might have been built in each core.

Figure 5.8 PE-BIST distributed test pattern generation

5.5 Cast Study

We use a simple example to illustrate how to set up a PE-BIST solution for a given interconnect structure. Figure 5.9 shows a routing solution. The routing area is divided into 3x3 routing region. This is similar to a global routing bin (We will cover global routing concept in the next Chapter). For illustration purpose, we assume that any two nets will have crosstalk effect with each other, only when they are adjacent and overlap in at least one of the routing region.

Figure 5.9 Example routing solution

Figure 5.10 RNIG of Example Routing

First, we draw the RNIG for each region. There are six regions where nets are considered as interfering with each other. The six RNIGs are shown in Figure 5.10. Combine all six RNIGs, we get the GNIG for the example routing in Figure 5.11.

Figure 5.11 GNIG of Example Routing

The next step is to determine the PE-BIST test cone size and assign a test channel to each net. The max order of the vertex in the GNIG is three. So, we first set PE-BIST test cone size equal four. Then, we begin to label each net with test channels from 1 to 4.

Figure 5.12 Channel Assignment of Example Routing

Figure 5.12 shows the assigned test channels. As we can see, the test channels are different for any two nets which have interference with each other. This way, we can guarantee that pseudo-exhaustive test patterns will cover all effective aggressors.

Figure 5.13 PE-BIST of Example Routing

The final testing scheme is shown in Figure 5.13. We show the assigned channel for each net on the routing graph. Those inputs will be connected to the corresponding channels in the PE-BIST test pattern generators, while the outputs will be connected to the corresponding signature analyzer ports. The pseudo-exhaustive test pattern set has size $2^3=8$ for one type of signal integrity fault (e.g. crosstalk glitch fault). Test patterns are shot in parallel and several

victims labeled by the same channel number are tested at the same time. The final PE-BIST solution is shown in Figure 5.13.

5.6 Experimental Results

We applied the PE-BIST GNIG construct scheme on a set of MCNC benchmark circuits. The specification of each circuit is listed in Table 5.1. The grid size refers to the size of a global routing graph which is specified by the numbers of rows and columns. Each MCNC benchmark is placed by a SA-based floorplanner [43], and then gridized to form a global routing graph. Since the power lines have been pre-routed in our over-the-cell (OTC) model, we skip the routing of possible power lines in the net list. The global routing is completed by a maze router [44]. The horizontal and vertical channel capacity is set to 15. The signal net is placed on track by the order of the net routed. The GNIG construction algorithm for PE-BIST has been implemented in C++ and tested on a P4, 2.6 GHz, 1G memory IBM PC.

Table 5.1 List of MCNC benchmark circuits

Chip	# Macro Cell	# Nets	# Pins	Grid Size
ami33	33	119	442	28x23
ami49	49	408	953	100x100
apte	9	94	266	52x52
hp	11	83	309	25x21
xerox	10	195	696	29x32

Locality serves as a measurement of influence on the victim line from nearby aggressors. As the cut-off locality of an interconnect structure increases, more nearby aggressors are considered in PE-BIST testing. In this experiment, we only consider RC interconnect as locality in RC interconnect is defined naturally by distance. We will consider RLCK interconnect in subsequent chapters as it will touch the topic of shield insertion which will be discussed in detail in the next chapter.

Figure 5.14 shows the test cone size for circuit Xerox under different cut-off locality for RC interconnect. As we can see, the PE-BIST test cone size increases as cut-off locality increases. This is because larger cut-off locality means more effective aggressors to the victim line which in turn result in larger test cone size.

Figure 5.14 PE-BIST test cone size VS cut-off locality (Circuit xerox)

Other MCNC benchmark circuit results are listed in Table 5.2 with cut-off locality set to 0. The smallest test cone size is 5 (circuit apte) and the largest test cone size is 50 (circuit ami49). Due to the test time constraint, it is practical that PE-BIST test cone size is limited to 30 or less. In the next chapter, we will discuss methods which can be used to control test cone size.

Table 5.2 MCNC benchmark circuits PE-BIST result

Chip	Test Cone Size
ami33	23
ami49	50
apte	5
hp	30
xerox	35

Summary

In this chapter, we apply PE-BIST to an arbitrary complex interconnect structure. We use the Net Interference Graph to model the crosstalk relationship between different nets. The PE-BIST test cone can be derived from the GNIG. After we have determined test cone size, we then assign nets into different test channels such that any victim and its aggressors are assigned different test channels. Pseudo-exhaustive test patterns will cover all effective aggressors of a victim, and thus achieve the desire test quality. The test architecture is very simple and almost identical to traditional BIST structures with a little modification in the test pattern generation and delivery circuits.

Chapter 6

PE-BIST Test Cone Size Control

In Chapter 4 and 5, we have shown how to apply PE-BIST to simple and complex interconnect structures. By constructing a GNIG, we can determine the PE-BIST test cone size. But in reality the test cone size cannot be too large. Constrained by test time, the total number of pseudo-exhaustive test patterns for each test cone cannot exceed a specific number, e.g., 2^{30} . If we have a GNIG with the maximum degree of all vertices larger than 30, we have to find a way to control the degree of that vertex in GNIG so that it meets the test time constraint.

The way to control PE-BIST test cone size is essentially to control the number of effective aggressors for a victim net. That is, there are many ways to control PE-BIST test cone size. In this chapter, we will discuss how to use shield insertion to limit PE-BIST test cone size after a PE-BIST solution has been formed. This method applies to existing routing solution with extra tracks available for shield insertion.

6.1 PE-BIST Test Cone Size Reduction

In Chapter 5, we constructed PE-BIST GNIG based on net interference information. Each vertex in the GNIG represents a net. An edge between two vertices represents crosstalk coupling. The PE-BIST test cone size can be inferred from the max vertex order in GNIG to fully test all nets. But if the test cone size is set to less than or equal to the maximum degree, any vertex with

its order equal to or greater than the test cone size cannot be fully tested. This introduces the concept of *PE-BIST test coverage*:

PE-BIST test coverage is defined as the number of nets fully testable by PE-BIST test cones divided by the total number of nets.

To illustrate the concept of PE-BIST test coverage, we use the same GNIG constructed from circuit listed in Figure 5.13. Figure 6.1 shows the GNIG. The max vertex order is 3, so the minimum PE-BIST test cone size to achieve 100% PE-BIST test coverage is 4. If we use cone size 3 in testing (i.e. exhaustive test patterns for up to two aggressors), net2 and net3 will not be fully tested because they both have three effective aggressors according to the GNIG in Figure 6.1. In this case, the test cone size 3 can fully test 6 nets out of 8 total nets (except net2 and net3). The PE-BIST test coverage for test cone size 3 is thus $6/8=75\%$. Note that even though net2 and net3 are not fully tested, they will still be partially tested when we apply pseudo-exhaustive test patterns with test cone size 3.

Figure 6.1 Sample PE-BIST GNIG

PE-BIST test cone size is directly related to the maximum degree of vertices in the

GNIG. If we can reduce the maximum degree, we can reduce the PE-BIST test cone size or improve test coverage for a given test cone size. Shield insertion is known to reduce crosstalk between coupled nets [30]. As the edge in GNIG represents the crosstalk coupling between nets, we need to break the coupling if we want to remove an edge connecting to a net in GNIG. The test cone size is inferred from the maximum node order in GNIG. In practice, the test cone size can not exceed 30. So the max node order in GNIG should not exceed the test cone size limit. If we can decouple two nets, we can remove an edge in GNIG thus reduce the node order. This is the basic idea behind PE-BIST test cone size control.

Now, net3 and net2 are out of the test cone size (i.e., 3). We need to reduce the degrees of net2 and net3 in Figure 6.1 by one to achieve 100% PE-BIST test coverage with test cone size 3. There are five edge candidates available to cut: net2-net1, net2-net5, net2-net3, net3-net1, and net3-net4.

Since each edge which represents a coupling between two nets in different regions, shield needs to be inserted in each routing region to decouple two nets. For example, net1 and net3 have coupling in region 2 and 3, so the cost to decouple net1 and net3 is 2 (essentially shield length). We have marked the associated shield length to remove each edge in Figure 6.1.

In choosing an edge to remove, we consider two factors: 1. the shield length to remove the edge 2. the total number of order reduced for nets with order out of test cone size. The final shield insertion cost is shield length divided by the number of order reduced. Table 6.1 shows the calculation process. As we can see, net2-net3 has the least shield insertion cost. Not only the shield length required to remove net2-net3 is as low as 1 but also it can reduce the vertex order of

both net2 and net3 which are out of test cone size. Edge net2-net3 has the minimum shield insertion cost. The shield will be inserted between net2 and net3 in region R1, net2 and net3 will no longer interfere with each other. This will remove the coupling edge between net2 and net3 in the GNIG.

Table 6.1 Shield Insertion Cost Calculation

Edge	Shield Length	# Net Order Reduced	Shield Insertion Cost
net2-net1	1	1	1
net2-net3	1	2	0.5
net2-net5	1	1	1
net3-net1	2	1	2
net3-net4	3	1	3

Figure 6.2 shows the changed GNIG after shield insertion between net2 and net3. After shield insertion, the maximum vertex degree in the GNIG is reduced to 2, and the minimum PE-BIST test cone size to achieve 100% test coverage is reduced from 4 to 3 as well. On the other hand, the PE-BIST test coverage for test cone size 3 will increase from 75% to 100%.

Figure 6.2 Sample PE-BIST GNIG After Shield Insertion

In the case of routing channel overflow and shield can not be inserted in certain routing region, we may not be able to remove a specific edge we choose. In this case, we can choose another edge in the list to remove. For example, region R1 is full and can no longer accommodate

shield insertion. We will not be able to choose net2-net3 edge to remove because it requires shield insertion in region R1. In this case, we have to choose the second candidate net2-net1 to remove. But it also requires shield insertion in region R1 which is not possible. So we have to choose next candidate net2-net5 edge to remove. Fortunately, region R4 is not congested so it will accommodate the shield insertion. Figure 6.3 shows the new GNIG after shield insertion.

Figure 6.3 Sample PE-BIST GNIG After Shield Insertion (Channel Overflow 1)

As we can see, net2 order is now within test cone size but net3 is still out of test cone size. We need to further insert shield to remove more edges to control the order of net3. Table 6.2 shows the candidate edge list and its shield insertion cost calculation.

Table 6.2 Shield Insertion Cost Calculation

Edge	Shield Length	# Net Order Reduced	Shield Insertion Cost
net3-net2	1	1	1
net3-net1	2	1	2
net3-net4	3	1	3

Notice that the net3-net2 shield insertion cost is changed to 1 instead of 0.5 because net2 is already testable within test cone size after we decouple net2 and net5. We repeat the same process for edge removal and we finally find net3-net1 can be removed with shield insertion in region R2 and R3. Figure 6.4 shows the GNIG after shield insertion. As we can see, the resulted GNIG is different from Figure 6.2 due to channel overflow considerations. The final shield

insertion cost is $1+2=3$ which is substantially higher. The complete shield insertion algorithm is listed in Figure 6.5.

Figure 6.4 Sample PE-BIST GNIG After Shield Insertion (Channel Overflow 2)

In applying the PE-BIST method to an interconnect structure, there are trade-offs between test coverage, test time (test cone size), and shielding overhead. Given an interconnect architecture, a larger test cone size usually means higher test coverage (until it reaches 100% test coverage), but it also means longer test time. To reduce the test cone size (test time), one can either accept lower test coverage or use proper shield insertion.

