國立交通大學

電子工程學系電子研究所

博士論文

以交連線為中心加強系統晶片可測試性及良率之震盪環結構與演算法

Interconnect-Centric Oscillation Ring Architectures and Algorithms for SoC Testability and Yield Enhancement

研 究 生 ：李 淑 敏

指導教授 ：李 崇 仁 博士

中華民國九十五 年 一 月

INTERCONNECT-CENTRIC OSCILLATION RING ARCHITECTURES AND ALGORITHMS FOR SOC TESTABILITY AND YIELD ENHANCEMENT

by

Katherine Shu-Min Li, B.S., M.S.

DISSERTATION

Presented to the Faculty of the Graduate School of

National Chiao Tung University

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

NATIONAL CHIAO TUNG UNIVERSITY AT HSINCHU

January 2006

To my parents, Yoan-Jian Lee and Ru-Huai Xue.

摘 要

交連線在深次微米及奈米技術中日益重要，因此，交連線的可測試性及良率之問題引起眾多學者投入研究。本博士論文是以震盪環測試結構與演算法(Oscillation Ring Architectures and Algorithms)來解決交連線的可測試性及良率之問題。我們所提出的震盪環測試機制符合IEEE1500標準並用以測試與診斷系統單晶片 (System on Chip,SoC)的交連線。

我們面對兩項技術挑戰，第一項、設計複雜度使得交連線的可測試性及良率之問題不可避免。第二項、串音雜訊使得交連線的訊號整合性及延遲錯誤之問題受到重視。

為了面對第一項設計複雜度的挑戰，我們的作法是將震盪環測試機制嵌入多接繞線器中以增進交連線的可測試性，並且提出以降低與平均繞線壅塞度的方法來增進交連線的良率之問題。

為了面對第二項交連線的訊號整合性及延遲錯誤的挑戰，我們的作法是進行一些基礎分析與研究，茲列舉如下。

(1) 為了建立交連線震盪環結構與演算法的分析架構，我們擴充早先可在交連線上加入緩衝器的研究，所以我們所提出的震盪環測試機制是可與常用的緩衝器加入技術(buffer insertion)相容。

(2) 為了測試交連線的訊號整合性及延遲錯誤，我們提出交連線串音整合偵測機制，其作法是不直接測試延遲錯誤，而是利用串音突波與串音延遲的關係來直接測試串音突波。為了證

明此測試機制的有效性，我們以蒙地卡羅模擬來說明此機制即使在百分之二十的製程飄移下仍達成百分之九十二以上的測試良率。此處所發展的串音突波偵測器電路設計可用於交連線震盪環測試結構中。

(3) 為了提高交連線的可測試性，提出交連線震盪環測試結構與演算法。我們先提出基本的測試方法論，再進一步提出另一個交連線診斷與最佳化的方法與技術。

(4) 最後我們將交連線震盪環結構與演算法加以修改應用到全晶片繞線器上以及同步序向電路中。

本博士論文提出以交連線為中心加強系統晶片可測試性及良率之震盪環結構與演算法來解決交連線的可測試性及良率之問題。綜結以上各觀點，完成了五篇國際論文與提交了六篇期刊論文。

INTERCONNECT-CENTRIC OSCILLATION RING ARCHITECTURES AND ALGORITHMS FOR SOC TESTABILITY AND YIELD ENHANCEMENT

Katherine Shu-Min Li, Ph.D.

National Chiao Tung University, 2006

Supervisor: Dr. Chung-Len Lee

Interconnects play a dominant role in deep-submicron and nanotechnologies. As a result, testability and yield problems of interconnects attract increasing attention.

The paradigm shift of the interconnect-related problems is indispensable to cope with two major challenges as technology advances into nanometer territory:

- The ever increasing design complexity of gigascale integration renders *testability (detection and diagnosability)* and *yield enhancement* inevitable.
- The complicated physical effects inherent from the scaling effects in nanoscale technology make *crosstalk noise* (*crosstalk-induced glitch faults* and *crosstalk-induced delay*) inevitable, and thus *signal integrity* and *delay faults* can no long be ignored.

The motivation of this research is targeted at testability and yield enhancement with test time reduction at design stages by our proposed Oscillation Ring (OR) test mechanism. These advantages of the oscillation ring test mechanism have made interconnects detectable and diagnosable through a systematic graph modeling approach. As a relatively novel methodology, OR mechanism for system-level interconnects should be compliant to IEEE Std. 1500. Thus, it is desirable to consider

test architectures and algorithms for interconnect testing for System on Chip (SoC) under IEEE Std. 1500, and develop interconnect-centric computer-aided-design tools including design, detection, and diagnosis.

To handle the first challenge, the ever increasing design complexity of gigascale integration, we integrate our proposed oscillation ring test techniques into a signal-integrity-aware router. We propose an integrated multilevel full-chip routing algorithm that improves testability and diagnosability, manufacturability, and signal integrity for yield enhancement. Two major issues are addressed.

(1) An oscillation ring test and diagnosis scheme for interconnects, based on IEEE Std. 1500, is integrated into the multilevel routing framework to achieve testability enhancement. We augment the traditional multilevel framework by introducing a preprocessing stage of Interconnect Oscillation Ring Detection (*IORT*) that analyzes the oscillation ring structure for better resource estimation before the coarsening stage, and a postprocessing (final) stage of Interconnect Oscillation Ring Diagnosis (*IORD*) after uncoarsening that improves testability to achieve 100% interconnect fault coverage and maximal diagnosability.

(2) We present a heuristic to balance routing congestion, and the goals of this router include minimizing multiple-fault probability, reducing crosstalk effects, and improving yield for both chemical-mechanical-polishing (CMP) and optical-proximity-correction (OPC) induced manufacturability problems. Experimental results on the MCNC benchmark circuits demonstrate that the proposed OR method achieves 100% fault coverage and the optimal diagnosis resolution for interconnects, and the multilevel congestion-driven routing algorithm effectively balances the routing density to achieve 100% routing completion. Experimental

results show that our method significantly improves routing quality for testability and yield enhancement.

To deal with the second challenge for signal integrity problem, the crosstalk-induced faults have caused significant impact on interconnect performance as technology advances into nanometer era. The crosstalk is a phenomenon of parasitic capacitance caused by continuous scaling effects. It directly influences reliability, manufacturability and yield of VLSI circuits.

(1) We present buffer planning techniques for designing and analyzing crosstalk noise together with performance during floorplanning, and show theoretically and experimentally that our interconnect-aware floorplanner outperforms currently available ones with simultaneously considering crosstalk and timing as our preliminary work which paves the base for *IORT* and *IORD*.

(2) There are two types of crosstalk: *crosstalk-induced glitch* and *crosstalk-induced delay*. We analyze and design the detection of crosstalk faults for interconnect bus, and show experimentally that the unified detection scheme for crosstalk-induced glitch and crosstalk-induced delay is feasible and effectively. This scheme is based on a built-in pulse detector with an adjustable threshold voltage, and we show that this design works well under process variations.

Furthermore, the pulse detector in the crosstalk unified detection scheme is embedded into IEEE Std. 1500 wrapper compliant cells so that oscillation ring test for the interconnect test can handle the delay fault, which poses challenges to system performance.

(3) We study interconnect detection and diagnosis problems for interconnects. We show a class of oscillation ring approximation algorithms for an interconnect

detection and diagnosis problem and prove that oscillation ring mechanism with IEEE Std. 1500 compliant test architecture guarantees 100% fault detection (by *IORT*) and the optimal diagnosis resolution (by *IORD*) not only under the fault models of traditional stuck-at and open faults, but also delay and crosstalk glitch faults. Solutions to the interconnect problems by applying oscillation ring methodology pave the way for developing a novel integrated multilevel routing framework with a congestion metric for routing as mentioned above.

(4) Finally, the oscillation ring test method has been successfully modified and applied to synchronous sequential circuits to facilitate at-speed test for delay fault detectable in addition to traditional stuck-at and open fault models.

In summary, both testability and signal integrity issues have significant impact on interconnect design and test. In my PhD dissertation, an interconnect-centric oscillation ring architectures and algorithms targeted for SoC testability and yield enhancement is proposed to deal with system-level interconnect test and diagnosis, full-chip integrated multilevel router framework, and RTL (register transfer level) synchronous sequential circuits for at-speed testability.

Acknowledgements

I am greatly indebted to my thesis advisor, Professor Chung Len Lee, for his tremendous support, encouragement, and guidance throughout my graduate study. Many discussions with him helped me progress in the right directions. I am also very grateful to him and Ms. Lee, also known as Prof. Hsu, for their help in many other matters, academically and socially. I would like to thank Professor Chauchin Su, Professor Yao-Wen Chang, for inviting me to join in their research groups and to be exposed to the bright research environments. They provided me with invaluable advice and encouragement. I wish to thank Professor Jwu E Chen in Department of Electrical Engineering of National Central University for his constant, immense encouragement and support, and inviting me to join multiple participation of EDA Forum between Academia and Industries.

I greatly appreciate the members of my dissertation committee. Professors Shih-Chieh Chang, Tsin-Yuan Chang, Yao-Wen Chang, Jwu E Chen, Chung-Len Lee, Jing-Yang Jou, Chauchin Su and Chen-Wen Wu for their interest in my work, for their invaluable comments and suggestions, and for their kind assistance in many occasions.

Thanks are due to all members of the SoC Testing & DFT group for providing a forum to discuss my research. Specifically, I wish to thank Dr. Wen-Ching Wu and Dr. Yeong-Jar Chang, Prof. Soon-Jyh Chang, Ming-Sheu Wu, Sheu Ping Lin. I especially appreciate Yoyo Chang at the department's Graduate Office for her professional assistance in all administrative and project matters.

Special thanks go to EDA Lab leading by Professor Yao-Wen Chang including Dr. Tsung-Yi Ho, Tai-Chen Chen, Tung-Chen Chen of National Taiwan University for their pioneering work on Routing and for providing me with the LEDA packages for my comparative studies.

Also special thanks go to Mixed-Signal Circuit Lab leading by Professor Chauchin Su for all the valuable discussion about circuit related characteristics with Hung Wen Lu, Hung Kai Chen, Ren-Qian Xu, Yu Hwai Tseng in Department of Electrical and Control Engineering of National Chiao Tung University.

Thanks also go to many wonderful friends who made my life in Hsin Chu so pleasant and unforgettable. Especially, I would like to express my appreciation to Prof. Pu Hsu for her assistance and support. Also, many thanks to my co-labs and classmates in SoC Testing & DFT Lab in Department of Electronics Engineering, National Chiao Tung University and EDA Lab in Department of Electrical Engineering, National Taiwan University (also previously known as VLSI&EDA Lab in Department of Computer and Information Science, Chiao Tung University).

My deepest appreciation goes to my parents, sisters and brothers for their unending love and support, and my uncle Yuan-Cai Lee and aunt Li-Zhou Gao, uncle Ru-Mao Xue and aunt Mei-Yun Zheng for their encouragement and help. Last, but not least. I am very grateful to the rest of my family: without their sacrifices and patience, the completion of this thesis would not have been possible.

Katherine Shu-Min Li
National Chiao Tung University
January 2006

Table of Contents

Chapter 6. Oscillation Test for Synchronous Sequential Circuits 140

List of Tables

List of Figures

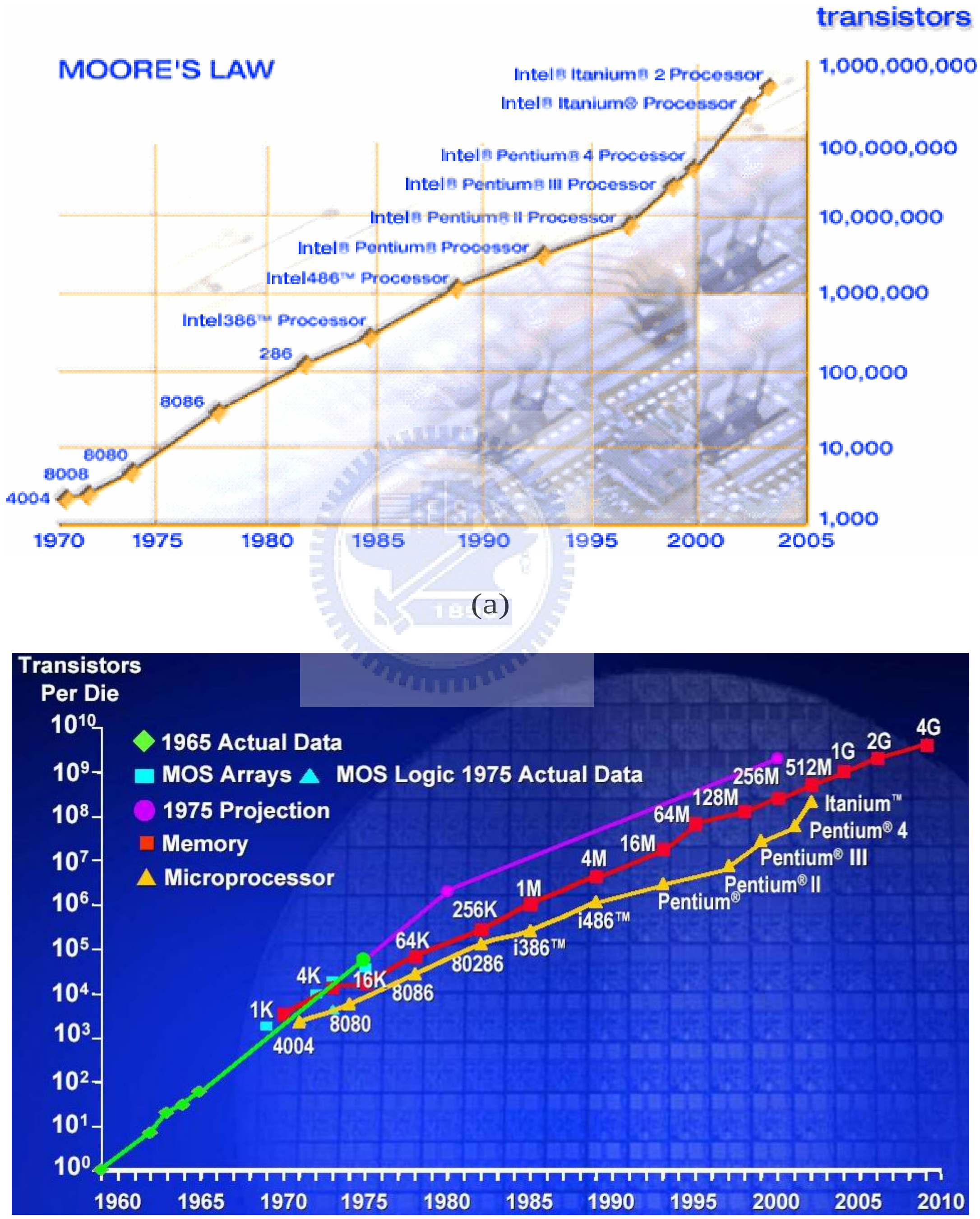
Chapter 1

Introduction

Interconnect becomes the most critical concern in handling performance demand, design complexity and signal integrity, which are the most crucial challenges for designers in nanotechnology. However, to meet all the challenges in performance, complexity, cost, time-to-market, and nanotechnology related issues, the development of sophisticated testability methodology and Electronics Data Automation (EDA) tools for interconnects is essential. This thesis addresses issues on optimizing interconnect-centric oscillation testability and yield enhancement by architectural and algorithmic approaches.