Algorithm: PE-BIST Test Cone Size Control by Shield Insertion

Input:

1. Global routing region information: R ;
2. Track assignment in reach routing region: T ;
3. Pseudo-exhaustive test cone size: n ;

Output: Shield insertion with 100% test coverage;

Variable: gNIG : global NIG;

track : net track assignment for R ;

Begin

Construct PE-BIST gNIG from routing solution R and track assignment solution T;

While (any vertex order in gNIG $\geq n$)

Foreach (vertex with order $\geq n$)

Foreach (edge of the vertex)

 Calculate the shield length (L_S) to remove the edge

 Calculate max number of order reduction for any vertex order $\geq n$ (E_R)

 Shield insertion cost = L_S/E_R

 Add the edge and shield insertion cost to edge removal candidate list

End Foreach

End Foreach

 Sort the edge removal candidate list by shield insertion cost

 Select the edge with lowest cost to remove

 Insert shield in each region to remove the edge

If(shield insertion causes routing channel overflow)

 Select the next available edge and repeat the above shield insertion process

End If

 Update gNIG after shield insertion

If (No more shield can be inserted **AND** gNIG still has vertex order $\geq n$)

 Flag fail and exit

End If

End While

End Algorithm

Figure 6.5 PE-BIST test cone size control algorithm by shield insertion

6.2 Experimental Results

We applied the PE-BIST GNIG construct scheme on a set of MCNC benchmark circuits. The specification of each circuit is listed in Table 6.3. The grid size refers to the size of a global routing graph which is specified by the numbers of rows and columns. Each MCNC benchmark is placed by a SA-based floorplanner [43], and then gridized to form a global routing graph. Since the power lines have been pre-routed in our over-the-cell (OTC) model, we skip the routing of possible power lines in the net list. The global routing is completed by a maze router [44]. The horizontal and vertical channel capacity is set to 15. The signal net is placed on track by the order of the net routed. The GNIG construction and shield insertion algorithm for PE-BIST has been implemented in C++ and tested on a P4, 2.6 GHz, 1G memory IBM PC.

Table 6.3 List of MCNC benchmark circuits

Chip	# Macro Cell	# Nets	# Pins	Grid Size
ami33	33	119	442	28x23
ami49	49	408	953	100x100
apte	9	94	266	52x52
hp	11	83	309	25x21
xerox	10	195	696	29x32

6.2.1 Experimental Results for RC Interconnects

1) Effect of Cut-off Locality

Locality serves as a measurement of influence on the victim line from nearby aggressors. As cut-off locality increases, more nearby aggressors are considered in PE-BIST testing. In practice, locality can be determined by circuit simulation for desired test accuracy. In this experiment, we used different cut-off locality to demonstrate its effect on the PE-BIST test coverage. Figure 6.6 shows the PE-BIST test coverage under different cut-off locality for circuit Xerox. It can be observed that the PE-BIST test coverage decreases steadily as the cut-off locality increases. This is due to the fact that a net is exposed to more aggressors as the cut-off locality increases. This results in more edges in the NIG under construction, and thus more shields are used to limit the degree of each vertex in the NIG. We emphasize again that the cut-off locality determines whether two nets form an aggressor-victim pair. The larger the cut-off locality is, the more likely these nets are potential aggressor-victim pairs. This results in more connections between nodes in the NIG. In PE-BIST, the test cone size must cover all effective aggressors for each net. As the cut-off locality increases, each net is exposed to more aggressors, which in turn reduces the PE-BIST test coverage.

As discussed in section 6.1, we can always achieve 100% PE-BIST test coverage by shield insertion. Figure 6.7 shows the number of shields needed to achieve 100% PE-BIST test coverage with preset test cone size 30. The shielding overhead is defined as the shield length

divided by the total signal net length. As we can observe from Figure 6.7, the shielding overhead increases steadily with increasing cut-off locality.

Table 6.4 Shielding overhead with different cut-off locality

Xerox, test cone size=30		
Cut-Off Locality	Test Coverage	Shielding Overhead with 100% Test Coverage
0	95.38%	2.4%
1	94.87%	2.9%
2	93.84%	5.0%
3	92.15%	6.4%
4	90.77%	6.7%

Figure 6.6 Test coverage VS cut-off locality for RC interconnects.

Figure 6.7 Shield insertion VS cut-off locality for RC interconnects

Table 6.5 test coverage VS test cone size for RC interconnect

Xerox, cut-off locality=0		
Cut-Off Locality	Test Coverage	Shielding Overhead with 100% Test Coverage
30	95.38%	2.4%
24	87.13%	5.2%
20	75.23%	8.8%
14	71.61%	9.3%

2) Effect of PE-BIST Test Cone Size

The test cone size also has effects on the PE-BIST test coverage. As the test cone size decreases, more nets in the GNIG cannot be fully covered by the test cone which results in decreasing test coverage as shown in Figure 6.8. The corresponding test coverage decreases from

95% to 70% while the shielding overhead to achieve 100% test coverage increases from 2.4% to 9.3% (Figure 6.9).

Figure 6.8 Test coverage VS test cone size for RC interconnects (Cut-off locality=0)

Figure 6.9 Shielding overhead VS test cone size for RC interconnect (Cut-off locality=0)

3) Results of MCNC Benchmark Circuits

Other MCNC benchmark circuit results are listed in Table 6.6 and Table 6.7 with cut-off locality set to 0 and 1 respectively. The PE-BIST test coverage without shield insertion is generally more than 94%. For ami33, apte and hp, the test coverage is 100% without shield insertion at cut-off locality 0. For other MCNC benchmark circuits, the shielding overhead to achieve 100% test coverage is generally less than 5%. The cut-off locality 1 always results in equal or higher shielding overhead compared to cut-off locality 0.

Table 6.6 MCNC benchmark results (cut-off locality= 0, test cone size= 30)

Chip	Test Coverage	Shielding Overhead With 100% Test Coverage
ami33	100%	0%
ami49	95.10%	3.0%
apte	100%	0%
hp	100%	0%
xerox	95.38%	2.4%

Table 6.7 MCNC benchmark results (cut-off locality= 1, test cone size= 30)

Chip	Test Coverage	Shielding Overhead With 100% Test Coverage
ami33	100%	0%
ami49	94.52%	3.3%
apte	100%	0%
hp	98.79%	0.2%
xerox	94.87%	4.3%

6.2.2 Experimental Results for RLCK Interconnects

1) Effect of PE-BIST Test Cone Size

The same set of MCNC benchmarks shown in Table 6.3 is applied to the RLCK interconnects. It is observed that inductive coupling for local interconnect is insignificant, however, it is significantly higher in long interconnects [67]. In our experiment, only nets with overlap length greater than 700 μ m within the inductive coupling range are subject to inductive coupling. For RLCK interconnects, both capacitive and inductive coupling are considered. To simplify the analysis of inductive coupling, we only consider capacitive coupling between adjacent nets. The cut-off locality listed in this section refers to inductive coupling locality which defines effective inductive aggressors. For RLCK interconnect, two nets are coupled if they are either capacitively coupled (adjacent nets) or inductively coupled (i.e. within the inductive coupling range).

Table 6.8 test coverage VS test cone size for RLCK interconnect

Xerox, cut-off locality=0		
Test Cone Size	Test Coverage	Shielding Overhead With 100% Test Coverage
30	94.87%	4.9%
24	86.67%	8.3%
20	66.67%	13.8%
14	36.92%	21.2%

Figure 6.10 shows the impact of test cone size on PE-BIST test coverage with the inductive cut-off locality equal to 0. As we can observe, the PE-BIST test coverage decreases

with the decreasing test cone size, while the shielding overhead to achieve 100% test coverage increases. Figure 6.11 shows the shielding overhead goes higher when test cone size decreases.

This is quite similar to RC interconnects.

Figure 6.10 Test coverage VS test cone size for RC/RLCK interconnects (Cut-off locality=0)

2) Comparison between RC and RLCK interconnect

Figure 6.10 shows head to head comparison on PE-BIST test coverage between RC and RLCK interconnect with cut-off locality equal to 0. Generally, RC interconnect has higher test coverage than RLCK interconnect. This is due to the fact that inductive aggressors are considered in RLCK interconnect but not in RC interconnect. The inductive aggressors lead to additional edges to the *global net interference graph (GNIG)*. Some vertex in the graph has a higher order than its counterpart in RC interconnect case. In Figure 6.10, for test cone size 20, PE_BIST test

coverage for RC interconnect is 75% VS 66% for RLCK interconnect. The inductive coupling actually affects 9% nets by increasing their number of aggressors from less than 20 to more than 20. Accordingly, the shielding overhead increases more rapidly in RLCK interconnect.

Figure 6.11 Shielding overhead VS test cone size in RC/RLCK interconnect (Cut-off locality=0)

3) Effect of Pre-existing Shield

In our previous experiment, we did not consider any pre-existing shield in the design phase. For high-speed interconnects where inductive coupling is dominant, it has been recommended that one shield should be present in every four lines for high speed RLCK interconnects [45]. A typical shield insertion design scheme gives 10%-16% shielding overhead with 50% signal sensitivity rate [68]. As a matter of fact, pre-existing shield can greatly limit coupling thus improve PE-BIST test coverage. To test the influence on PE-BIST by pre-existing shields, we randomly insert shields into interconnect structure to simulate a pre-existing shield

insertion scheme in the design phase. Then we apply PE-BIST method to the final interconnect. The result for circuit xerox is shown in Table 6.9. As we can see, once we have shields in the interconnect structure, the test coverage improves substantially. A 20% pre-existing shield can improve test coverage from 95% to 99%.

Table 6.9. Effect of Pre-existing Shields (Cut-off locality=0)

Pre-existing Shield	Test Coverage	Shielding Overhead With 100% Test Coverage
0%	94.87%	4.9%
10%	97.94%	1.6%
20%	98.97%	0.9%

4) Effect of Net Sensitivity

Two nets can crosstalk with each other only when one net's switching will cause the other net to malfunction. This introduces the concept of net sensitivity [68]. Two nets are considered sensitive if a switching event on the aggressor happens during a sample time window for the victim. Otherwise, two nets are considered insensitive. Table 6.10 shows the impact of net sensitivity on PE-BIST test coverage. The net sensitivity is represented with a sensitivity matrix S of size $n \times n$, where n is the number of signal nets and an entry S_{ij} of 1 or 0 in location (i, j) indicates that S_i and S_j are sensitive or not sensitive, respectively, to each other. In this experiment, we randomly set the net sensitivity information for circuit xerox. As shown in Table 6.10, the PE-BIST test coverage increases constantly when the net sensitivity decreases. We can achieve 100% PE-BIST test coverage at 50% net sensitivity.

Table 6.10 Effect of Net Sensitivity on PE-BIST Test Coverage

Xerox, cut-off locality=0		
Net Sensitivity	Test Coverage	Shielding Overhead With 100% Test Coverage
100%	94.87%	4.9%
90%	97.43%	3.7%
80%	97.62%	2.4%
70%	97.94%	1.9%
60%	99.18%	0.3%
50%	100%	0%

Table 6.11 MCNC benchmark results (cut-off locality=0, test cone size=30)

Chip	Test Coverage	Shielding Overhead With 100% Test Coverage
ami33	97.47%	3.5%
ami49	94.32%	4.4%
apte	100%	0%
hp	100%	0%
xerox	94.87%	4.9%

5) Results of MCNC Benchmark Circuits

Other MCNC benchmark circuit results are listed in Table 6.11 with cut-off locality set to 0. The results are obtained without considering any pre-existing shield insertion. The PE-BIST test coverage is generally greater than 94%. To achieve 100% test coverage, the shielding overhead is less than 6% for the listed MCNC benchmark circuits. If we consider any pre-existing shields added in the design phase, the PE-BIST test coverage will be higher and the shielding overhead will be smaller.

Summary

In this chapter, we proposed a shield insertion scheme to reduce PE-BIST test cone size. Shield insertion can effectively decouple aggressor-victim nets thus control PE-BIST test cone size. Given an interconnect solution and its PE-BIST solution, a larger test cone size usually means higher test coverage (until it reaches 100% test coverage) but at the same time it means the test time is higher. In practice, there is a trade off between test coverage and test time. If we want to achieve both objectives (test coverage and test time), shielding overhead has to be introduced. Our simulation results show the trade off between test coverage, test time (test cone size) and shielding overhead.

Chapter 7

Post Global Routing for PE-BIST

In Chapter 4, 5 and 6, we have showed how to apply PE-BIST to simple and complex interconnects as well as test cone size control by shield insertion. In this chapter, we will explore post global routing track assignment technique to control the PE-BIST test cone size and reduce the shielding overhead.

7.1 Over-the-cell Routing Model

In modern routing technologies, the routing problem is usually solved by using a two-stage approach: global routing followed by detailed routing. The detailed routing step includes track assignment and detailed net routing. The relative physical track of each net plays a crucial role in determining the NIG, which subsequently determines the PE-BIST test cone. In order to make an interconnect structure PE-BIST testable with a reasonable test cone size, we must have control in the routing phase, especially track assignment. So, we detach the track assignment process from the detailed net routing process as an independent and intermediate step between global routing and detailed routing. This chapter will discuss how to finish track assignment in a way that a PE-BIST solution with a preset test cone size is possible.

As more metal layers are available in current technology, Over-the-cell (OTC) routing has been used extensively for large design cases [41][42]. In the dedicated routing layer, P/G network is usually designed as a mesh structure. In this chapter, we assume that the routing area is divided by the prerouted power/ground (P/G) networks into rectangular partitions as *global bins* [see figure 7.1]. The cell/core terminals are snapped to the nearest global bins in the GRG (Global Routing Graph). The routing problem is usually solved by using the two-stage approach, global routing followed by detailed routing. The detailed routing step includes track assignment and detailed net routing. In our approach, we detach the track assignment from the detailed net routing as an independent intermediate step between global routing and detailed routing. The detailed routing following the track assignment step merely connects each net to its actual physical terminal which is a local operation on the net route while the global routing stage followed by track assignment determines most of the physical routing of the global net. As discussed before, the coupling (both capacitive and inductive) between nets are proportional to their coupling length. This way, we can safely ignore the coupling effect introduced by the local detailed router to the global signal net, thus, we can focus our solution to the global routing and track assignment stage when dealing with the signal integrity problem for global interconnects.

Figure 7.1(a) shows an example layout that illustrates the definition of global bins and a routing for a two terminal net. Figure 7.1(b) is the routing graph generated from (a). The routing area can be modeled by an undirected graph $G(V,E)$ in Figure 7.1(b), where each vertex $v \in V$ represents a global bin, and each edge represents the routing area between two adjacent bins. Each edge has a length which represents the routing length between the center of global bins and

a capacity which represent the maximum number of tracks available for routing in this region.

The capacity is limited by circuit design and technology. We consider a two layer routing – one for horizontal routing and the other for vertical routing. Layers are connected through vias (Figure 7.2). The basic routing problem can be described as a rectilinear Steiner minimal tree (RSMT) problem of specified nodes in GRG.

Figure 7.1 Global Routing

Figure 7.2 Two Layer Routing

7.2 PE-BIST Track Placement Problem

After global routing, each net is assigned to the individual routing region but not yet put on a track. The track placement step will determine the coupling between nets which in terms determines the NIG (both RNIG and GNIG). If we want to have nets to be aggressor-victim pairs, we can place them close to each other within the cut-off locality range. On the other hand, if we want to decouple two nets, we can put them far away from each other. This will alter the topology of the NIG, which subsequently decides the PE-BIST test cone size. By placing nets into the right tracks, we can have control over the NIG and ultimately over the PE-BIST test cone size. This is why we choose to construct PE-BIST testable interconnects in the post global routing phase.

Formulation of the PE-BIST Routing Problem: Given a set of signal nets, their global routing solution, and a specific pseudo-exhaustive test cone size with the desired cut-off locality, the PE-BIST routing problem is to find a net-track assignment solution so that all nets are PE-BIST testable with minimum shielding overhead.

In the process of building GNIG, we first build RNIG for each region individually, and then combine all RNIGs into the GNIG. In our track placement algorithm, we follow the same bottom-up building process: starting with RNIG construction and then accumulating RNIGs into the GNIG gradually. First, we start with a routing region which has the smallest number of nets. We choose such a routing region to start because it has the least flexibility in placing those nets in

the region. We place nets into tracks one by one for each region. A net is chosen each time by a specified cost function such that the one with the least cost and satisfying the GNIG PE-BIST constraint will be selected. The cost calculating process is repeated for every remaining net to dynamically reflect the cost change after a net is placed. If no candidate net can be found for a certain track because of the GNIG PE-BIST constraint, a shield can be inserted to decouple certain nets placed in the region. In the following discussion, we present the algorithm in detail with an example for RC and RLCK interconnect individually. A complete algorithm description is also included.

7.3 PE-BIST Track Placement Algorithm

Throughout the algorithm, we maintain two NIGs: GNIG for the whole interconnect structure and RNIG for the region under routing. After we have the global routing solution, we know the regions each net will be routed through but has not been placed on individual track yet. Capacitive coupling is a short range effect. The physical track a net will be placed on will determine its capacitive coupling aggressors so we consider use the capacitive coupling to guide the track assignment. Initially, GNIG and RNIG contain net nodes without any connection, since we have not placed any net in any routing region yet. We start with the routing region with the smallest number of nets. As we build the RNIG for each region, the GNIG is also updated after the track placement for the entire region is determined. The net selection and placement process is illustrated by an example.

Assume we have the GNIG processed so far given in Figure 7.3. That is, currently, seven nets 1 to 7 have been assigned to tracks in one or more regions, and their corresponding relationship is presented in Figure 7.3. The test cone size is preset to 5 with the capacitive cut-off locality set to 1. That means we consider capacitive coupling for its nearest two aggressors.

In this example, we try to process a region which contains nets 4, 5, 6, 7, 8 after the global routing phase. Our mission is to find a track placement and shield insertion solution so that the new GNIG still satisfies the PE-BIST requirement, i.e., the max degree of each vertex is less than the pseudo-exhaustive test cone size: 5.

Figure 7.3 An example GNIG for RC interconnects

First, we extract the available NIG information for the nets in the processed region from the GNIG. It is essentially a sub-graph of the GNIG except that we add a new vertex, if its corresponding net was not present in the GNIG. Figure 7.4 shows the NIG information extracted from the GNIG with a new node 8.

Our next step is to place nets (4, 5, 6, 7, 8) into different tracks so that the resultant GNIG still satisfies the PE-BIST requirements. If we can find a net order such that the GNIG

remains the same, we say that it is *cost-free* to incorporate this region into the GNIG. Most of the time, we need to either modify the GNIG by adding new edges (newly formed aggressor-victim relationships) or we need to add extra shields to prevent nets from interfering with each other. To illustrate this point, we use the same example.

Figure 7.4 Extracted NIG from GNIG

First, we need to find the max clique in the extracted NIG. A clique of a graph G requires that every node in the clique (which is a sub-graph of G) is connected to each other. In the NIG, this means every node (representing each individual net) is allowed to interfere with each other. Since we use a bottom-up approach, the clique information can be updated every time when we add a new edge to the graph, without exhaustively searching for cliques. We have a clique in Figure 7.4 formed by nets 4, 5, 6. If several cliques are found with the same number of nodes, we select the one whose sum of degrees is maximal. If a clique is not found, we select a two-node pair whose sum of degrees is maximal. In our example, we have only one clique which contains nets 4, 5, 6. In this clique, we further sort the vertices by their degrees.

It is required that the degree of every vertex in the GNIG be smaller than the test cone size. On the other hand, if a vertex's degree is smaller than the test cone size, the net represented by that vertex can always be pseudo-exhaustively tested. While we must limit the maximum degree of each vertex, we can add more edges to the vertex as long as its degree is smaller than the test cone size. If a vertex has the maximum degree, the corresponding net is placed first since it has fewer freely available connections. However, if two nets have the same degree, we place the one with fewer connections to nets in this region first. In our example, both net5 and net6 have the highest degree, followed by net4. But net6 has two connections to nets (4, 5) in this region, while net5 has three connections to nets (4, 6, 7) in this region. So, net6 is placed first, followed by net5. The reasoning behind this is that a net with fewer connections to nets in the region under consideration has a greater probability of introducing new edges in the NIG, when interacting with the rest of the nets in the region. Therefore, it is placed first to reduce its exposure to those nets.

Now, we have an initial track assignment 6-5-4. Since we set the cut-off locality to 1, if we add another net adjacent to net4, that net will interfere with both net5 and net4. If adding such interference causes the degree of a vertex greater than or equal to the pseudo-exhaustive test cone size, it is not allowed to be added and shield(s) must be added to avoid that interference. On the other hand, if it does not result in that way, the net can be placed adjacent to net4. To identify which net is the best candidate to be placed aside net4, we first sort all available nets according to the cost of placing each of them adjacent to net4.

The cost considers two factors: (1) the number of edges to be added into the NIG, and (2) the number of shields to be added to separate the nets. In our example, to add net7 aside net4, we have to add a new edge between net7 and net4 in Figure 6.4. While to add net8 aside net4, we need to add two edges: one is between net8 and net4, while the other is between net8 and net5. In the latter case, net5 will have a degree of 5 which is equal to the test cone size. So in order to put net8 aside net4, a shield must be added between net8 and net4. The cost of adding a shield is more than the cost of increasing number of edges in the NIG. Net7 has the minimum cost, so we place net7 adjacent to net4 which results in Figure 6.5. Note that we added one new edge between net7 and net4. Our track assignment becomes nets 6-5-4-7.

Figure 7.5 An example GNIG for RC interconnects (cont.).

Now, only net8 is left. We have to place net8 beside net7. But if we place net8 beside net7, net8 will have interference with both net7 and net4 which results in adding two new edges in the NIG (Figure 7.5). It will force the vertex representing net4 to have a degree of 5 which is equal to the test cone size 5. This is prohibited. So, we have to place a shield somewhere between net8 and net4 so that net4 does not exceed the max order. There are two ways to place the shield: 6-5-4-7-SHIELD-8 and 6-5-4-SHIELD-7-8. Obviously 6-5-4-SHIELD-7-8 is a better option since

it decouples both 7-4 and 4-8 while 6-5-4-7-SHIELD-8 only decouples 7-8. The final track assignment is thus 6-5-4-SHIELD-7-8, and the resultant GNIG is shown in Figure 7.6. Comparing Figure 7.4 and Figure 7.6, one can find out that only one new edge (shield) is added into the GNIG (the nets) after the track assignment in this region.

Figure 7.6 Final GNIG and channel assignment

The last step is channel assignment. As we have explained in Section 4, for a given test cone size n , our PE-BIST test generator must have n channels. For any net in the GNIG, as long as the net and all its aggressors are assigned different channels, all effective aggressors are guaranteed to be involved in generating the pseudo-exhaustive test patterns for the net. For the above example, the test cone size is 5 so there are five channels C1, C2, C3, C4, C5. We start with a vertex with the maximum degree, net5 for example, and assign it to channel C1. Then, its adjacent nets 1, 4, 6, 7 can be assigned to C2-C5 (Figure 7.6). We have net4 and net6 assigned to C3 and C4, so net2 must be assigned to either C1 or C2 or C5. We randomly choose C1. We keep assigning channels until all nets are assigned. The channel assignment problem can be solved by graph-coloring methods [19]. In summary, two nets can be assigned the same channel, if they are

not connected in the GNIG. The final channel assignment is shown in Figure 7.6, while the algorithm is shown in Figure 7.7.