1.1 Interconnect-Centric Study

The motivation arises in dominant effects of interconnects (Figure 1.1), especially for square scaling effects in global interconnects (Figure 1.1(b)), and a more obvious trends appears with the nanotechnology in Figure 1.2 since for 90 nm technology, interconnect delay will account for 75% of the overall delay. A limitation of global interconnect routing (Figure 1.3) specially for SoC lies in their high complexity and density—due to the restricted nature of the interconnect structures, the complexity of the SoC ICs grow too quickly as the number of transistors increase due to Moore’s Law (Figure 1.1(a)). One feasible approach to significantly improving chip capacities based on interconnect architectures is to incorporate testability and diagnosability on an SoC chip with IEEE Std.1500 standards. To deal with a very high complexity and criticality of interconnect structure, it is desirable to develop a new technology and methodology for interconnect testing and diagnosis.



(a)

(b)

Figure 1.1 Interconnect (a) Moore’s Law (b) Scaling effects on memory and microprocessor (*Source*: *Intel for* (a); *Intel at ISSCC*-03 for (b))

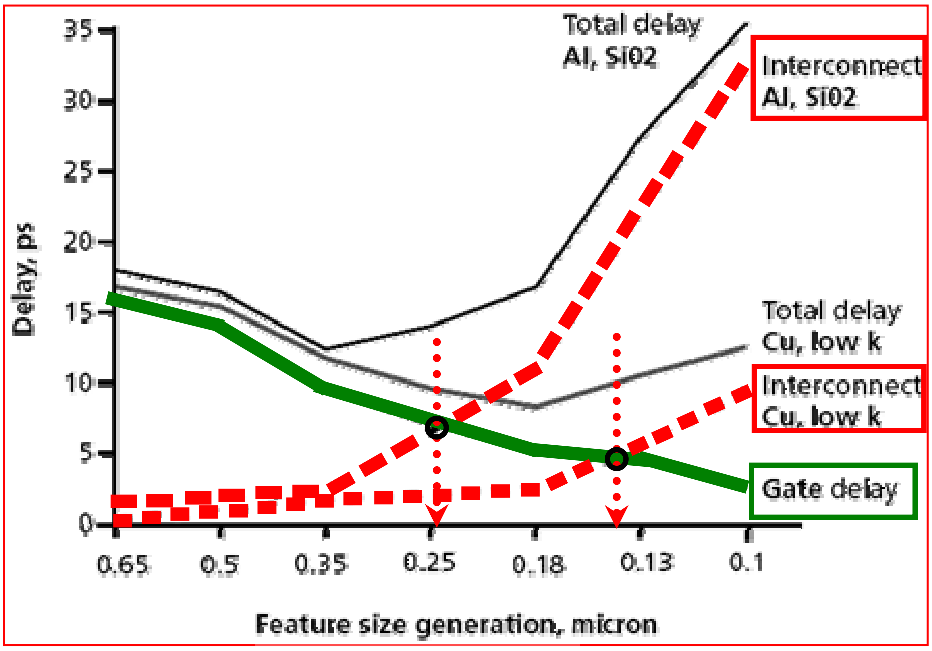
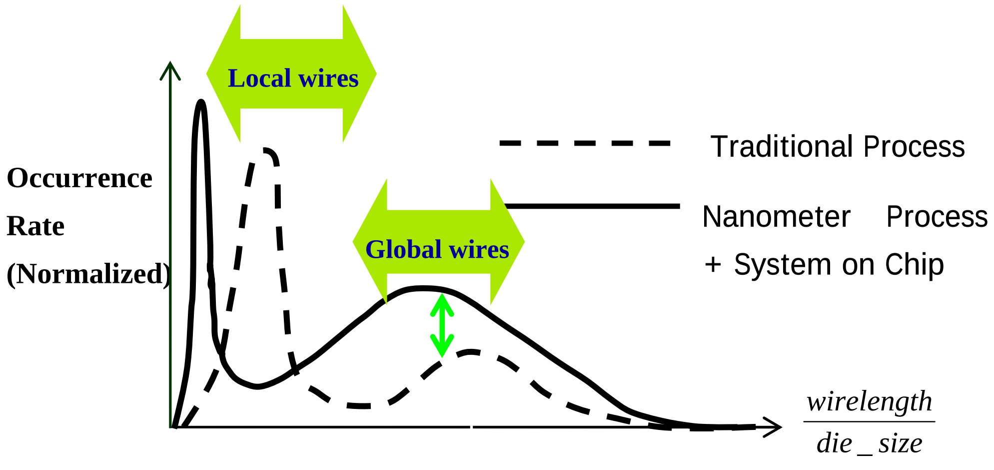


Figure 1.2 For 90 nm technology, interconnect delay will account for **75%** of the overall delay. (*Source: Cadence Design System)*

In Figure 1.3, by observing the relative relationship of the interconnect wirelength and device size (die size), the global wires dominates in nanometer process and SoC eras since intrinsic delay of device scales down by a factor of *s*, local interconnect delay remains the same, and the global interconnect delay increases by square of the scaling factors. The occurrence rates in both local and global wires of nanometer process are more than traditional process technology, with especially obvious difference in global wires.



Source: Tutorial of ICCAD '00

Figure 1.3 Important effects of global interconnects (*Source*: *Tutorial of ICCAD* '00).

1.2 Challenges of Interconnect-Centric Study

The challenges in interconnects are listed as follows:

- Design complexity, performance, and time-to-market (pull force): make interconnects critical in deciding performance.
    - Testing and diagnosability (DfT): It combines the scalable interconnection structure in SoC with considerations of compliant IEEE Std.1500 for DfT (Design for Test), congestion for DfY (Design for Yield), and their applications in physical design including floorplanning and routing frameworks.
    - Testability and yield enhancement (DfM):
        - Furthermore, testability and yield enhancement is important in dealing those CMP, OPC related issues in DfM. Thus, our approach is that interconnect congestion influences multiple fault probability, CMP and

OPC issues, which could be optimized by testability and yield enhancement technologies [81-82].

  - Research has shown that decisions made during the design period determine 70% of the product's costs while decisions made during production only account for 20% of the product's costs. Further, decisions made in the first 5% of product design could determine the vast majority of the product's cost, quality and manufacturability characteristics. This indicates the great leverage that DfM can have on a company's success and profitability [33].

- Signal Integrity: Crosstalk, process variation and related issues in design for manufacturing are caused by deep submicron and nanotechnology (push force). The motivation of our study focuses in analysis and detection of crosstalk effects including crosstalk-induced delay and glitches as shown in Figure 1.5.

A typical interconnect-centric SoC circuit as illustrated in Figure 1.4 is composed of three major components: modules or IPs, routing resources, and input/output (I/O) cells. In an SoC, a two-dimensional of IP/modules is surrounded by general interconnect/routing bounded by I/O cells. The modules or IPs contain combinational and sequential circuits that implement logic functions. The interconnections between the modules or IPs and the I/O cells are general and independent topology. We develop our proposed oscillation ring test methodology targeted for system-level interconnects in SoC circuits [77-80].

In physical design, the optimized circuits are then converted into geometric patterns called *layouts.* The Interconnect-centric physical design process consists of three steps: technology mapping, placement, and routing. A technology mapper maps the optimized circuits into a circuit of logic gates. Thus, we develop an oscillation ring test mechanism targeted for at-speed testability in logic-level synthesis [83-84]. Then, the logic modules are placed by a placement program. In the final step of the physical design, a router assigns interconnects to establish the required connections among the modules or IPs. We integrated our proposed oscillation ring test architectures and algorithm into a congestion-driven multilevel router to enhance testability and yield [81-82].

Note that, in order to meet all the mentioned challenges, we propose an oscillation ring test schemes and test architectures, an effective DfT technique. Though not mentioned in details in the above paragraphs, testing and diagnosibility are integrated into many stages of the design process to show the effectiveness of oscillation ring test methodology.

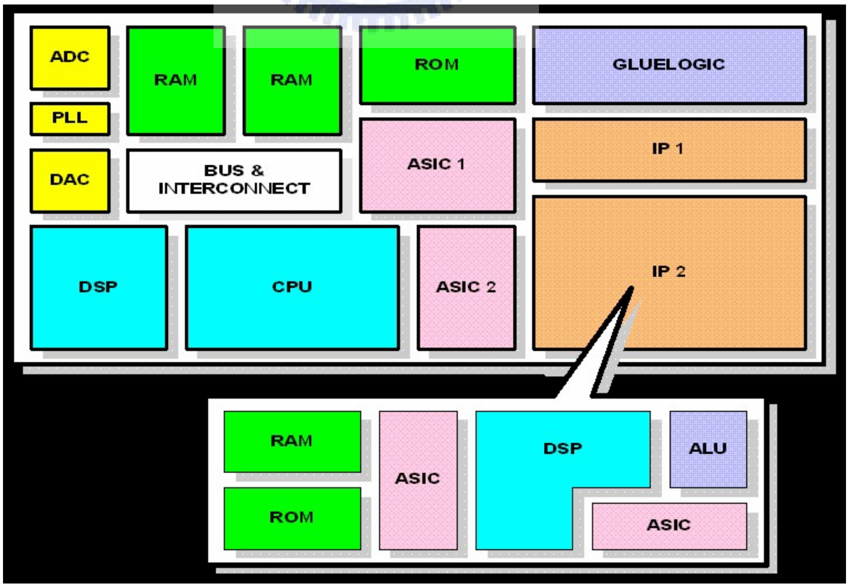
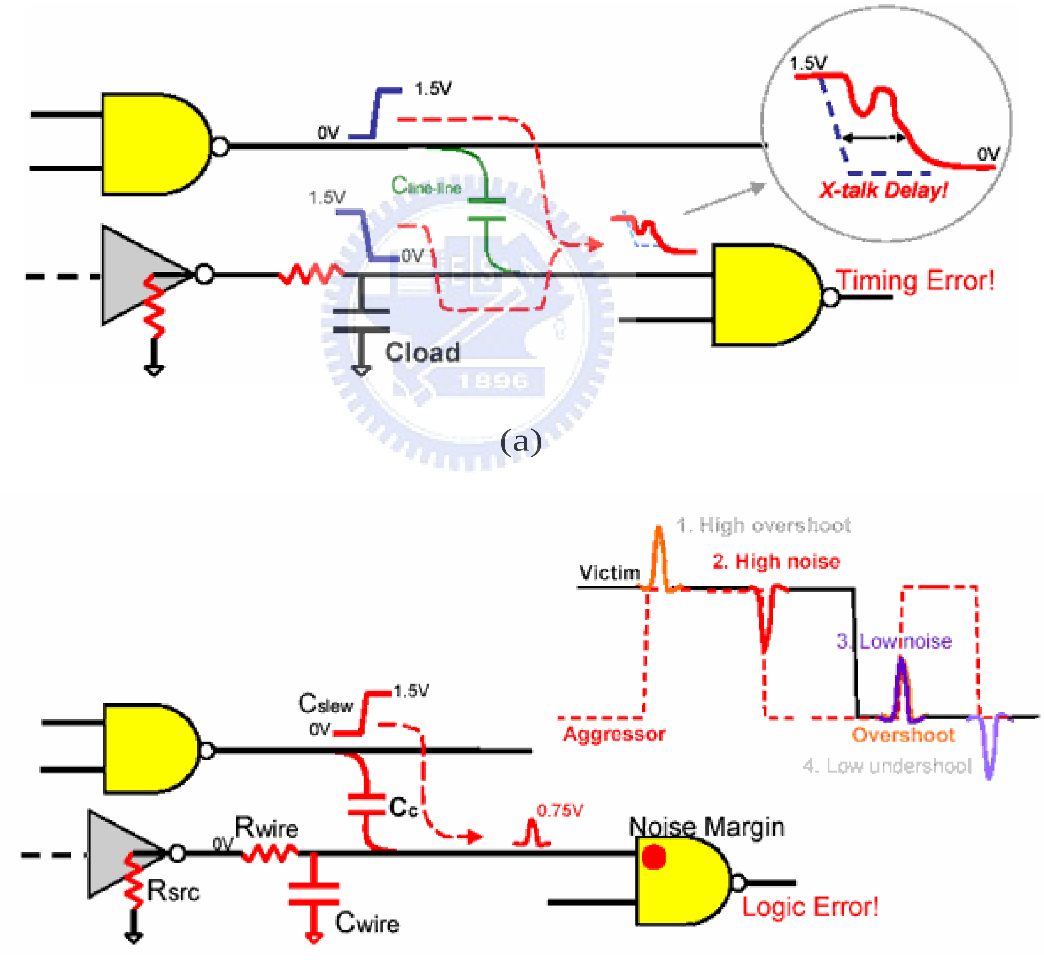


Figure 1.4 An SoC circuit. (*Source: on the courtesy of Prof. K. –J. Lee*)

While the system-level interconnect design and test process is important, currently existing testing approaches for traditional interconnects do not apply to delay and crosstalk glitch faults. This is due to the intrinsic difference between the architectures and algorithms for oscillation-based and those for traditional interconnect test methods. Therefore, it is desirable to develop specific computer-aided-design (CAD) tools for the interconnect-driven oscillation ring methodologies with applications in other design stages, especially for physical design.



(a)

(b)

Figure 1.5 Crosstalk Effects (a) Crosstalk-induced Delay, (b) Crosstalk-induced Glitch. (*Source*: *Magma Design Automation, Inc.*)

As a relatively novel application of ring oscillator technology in digital domains, oscillation ring test architectures are constantly evolving, and so are Oscillation-Ring-specific CAD tools. Therefore, it is of particular importance to design interconnection detection and diagnosis architectures, to develop OR-specific CAD tools for the new test architectures, and to explore the interaction between the architectures and the CAD tools.

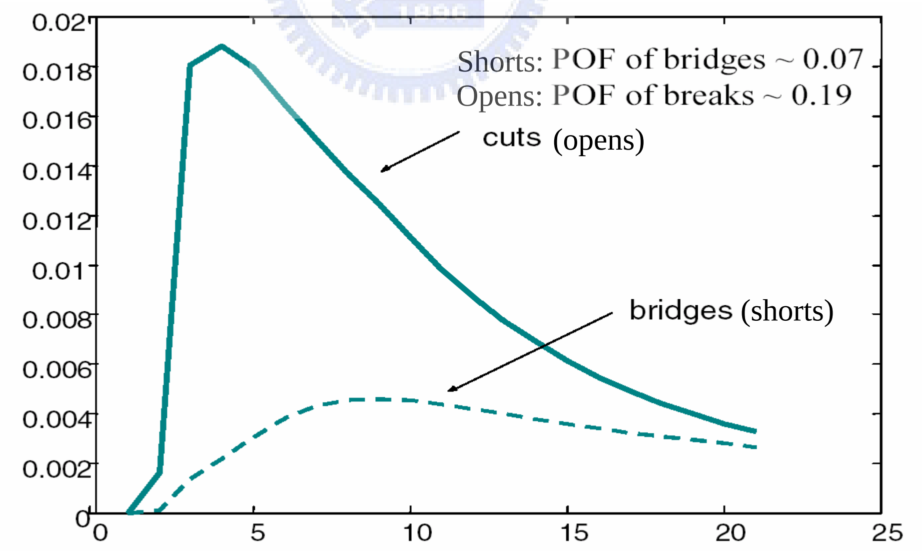
1.3 Interconnect Issues in Design and Test Process

- Signal Integrity Problem for Interconnects: (1) crosstalk noise, (2) crosstalk-induced glitch, crosstalk-induced delay.
    - We present techniques for designing and analyzing crosstalk noise together with performance for interconnects during floorplanning, and show theoretically and experimentally that our interconnect-aware floorplanner outperform currently available ones with simultaneously considering crosstalk and timing [73-74].
    - There are two types of crosstalk: *crosstalk-induced glitch* and *crosstalk-induced delay*. We analyze and design for detection of interconnect bus, and show experimentally that the unified detection scheme for crosstalk-induced glitch and crosstalk-induced delay is feasible and effectively [75-76].
- Testability Enhancement for Interconnect Detection and Diagnosis Problem
    - We show a class of oscillation ring approximation algorithms for an interconnect detection and diagnosis problem and prove that oscillation ring (OR) mechanism with IEEE Std.1500 compliant test architecture guarantees

100% fault detection and the optimal diagnosis resolution not only to traditional stuck-at and open faults, but also to delay and crosstalk glitch faults [77-80].

  - ➢ The motivation of open faults is that open faults are significantly (3x) more likely to occur in Figure 1.6.

- Yield Enhancement for interconnects by congestion-driven approach
  - ➢ Solutions to the interconnect problems by applying oscillation ring methodology pave the way for developing a novel integrated multilevel routing framework with a congestion metric for routing. Experimental results show that the new metric significantly improves a router's average and balanced congestion [81-82].



(Source: de Gyvez, SLIP01)

Figure 1.6 Comparison of probability of faults between short and open faults (*Source*: *de Gyvez*, *SLIP*01).

More specific, our approaches to study the related interconnect issues are listed as follows:

- *An analysis of crosstalk noise framework for system-level interconnects:* as preliminary study to study the problem of crosstalk effects on interconnects among modules or intellectual property (IP)
    - Interconnect-Driven Floorplanning with Noise-Aware Buffer Planning [73-74].
- *An fundamental analysis of crosstalk noise detection scheme for system-level interconnect:* to study the problem of crosstalk effects on interconnects among modules or IPs
    - A unified approach to detecting and optimizing [75-76].
- *An Oscillation Ring Detection Methodology for System-Level Interconnects:* to study the problem of crosstalk effects on interconnects among modules or IPs
    - Oscillation ring based interconnect test scheme for SOC [77-78].
- *An Oscillation Ring Diagnosis Methodology for System-Level Interconnects:* to study the problem of crosstalk effects on interconnects among modules or IPs
    - IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults [79-80].
- *An Oscillation Ring Testability for Logic Synthesis:* to study the problem of crosstalk effects and delay issues on interconnects among modules or IPs at gate level
    - Finite State Machine Synthesis for At-Speed Oscillation Testability [83-84].
- *An Oscillation Ring Testability and Diagnosability for Routing:* to study the problem of crosstalk effects on interconnects among modules or IPs

- Multilevel Full-Chip Routing with Testability and Yield Enhancement [81-82].

1.4 Review of the Dissertation

In this thesis, we focus on two closely related issues of Oscillation Rings: interconnect architectures and interconnect algorithms. Interconnect is a crucial step in implementing circuits. Researchers have shown that the feasibility of interconnect design is most constrained by routing resources [70]. and routing delays dominate the performance (Figure 1). Congestion induced by scaling effects results in limited interconnect testability and diagnossability. On the other hand, congestion in interconnect would reduce routability. Thus designing interconnects to maximize their routability, testability and diagnosability under the area and delay constraints is desirable. Routing in the interconnect environment is more complicated than that in the traditional IC technologies by the constraints that all of the available routing resources are fixed in place. This constraints present new challenges in interconnect design, test and diagnosis in design stages including floorplanner and router which have not been encountered in the traditional IC technologies before and become dominant in SoC and nanotechnology.

This thesis addresses two classes of problems in frameworks of system-level, logic-level and physical design (P&R): testability/diagnosability/yield enhancement designs and signal integrity for interconnects. More specifically, we consider the following issues and each topic is briefly introduced in the following subsection followed by detailed description in the following chapters.

- Crosstalk analyses and resulting in a unified detection scheme (Chapter 3)

- System-level interconnect test structure designs by Oscillation Ring test scheme (Chapters 4 and 5)
- Synthesis for gate-level structure by Oscillation Ring test method (Chapter 6)
- Interconnect routing with oscillation ring detection and diagnosability technologies (Chapter 7).

1.4.1 Interconnect Detection Analysis

Crosstalk-induced faults are the most important fault-models to study the signal integrity effects on interconnects in nanotechnology. The analysis problem, as discussed in Chapter 3 is informally described as follows. Crosstalk-induced delay consists of an original signal waveform and a crosstalk-induced glitch. The question is to determine if there exists both a feasible and unified detection mechanism for interconnect buses. The analysis problem focuses on an interconnect bus structure since the bus structure has parallel and long enough lines to induce excessive coupling capacitance, as is the main source of crosstalk faults. The above analysis has important applications to:

- The system-level interconnect design (Chapters 4 and 5).
- Gate-level interconnect synthesis (Chapter 6).
- Interconnect routing framework: The interconnect routability evaluation considering crosstalk fault detection and diagnosability.

1.4.2 Interconnect Test Structure Design

Given an interconnect architecture and a set of nets, the interconnect detection problem is to find the faulty ring in which the faulty net belongs to under single fault assumption. Therefore, we design the oscillation ring test architectures according to IEEE Std. 1500 in Figure 4.1 and its compliant wrapper cell design in Figure 4.3.

As to oscillation ring algorithms, an exact algorithm based on *cover covering problem based on graph theory techniques* was presented in our works [77-78]. However, since test ring generation problem as an automatic test pattern generation (ATPG) is NP-complete [44], thus QM-based (Quinn MacClusky) algorithm in the worst case is computationally expensive. Whether the analysis problem can be solved in polynomial time is still open.

In Chapter 3, we first consider an approximation algorithm for analyzing the testability of interconnect detection modules in SoC circuits. We show that the proposed algorithm has provably good performance with 100% fault coverage for any interconnect for the two types of target faults of *delay and crosstalk glitch faults,* respectively, in additional to *traditional stuck-at and open faults*. Extensive experiments show that the algorithm is highly accurate and runs much faster than the exact QM algorithm.

In Chapters 4 and 5, we consider the interconnect design problems, respectively. The main consideration in the interconnect design is the trade-off between the testability, diagonsability, congestion, routability, and yield of any interconnect structure. In Chapter 4, we study the oscillation ring test architectures and methodologies for general interconnect topologies, thus, 100% interconnect detection fault coverage achieved. In Chapter 5, we further explore the interconnect diagnosability and the optimal interconnect diagnosis resolution achieved by our algorithms.

Interconnects usually occupy large areas with specially long and parallel global bus routing, and hence the testability and diagnosability of interconnects in SoC ICs is usually limited and results in reduction in yield. On the other hand, fewer interconnects would reduce its routability. Thus, it is desirable to design interconnect test structures and algorithms such that the number of its routable interconnects is maximized with provable testability and diagnosabilty, subject to the area and performance constraints. Experimental results show that our approach consistently outperforms the recent work in [106] and the works before [106] by a large margin in fault models, interconnect topology constraints and test clock control overhead.

1.4.3 Interconnect Diagnosis Analysis

Given an interconnect architecture and a set of nets, the interconnect diagnosis problem is to find the wire segments for each net so that all faults of the net are located in addition to fault detection under single fault assumption. Interconnect diagnosis is a very complex problem. In order to make it manageable in any interconnect structure, the diagnosis problem is often solved using the two-stage method of detection followed by diagnosis. The goals of these two stages are: detecting the faulty oscillating ring of interconnects and diagnosing the faulty wire net segment in the faulty oscillating ring respectively. Unlike existing previous interconnect diagnosis schemes for traditional stuck-at or bridging faults, such as walking-0/walking-1, counting sequence, maximal independent test sets, we target at delay and crosstalk glitch faults [8-10]. There are much important interconnect diagnosis, such as FPGA or bus structures which can take full advantage of the special structures of the interconnect topology. Those two previously researches of traditional fault models and special structures (FPGA [46], bus-driven [108], sparsely

[26]) interconnect diagnosis algorithms do not resemble their counterparts in the general interconnect structure or topology technologies. In particular, the diagnosis resolution information of interconnect structural limitation essentially is still measured by the numbers of test rings. Since the internal architecture of an interconnect topology decides what can generate oscillation test rings through the grids, the traditional measure of interconnect diagnosis capacity is no longer accurate especially for delay and crosstalk faults.

The interconnect diagnosis resolution problem was previously considered by Shi and Fuchs [106]. In [106], a heuristic algorithm based on behavioral diagnosis techniques was proposed: however, the algorithm is only targeted at short, open and stuck-at faults. Our approach is targeted at crosstalk faults and delay faults without involving two-pattern clock control problem, and thus outperforms the traditional diagnostic approaches.

1.4.4 Finite State Machine Synthesis for At-Speed Oscillation Testability

To study the delay test by using oscillation test mechanism, we study the problem of synchronous sequential test. However, gate-level structure limits oscillation testability. And thus, interconnect topology in logic level mainly determines the fault coverage. In order to release the interconnect structural limitations, we propose finite state machine synthesis with target faults of delay faults in addition to traditional stuck-at and open faults.

Our proposed oscillation ring test mechanism in logic level is how to construct oscillation ring conditions by given a Finite State Machine (FSM). The idea is to choose candidates primary outputs of oscillating candidates “0” and “1” and force their next states to alter mutually. This approach is to release the interconnect structural limitations by the DfT techniques of embedding oscillation characteristics in logic synthesis scheme. The main difficulty is that this approach is only suitable to sufficient primary outputs or else oscillation signals can not propagate properly.

The proposed method has three major advantages over the scan test. (1) It enables at-speed test, since oscillation test is triggered by system clock and thus operates at normal speed. (2) Faults are detected if outputs fail to oscillate, thus it is not necessary to store and analyze output response. Thus, the communication bandwidth between the automatic test equipment (ATE) and CUT is greatly reduced, which partly solves the problem of test data compression in SOC testing. (3) Our method does not need complex test clocks, which is required for two-pattern tests used in transitional delay tests. Test vectors can be derived directly from the finite-state machine (FSM) model in our OTPG algorithm, and it greatly simplifies the ATPG process accordingly.

1.4.5 Multilevel Full-Chip Routing with Testability and Yield Enhancement

Interconnection delay plays an increasingly significant role in determining circuit performance, and thus timing-driven routing has received much attention recently. A model of graph-theoretic problem for finding minimum spanning trees with bounded diameter/radius was studied in [49]. This model assumes that the maximum signal delay, denoted by the tree diameter/radius, is in proportion to wiring length (path length in a tree). To improve circuit performance and maintain reasonable routability simultaneously, the congestion in routing channels is usually dense, and thus routing tracks consist of wires with a versatile set of lengths. Researchers have shown that the number of wire segments, instead of wire-length, used by a net is the most critical factor in controlling routing delay [50]. In other words, due to the segmented routing architectures, a signal delay is not necessarily in proportion to the geometric distance (wire-length) of the signal. Therefore it is desirable to consider the timing-driven routing-tree problem with four *independent* weights, two for the traditional signal delay and area, and the other two for the routability and the routing cost of congestion.

To precisely capture the interconnect nature, we show in Chapter 7 a novel way to measure the congestion at individual routing blocks. We model interconnects as a weighted graph. The weights on the edges are proportional to the congestion on the corresponding resources. In particular, the routing congestion information of interconnect structural limitations essentially is still measured by the numbers of available grid boundaries in the interconnect topologies. Since the internal architecture of an interconnect topology decides what can route through the grids, the traditional measure of interconnect routing capacity is no longer accurate. We need to develop a new cost function and weight to include congestion-driven interconnect routing. Thus, we develop a technique that dynamically updates the weights based on the available resources. Experiments show an average congestion improvement of 1.0X-4.52X in the routing required to route MCNC benchmark circuits compared with an algorithm based on the traditional methods for density control [70].

In Chapter 7, recent works [81-82] have also shown that the higher the interconnect routability, the more difficult the testability and yield to achieve 100% fault coverage. Hence, it is of significant importance to consider the detection analysis for interconnects and we propose interconnect routing framework considering oscillation testability accordingly.

In summary of Chapter 7, we consider a model of congestion-driven routing, based on the idea of finding minimum average congestion and variance of congestion in spanning trees (minimum congestion cost) with bounded delays, in a multiple weighted graph. We explore the complexity in two prospectives: (1) testability enhancement: this oscillation detection and diagnosability problem in multilevel routing framework, (2) yield enhancement: congestion-driven routing metric; and present simple, yet efficient and effective approximation algorithms for the problem. Experimental results show that our algorithms are very promising compared with previous works [49-50, 70].

1.5 Organization of the Dissertation

The rest of this dissertation is organized as follows. Chapter 2 presents related issues for interconnects, problem formulation, some backgrounds on oscillation ring architectures and algorithms, and some fundamental frameworks to discuss interconnect-driven floorplanning with noise-aware buffer planning and multi-level routing. Chapter 3 discusses a unified approach to detecting and optimizing on-chip buses for speed and acceptable crosstalk problem. Chapter 4 introduces IEEE Std.1500 compatible oscillation ring interconnect delay and crosstalk test methodology and Chapter 5 explores the design problem for the IEEE Std.1500 compliant interconnect *diagnosis* for delay and crosstalk faults. Chapter 6 addresses the oscillation-specific routing problem and its application in multilevel full-chip routing with testability and yield enhancement. Chapter 7 studies the logic synthesis with oscillation testability to cover at-speed test problem. Finally, Chapter 8 concludes this dissertation and gives future research directions.

In order to make the following chapters self-contained, the interconnect models used in each chapter and some notations and definitions associated with the oscillation ring detection and diagnosis formulation are repeated in each chapter.

Chapter 2

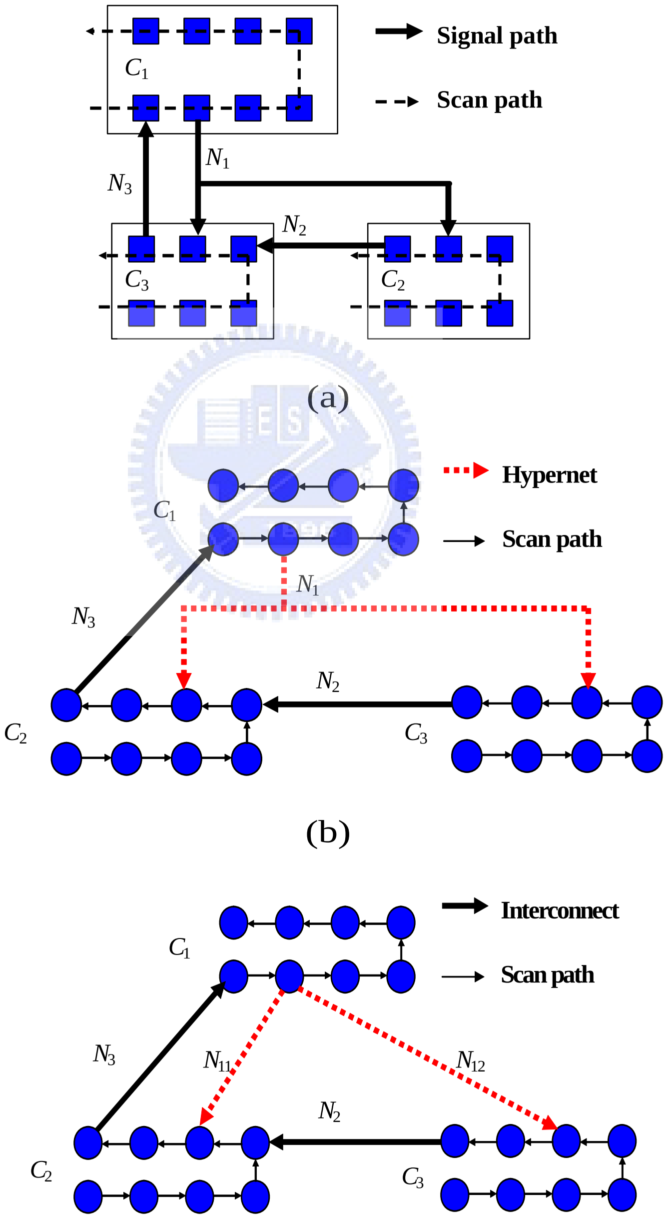
Preliminaries

This chapter gives the models of interconnect detection and diagnosis based on oscillation ring test scheme and the test architecture for system-level interconnects, crosstalk detectors for interconnect buses, synchronous sequential circuits based on the oscillation ring scheme, fundamentals of interconnect routing framework and technologies including the routing model and the cost function, the assumptions and limitations of our formulation, a brief survey on the related research in interconnect detection and diagnosis architectures and algorithms, previous work on both interconnect issues and oscillation test.