If there is a channel capacity constraint, i.e. there may not be enough available tracks to insert shield in the routing region, we may get a net order larger than test cone size temporarily. In this case, we will allow this exception to exist in GNIG. The idea is that even one routing region does not have the available tracks to do shield insertion, other routing region may have. So the solution is to keep this exception until we get the final GNIG. Then we apply the general shield insertion algorithm we developed in Chapter 6 to do shield insertion. Since we try all of the net's aggressor edges and all routing regions, there is a much better chance to find a region to do shield insertion.

If inductive coupling is considered, the inductive coupling will be determined after we get the final track assignment after the track placement algorithm. The extra inductive edges will then be added to generate the final GNIG. Then we apply the general shield insertion algorithm we developed in Chapter 6 to do shield insertion if there is any net not fully tested by the test cone size.

Algorithm: PE-BIST Post Global Routing Track Placement

Input: 1. Routing region information: R ;

2. Pseudo-exhaustive test cone size : n ;

Output: Track assignment for each net and possible shield insertion in each region;

Variable: gNIG : global NIG;

rNIG : regional NIG;

track : net track assignment for R;

netProcessList : list of nets to be processed;

Sort routing regions in R by the number of nets placed in each routing region;

While (R not empty)

 Select routing region R_i with the least number of nets;

 Insert all nets of R_i to netProcessList;

 Extract rNIG from gNIG for region R_i ;

 Find the max clique in rNIG;

 Sort vertices in the clique by the degree of each vertex, and the number of its connections to vertices in current rNIG;

 Assign vertices in the clique to tracks;

 Delete all nets in the clique from netProcessList;

While (netProcessList not empty)

For (each vertex in netProcessList)

 Placement cost = number of edges and shields added if placed in the track under consideration;

 Select vertex V_j with the least cost;

If (degree of any vertex in gNIG $\geq$ cone size n AND there are available tracks)

Foreach(track position between V_i and V_j)

Calculate the weight of adding a shield on the track position
 (Weight=number of capacitive edge cut)

End Foreach

Add one shield into track position which has the largest weight;

End If

Assign vertex V_j into track;

Delete vertex V_j from netProcessList;

Update rNIG and gNIG to reflect the added edges and shields;

End For

End While

End While

Scan track placement in all regions and add inductive coupling edges to gNIG

If (Any node order $\geq$ test cone size n)

Apply general shield insertion algorithm (see Chapter 6)

End If

Assign channels to each net with the final gNIG;

End Algorithm

Figure 7.7 PE-BIST routing algorithm

7.4 Experimental Results

The PE-BIST track placement algorithm has been implemented in C++ and tested on a P4, 2.6 GHz, 1G memory, IBM PC. The specification of each circuit is listed in Table 7.1. The grid size refers to the size of a global routing graph which is specified by the numbers of rows and columns. Each MCNC benchmark is placed by a SA-based floorplanner [43], and then gridized to form a global routing graph. Since the power lines have been pre-routed in our over-the-cell (OTC) model, we skip the routing of possible power lines in the net list. The global routing is completed by a maze router [44]. The horizontal and vertical channel capacity is set to 15. The signal net is placed on track by our PE-BIST track placement algorithm. The *global net interference (GNIG)* is then constructed and shielding overhead is determined.

Table 7.1 List of MCNC benchmark circuits

Chip	# Macro Cell	# Nets	# Pins	Grid Size
ami33	33	119	442	28x23
ami49	49	408	953	100x100
apte	9	94	266	52x52
hp	11	83	309	25x21
xerox	10	195	696	29x32

Table 7.2 Shielding overhead with different cut-off locality

Xerox, test cone size=30, PE-BIST Track Placement	
Cut-Off Locality	Shielding Overhead
0	0.4%
1	2.0%
2	3.6%
3	6.3%
4	6.4%

7.4.1 Experimental Results for RC Interconnects

1) Effect of Cut-off Locality

Table 7.2 shows how cut-off locality affects shielding overhead for PE-BIST track placement algorithm. The shielding overhead increases steadily when cut-off locality increases. This is quite similar to normal track placement. Figure 7.8 shows the comparison between PE-BIST track placement and normal track placement. It is observed that PE-BIST track placement yields less shielding overhead than normal track placement.

Figure 7.8 Shielding overhead vs cut-off locality for RC interconnects

2) Effect of PE-BIST Test Cone Size

The test cone size also has effects on the shielding overhead. As the test cone size decreases, the maximum allowable aggressors (i.e., the vertex degree) are limited, and thus more shield insertions are required to control the maximum degree of vertex in the NIG. Table 7.3 shows the shielding overhead with the test cone size varying from 14 to 30 and the cut-off locality equal 0. Figure 7.9 shows the comparison between PE-BIST track placement and normal track placement. The PE-BIST track placement algorithm generally yields less shielding overhead.

Table 7.3 Shielding overhead vs test cone size

Xerox, cut-off locality=0, PE-BIST Track Placement	
Test Cone Size	Shielding Overhead
30	0.4%
24	1.9%
20	3.9%
14	4.2%

3) Results of MCNC Benchmark Circuits

Other MCNC benchmark circuit results are listed in Table 7.4 and Table 7.5 with cut-off locality equal to 0 and 1 respectively. As we can see, with the test cone size 30, the shielding overhead is very small, less than 3%. For ami33 and apte, no shielding overhead is induced. Our PE-BIST track placement yields less shield insertion compared to normal track placement in all benchmark circuits.

Figure 7.9 Shielding overhead vs test cone size for RC interconnects

Table 7.4 MCNC benchmark results with cut-off locality equal to 0 and test cone size=30

Chip	Shielding Overhead PE-BIST Track Placement	Shielding Overhead Normal Track Placement
ami33	0%	0%
ami49	2.0%	3.0%
apte	0%	0%
hp	0%	0%
xerox	0.4%	2.4%

Table 7.5 MCNC benchmark results with cut-off locality equal to 1 and test cone size=30

Chip	Shielding Overhead PE-BIST Track Placement	Shielding Overhead Normal Track Placement
ami33	0%	0%
ami49	2.4%	3.3%
apte	0%	0%
hp	0.2%	0.2%
xerox	2.0%	4.3%

7.4.2 Experimental Results for RLCK interconnects

The same set of MCNC benchmarks shown in Table 7.1 is applied to RLCK interconnects. It is observed that inductive coupling for local short interconnects is insignificant; however, it is significantly higher in long interconnect [67]. For this experiment, we consider capacitive coupling between two adjacent nets only (cut-off locality 0) while only nets with overlap length greater than 700 μ m within the inductive coupling range (cut-off locality) are subject to inductive coupling. Two nets are coupled if they are either capacitively coupled or inductively coupled. The cut-off locality indicated in this section refers to inductive coupling cut-off locality while capacitive cut-off locality is fixed at 0 to simplify the analysis.

Table 7.6 Shielding overhead vs test cone size for RLCK interconnects

Xerox, cut-off locality=0, test cone size 30	
Test Cone Size	Shielding Overhead
30	2.5%
24	5.3%
20	8.8%
14	12.5%

Table 7.7 MCNC benchmark results with cut-off locality 0 and test cone size 30

Chip	Shielding Overhead PE-BIST Track Placement	Shielding Overhead Normal Track Placement
ami33	1.4%	3.5%
ami49	3.8%	4.4%
apte	0%	0%
hp	0%	0%
xerox	2.5%	4.9%

1) Effect of PE-BIST Test Cone Size

Table 7.6 shows the impact of test cone size on shielding overhead using benchmark Xerox, while Figure 7.10 plots this relationship. As we can see, the shielding overhead decreases rapidly with the test cone size. Basically, it shows the same trend in terms of shielding overhead vs test cone size as in normal track placement case. However, PE-BIST track placement yields much less shielding overhead, especially when test cone size is small.

Figure 7.10 Shielding overhead vs test cone size for RLCK interconnects

2) Results of MCNC Benchmark Circuits

Table 7.7 lists the results for five benchmark circuits with RLCK interconnects. The shielding overhead is generally less than 5% with test cone size set to 30. This is quite acceptable for high speed RLCK interconnect design as it has been recommended that a shield should be present for every four lines for high speed interconnect [45]. As we can see, for each of the benchmark circuits, PE-BIST track placement always yields less shielding overhead compared to normal track placement.

Summary

In this chapter, we proposed a powerful post global routing track placement method for PE-BIST to reduce shielding overhead. PE-BIST track placement tries to process each routing region to find the optimal track placement so that PE-BIST test cone size and shield insertion can be minimized. Simulation results show that it can greatly reduce shielding overhead compared to normal track placement.

Chapter 8

PE-BIST Power Aware Testing

In PE-BIST, multiple victims are tested simultaneously. To create the worst case scenario, all aggressors are switching aggressively. It is very likely to have the test power exceed the limit of power consumption in a SoC design. Hence, power aware test scheduling is very important in order to apply PE-BIST to a power-limited SoC design.

8.1 Background

Typical designs from the *Hyper* synthesis system [48] have shown that buses may alone consume 5 to 40% of the total power, and the interconnect elements together may contribute 25-50% of the total power. Traditionally, $\frac{1}{2} CV_{DD}^2$ expression has been widely used in interconnect power analysis, assuming interconnect as a single capacitor model. For current high speed interconnects, we can no longer neglect the resistance of each interconnect thus this model lacks in accuracy. Several methods to improve the accuracy have been introduced [77]. In [76], it is proposed to analyze the power distribution in interconnects in the frequency domain using poles and residues. However, high complexity is inevitable when calculating the power dissipation of interconnects since poles and residues for current flowing through each resistance have to be calculated. In [78], dynamic power dissipation of on-chip interconnects is estimated with a reduced order model. The work in [53] presented closed-form expressions to estimate the

coupling power based on analysis of lossy transmission lines and distributed RLC circuits. However, their power model depends on circuit-level parameters which make these models suitable for low-level power estimation only. In [54][55], high level design (HLD) and system-on-a-chip (SoC) synthesis have taken interconnect power consumption into account, and primitive interconnect power models are used in calculating the power consumption. In [56], the authors presented a comprehensive treatment of interconnect power consumption during HLS without addressing the problem of interconnect power modeling. In [76], closed-form solutions are presented for inserting repeaters into RLC lines, and the results are highly accurate when compared with numerical solutions. It optimizes repeater insertion to save interconnect power consumption. Recently, low power interconnect design for reconfigurable architecture is discussed in [108-110].

There has been considerable work done in developing power models for interconnects. In [50] [51] [52], the authors have discussed a suite of tools to analyze the interconnect requirements of a chip and provided designers with estimates of power consumption. A new energy model has been proposed by considering coupling effects. The energy model introduces a new parameter, called *charge time*, which represents the correlation time length between two events. By this model, each interconnect is modeled as a set of wire segments and each segment is modeled by a lumped *RLC* element. The coupling effect is modeled mainly based on coupling capacitance. In [53], the transmission line effects on energy dissipation and electromagnetic coupling on the interconnect energy dissipation have been considered in their energy model. A stable circuit that is capable of modeling the transmission line for a broad range of frequencies is synthesized. The

energy is calculated using an approximated expression for the driving-point impedance of lossy coupled transmission lines.