2.1 Interconnect Models

2.1.1 Interconnect Model for Detection

The interconnect detection model used in this thesis is shown in Figure 2.1. In order to simplify the problem of interconnect test, we represent the circuit interconnection by using an abstract *hypergraph* to represent the SoC circuit in Figure 2.1 (a) and a *hypernet* to represent a multiple-terminal signal net in Figure 2.1 (b). We define the terminology formally in Chapter 4.4.1. However, this hypernet graph model is not good enough for interconnect detection problem. It is obvious that the two branches of N_1 in Figure 2.1 (a) should belong to two different rings, and they cannot be tested simultaneously. Therefore, we consider each branch of a hypernet individually, and decompose each branch of a hypernet to a 2-pin net. For example, nets N_{11} and N_{12} in Figure 2.1(c) are two 2-pin nets for hypernet net N_1 in Figure 2.1 (b). Without loss of generality, an *n*-terminal hypernet is thus broken into (*n*–1) 2-pin nets as shown in Figure 2.1 (c), and transform our interconnect detection problem into edge-covering problem by using the 2-pin net graph modeling.



(a)

(b)

(c)

Figure 2.1 (a) The interconnect diagram for SoC, (b) hypernet graph, (c) interconnect graph model with 2-pin nets for detection.

For our graph modeling of interconnect test, we have a normal graph $G = (V, E)$, where E is the set of 2-pin nets. There are two rings in Figure 2.1 (c): $R_1 = \{N_{11}, N_3\}$, $R_2 = \{N_{12}, N_2, N_3\}$.

A complete test for stuck-at faults and open faults for all interconnections is thus reduced to a problem of finding a set of rings that cover all edges corresponding to interconnection structure in the graph G. A minimum test is thus the set of rings with minimum cardinality.

To model the delay fault, we use a *weighted graph* $G = (V, E)$ consisting of a vertex set V and an edge set E. In E, each edge, $e_i \in E$, is an ordered pair (u, v), where $u, v \in V$, and has a *weight* w_i. For the delay fault testing, signal delay on each net along the ring is considered. To deal with the delay fault, a weight w_i, which is the timing specification, on a 2-pin net e_i by a 2-tuple $w_i = (l_i, u_i)$, where l_i and u_i are lower and upper bound on the distribution of normal path delay respectively, is defined in Figure 2.2.

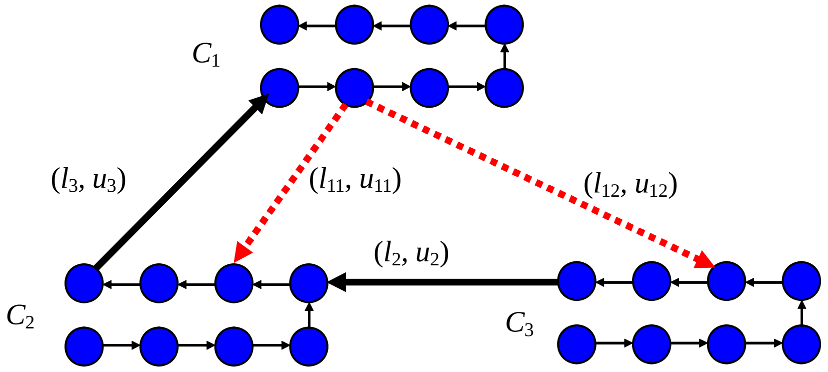
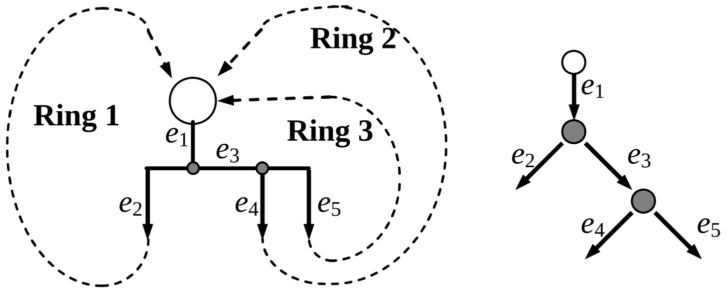


Figure 2.2 Graph model for delay faults.

2.1.2 Interconnect Model for Diagnosis

For interconnect diagnosis problem, our proposed oscillation ring test scheme can also be used for interconnect diagnosis, the process of locating the exact fault site in interconnects. However, the two-pin net model for hypernets is not sufficient for diagnosis. Therefore, the interconnect structure is transformed into a diagnosis graph model as follows. The scan path and wrapper cells in a core are lumped into a single terminal node, as we assume that they are fault-free. The fanout points of a hypernet form dummy intermediate nodes, and a wire segment connecting two nodes is an edge. For example, the diagnosis graph model for the hypernet of Figure 2.3 (a) is shown in Figure 2.3 (b), in which the white node is a terminal node and gray nodes are intermediate nodes. An edge is the smallest unit of a wire segment that can be uniquely diagnosed. In Figure 2.3 (b), any stem affects all the downstream nodes and edges. If edge e1 is faulty, all three rings will not oscillate correctly. A faulty e3 affects rings 2 and 3, while faults on edges e2, e4, and e5 affect rings 1, 2, and 3, respectively. For diagnosis purpose, all these five segments are different. For a test set, our goal is to diagnose the single fault in any edge in the interconnect diagnosis model, and thus the optimal diagnosis resolution is achieved.



(a) (b)

Figure 2.3 (a) a multiple-sink hypernet, and (b) an interconnect diagnosis graph model.

2.2 Oscillation Ring Test Methodology

In this subsection, we will give a overall view of our proposed oscillation ring test scheme (OR) including test architecture & operations, effectiveness, IEEE 1500 compliant wrapper cell design and finally delay measurement formula. As to the details, please refer to Chapter 4.

2.2.1 Oscillation Ring Test Architecture & Operations

In this subsection, we propose the architecture for the oscillation ring test for interconnects as shown in Figure 2.4. Figure 2.4 shows the proposed architecture, where C’s are circuit cores implemented with boundary scan cells and a local counter, which is to capture the induced glitches for crosstalk fault detection and to measure delays of oscillation rings for delay measurement. For this architecture, oscillation ring(s) will be formed as shown during the testing mode. If the formed oscillation ring fails to oscillate, it implies that there exists stuck-at or open fault(s) in components of the oscillation ring. If there is a crosstalk fault between a victim interconnect line and the oscillation ring interconnect lines, glitches will be induced on the victim interconnect line. Figure 2.4 shows the oscillation signal at the oscillation ring and the induced glitches at the victim interconnect line. These induced glitches will be captured by the local counter of the core and be shifted out for observation. To test the delay fault, the delay of the oscillation ring will be measured through using the local counter and the central counter of TAM of the SOC. At this time, the central counter is enabled by signal *OscTest* and triggered by the system clock, and a local counter is connected to one wrapper cell of the oscillation ring so that the oscillation signal is fed to the local counter. When the oscillation test session starts (*OscTest* = 1), the central counter as well as all local counters in cores are enabled. After the counter in TAM counts to a specific number n, the oscillation test session terminates and all local counters are disabled (*OscTest* = 0). The counter contents are shifted out to an ATE for inspection.

Assume that the frequency of the system clock to be f and the local counter content of the ring to be n_i. The ring's oscillation frequency, *fi*, is:

$$f_i = f \times n_i / n \tag{2.1}$$

According to the timing specification, for a good oscillation ring connected by interconnect lines and boundary scan cells, $f_{min} \le f_i \le f_{max}$. That is: $n_{min} \le n_i \le n_{max}$.

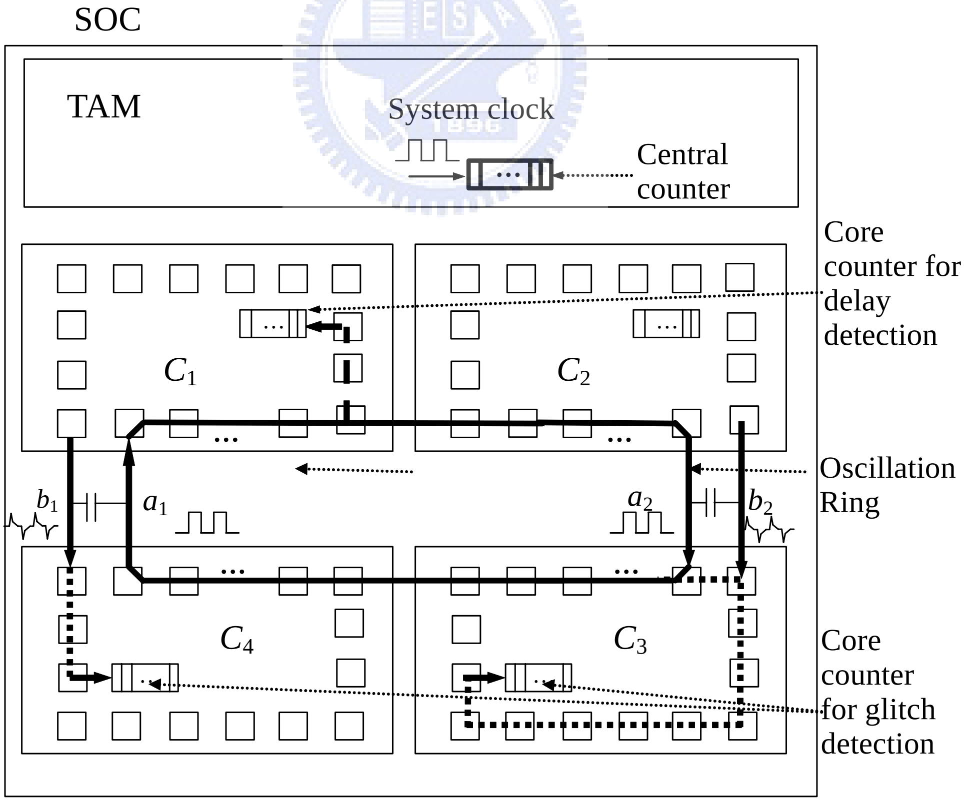
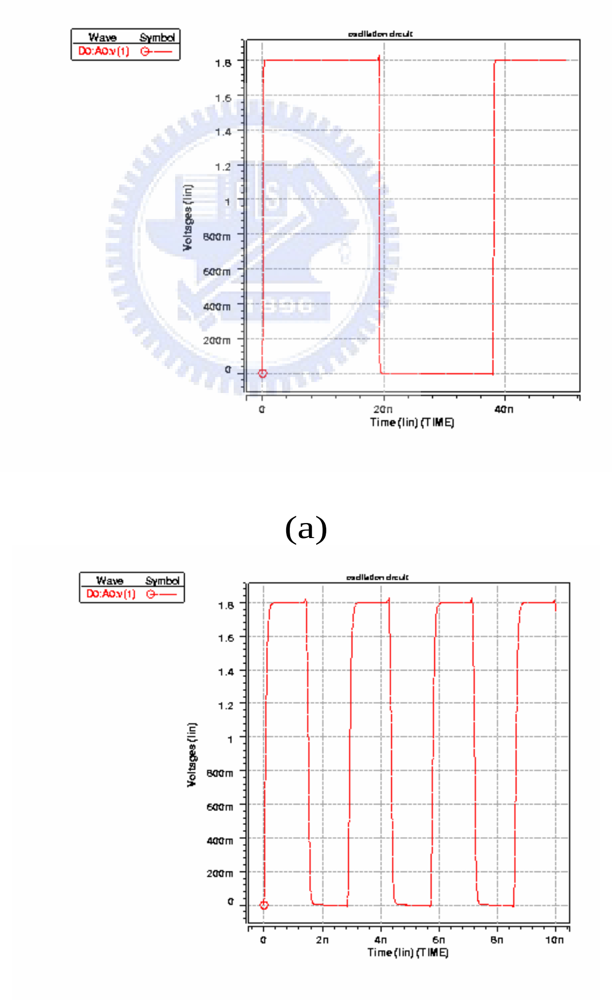


Figure 2.4 Test architecture of system-level interconnect test for SoC ICs.

2.2.2 Effectiveness of Oscillation Ring Test Scheme

2.2.2.1 Effectiveness for Delay Fault

The simulation results are shown in Figure 2.5, where Figure 2.5(a) shows the oscillation signal of the longest ring; and Figure 2.5(b) shows the oscillation signal of the shortest rings. The cycle time of the longest rings (with nine interconnects) is about 38ns and that of the shortest rings is about 2.8ns. Thus, the oscillating frequency ranges from 26 MHz to 357 MHz, and this shows that this oscillation detection scheme is feasible.



(b)

Figure 2.5 Simulated waveforms of the longest (a) and shortest rings (b) of benchmark circuit *hp*.

2.2.2.2 Effectiveness for Crosstalk Faults

In order to verify that the proposed architecture can be applied to detect crosstalk-induced glitches, we conduct HSPICE simulation with TSMC 0.18μm technology. An oscillation signal is generated on a ring as shown in Figure 2.4, and a 1 mm wire with three times of normal coupling capacitance is assumed. The results are shown in Figures 2.6 and 2.7. Figure 2.6 shows the oscillation signal on the ring, the induced glitches on the victim net, and the output of the counter. The crosstalk-induced glitch shown in Figure 2.6 can be detected and verified since the counter changes the state on every positive glitch.

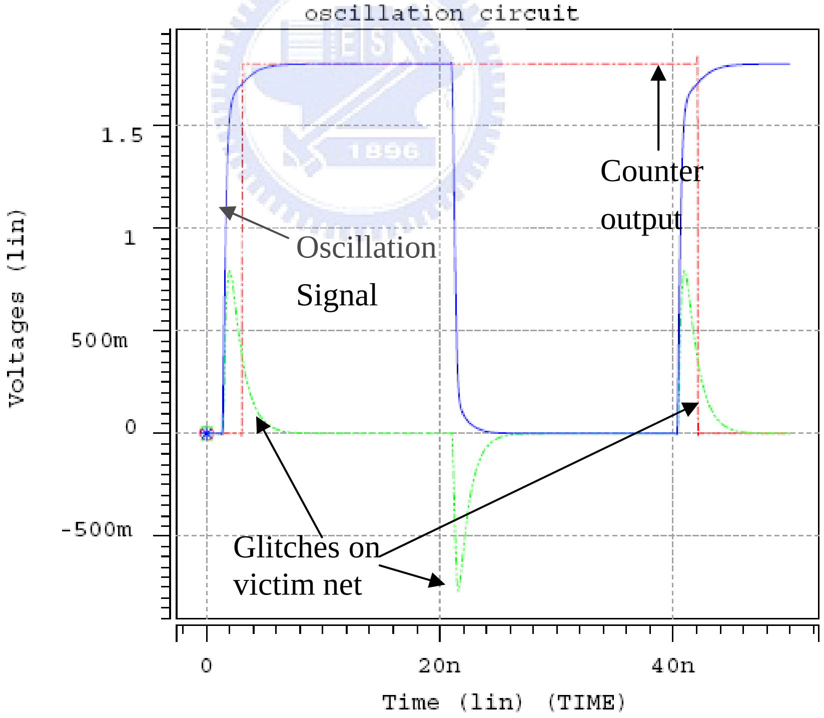


Figure 2.6. Oscillation signal on the ring, induced glitches on the victim net, and counter output.

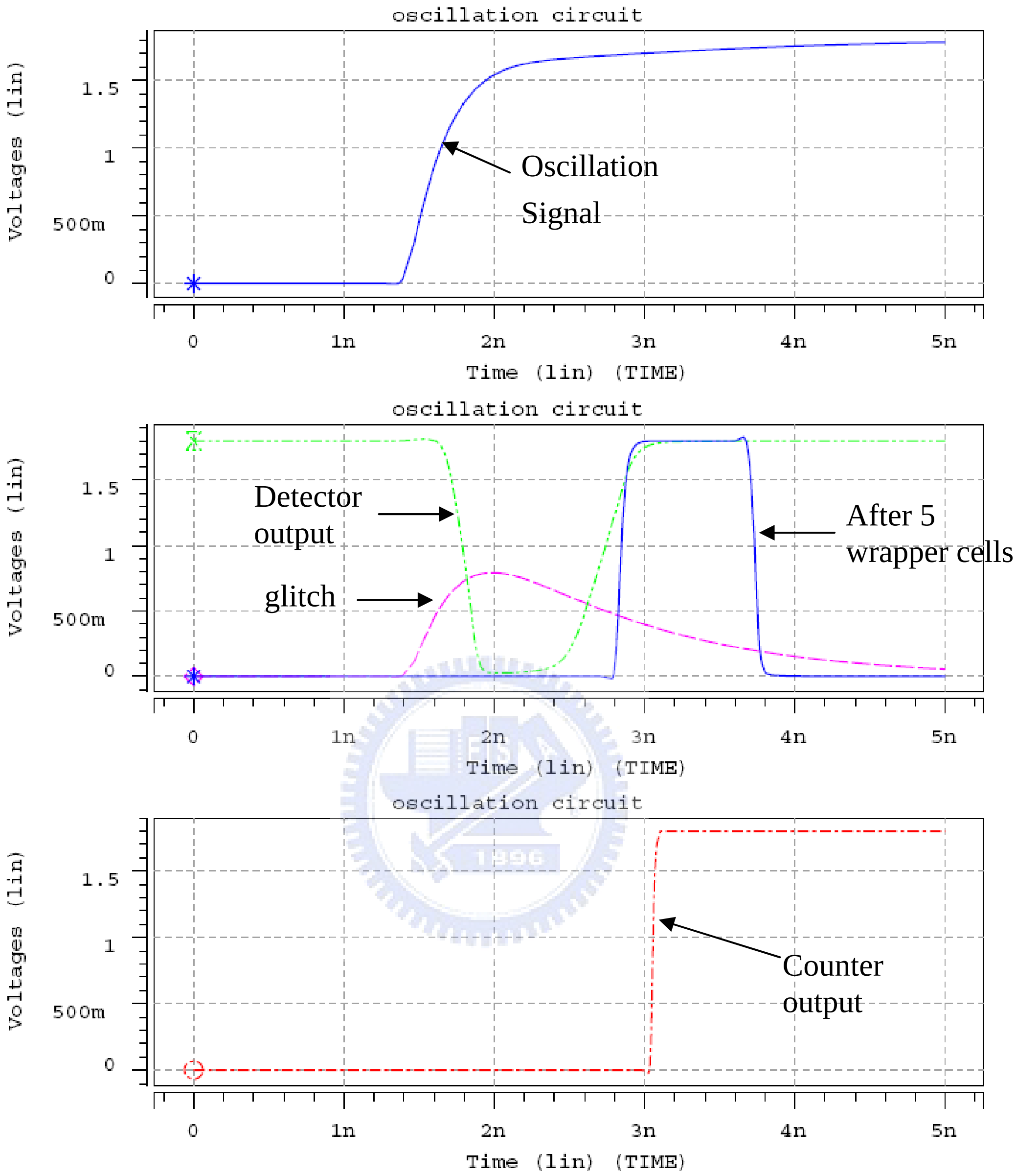


Figure 2.7. Illustration on how the glitches are detected, an oscillation signal (top), the resulting crosstalk-induced glitch, the detector output, and the signal after 5 wrapper cells (middle), the counter output with the verified state change (bottom).

Figure 2.7 gives an illustration on how to detect the glitches. The oscillation signal is shown in top of Figure 2.7, and the induced positive glitch, whose peak value is about 0.8V, is shown in the middle set of figures. This glitch is amplified by a detector, which is a specially designed inverter in our P1500-compliant input wrapper cell. We may adjust the *W*/*L* ratio of the detector's transistors to determine the detection threshold of glitches [75-76]. For example, in our experiment we set $(W/L)_{pu}/(W/L)_{pd}$ to be 1/4. In other words, the width of the pull-down nMOS is four times that of the pull-up pMOS, while the channel lengths of both transistors are set to the minimum. Since the positive crosstalk glitch and the negative glitch are symmetric, we only need a design to detect either a positive glitch or a negative glitch. Here we just give the basic detection principles for the positive glitch detection shown in Figures 2.6 and 2.7.

The detector's output is passed through a chain of wrapper cells. In our experiment, there are five wrapper cells in the chain, and it can be seen that a near rectangular pulse is formed. This pulse is used to trigger a two-port T-type flip-flop (2P-TFF) successfully without causing any setup/hold time violation. The 2P-TFF can be triggered by two different signals, one port is triggered by the crosstalk glitch signal and the second port is trigger by the system clock to scan out the counter contents. In the oscillation test mode, this 2P-TFF is triggered by the amplified glitches and acts as a counter. When we need to scan out the counter contents, it is triggered by the system clock. All the transistors, except for the detector, are minimum-sized.

The crosstalk is caused by excessive coupling capacitance between adjacent wires, and it can incur two types of errors: glitch and delay [75-76, 108]. When there is a signal transition in the aggressor while the victim signal is stable, a crosstalk-induced glitch appears in the victim net. On the other hand, a crosstalk-induced delay occurs when the victim net makes a signal transition opposite to the direction of the aggressor net's signal at roughly the same time. The crosstalk-induced delay is just a superposition of the original signal in the victim and the glitch induced by the aggressor [75-76]. Therefore, it is possible to detect crosstalk-induced delay simply by detecting induced glitches [75-76].

2.2.3 IEEE 1500 Compliant Wrapper Cell Design

This subsection demonstrates the detection unit design for crosstalk glitch fault, and how we apply it for interconnect detection and diagnosis in system-level interconnects. Section 2.2.3.1 shows the pulse detector design for crosstalk glitch, and Section 2.2.3.2 shows IEEE 1500 compliant wrapper cell design which embeds the pulse detector for interconnect crosstalk glitch detection.

2.2.3.1 Pulse Detector

In order to detect the crosstalk glitch in interconnects, we design a pulse detector to latch the glitch signal. For details, please refer to Chapter 3.

A glitch is a pulse, and it can be detected by a pulse detector. In order to detect the crosstalk-induced glitch with a given glitch peak, the pulse detector should be able to be adjusted its detection threshold. A pulse detector (PD) with an adjustable detection threshold is shown in Figure 2.8. The detector consists of two major components: an inverter (INV1) that is used to adjust the detection threshold, and a pseudo static latch (the remaining part) that is locked to “1” once a pulse is detected. The enabling of the latch comes from V_{DD}, and the input V_{DD} is controlled by a pass transistor. The pass transistor is controlled again by two inverters; one of which, i.e., INV1, is able to be adjusted its detection threshold by changing its W/L values for pull-down nMOS and/or pull-up pMOS. The latch can be reset by a reset input. Whenever a glitch whose amplitude is high enough to be picked up by INV1, the pass transistor will be turned on and the latch is set to “1”, indicating a glitch is detected.

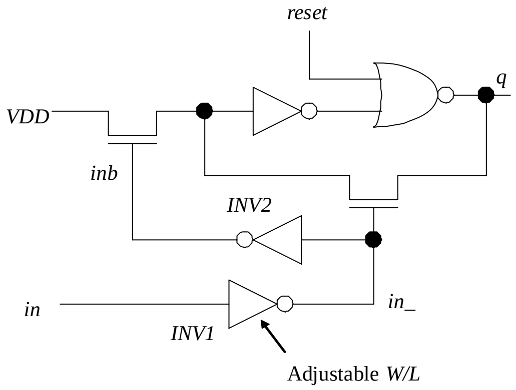


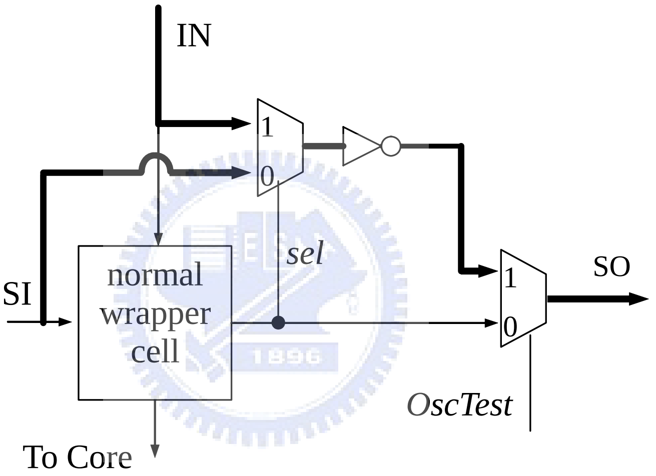
Figure 2.8. A pulse detector (PD) with an adjustable threshold by W/L ratio of INV1.

2.2.3.2 Wrapper Cell Design with Embedded Pulse Detector

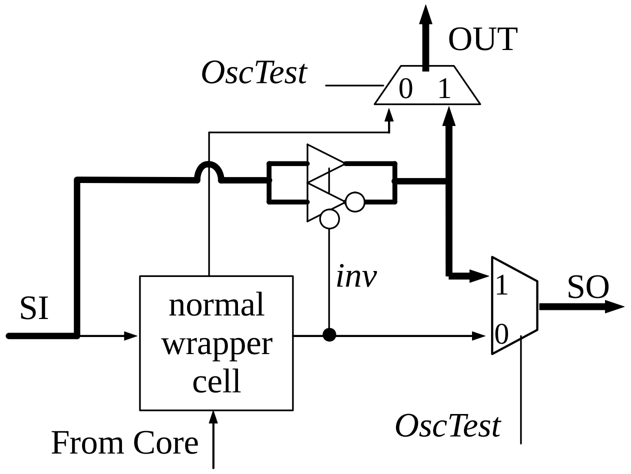
An oscillation ring for interconnect test consists of interconnect wires and part of the scan path in each core where the ring passes. Therefore, a wrapper cell must provide a path between input/output ports and scan in/scan out ports. If oscillation test is used to test wires attached to/from pads, the boundary scan cells also have to be modified in a similar way. In order to facilitate the scheme, the IEEE Standard 1500 compliant boundary wrapper cells need to be modified. In this subsection, the modified wrapper cell design is presented.

A normal wrapper cell provides two types of paths: a scan path connecting all wrapper cells into a shift register, and an interface buffering between core internal and the wire connected to the pin. Whenever oscillation test is applied, a third combination path must be provided. For an input pin, the wrapper cell must connect the pin input (IN) to scan output (SO), while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session.

The modified wrapper cell design is shown in Figure 2.9 for input and output cells. In each cell, two MUXs are added for path selection. For an input wrapper cell, the extra paths are SI→SO and IN→SO, while for an output wrapper cell the extra paths are SI→SO and SI→OUT. The added inverting and non-inverting buffers in output cells are used to provide odd inversions on the oscillation ring path to generate oscillation signals for the OR test. *OscTest* is a global control signal, while *sel* is used in the input wrapper cell and *inv* is used in the output wrapper cell. Signals *sel* and *inv* are individually set and are scanned into the wrapper cells before an OR test session.



(a)



(b)

Figure 2.9 Modified wrapper cells: (a) input cell (b) output cell.

In either the normal mode or the IEEE Std.1500 test mode (*OscTest* =0), modified cells act as normal wrapper cells. In the OR test mode (*OscTest* =1), the part of “normal wrapper cell” is bypassed. For an input cell, *sel* is used to select either SI→SO or IN→SO, depending on the position of the input wrapper. If the cell connects an external interconnect to the internal scan path, it is configured as IN→SO. Otherwise, it is configured as SI→SO. For an output cell, the bit information stored in the cell is used for inversion control *inv*, which decides whether the passing signal should be complemented. This is also applied to buffered interconnects where inverters are used for timing closure and signal amplification.

A summary of control signals for the modified wrapper cells shown in Figure 2.9 is given in Tables 2.1 and 2.2, respectively

Table 2.1 Control signals for the modified input wrapper cell.

OscTest	*Sel*	Comments
1	1	~IN → SO (OscTest Mode)
1	0	~SI → SO (OscTest Mode)
0	–	normal or IEEE Std. 1500 test mode

Table 2.2 Control signals for the modified output wrapper cell.

OscTest	*inv*	Comments
1	1	SI → SO and SI → OUT (OscTest Mode)
1	0	~SI → SO and ~SI → OUT (OscTest Mode)
0	–	normal or IEEE Std. 1500 test mode

2.2.4 Delay Measurement Formula

Further to our formula of counter-based delay detection scheme in (2.1), we explore the delay measurement in this subsection. Since the frequency of each ring is predetermined during the design phase, a wire delay fault is detected and measured by inspecting the contents of the delay counters. Let the oscillation frequency of the rings, according to the timing specification, be $f_{min} \le f_i \le f_{max}$, with the unit of measuring T_0 ($= n/f$). Thus, we have $n_{min} \le n_i \le n_{max}$, where $n_{min} = f_{min} \times T_0$ and $n_{max} = f_{max} \times T_0$. Let ξ be the resolution of delay measurement, and ε be the maximum measurement error. Since a counter's maximum measurement error is ±1, the requirement for ε should be the reciprocal of f_{min} and T_0.

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \le \zeta \qquad (2.2)$$

An example for delay measurement is given as follows. Let the frequency specification of the oscillation rings be 4 MHz to 400 MHz and ξ be 0.001, implying the counter content n_{min} is at least 1000. From (2.2), we have the required T_0 to be 250μs. This example illustrates the feasibility of the oscillation test scheme from a measurement prospect, and this frequency specification is actually compliant with ATE specifications.