Some significant deep sub-micron (DSM) effects are caused by increasing cross-coupling capacitance and inductance, the effects being most significant in global interconnects [49]. In [50], interconnect crosstalk effects on energy dissipation have been studied. It was observed that crosstalk contributes significantly to the total power dissipation of interconnects. In Figure 8.1, the results clearly show that DSM crosstalk effects worsen the power consumption in interconnects, especially in long interconnects such as busses and inter-core communication structures in a system-on-chip.

Figure 8.1 Crosstalk effects on energy dissipation in interconnects [50]

As shown in Figure 8.2, energy consumed by the coupling capacitance between a pair of interconnects can be given by equation $E = CV_{DD}^2(1 + \exp(1+\epsilon/(RC)))$ or $E = CV_{DD}^2(1 -$

$\exp(1+\varepsilon(RC))$ depending on the correlation of signal transitions. Note that ε is the difference in time between transitions as shown in Figure 8.2, C is the coupling capacitance between two interconnects, while V is the voltage swing for each transition. If two transitions have the same direction (Figure 8.2 (a)), then the latter equation holds. However, if two transitions have different directions (Figure 8.2 (b)) then the former equation holds. In PE-BIST test architecture, each transition is generated by two physically adjacent registers, so this design makes the value of ε approaches 0 for each pair of adjacent lines. Thus, the power dissipation incurred by crosstalk capacitance is either 0 or $2CV_{DD}^2$ depending on the signal correlation. Based on this model, the worst-case coupling energy occurs when each pair of neighboring interconnects receives transitions with different directions, such as (rising, falling, rising, ...) or (falling, rising, falling, ...).

Figure 8.2 Crosstalk power dissipation for coupling interconnects

For DSM interconnects, it is not enough just to count number of transitions. Figure 8.3 shows how different types of transitions can have a profound effect on the amount of energy dissipated. If we only count the number of transitions, a RFRFR bus event should consume 1.2

times the power of a RR0RR bus event, where R is a rising transition, F is a falling transition, and 0 indicates no transition. However, as shown in Figure 8.3, an RFRFR transition consumes 6.7 times the energy of an RR0RR transition for a set of 5mm interconnects. The difference in energy consumption between different types of transitions increases as the interconnect length increases.

Figure 8.3 The effect of different types of transitions on energy consumption in interconnects [50]

It is formally proven that, under some reasonable assumptions on the timing and the drivers, energy dissipation depends only on the admittance matrix of the capacitive part of the bus [69]. That is, for DSM interconnects, we can ignore the inductive coupling effects on power analysis. The adjacent capacitive coupling crosstalk has the largest impact on power dissipation. In [50], a simulation-based three wire power lookup table is used to estimate interconnect power with capacitive coupling which yields average error less than 11% compared to SPICE simulation.

During circuit testing operation, test mode power consumption is usually larger than normal mode power consumption [104]. Several circuit modules are usually tested simultaneously. The switching activity is usually higher in ATPG test patterns and less correlated than normal mode operation [103]. Those switching activities will cause excessive power consumption and thus cause circuit reliability issue, peak power spikes, IR drop as well as yield loss [95][98]. There are generally two test power issues: heat related reliability safety issue and power supply and power noise related issue. The heat related issue is more predictable and can be effectively alleviated by practical techniques like dummy test pattern insertion in ATPG process which prolongs the testing process thus reduces average power consumption [95] and boundary scan segmentation [99][105]. On the other hand, the power supply and power noise related issue is less predictable and more dependent on accurate circuit switching activity caused by test patterns [100][101][102]. In [96], a stimuli generation framework is proposed to generate max switching activity of a given design, which can be used to determine the test power limits. This framework is extended in [97] to define test power limits more accurately with a functional power evaluation flow. In this thesis, we focus primarily on interconnect power dissipation. Unlike the complicated logic inside a circuit, interconnect switching activities are fully predictable thus power can be easily estimated with given ATPG. We target the max power dissipation during PE-BIST testing to avoid testing power supply issues.

8.2 PE-BIST Power Budgeting

To limit the power consumption of a circuit, we can either totally shut down portion of the testing circuits or inhibit the signal transition in some of the circuits so that the dynamic power consumption can be minimized. At the same time, we should avoid shutting down too much of the test circuit since it will prolong the testing process. There is a tradeoff between the test time and test power consumption.

The first step is to establish a high-level estimation model for power dissipation, so that we can quickly estimate the PE-BIST power dissipation. We adopted the well-known simulation-based three wire power lookup method proposed in [50] to estimate the PE-BST test power with crosstalk effects. The three wire model is based on SPICE simulation so it is accurate. The lookup method makes power estimation very fast. It is our ideal way to estimate power dissipation during PE-BIST for simulation accuracy and speed. In this work, we only focus on budgeting power dissipation on interconnects themselves, and thus three wire power lookup model perfectly fits our needs.

8.2.1 Three Wire Power Estimation Model

To estimate power consumption quickly and accurately, we used the three wire power lookup model developed in [50]. In the three wire power estimate mode, a *three-wire power lookup table* is created for all possible transitions by SPICE simulation. Different lengths of three wire interconnects are also simulated. By looking up the type of transition which occurs on three interconnects in the lookup table along with the wire length, an accurate estimate of the center

interconnect's energy consumption is returned. Figure 8.4 shows how to obtain the three wire power lookup table. The three wire power lookup model yields power estimation for an average error of 11% with standard derivation of 6% [50]. In [70], the three wire power estimation model is used on several benchmark circuits and it yields an average error of 6%. This is a quite accurate power estimation model.

Figure 8.4 Three Wire Power Estimation Model

It has been shown that with the presence of crosstalk, the maximum interconnect power consumption occurs when adjacent wires switching in opposite direction. In the case of a three wire bus, it was observed that the maximum power is consumed when adjacent wires are switching in opposite direction: 010=>101 and 101=>010 [69].

The power budgeting of PE-BIST testing is to prevent power overuse during testing. We only need to focus on the maximum power consumption by nets under testing when pseudo-exhaustive test patterns are applied. So we can use the all opposite switching patterns to

estimate the maximum test power required for PE-BIST, instead of estimating the power consumption for all possible patterns.

8.2.2 PE-BIST Power Budgeting Algorithm

As we have discussed in previous chapters, a PE-BIST solution can be modeled as a GNIG (*Global Net Interference Graph*). GNIG shows how many significant aggressors a victim line has. To test a victim line, all aggressor lines must be activated. PE-BEST can naturally do parallel testing on several victim lines. This is very good to reduce test time but it poses problems in power dissipation. To limit power consumption, we may not be able to test all victim lines at once. The problems boils down to which victim nets we test altogether and how to schedule our testing.

Formulation of the PE-BIST Power Budgeting Problem: Given a PE-BIST solution and a power dissipation limit, the PE-BIST power budgeting problem is partition original PE-BIST solution into several child PE-BIST solutions such that each child solution can satisfy the power requirement and the number of child solution is minimized.

To test each victim line, all of its significant aggressors must be activated. The power consumption of testing a victim line is the sum of test power on the victim line plus all its aggressors. Since PE-BIST tests victim nets in parallel, we can save power by simultaneously

testing multiple victims which share the same aggressors. For example, Figure 8.5 shows a sample PE-BIST GNIG with the maximum power for each interconnect if activated (unit: μW). Here to better illustrate our power budgeting algorithm, we simply use a number to represent the power required to activate a wire during testing. In our experiments, however, we use three wire power lookup model to accurately estimate the actual power consumption.

In Figure 8.5, net2 has power consumption of $1\mu\text{W}$. To test net2, we have to activate test patterns from net 1, 3, 4 and 5. So the power consumption to test net2 is estimated the sum of power for nets 1, 2, 3, 4 and 5, i.e. $1+1+3+1+2=8\mu\text{W}$. As can be observed, while we are testing net2, net1 and net3 are also tested in the same test section as all of their aggressors are covered by PE-BIST test patterns. Net5 is not fully tested because it requires activating its aggressor net8. So in order to get net5 finally tested, net8 must be activated. So the total power to test net 1, 2, 3, 5 is $8+1=9\mu\text{W}$. So for a merely $1\mu\text{W}$ increase, we can test net5 in the same test section with nets 1, 2 and 3. On the other hand, to test net5 alone, we need to activate net 2, 3, 4 and 8 which has total power consumption of $3+1+1+1+2=8\mu\text{W}$. When we are already testing nets 1, 2, and 3, we can test net 5 at an extra cost of $1\mu\text{W}$, instead of $8\mu\text{W}$ if net 5 is tested alone. So it is more efficient to test multiple nets in the same test section if they share the same aggressors.

We use the same PE-BIST GNIG to illustrate how our power budgeting algorithm works with power limit no larger than $9\mu\text{W}$. The complete algorithm is listed in Figure 8.7.

Figure 8.5 Sample PE-BEST GNIG with test power required for each net

We start by adding net1 to testing list. To test net1, we need to activate net2. So the total peak power consumption is:

Testing Nets (1), Aggressor Netss (2), Power= $2\mu W$ [STEP 1]

Now both net1 and net2 is activated but so far only net1 is fully tested. To test net2, we need to include net 3, 4, 5 and 1 (already activated). The extra power required is $2+3+1=6\mu W$.

Testing Nets (1, 2, 3), Aggressor Nets (4, 5), Power= $8\mu W$ [STEP 2]

Notice that net3 is put in the testing net list because both of its aggressors net2 and net5 are activated during testing net2. So net3 is tested for free. Now both net4 and net5 are in the aggressor net list (activated but not tested).

To test net5, its aggressors are net 2, 3, 4 and 8. Since nets 2, 3, 4 and 5 are already in the activated net list, it only needs to activate net8. What we can observe is that testing net 5 alone would require $1+1+3+2+1=8\mu W$, but here we just need to activate one more net which is only $1\mu W$ extra. This results a saving of $7\mu W$. If we repeat the similar analysis for net 4, we find that testing net 4 alone requires $1+1+2+1=5\mu W$, but here we need additional $1\mu W$ to test net 4. This results a saving of $4\mu W$. It is more economical to add net 5 in the testing list rather than net 4 as the next choice. So we activate net 8 which results in

Testing Nets (1, 2, 3, 5), Aggressor Nets (4, 8), Power= $9\mu W$. [STEP 3]

Now we have reached the power limit. We start over on the GNIG to budget the rest of the nets. The left nets are net 4, 6, 7, 8 and 9. We start with net 4:

Testing Nets (4), Aggressor Nets (2, 5, 6), Power= $5\mu W$ [STEP 4]

Then we add net6 to the testing suite:

Testing Nets(4, 6, 7), Aggressor Nets (2, 5), Power= $6\mu W$ [STEP 5]

Notice that net7 is added to the testing nets automatically, since it is tested for free if both net6 and net7 are activated. Then we start to add rest of the nets to the testing suite:

Testing Nets (4, 6, 7, 8, 9), Aggressor Nets (2, 5), Power= $8\mu W$ [STEP 6]

Now all nets are added to the tested net list, and we finished our algorithm. The original global GNIG is now partitioned into two smaller GNIGs: (1, 2, 3, 5) and (4, 6, 7, 8, 9), each satisfying the test power requirement. Note that nets 2, 4 and 5 are activated twice because of the partitioning, and both child GNIGs are shown in Figure 8.6.

Figure 8.6 GNIG is partitioned into two child GNIGs

To illustrate how our algorithm saves test time, we re-evaluate step 3. If we do not follow our algorithm, instead of adding net5, we add net4 to test suite in step 3. We will end up with the following test section:

Testing Nets (1, 2, 3, 4), Aggressor Nets (5, 6), Power= $9\mu W$

We reached the power limit, and start to construct the second test section with net 5:

Testing Nets (5), Aggressor Nets (2, 3, 4, 8), Power= $8\mu W$

We then add net8 to the test suite:

Testing Nets (5, 8, 9), Aggressor Nets (2, 3, 4), Power= $9\mu W$

We reach the power limit again. The rest of the nets (6, 7) can be put into one test section. This time, we end up with three test sections: (1, 2, 3, 4), (5, 8, 9) and (6, 7)., and this greatly prolongs the test time.