2.3 Interconnect-driven Floorplanning

Since one of our main target is to detect and diagnose interconnects, we would like to explore crosstalk noise and how this signal integrity of interconnects affects interconnect performance and delay fault detection, which serves as system-level framework for our proposed oscillation ring test.

2.3.1 Crosstalk Noise and Signal Integrity

In [73-74], crosstalk-induced noise has become a key problem in interconnect optimization when technology improves, spacing diminishes, and coupling capacitance/inductance increases. Buffer insertion/sizing is one of the most effective and popular techniques to reduce interconnect delay and decouple coupling effects to meet timing specification.

For any interconnect wire, we show a buffer driver model and a wire π-model in Figure 2.10. Then, for any adjacent wires, we show a crosstalk glitch noise with an amplitude of χ from a crosstalk-induced current due to coupling capacitance in Figure 2.11. Further, we consider a 2-dimension floorplanning with feasible regions of buffer insertion and interconnect routing, and show the intersection of both timing and noise slack constraints in Figure 2.12 with consideration of blockages in interconnects.

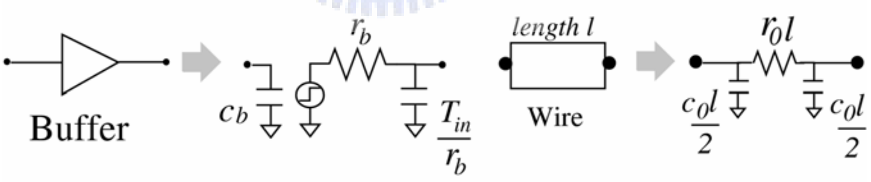


Figure 2.10 Switch-level RC circuits for buffers and wires.

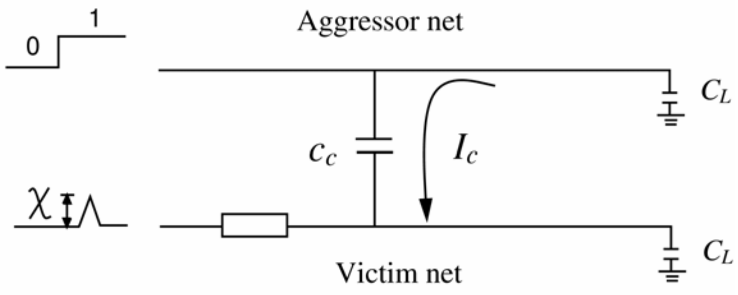


Figure 2.11 Noise due to crosstalk-induced current.

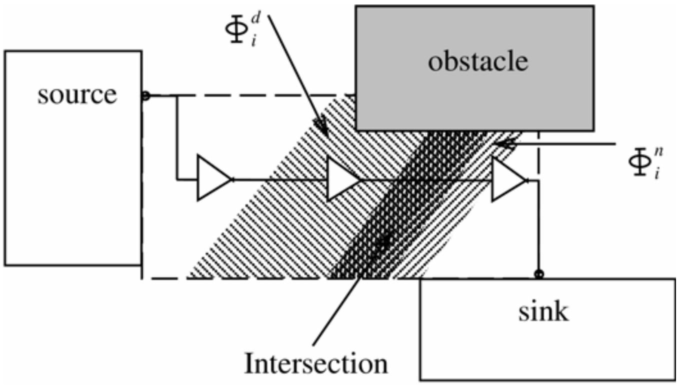


Figure 2.12 Respective feasible regions Φ_i^n, Φ_i^d and $\Phi_i^n \cap \Phi_i^d$ for inserting a buffer that meet the delay, noise and both delay and noise constraints.

In Figure 2.13, we show how we place buffers to satisfy timing constraint in any interconnect. For crosstalk-induced glitch noise, we show coupling capacitance among the victim line and its neighboring aggressor nets, and the coupling distance and length of coupling capacitance in interconnect topology.

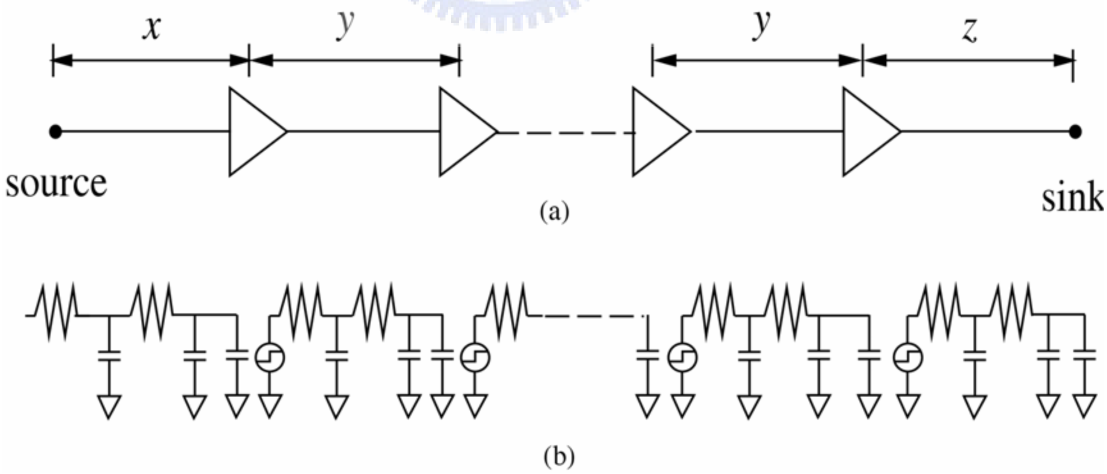


Figure 2.13 (a) Buffer placement: *x* is the optimized length between the source node and the first buffer, *y* is the optimized length between every pair of neighboring buffers, and *z* is the length between the last inserted buffer and the sink node. (b) The corresponding buffer model and wire (π) model.

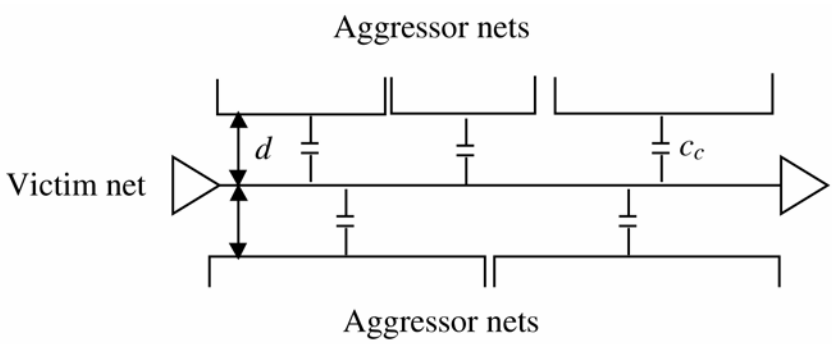


Figure 2.14 The victim net suffers from multiple aggressor nets for the coupling capacitance.

In Figure 2.15, we show all the possible cases in the intersection of respective feasible regions timing (Φ_i^d) and noise (Φ_i^n) constraints.

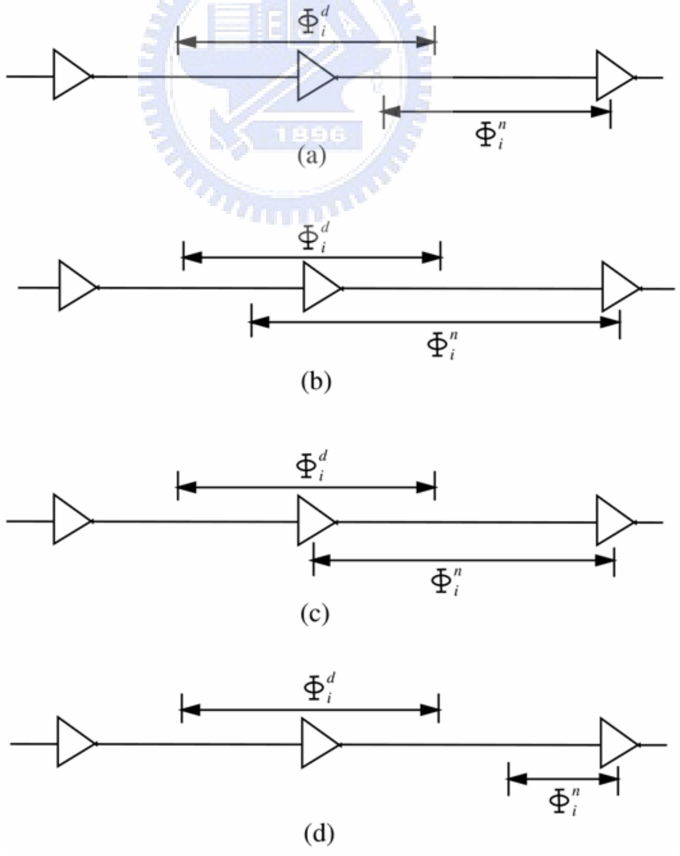


Figure 2.15 Four cases for the intersection of Φ_i^d and Φ_i^n.

Above figures show how we consider crosstalk noise and signal integrity issues in interconnects, and buffer insertion techniques to meet the timing and noise constraints. Thus, noise-aware buffer planning for interconnect-driven floorplanning can improve timing closure and design convergence. This work paves the way for our central oscillation ring test scheme in system-level interconnect detection and diagnosis problem with buffer insertion techniques to meet timing, noise and other possible constraints, is any.

2.3.2 System-Level Framework for Oscillation Ring Test

As a fundamental framework for system-level interconnects study, this work shows the feasibility and effectiveness of applying our proposed oscillation ring scheme in system-level interconnects [72, 77-80] with buffer insertion in [73-74]. In Figure 2.4, we show the system-level framework for our proposed oscillation ring test architecture targeted at interconnect faults including delay and crosstalk-induced glitch faults in additional to traditional stuck-at and open faults. This test architecture implements the IEEE Std. 1500 compliant core test standard.

For detailed description of test operations, we will formally define in Chapter 4 for complete oscillation ring test methodology and in Chapter 5 for thoroughly exploration theoretically and experimentally of interconnect diagnosis to achieve the optimal diagnosis resolution and the maximal diagnosability.

2.4 Interconnect-driven Routing

In [81-82], we study interconnect routing problem with testability and yield enhancement. We have two approaches; one is to apply the oscillation ring test methodology as Design for Test (DfT) technique, and the other is to reduce and balance the interconnect congestion to improve yield.

Interconnect global routing in nanotechnology contributes the congestion variation in routing stage, and typical interconnects may contain thousands of thousands in routing complexity among increased metal layers and vias. Therefore, we should study the properties of interconnects, such as manufacturing technology, scaling effects on physical sizing, etc.

- Nanotechnology Manufacturing Process
  - Scaling Effects
  - Process Variation
- Congestion Effects
  - Chemical Mechanical Polishing (CMP)
  - Optical Proximity y Correction (OPC)
  - Multiple Fault Probability
  - Crosstalk and signal integrity

In addition to the above, adaptive (dynamically regenerated test rings) and concurrent approaches are also two attractive features because they makes it possible to dynamically reconfigure test pattern generation algorithms which result in reducing test time and allowing easy interconnect test ring construction changes.

2.4.1 Applications of the Oscillation Ring Test Methodology as a DfT Technique

The oscillation ring (OR) test and its diagnosis scheme for interconnects based on the popular IEEE Std. 1500 are integrated into the multilevel routing framework to achieve testability enhancement. We augment the traditional multilevel framework of coarsening followed by uncoarsening by introducing a preprocessing stage of interconnect oscillation ring detection (IORT) that analyzes the oscillation ring structure for better resource estimation before the coarsening stage, and a final stage of interconnect oscillation ring diagnosis (IORD) after uncoarsening that improves testability to achieve 100% interconnect fault coverage and maximal diagnosability.

2.4.2 Congestion versus Design for Yield

We present a heuristic in [81-82] to reduce and balance routing congestion to optimize the multiple-fault probability, chemical mechanic polishing (CMP) and optical proximity correction (OPC) induced manufacturability, and crosstalk effects, for yield improvement. Our target is to show that our congestion-guided router can reduce and balance routing density to improve routing quality for yield enhancement through average congestion reduced and standard deviation of routing congestion smaller in MCNC benchmark circuits in addition to achieve 100% routing completion.

2.4.3 Previous Multilevel Routing Framework

In [70], Lin and Chang propose a multilevel approach for full-chip routing, which considers both routability and performance. This framework integrates global routing, detailed routing, and resource estimation together at each level, leading to a more accurate routing resource estimation during coarsening and thus facilitating the solution refinement during uncoarsening.

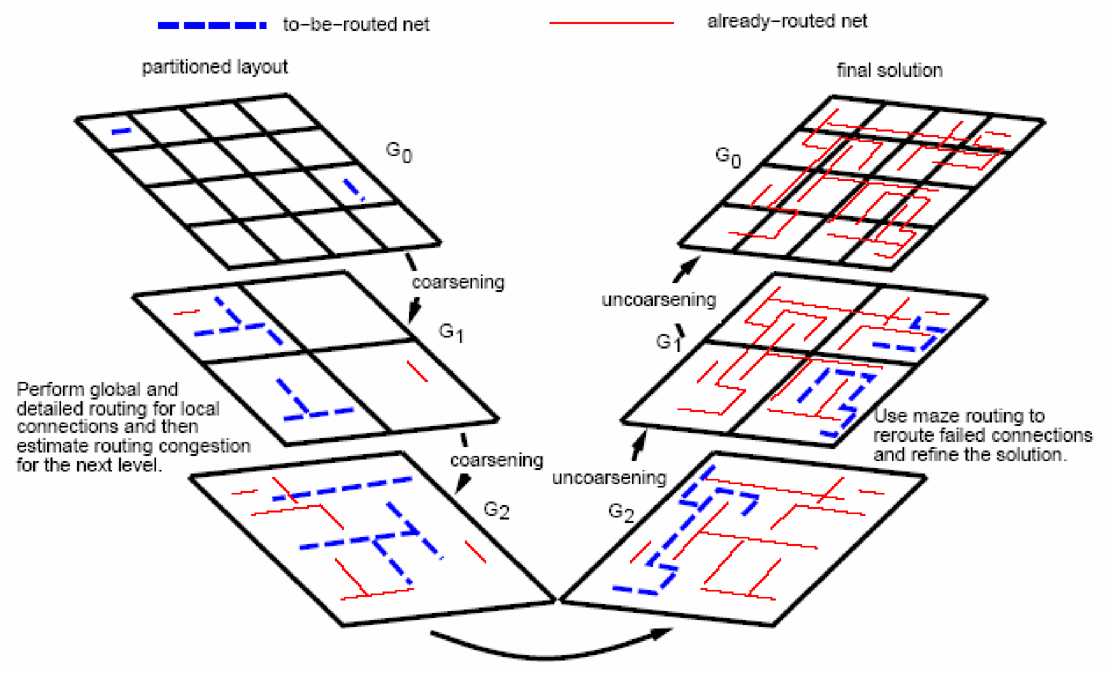


Figure 2.16. Previous Multilevel Routing Framework Flow of [70].