Algorithm: PE-BIST GNIG Partition with Power Requirement

Input: 1. Test power Limit: $P_{\max}$;

2. PE-BIST GNIG: gNIG;

3. Three wire power lookup table P_{table} ;

Output: child NIGs which satisfy the power requirement;

Variable: testSet: list of nets in the current test section;

candidateNetList: list of nets as candidate to be added to current test section;

While (gNIG not empty)

If (testSet is empty)

Look up P_{table} to find the net with max test power requirement;

If multiple nets have the same power requirement, use the net with max order;

Add the net to testSet;

End If

Foreach (net connecting to nets in testSet)

Calculate additional power required to test the net $P_{\text{additional}}$;

Calculate the power required if the net is tested standalone $P_{\text{standalone}}$;

Calculate total power after net is added to existing test set P_{total} ;

If ($P_{\text{total}} < P_{\max}$)

```


$$P_{\text{save}} = P_{\text{standalone}} - P_{\text{additoinal}};$$


Add net to the candidateNetList;

End If

End Foreach

If (candidateNetList is not empty)

    Select the net with maximum  $P_{\text{save}}$  from the candidateNetList;

    If two nets have the same  $P_{\text{save}}$ , choose the one with larger  $P_{\text{standalone}}$ ;

    Add net to testSet;

Else

    Output nets in testSet to form a child NIG;

    Delete nets in testSet from gNIG;

    Empty testSet;

End If

End While

End Algorithm

```

Figure 8.7 PE-BIST GNIG power budgeting algorithm

8.2.3 PE-BIST Test Scheduling

Once we partition the PE-BIST GNIG into several child NIGs, we can test each child NIG in sequence to fulfill the test power requirement. In order to independently test each child

NIG, we need to shut down part of the circuit so that only nets in the child NIG are delivered with the PE-BIST test patterns while other nets remain quiet. This requires minor modification to the existing PE-BIST test pattern generator design as shown in Figure 8.8. A new test pattern scheduler component is added to the existing PE-BIST test pattern generator. The test pattern scheduler controls a set of switches so that only the nets in the child NIG under testing are delivered with test patterns while all other nets remain quiet.

Figure 8.8 PE-BIST test pattern delivery with power scheduling

8.3 Experimental Results

We applied the PE-BIST power budgeting algorithm to PE-BIST solutions in Chapter 6. The PE-BIST capacitive cut-off locality is set to 0 and test cone size is set to 30. The three wire power lookup table is constructed for different interconnect length in 50 μ m increments. Table 8.1 shows the three wire power lookup table for interconnect length 50 μ m. HSPICE is used to simulate the three wire interconnect model to determine the power dissipation of different types for transition with different interconnects. The total power is estimated by looking up the three wire power lookup table with specific length and switching pattern. If a specific length is not found in the table, the two three wire length entries which are closest are used for linear

interpolation. Table 8.2 lists the MCNC benchmarks and the max interconnect power dissipation for PE-BIST testing estimated by the three wire power lookup model without power limitation.

Table 8.1 Three Wire Power Lookup Table (0: quiet, R: rise, F: fall)

Transition	Power(μ W)	Transition	Power(μ W)
000	0.0	RRR	0.06
0R0	5.2	0RR	1.5
00R	1.1	RRF	5.1
R0R	4.4	0RF	10.9
R0F	0	FRF	18.6

Table 8.2 List of MCNC benchmark circuits

Chip	Interconnect Power Dissipation (mW)
ami33	14.79
ami49	353.35
apte	98.32
hp	31.47
xerox	107.81

Figure 8.9 shows the number of child NIGs generated by our power budgeting algorithm with the power limit set between 60%-100% of the full interconnect power dissipation shown in Table 8.1. As the power limit decreases, the number of child NIGs increases more rapidly. As we discussed in Section 8.2.2, there are aggressors activated multiple times between different child PE-BIST solutions. The total power of all child solutions will exceed the original PE-BIST solution power. The more number of child solutions we form, the larger the difference can be observed. This explains why we have 8 child solutions while power limit is set at 60%.

Figure 8.9 Number of child NIGs vs test power limit (circuit xerox)

Table 8.3 shows MCNC benchmark simulation results. Each power limit is set to 80% of total interconnect power dissipation. Circuit ami49 has the largest number of child NIGs formed, while circuit apte has the smallest.

Table 8.3 List of MCNC benchmark circuit results with test power limit

Chip	Test Power Limit (mW)	Number of Child NIGs
ami33	11.83	4
ami49	282.68	6
apte	78.66	2
hp	25.18	4
xerox	86.25	5

Summary

In this chapter, we studied how to limit power consumption during PE-BIST testing. Our focus is on interconnect switching power dissipation which depends on how we test the interconnects. We used a simulation-based three wire power lookup model to estimate the power consumption on interconnects during testing. An efficient algorithm has been proposed to partition an existing PE-BIST solution into several child PE-BIST solutions. Each child PE-BIST solution can meet a preset interconnect test power dissipation limit.

Chapter 9

Conclusion and Future Research

9.1 Conclusion and Contribution

In dealing with the signal integrity testing problem for interconnects, traditional fault models like maximum aggressor (MA) [5] are no longer valid in the presence of inductive effects. Many work on interconnect testing is based on the MA model. This includes various BIST methods for interconnects and their derivatives like a low-cost self-test scheme for SoC logic cores and interconnects called LI-BIST [9]. When the effect of inductance is considered, there is no efficient fault model for RLCK interconnect testing. Multiple Transition Model (MT) upgrades MA patterns by adding a few more transition patterns. But it can hardly model the long range inductive coupling. It is even concluded in [11] that random test patterns are more qualified than those based on conjectured models to create the worst-case integrity test.

To the best of our knowledge, this is the first time that pseudo-exhaustive built-in self testing (PE-BIST) is proposed to test high-speed interconnects. The motivation of this research comes from the fact that generating test patterns for signal integrity faults of Giga Hertz RLCK interconnects is extremely difficult. There is little research effort done in testing signal integrity considering long range inductive coupling.

In Chapter 3, we validated the concept of crosstalk locality in both RC and RLCK interconnects. Base on the validation, we can partition interconnects based on the concept of locality, and thus pseudo-exhaustive testing is applicable to high-speed interconnects.

In Chapter 4, we applied PE-BIST to simple interconnect bus structures. Due to the regularity of a simple interconnect bus structure, test patterns can be easily delivered through a serial scan chain which can be integrated with existing boundary scan testing techniques. A small modification of the standard boundary scan cell is proposed to accommodate PE-BIST to any boundary scan technique. The test application time can be well controlled by a crosstalk localization technique (e.g. shield insertion). The hardware cost of PE-BIST is very low because it can fully utilize existing testing structures. In this way, PE-BIST is suitable for testing modern SoC interconnect buses.

In Chapter 5, we targeted arbitrary interconnect buses. We proposed a *Net Interference Graph (NIG)* to model the crosstalk between interconnect wires. With the aid of NIG, we can determine the test cone size for PE-BIST and assign each individual net to different test pattern channel. Once all nets are assigned test channels, our PE-BIST test architecture can guarantee that each net is exhaustively tested by its effective aggressors. Since our PE-BIST architecture largely reuses existing BIST structures with minimum modification, the hardware overhead is low and can be easily integrated with an existing on-chip BIST structure.

In Chapter 6, we used shield insertion to control the test cone size. With proper shield insertion, the net order in a GNIG can be effectively controlled. There is a trade off between test time (test cone size), test coverage, and shielding overhead. Simulation results show that

PE-BIST test cone size can be controlled to practical limit (30) with relatively small shielding overhead.

In Chapter 7, we tried to take advantage of the post global routing track placement phase to further reduce shielding overhead and control the PE-BIST test cone size. Experimental results show that PE-BIST track placement yields less shielding overhead compared to normal track placement.

In Chapter 8, a power limit scheme is proposed to partition PE-BIST GNIG into several child NIGs so that each child NIG can be tested within a preset power limit.

The proposed pseudo-exhaustive testing scheme is simple but effective for high-speed interconnect testing. The test quality can be guaranteed if the cone size is properly determined by Hspice simulation. The test structure is simple and can be integrated into existing boundary scan techniques in a SoC chip or existing on-chip BIST structures. The deployment is quite flexible. The test application time is mainly determined by the test cone size, and is weakly dependent on the number of interconnects. Thus, this method can be used to deal with a large number of interconnects in SoC designs.

9.2 Future Work

In this thesis, we have discussed a framework to construct a PE-BIST testing scheme for high speed SoC interconnects. There are several topics which may attract further research efforts in the following discussions.

Multi-layer routing and 3D routing. As the manufacturing technology advances, more metal layers become available and 3D routing also becomes possible. The coupling interaction mechanism of 3D nets will become different. More metal layers also mean more available routing channels. It is interesting to extend PE-BIST to new 3D routing schemes.

Non-uniform interconnect structure. Throughout this thesis, we assume interconnects to be uniform, i.e. the same width, wire spacing, metal layer etc. The locality is determined by simulation. This is a reasonable assumption for high level testable design, when interconnects are uniform. But it can be improved if we can have more information regarding the physical information of actual interconnect layouts. An inductive and capacitive coupling scanning scheme can be developed with the physical design information. This will allow the proposed PE-BIST method to have better net coupling information and is able to deal with non-uniform interconnect structures.

PE-BIST for Network on Chip (NoC). In this thesis, we mainly deal with System on Chip (SoC) interconnects. A major challenge in SoC is to provide reliable means for component interactions. Network on Chip (NoC) design solves this issue by decoupling the communication tasks from the computation tasks, and offers communication protocols to handle packet switching between computation tasks. NoC design transmits data using packets. To apply PE-BIST, specially designed test packets and test sequences must be applied. This poses interesting research on adapting our proposed PE-BIST to NoC interconnects.