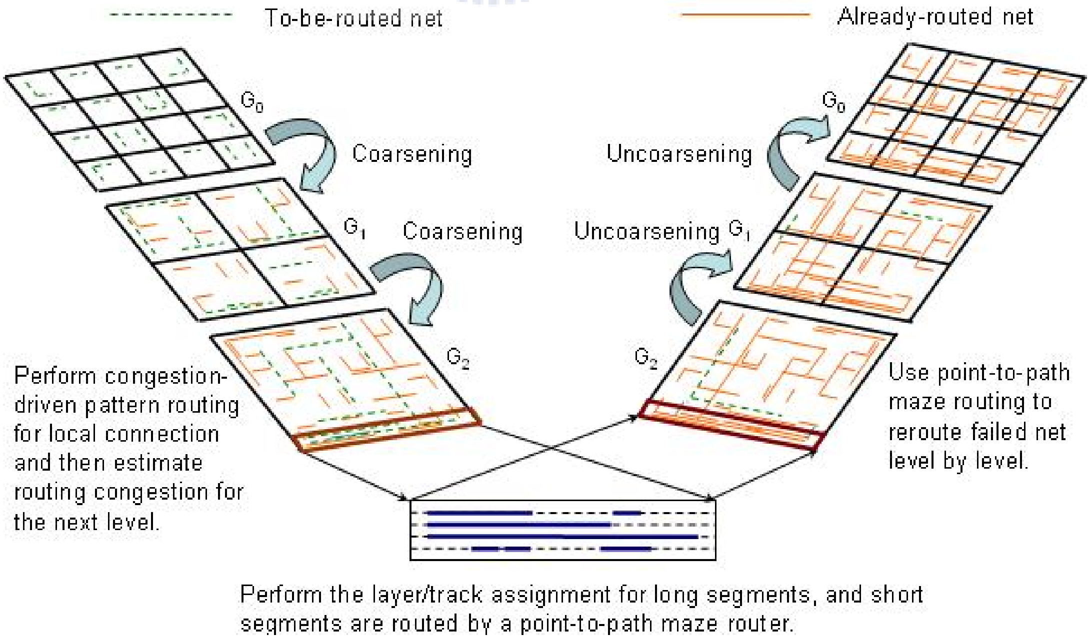


Figure 2.17 Crosstalk-Driven Multilevel Routing Framework Flow [49].

In [49], Ho and Chang propose a novel framework for fast multilevel routing considering crosstalk and performance optimization. To handle the crosstalk minimization problem, they incorporate an intermediate stage of layer/track assignment into the multilevel routing framework.

2.4.4 Our Integrated Multilevel Routing Framework

In Figure 2.13 and [81, 82], we incorporate the traditional multilevel framework by introducing a preprocessing stage of Oscillation Ring Detection (ORT) that analyzes the oscillation ring structure for better resource estimation before the coarsening stage, and a postprocessing (final) stage of Oscillation Ring Diagnosis (ORD) after uncoarsening that improves testability to achieve 100% interconnect fault coverage and maximal diagnosability.

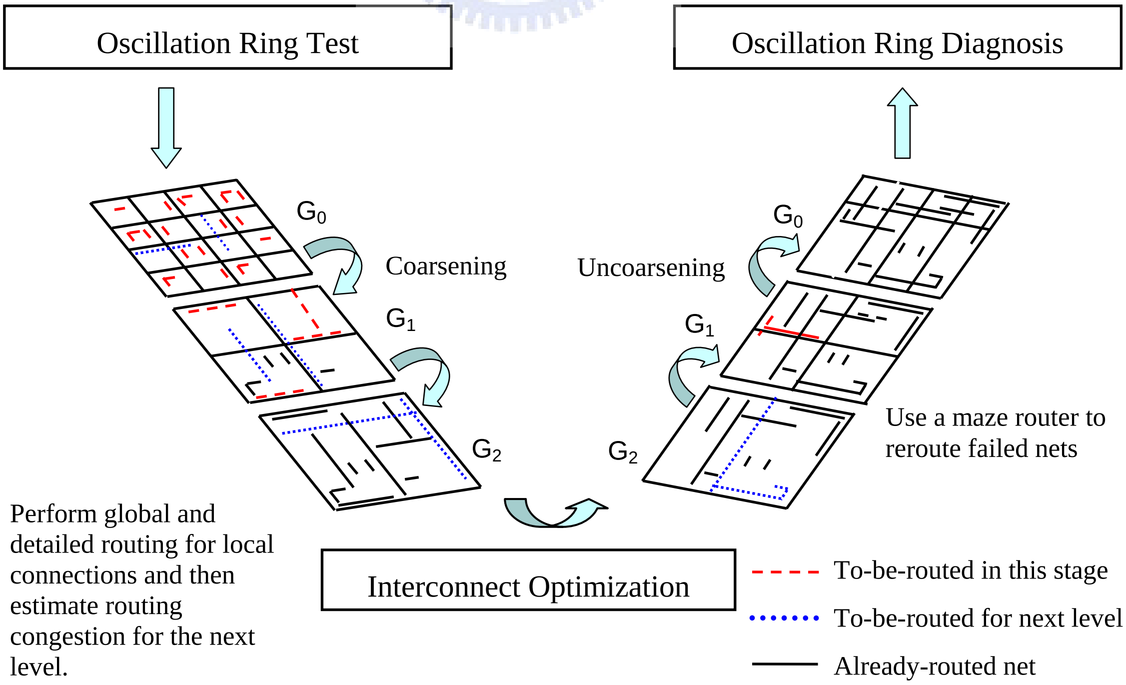


Figure 2.18. Our Integrated multilevel routing framework with Testability and Yield Enhancement.

2.5 Assumptions and Limitations

This section gives the assumptions and limitations of our study. The assumptions described below are used throughout this thesis, unless stated otherwise.

- We assume that at most single fault can be detected and diagnosed, i.e. single-fault diagnosable. This restriction represents a suitable balance between testability, diagnosability and program complexity for any interconnect structure, and is thus a reasonable assumption for the purpose of interconnect testability and diagnosability design and analysis. Based on this assumption, for interconnects, the detection goal is to achieve 100% fault coverage, and the diagnosis goal is to achieve optimal diagnosis resolution to uniquely identify and locate any faulty interconnect net.
- For simplicity, we shall focus on interconnect bus structure for study the unification detection scheme in this thesis, especially for high-speed synchronous circuits. For study of interconnect testing by oscillation ring scheme, generalization to any interconnect topology is achieved for both the interconnect detection problem and the interconnect diagnosis problem.
- Our interconnect models of theoretical studies deal with two-terminal nets for interconnect detection purpose in Chapter 4, and multiple-net-segments for interconnect diagnosis purpose in Chapter 5. All our theoretical findings, however, we have been justified by experiments using the benchmark circuits composed of two- as well as multi-terminal nets.

2.6 Interconnect-Centric Detection and Diagnosis Technologies

We explore the target problem of interconnect detection and diagnosis, and develop an oscillation ring technique to test interconnects. Below, we discuss briefly the interconnect detection problem in Section 2.5.1 and give formal academic research in Section 4. Further, we study the interconnect diagnosis problem briefly in Section 2.5.2 and completely in Section 5. Finally, we summary interconnect techniques in Section 2.5.3.

2.6.1 Interconnect Detection Technology

We propose an Oscillation Ring Test Methodology for interconnect test to achieve 100% interconnect detection under single fault assumption [77, 78]. The formal interconnect detection model, formal definitions and related theorems are introduced in Chapter 4. For SoC interconnect testing, IEEE Std. 1500 greatly affects the interconnect architecture decision and wrapper cell designs.

Specifically, the following three properties of OR schemes for interconnect tests are desirable: (1) The fault coverage of Oscillation Ring Test Scheme is approximately linearly dependent on test time in terms of the number of test rings, and the experimental results show this linearity characteristics of OR hold for every benchmark, (2) The different slopes of the linearity is up to interconnect structures, and more strict slope between fault coverage and the number of test rings shows that the interconnect structure is suitable for longer ring construction property, (3) From the above two linearity and slope characteristics, we distinguish the important properties between OR and traditional logic test: OR holds linearity of fault coverage but traditional logic testing saturates. In other words, OR does not have effective test patterns, but traditional logic testing does have effective test patterns. Therefore, the effectiveness of test rings is mainly determined by the interconnect topology of SoC circuits, and is observed by the sharpness of slope.

2.6.2 Interconnect Diagnosis Technology

We propose an Oscillation Ring Diagnosis Methodology for interconnect diagnosis problem to achieve the optimal diagnosis resolution or the maximal diagnosability under single fault assumption [79, 80]. The formal interconnect diagnosis model, formal definitions and related theorems are introduced in Chapter 5. For interconnect diagnosis problem, IEEE Std. 1500 compliant OR test architecture and interconnect detection algorithms are still applied. However, we explore further to locate the faulty net segment in addition to detect the faulty test ring.

Specifically, the following three properties of OR diagnosis schemes for interconnect tests are desirable: (1) Oscillation Ring Test Scheme (ORT) is regarded as a preprocessed stage in interconnect diagnosis problem. (2) For our interconnect diagnosis problem, we include diagnosis test rings in addition to our original detection test rings in ORT stage, and we refer to total test rings in this straightforward ring construction as Predetermined Interconnect Oscillation Ring Diagnosis (PIORD). (3) In order to further optimize interconnect diagnosis time, we propose two interconnect optimization techniques: Adaptive approach and Concurrent approach. We expect that adaptive IORD will greatly improve interconnect diagnosis cost dynamically, concurrent IORD will also improve PIORD but limited by interconnect structures.

2.6.3 Summary of interconnect Technologies

We study the interconnect detection problem due to crosstalk faults, and propose a unified detection scheme together with a crosstalk glitch detector design [75, 76]. Then, we explore the interconnect detection [77, 78] and diagnosis [79, 80] problems by our proposed oscillation ring test methodology. Also, we applied IEEE Std. 1500 compliant system-level interconnect detection and diagnosis scheme in routing stage to enhance interconnect testability [81, 82]. Finally, we research the gate-level wire detection problem also by oscillation scheme with finite state synthesis [83, 84].

2.7 Previous Work

This subsection shows the related works covering two major targets of interconnect detection and diagnosis in Section 2.7.1 and oscillation ring scheme (also known as ring oscillator in some fields) in Section 2.7.2.

2.7.1 Interconnect Detection and Diagnosis Test Architectures & Algorithms

Interconnect test and diagnosis for various applications, such as printed circuit board (PCB), multi-chip module (MCM), and systems in package (SiP), have been studied extensively in the literature [7, 14, 61, 62, 89, 122, 124].

Previous works on interconnect test, including fault detection and diagnosis, focus mainly on traditional fault models including stuck-at and bridging faults. Those diagnosis algorithms include counting sequence, walking-0 and walking-1 sequence, maximum anti-chain, maximal independent test set [20, 21], etc. An efficient way to apply these tests is to exploit the boundary-scan architecture [58, 111, 117]. Many diagnosis algorithms presented in previous works focus mainly on special interconnect structures, especially for bus-oriented systems [108], sparsely interconnected systems [26], or FPGA designs [1, 46, 114]. The diagnosis of wire delay and crosstalk faults, often considered the most important segments of interconnect diagnosis, has attracted increasing attention since the process technology enters the deep submicron era. Much work has been done in these areas, including the development of fault models, test generation algorithms, and test methodology for delay tests [63] and BIST schemes for crosstalk faults [109, 111, 117]. However, the counting sequence and the maximal independent test set detect faults *without* diagnosing the faulty positions. On the other hand, the walking-one sequence can detect and diagnose all faults, but its test length is too long. Therefore, an interconnect test algorithm to diagnose all faults within a short testing period is not only desired but required. The new algorithm named the group, net, shifted net (GNS) sequence can detect and diagnose all faults within a much shorter testing period than previous diagnosis algorithms [62]. However, most previous work focus only on stuck-at, open or short faults, not crosstalk-induced glitches and delay faults due to nanotechnology effects.

There are many other pioneer researchers in interconnect test problem in additional to previous mentioned works. Shi *et al.* [106] studied the diagnosis problem on the interconnect diagnosis with randomized algorithm. They formulated the interconnect diagnosis problem as a two-dimensional graph problem and studied the complexity of that problem. Unlike the work in [21] which is based on graph mixing theory and adjacency analyses, they explored the behavior diagnosis on that architecture using the worst-case scenario—they proved that it is NP-complete to determine whether a given interconnects can have full diagnosis (detection and identification/location). Again, their target fault model is only short, and adjacency relations are known.

The conclusions in the above subsections lead to the architectural choice: general interconnect structure combined with a general graph topology analysis which gives the best area and routability trade-offs. A theoretical study of flexibility and routability was later presented based on a routing framework and model with congestion-guided weight which confirms the experimental results in [81, 82].

The worst cases occur when most nets are very long and are routed in some specially designed topologies, a complete tree structure of an entire SoC chip (see Figure 2.19 for a worst-case instance shown in [77-80]). Especially, global interconnects are often very long and often takes cycles to communicate in SoC ICs which leads to the motivation of our systematic study on interconnect diagnosis problem with the optimal resolution. Later in Chapter 5, we will show the formal definition of the optimal resolution of interconnect diagnosis which is roughly referred as uniquely identifying any fault. This work provides a theoretical insight to the *worst-case* performance of interconnect diagnosis problem by using that oscillation ring test architecture.

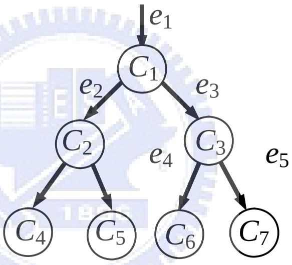


Figure 2.19 A worst-case scenario of interconnect structure or topology in SoC.

We considered a variation of the interconnect detection architecture modeled in Chapters 4 and 5 [77-80]. See Figure 2.4 for an illustration of the architecture and an interconnect routing on that architecture.

2.7.2 Oscillation Test Schemes

Oscillation based test is an efficient and effective method to detect faults in a circuit or a device [6, 58]. Recently, oscillation ring test is applied for system-level interconnects for delay faults and crosstalk glitch faults [77-78]. The proposed oscillation test methodology attacks the testing problem from a different perspective.

It modifies the storage elements such that oscillation signals can be generated according to the functional specifications of a given circuit.

2.7.2.1 Analog and Mixed Signal Domain

Ring Oscillator related techniques are developed in many prospects [6], [52], [58], [103] such as Selective Process Bias (SPB) of interconnects, Phase Noise, and etc. There are many works on oscillation tests. For example, Kaneko and Sakaguchi proposed an oscillation fault diagnosis method for analog circuits based on boundary search with perturbation model [58].

2.7.2.2 Digital Domain

With contrast with [6], Lee and Wu develop an oscillation test scheme in logical level (gate-level) [119-120]. [120] proposes a new test scheme, oscillation ring test, and its associated test circuit organization for delay fault testing for high performance microprocessors. For this test scheme, the outputs of the circuit under test are connected to its inputs to form oscillation rings and test vectors which sensitize circuit paths are sought to make the rings oscillate. High speed transition counters or oscillation detectors can then be used to detect whether the circuit is working normally or not. The sensitizable paths of oscillation rings cover all circuit lines, detecting all gate delay faults, a large part of hazard free robust path delay faults and all the stuck-at faults. It has the advantage of testing the circuit at the working speed of the circuit. Also, with some modification, the scheme can also be used to measure the maximum speed of the circuit. The scheme needs minimal simple added hardware, thus ideal for testing, embedded circuits and microprocessors. Further in

[119], a test scheme for the crosstalk fault based on the oscillation signal is proposed. It uses an oscillation signal applied on an affecting line and detects induced pulses on a victim line if a crosstalk fault exists between these two lines. It is simple and eliminates the complicated timing issue during test generation for the crosstalk fault in the conventional approaches. The test generation and fault simulation based on the scheme are described.