Bibliography

- [1] G. E. Moore, "Cramming more Components onto Integrated Circuits," In: *Electronics*, vol. 38 no. 8, April 9, 1965.
- [2] R. H. Dennard *et al.*, "Design of ion-implanted MOSFET's with very small physical dimensions," *IEEE J. Solid-State Circuits*, pp. 256–268, Oct. 1974
- [3] K. L. Shepard and V. Narayanan, "Noise in deep submicron digital design," in *Proc. ICCAD*, pp. 524–531., 1996
- [4] M. Nourani and A. Attarha, "Signal Integrity: Fault Modeling and Testing in High-Speed SoCs," *Journal of Electronic Testing: Theory and Applications (JETTA)*, pp. 539-554, 2002.
- [5] M. Cuvillo, S. Dey, X. Bai, and Y. Zhao, "Fault modeling and simulation for crosstalk in system on chip interconnects," In *Proc. International Conf. on Computer-Aided Design*, pp. 297-303, 1999
- [6] L. Chen, X. Bai and S. Dey, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," in *Proc. Design Automation Conf.*, pp. 317-322, 2001.
- [7] Y. Zhao and S. Dey, "newblock Analysis of interconnect crosstalk defect coverage of test sets," In *International Test Conference*, pp. 492--501. IEEE, 2000.
- [8] L. Chen, X. Bai, and S. Dey, "Testing for interconnect crosstalk defects using on-chip embedded processor cores," In *Design Automation Conference*, pp. 317–322. ACM/IEEE, 2001.
- [9] K. Sekar and S. Dey, "LI-BIST: A low-cost self-test scheme for SoC logic cores and interconnects," In *VLSI Test Symposium*, pp. 417-422. IEEE, 2002.
- [10] M. Nourani and A. Attarha, "Built-in self-test for signal integrity," In *Proc. Design Automation Conference*, pp. 792-797, 2001.
- [11] A. Attarha and M. Nourani, "Testing interconnects for noise and skew in giga Hertz SoCs," In *Proc. International Test Conference*, pp. 305-314, 2001.
- [12] A. Attarha and M. Nourani, "Test pattern generation for signal integrity faults on long interconnects," In *Proc. VLSI Test Symposium*, pp. 336-341, 2002.
- [13] W.H Chang, "Analytical IC-metal-line capacitance formulas," *IEEE Trans. Microwave Theory and Tech.* vol. MTT-24, pp. 608-611, 1976
- [14] U. S. Inan and A. S. Inan, "Engineering Electromagnetics," *Addison-Wesley Publishing Company*, 1999.
- [15] A. Ruehli, "Inductance calculations in a complex integrated circuit environment," *IBM Journal of Research and Development*, vol. 16, pp. 470–481, September 1972.
- [16] K. Gala, V. Zolotov, R. Panda, B. Young, J. Wang, and D. Blaauw, "On-chip inductance modeling and analysis," *Proceedings of 37th Design Automation Conference*, pp. 63–68, June 2000.
- [17] J. W. Nilson, "Electric circuits," *Addison-Wesley Publishing Company*, May 1993
- [18] Y. I. Ismail, E. G. Friedman, and J. L. Neves, "Figures of merit to characterize the importance of on-chip inductance," *Proceedings of 35th Design Automation Conference*, pp. 560–565, June 1998.

- [19] C.-K. Cheng, J. Lillis, S. Lin, and N. Chang, "Interconnect Analysis and Synthesis," *John Wiley and Sons*, 2000.
- [20] A. Deutsch et al., "When are transmission-line effects important for on-chip interconnections," *IEEE Transactions on Microwave Theory and Techniques*, vol. 45 no. 10 pp.1836–1846, October 1997.
- [21] M. Kamon, M. J. Tsuk, J. White, FastHenry: A, "Multipole-Accelerated Program 3-D Inductance Extraction," *IEEE Trans on MTT*, vol. 42, no. 9, pp. 1750-1758, 1994
- [22] E.J. McCluskey. "Design for autonomous test," *IEEE Trans. on Computers*, vol. 11, pp. 866-875, Nov. 1981.
- [23] E. J. McCluskey, "Verification Testing - a Pseudoexhaustive Test Technique", *IEEE Transactions on Computers*, vol. 6, pp. 541-546, June 1984.
- [24] L.Chen, X. Bai, and S. Dey, "Testing for Interconnect Crosstalk Defects Using On-chip Embedded Processor Cores," *IEEE/ACM Design Automation Conference*, pp. 317-322, 2001.
- [25] Y. Zhao and S. Dey, "Newblock Analysis of Interconnect Crosstalk Defect Coverage of Test Sets," *IEEE International Test Conference*, pp. 492-501, 2000.
- [26] R. Pendurkar, A. Chatterjee, and Y. Zorian, "Switching Aactivity Generation with Automated BIST Synthesis for Performance Testing of Interconnects," *IEEE Transactions on Computer-Aided Design*, vol. 20 no. 9 pp.1143-1158, 2001.
- [27] L. He, "Simultaneous Shield Insertion and Net Ordering under Explicit RLC Noise Constraint." *IEEE Design Automation Conference*, pp. 199-202, 2001
- [28] Y. Massoud et al., "Layout Techniques for Minimizing On-chip Interconnect Self Inductance," *IEEE Proc. 1998 Design Automation Conference*, pp. 566—571, 1998
- [29] J. Liu, W. B. Jone, and S. R. Das, "Pseudo-exhaustive built-in self-testing of signal integrity for high-speed SoC interconnects," *Proc. of Instrumentation and Measurement Technology Conference*, pp. 1-4, May 2007.
- [30] D. Yu, M. Petranovic, Shoba Krishnan, Kwyro Lee, Cary Y. Yang, "Loop-Based Inductance Extraction and Modeling for Multiconductor On-Chip Interconnects," *IEEE Transactions on Electron Devices*, vol. 53, no. 1, pp. 135-145, January, 2006
- [31] K. L. Shepard and Z. Tan, "Return-limited Inductances: A Practical Approach to On-Chip Inductance Extraction," *Proceedings of the IEEE Custom Integrated Circuits Conference*, pp. 453-456, 1999.
- [32] S. Kim, Y. Massoud, and S. S.Wong, "On the accuracy of return path assumption for loop inductance extraction for 0.1 μ m technology and beyond," *International Symposium on Quality Electronic Design*, pp. 401–404, March 2003.
- [33] S Wong, S.Y. Kim; C.P. Yue, R. Chang, F. O'Mahony, "Extraction and applications of on-chip interconnect inductance," *7th International Conference on Solid-State and Integrated Circuits Technology*, vol. 1, pp. 142 – 146, Oct. 2004
- [34] B. Young, "Digital Signal Integrity: Modeling and Simulation with Interconnect and Packages", *Englewood Cliffs, NJ: Preentice-Hall*, 2001
- [35]] C. Cheng, J. Lillis, S. Lin and N. Chang, "Interconnect Analysis and Synthesis", *John Wiley & Sons*, 2000
- [36] B. Sheehan, "Projective Convolution: RLC Model-Order Reduction Using the Impluse Response," *IEEE Proc. Design Automation Conference*, pp. 669-673, 1999

- [37] M. Xu and L. He, "An Efficient Model for Frequency-based On-chip Inductance," *IEEE/ACM International Great Lakes Symposium on VLSI*, pp. 115-120, 2001
- [38] Stephen H. Hall, et al., "High-Speed Digital System Design: A handbook of Interconnect Theory and Design Practices", *Wiley-Interscience*, 2000.
- [39] M. Kamon et al., "FASTHENRY: A multipole accelerated 3-D inductance extraction program," *IEEE Trans. Microwave Theory Tech* , vol. 42, pp. 1750-1758, September 1994
- [39] Tehranipour, M.H.; Ahmed, N.; Nourani, M, "Testing SoC interconnects for signal integrity using boundary scan," *IEEE VLSI Test Symposium*, pp.158–163, May 2003
- [40] N. Hanchate, N. Ranganathan, "Simutaneous interconnect delay and crosstalk noise optimization through gate sizing using game theory," *IEEE Trans. on Computers*, vol. 55, no. 8, pp. 1011-1023, August 2006.
- [41] J. Cong, Bryan Preas, C.L. Liu, "General models and algorithms for over-the-cell routing in standard cell design," *Proc. Design Automation conference*, pp. 709 - 715, 1991.
- [42] W. Swartz and C. Sechen, "A new generalized row-based global router," *In Proc. Design Automation Conf*, pp. 491-498, 1993.
- [43] <http://www.cse.ucsc.edu/research/surf/GSRC/progress.html>
- [44] C.Y. Lee, "An algorithm for path connection and its application," *IRE trans. on Electronic Computer*, vol. EC-10, pp. 346-365, September 1961
- [45] X. Huang et al. "RLC Signal Integrity Analysis of High-Speed Global Interconnects," *IEEE International Electron Devices Meeting*, pp. 731-734, 2000
- [46] M.H. Tehranipour, N.Ahmed, M. Nourani, "Multiple transition model and enhanced boundary scan architecture to test interconnects for signal integrity," in *Proc. Int. Conf. on Computer Design*, pp. 554-559, 2003
- [47] S.-P. Sim et al., "An effective loop inductance model for general non-orthogonal interconnect with random capacitive coupling. " *International Electrons Devices Meeting*, pp. 315–318, December 2002.
- [48] J. M. Rabaey, C. Chu, P. Hoang, and M. Potkonjak, "Fast Prototyping of Datapath-Intensive Architectures," *IEEE Design & Test of Computers*, pp. 40-51, June 1991
- [49] M. Cuviallo, S. Dey, X. Bai, and Y. Zhao, "Fault Modeling and Simulation for Crosstalk in System-on-Chip Interconnects", in *Digest of Technical Papers, ICCAD*, pp. 297-301, 1999.
- [50] C.N. Taylor, S. Dey, Y. Zhao, "Modeling and minimization of interconnect energy dissipation in nanometer technologies," *Design Automation Conference*, pp. 754-757, 2001
- [51] J. Cong, "An interconnect-centric design flow for nanometer technologies," *Proc. IEEE*, vol. 89, no. 4, pp. 505–528, April 2001.
- [52] T. Uchino and J. Cong, "An interconnect energy model considering coupling effects," *IEEE Trans. Computer-Aided Design*, vol. 21, no. 7, pp. 763–776, July 2002.
- [52] D. Sylvester and K. Keutzer, "System-level performance modeling with BACPAC - Berkeley advanced chip performance calculator," in *Proc. System-level Interconnect Prediction Conf.*, pp. 109–114, April 1999
- [53] P. Heydari and M. Pedram, "Interconnect energy dissipation modeling in high-speed ULSI circuits," in *Proc. Asia and South acific Design Automation Conf.*, pp. 132–140, January 2002

- [54] R. Mehra, L. M. Guerra, and J. M. Rabaey, "Low-power architectural synthesis and the impact of exploiting locality," *J. VLSI Signal Processing*, vol. 13, no. 8, pp. 877–878, August 1996.
- [55] S. Hong and T. Kim, "Bus optimization for low-power data path synthesis based on network flow method," in *Proc. Int. Conf. Computer-Aided Design*, pp. 312–317, November, 2000
- [56] L. Zhong and N. K. Jha, "Interconnect-aware low power high-level synthesis," *Computer-Aided Design of Integrated Circuits and Systems*, IEEE Trans on, vol. 24, no. 3, pp. 336–351, March 2005
- [57] D. Sylvester and K. Keutzer, "Getting to the bottom of deep submicron," in *Proc. ICCAD*, pp. 203–211, 1996
- [58] J. Xiong, L. He, "Full-chip routing optimization with RLC crosstalk budgeting," *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, vol. 23, no. 3, pp. 366–377, March 2004
- [59] N. Hanchate, N. Ranganathan, "Simutaneous interconnect delay and crosstalk noise optimization through gate sizing using game theory," *IEEE Trans. on Computers*, vol. 55, no. 8, pp. 1011–1023, August 2006.
- [60] S. Naffziger, "Design methodologies for interconnects in GHz ICs - Tutorial Lecture," in *Proc. Int. Solid-State Circuits Conf. (ISSCC)*, Feb. 1999.
- [61] L. T. Wang, M. Nourani, and T. M. Mak, "Coping with physical failures, soft errors, and reliability," Chapter 8, *System-On-Chip Test Architectures: Nanometer Design for Testability*, L. T. Wang, C. Stroud, and N. A. Touba (Editors), Morgan Kaffmann, 2008.
- [62] S. Hasan, A. K. Palit and W. Anheier, "Test pattern generation and compaction for crosstalk induced glitches and delay faults," in *Proc. International Conference on VLSI Design*, pp. 345–350, January 2010.
- [63] J. Lee and M. Tehranipoor, "A novel pattern generation framework for inducing maximum crosstalk effects on delay-sensitive paths, In *Proc. International Test Conf.*, pp. 1–10, October 2008.
- [64] J. Song, J. Han, H. Yi, T. Jung, and S. Park, "Highly compact interconnect test patterns for crosstalk and static faults," *IEEE Trans. on Circuits and Systems – II: Express Briefs*, vol. 56, no. 5, pp. 419–423, May 2009.
- [65] T. Rudnicki, T. Garbolino, K. Gucwa, and A. Hlawiczka, "Effective BIST for crosstalk faults in interconnects," in *Proc. Inter. Symp. on Design and Diagnostics of Electronic Circuits & Systems*, pp. 164–169, April 2009.
- [66] T. Xue, E. S. Kuh, and D. Wang, "Post global routing crosstalk synthesis," *IEEE Trans. on Computer-Aided Design*, vol. 6, no. 12, pp. 1418–1430, December 1997.
- [67] S. H. Choi, B. C. Paul, and K Roy, "Dynamic noise analysis with capacitive and inductive coupling," in *Proc. IEEE Asia and South Pacific Design Automation Conf.*, pp. 65–69, January 2002.
- [68] J. Ma, L. He, "Towards global routing with RLC crosstalk constraints," in *Proc. IEEE Design Automation Conf.*, pp. 669–672, June 2002.
- [69] P.P. Sotiriadis ; A.P. Chandrakasan. "A bus energy model for deep submicron technology," *IEEE Trans on Very Large Scale Integration (VLSI) Systems*, vol. 10, no. 3, pp. 341 – 350, June 2002
- [70] P. Gupta, L. Zhang, N.K. Jha, "A High-level Interconnect Power Model for Design Space Exploration," in *Proc. ICCAD*, pp. 115–122, 2003
- [71] S. Chun, Y. Kim, T. Kim, and S. Kang; "A high-level signal integrity fault model and test methodology for long on-chip interconnections," in *Proc. IEEE VLSI Test Symp.*, pp.152 – 157, May 2009.

- [72] M. H. Tehranipour, N. Ahmed, and M. Nourani, "Testing SoC interconnects for signal integrity using extended JTAG architecture," *IEEE Trans. on Computer-Aided Design*, vol. 23, no. 5, pp. 800-811, May 2004.
- [73] R. Kar, V. Maheshwari, A. Choudhary, A. Singh, A.K. Mal, A.K. Bhattacharjee, "Inductive coupling aware explicit cross-talk and delay formula for on-chip VLSI RLCG interconnects using difference model approach," *International Conference Computing Communication and Networking Technologies (ICCCNT)*, pp. 1-6, 2010
- [74] A.R. Aswatha, T. Basavaraju, A.B. Kalpana, "Efficient power modeling for on-chip global interconnects," *51st Midwest Symposium on Circuits and Systems*, pp. 458-461, 2008.
- [75] M.A. Elgamel, A. Kumar, M.A. Bayoumi, "Efficient shield insertion for inductive noise reduction in nanometer technologies," *IEEE Trans on Very Large Scale Integration (VLSI) Systems*, vol. 13, no. 3, pp. 401-405, 2005
- [76] Y. Shin, T. Sakurai, "Power Distribution Analysis of VLSI Interconnects using Model Order Reduction," *IEEE Tran. Computer-Aided Design*, vol. 21, pp.739-745, June 2002
- [77] Joong-ho Park, Bang-Hyun Sung, Seok-Yoon Kim, "An Efficient Method of Dynamic Power Dissipation on VLSI Interconnects", *LNCS Springer*, vol. 3985/2006, pp. 65-74., 2006
- [78] S. Sahoo, M. Datta, R. Kar, "An Efficient Dynamic Power Estimation Method for On-chip VLSI Interconnects", *2011 Second International Conference on Emerging Applications of Information Technology (EAIT) Digital Object*, pp. 379-382, 2011
- [79] C.Y. Chang, "Random pattern testing for MCM interconnect using computer generated hologram technology," *International Conference on Machine Learning and Cybernetics (ICMLC)*, vol. 5, pp. 2653-2657, 2010
- [80] Y. Kim, H. Kim, and S. Kang, "A new maximal diagnosis algorithm for interconnect test," *IEEE Transactions on VLSI*, vol. 12, no. 5, pp. 532-537, May 2004.
- [81] P. Min, H. Yi, J. Song, S. Baeg and S. Park, "Efficient interconnect test patterns for crosstalk and static faults," *IEEE Transactions on CAD*, vol. 25, no. 11, pp. 2605-2608, November 2006.
- [82] C.-H. Chiang, and S.K. Gupta, "BIST TPGs for faults in board level interconnect via boundary scan," *in Proc.VLSI Test Symp.*, pp. 376-382, April 1997.
- [83] K. S.-M. Li, C. L. Lee, C. Su, and J.E Chen, "Oscillation ring based interconnect test scheme for SoC," *in Proc. Asia and South Pacific Design Automa. Conf.*, pp. 184-187, January 2005.
- [84] T.H. Yeh, S.J. Wang, K.S. Li, "Interconnect test for core-based designs with known circuit characteristics and test patterns," *IEEE International Conference on IC Design & Technology (ICICDT)*, pp. 1-4, 2011
- [85] T. Rudnicki, T. Garbolino, K. Gucwa, A. Hlawiczka, "Effective BIST for crosstalk faults in interconnects," *12th International Symposium on Design and Diagnostics of Electronic Circuits & Systems (DDECS)*, pp. 164-169, 2009
- [86] V. Champac, V. Avendano, J. Figueras, "Built-In Sensor for Signal Integrity Faults in Digital Interconnect Signals," *IEEE Trans. Very Large Scale Integration (VLSI) Systems*, vol. 18, no. 2, pp. 256-269, 2010

- [87] K.K. Duganapalli, A.K. Palit, W. Anheier, "Test pattern generation for worst-case crosstalk faults in DSM chips using Genetic Algorithm," *Electronics System-Integration Technology Conference, ESTC 2008*, 2nd, pp. 393-402, 2008
- [88] <http://www.itrs.net>
- [89] Nathan Ida, "Engineering Electromagnetics", 2nd Edition, Springer, 2004
- [90] K. Lepak, M. Xu, J Chen and L. He, "Simultaneous Shield Insertion and Net Ordering for Capacitive and Inductive Coupling Minimization", *ACM Trans on Design Automation of Electronic Systems*, vol. 9, no. 3, Pages 290–309, July 2004.
- [91] W.H Chang, Analytical IC-metal-line capacitance formulas, *IEEE Trans.Microwave Theory and Tech.* vol. MTT-24, pp. 608-611,1976
- [92] T. Sakurai and K. Tamaru, Simple formulas for two and three dimensional capacitance, *IEEE Trans. Electron Devices*, vol. ED-30, pp. 183-185, 1983
- [93] Jue-Hsien Chern, Jean Huang, Lawrence Arledge, Multilevel Metal Capacitance Models for CAD Design Synthesis System, *IEEE Electron Device Letters*, vol. 13. No 1 pp. 32-34, 1992
- [94] <http://www.mosis.com/>
- [95] P. Girard, N. Nicolici and X. Wen (editors), Power-Aware Testing and Test Strategies for Low Power Devices, *Springer*, ISBN 978-1-4419-0927-5, 2009
- [96] A. Calimera, E. Macii, D. Ravotto, E. Sanchez, M. Sonza Reorda, "Generating power-hungry test programs for power-aware validation of pipelined processors," *Proc. of the 23rd symposium on Integrated circuits and system design*, pp. 61-66, September 2010
- [97] M. Valka, A. Bosio, L. Dilillo, L. Girard, S. Pravossoudovitch, A. Virazel, E. Sanchez, M. De Carvalho and M. Reorda, "A Functional Power Evaluation Flow for Defining Test Power Limits during At-Speed Delay Testing", *IEEE European Test Symposium (ETS)*, pp. 153 – 158, 2011
- [98] J. Saxena, K. M. Butler, V. Jayaram, S. Kundu, N. V. Arvind, P. Sreepakash, and M. Hachinger, "A Case Study of IR-Drop in Structured At-Speed Testing," *Proc. Int. Test Conf.*, pp. 1098–1104, 2003
- [99] L. Whetsel, "Adapting Scan Architectures for Low Power Operation," *Proc. ITC*, pp. 863-872, 2000
- [100] R. Chou, K. Saluja, and V. Agrawal, "Scheduling tests for VLSI systems under power constraints," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 5, no. 2, pp. 175–185, June,1997
- [101] K. Miyase, et al., "A Novel Post-ATPG IR-Drop Reduction Scheme for At-Speed Scan Testing in Broadcast-Scan-Based Test Compression Environment," *Proc. ICCAD*, pp. 97-104, 2009.
- [102] X. Wen, et al., "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing," *Proc. ITC*, Paper 25.1, 2007.
- [103] S. J. Wang, K. L. Fu, and K. S. M. Li, "Low peak power ATPG for n-detection test," *Proceedings of the IEEE International Symposium on Circuits and Systems (ISCAS '09)*, pp. 1993–1996, May 2009.
- [104] P. Girard, "Survey of low-power testing of VLSI circuits," *IEEE Design & Test of Computers*, vol. 19, no. 3, pp. 80–90, May 2002.
- [105] H. Fai and N. Nicolici, "Automated scan chain division for reducing shift and capture power during broadside at-speed test," *IEEE Trans on Computer-Aided Design of Integrated Circuits and Systems*, vol. 27, no. 11, pp. 2092–2097, 2008.

- [106] A. Lambrechts, P. Raghavan, M. Jayapala, B. Mei, F. Catthoor, and D. Verkest, "Interconnect exploration for energy versus performance tradeoffs for coarse grained reconfigurable architectures," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 151–155, 2009.
- [107] T.Yeh; S Wang and K. Li, "Interconnect Test for Core-based Designs with Known Circuit Characteristics and Test Patterns", *IEEE International Conference on IC Design & Technology (ICICDT)*, pp. 1-4, 2011
- [108] A. Lambrechts, P. Raghavan, M. Jayapala, B. Mei, F. Catthoor, and D. Verkest, "Interconnect exploration for energy versus performance tradeoffs for coarse grained reconfigurable architectures," *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, vol. 17, no. 1, pp. 151–155, 2009.
- [109] P. Raghavan, M. Jayapala, A. Lambrechts, J. Absar, and F. Catthoor, "Playing the trade-off game: Architecture exploration using COFFEE," *ACM Trans. on Design Automation of Electronic Systems*, vol. 14, no. 3, pp. 1–37, 2009.
- [110] G. Mehta, J. Stander, M. Baz, B. Hunsaker, and A. K. Jones, "Interconnect customization for a hardware fabric," *ACM Trans. on Design Automation of Electronic Systems*, vol. 14, pp. 1–32, 2009.
- [111] C. Lei, "Genetic-Algorithm- Annealing-Algorithm-Based Scheme for MCM Interconnect Test", *IEEE International Conference on Electronic Packaging Technology & High Density Packaging*, pp. 973-976, 2010
- [112] M Herve, P Aleida, F Kastensmidt, E Cota and M Lubaszewski, "Concurrent Test of Network-on-Chip Interconnects and Routers", *IEEE Test Workshop (LATW)*, pp. 1-6, March 2010
- [113] M Botelho, F Kastensmidt, E Cota and L Carro, "A Broad Strategy to Detect Crosstalk Faults in Network-on-Chip Interconnects", *IEEE/IFIP International Conference on VLSI and System-on-Chip*, pp. 299-303, 2010
- [114] C. Chang, "Random Pattern Testing for MCM Interconnect Testing Using Computer Generated Hologram Technology", *IEEE Proceedings of the Ninth International Conference on Machine Learning and Cybernetics*, pp. 2653-2657, July 2010.
- [115] M. Chan and C. Hsu, "A Strategy for Interconnect Testing in Stacked Mesh Network-on-Chip", *IEEE International Symposium on Defect and Fault Tolerance in VLSI System*, pp. 122-128, 2010