In our work, we have a total different approach and focus on system-level interconnects. However, later in Chapter 6, we adopt a different approach of finite machine synthesis to deal with gate-level oscillation ring test scheme.

Chapter 3

A Unified Approach to Detecting Crosstalk Faults of Interconnects in Deep Sub-micron VLSI

In this chapter, we consider a crosstalk detection problem for interconnect buses. The problem has important many applications to system-level interconnects, logic level design, congestion and routability of router in physical design. For the very deep sub-micron SOC VLSI, crosstalk becomes important in affecting performance and signal integrity of the circuit. In this chapter, two crosstalk fault effects, namely, glitches and the crosstalk-induced delay, in the SOC interconnect bus are analyzed and a unified scheme to detect them is proposed and demonstrated. The crosstalk induced delay is found to be superposition of the induced glitch and the applied signal at the victim line and is more important in affecting the circuit performance and being tested. A pulse detector with an adjustable detection threshold is proposed for detection of the glitch, consequently the induced delay. Several issues affecting the yield of the proposed testing scheme are discussed and Monte Carlo simulation experiments are conducted to show the feasibility of the scheme. Experimental results show that a testing yield of 92.915% can be achieved even under the 20% process variation.

3.1 Introduction

The crosstalk-induced noises have been attracting increasing attention as spacing between lines decreases and coupling capacitances increases on interconnection bus lines of the deep sub-micron VLSI. These noises affect the circuit performance in two ways: they may induce unexpected glitches, which may be captured by end latches to produce erroneous logic values, or they may cause unexpected signal propagation delay [10], [54]. These *crosstalk* issues should be considered during the design stage for performance, and they should be tested during the manufacture step.

The crosstalk noises on interconnect lines had been modeled and analyzed in many previous works. For example, they were studied by treating the interconnect lines as coupled lossy transmission lines [87],[125], and they were analyzed numerically [123],[124]. Simulation models for interconnect lines were also reported [14],[97]. Simplified lumped RC model for studying crosstalk noises was proposed and analyzed by many authors [23], [34], [92], [98]. Other issues for crosstalk, including fault avoidance, test generation, and test set evaluation had been reported in [65], [102], [116], [126]. In addition, various on-chip circuits for the measurement of crosstalk effects have been proposed [9],[100]. They can be used to measure glitch amplitude [100], or they can be used to characterize the crosstalk effects [9]. However, these circuits are generally sophisticated and their sizes are too large if they are to be considered in the BIST application.

In general, the two crosstalk effects, i.e., the induced glitches and the induced delay, require different techniques for their detection, which complicates the testing process and increases the testing cost. In this chapter, we investigate the origin of their occurrence, their relationship and identify their respective importance in affecting the circuit performance; and propose an approach which tests both glitches and delay in a unified way. Furthermore, the proposed method can be implemented with a very simple circuit. With this proposed method, fault detection for crosstalk noises can be greatly simplified.

This chapter is organized as follows. In the Section 3.2, a general distributed circuit model for the crosstalk faults for this study is first presented, and the timing issues caused the crosstalk effects are discussed. Then an analysis on the relationship between the two crosstalk fault effects is discussed in Section 3.3, where a unified test scheme for both faults is proposed. In Section 3.4, a simple pulse detector of an adjustable detection threshold for detecting glitch fault, then, the crosstalk-induced delay fault is presented. In Section 3.5, the issue of skews among aggressor signals on affecting this scheme is analyzed and discussed. In Section 3.6, the complete detection scheme with the Circuit under Test (CUT) and the proposed pulse detector is analyzed considering the manufacture process variation to show the effectiveness of the scheme. Finally, some concluding remarks are given in Section 3.7.

3.2 Preliminaries

In this chapter, we are concerned with the following fundamental problem:

- *Distributed circuit model for this study of the crosstalk fault*
- *Timing issues affecting the crosstalk effects*

3.2.1 Circuit Model for Crosstalk

For the current VLSI SOC chip interconnects, coupling crosstalk effects are mainly caused by parasitic capacitors between neighboring interconnection lines.

Hence, in this study, we focus on the capacitive crosstalk. Figure 3.1 shows the *bus circuit model* for analysis, where adjacent wires run in parallel. The middle wire is the *victim* net, while the other two wires are the *aggressor* nets. The wires are driven by inverters served as buffers with characteristic "ON" resistances $R_{on\text{-a1}}$, $R_{on\text{-a2}}$, and $R_{on\text{-}v}$, respectively. Each wire contains distributed wire resistance (R_{a1}, R_{a2}, and R_v) and capacitance (C_{a1}, C_{a2}, and C_v). A coupling capacitance (C_{c1}, C_{c2}) exists between two adjacent wires, and it causes the crosstalk effects. The output of each wire is connected with an inverter, which also serves as a buffer. These end inverters provide the load, C_{L1}, C_{L2}, and C_{Lv}, for the wires respectively. We assume that all wires are homogenous, i.e., $R_{a1} = R_{a2} = R_v = R_w$ for wire resistances, $C_{a1} = C_{a2} = C_v = C_w$ for wire capacitances, and $C_{L1} = C_{L2} = C_{Lv} = C_L$ for load capacitances, and $C_{c1} = C_{c2}$ for coupling capacitances. All aggressor signals are assumed to be synchronized first in order to maximize the crosstalk effects and later the skew effects between aggressor lines are discussed. The inverter attached to the end of each line consists of a pair of minimum-sized transistors.

In our analysis for the above interconnect structure, we conducted simulation with a TSMC 0.18μm technology. The wire length was set to be 1 mm, and the distance between adjacent wires is the minimum line spacing.

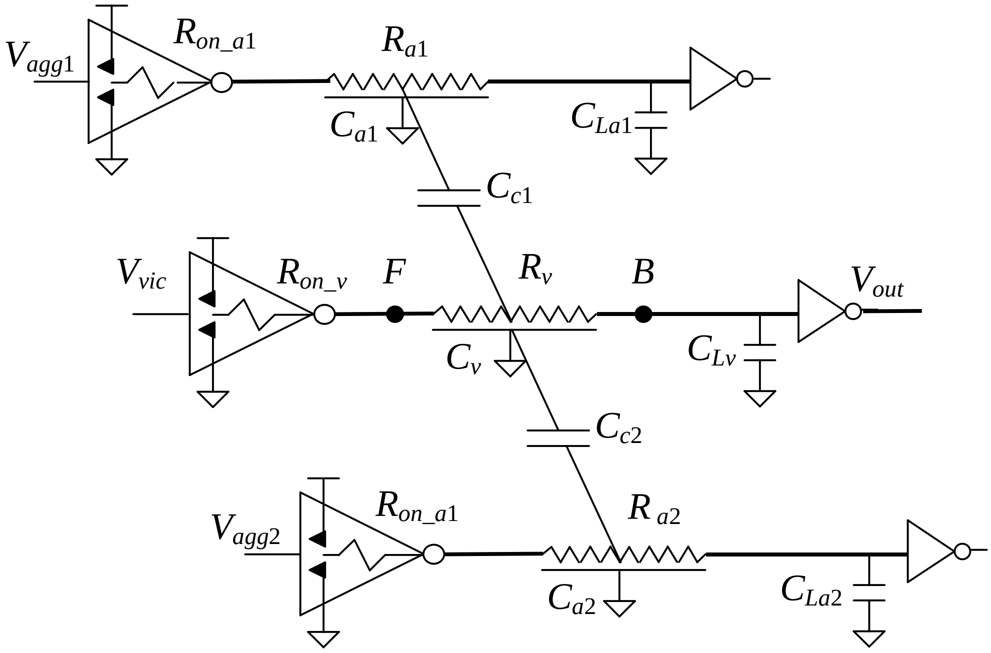
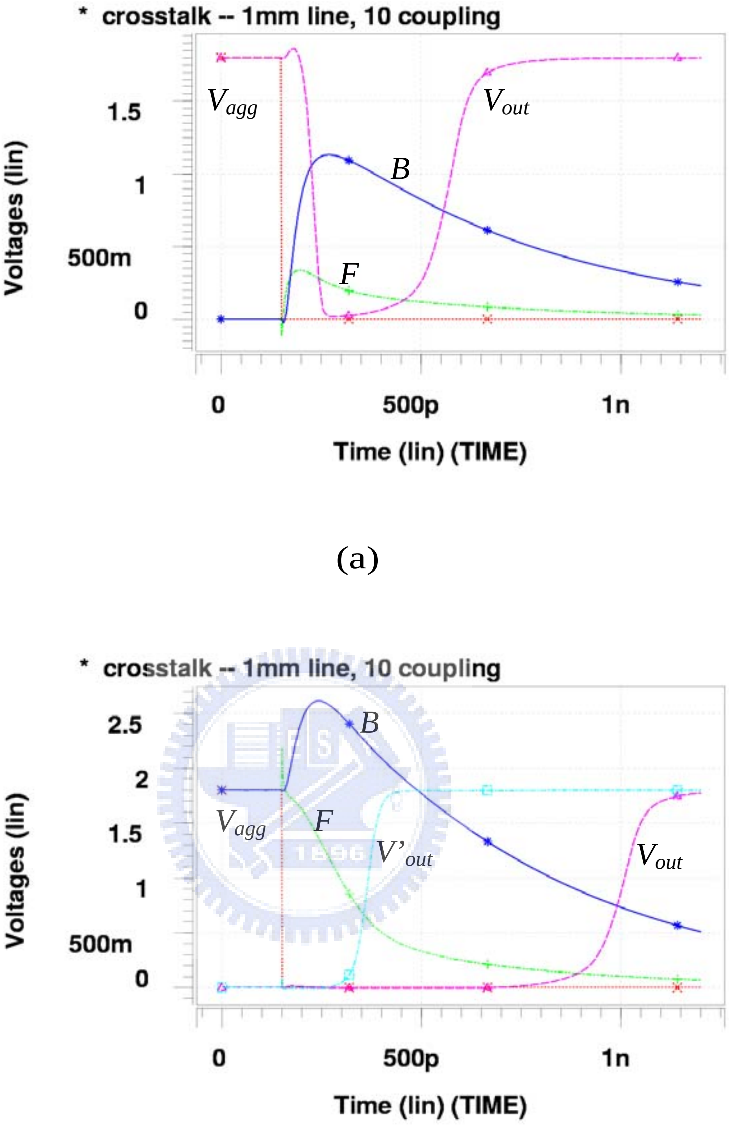


Figure 3.1 Circuit model for the crosstalk analysis.

3.2.2 Crosstalk Fault Effects

As mentioned previously, there are two types of crosstalk effects, namely, glitch and delay, on the victim line, depending on the signals applied. When a rising (or falling) transition is applied on the aggressor lines with the victim line sitting at a stable “low (or high)” signal, a positive (or negative) glitch is induced at the victim line. When the aggressor lines and the victim line are applied with opposite transition signals respectively, the signals at both the aggressor lines and the victim line will suffer a slow-down. Here we only consider the slow-down case of the victim line, since the signal slow-down of the aggressor is the same. Also, interconnects considered are busses which run in parallel with the same length.



(a)

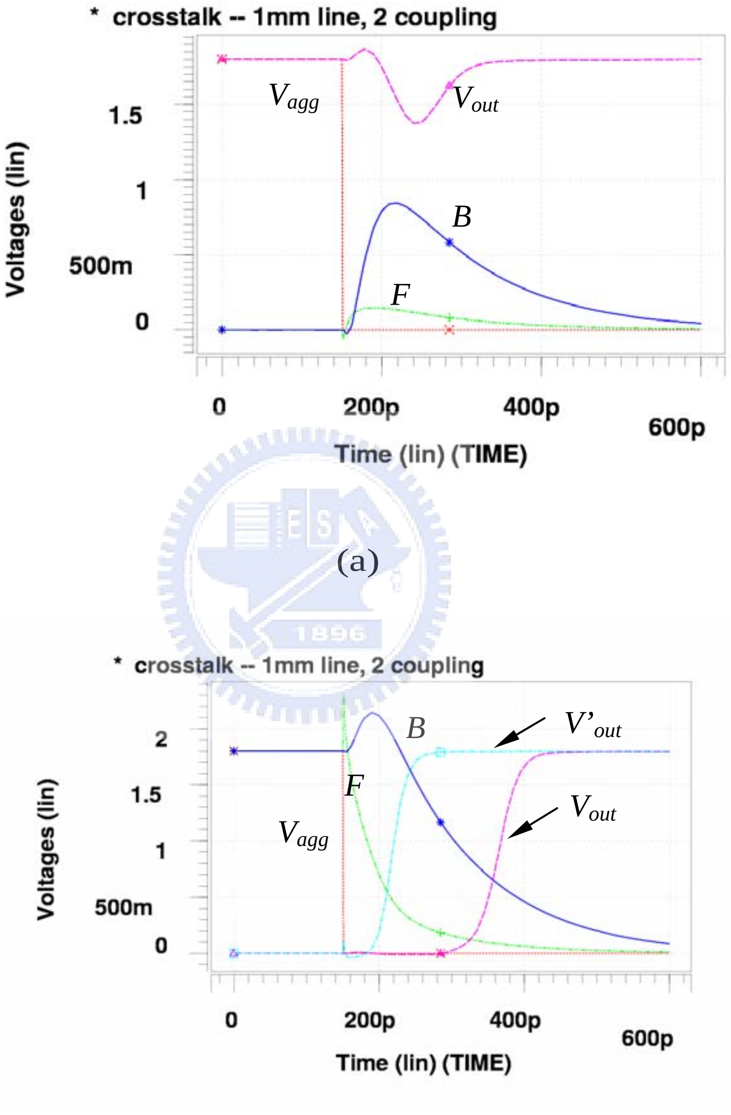
(b)

Figure 3.2 Simulated crosstalk effects for large enough coupling capacitance,
(a) the induced glitch and, (b) the induced delay.

Figures 3.2 (a) & (b) show the simulation results of glitches and delays under two different sets of inputs respectively. Figure 3.2(a) is the simulated waveforms of the induced glitches at the front end (node *F* in Figure 3.1) and the back end (node *B* in Figure 3.1) of the victim bus line, where the inputs of aggressor lines (V_{agg1} and

V_{agg2}) and the victim line (V_{vic}) are a step falling input and a d.c 1.8V, respectively. Signal V_{vic} is not shown in Figure 3.2 in order to improve the readability. The output, V_{out}, of the end-connected inverter is also shown in the figure. Since a distributed *RC* transmission line model was used for the analysis, the induced glitch at the front end (*F*) of the victim line initially is small and becomes larger at the back end (*B*) of the victim line. The glitch, when passing the end inverter, becomes a much more amplified negative pulse, V_{out}, which may be captured by a latch at the end of the victim interconnection line to cause an erroneous state. For Figure 3.2(b), in addition to the falling step input applied to the aggressor line, a rising step input is also applied to the victim line. Due to this applied rising step, it can be seen that the induced glitch at the victim line becomes a much slowly decreasing waveform. This glitch, when passing through the inverter, causes a large delayed waveform, V_{out}, of the inverter. (This can be seen by comparing this V_{out} with the waveform V'_{out}, which is the output of the inverter if no crosstalk effect is considered.) When the decreasing rate (i.e. glitch slope) of the glitch is slower, the larger the induced-delay will be.

Figures 3.3 (a) & (b) show another similar set of waveforms but with a smaller coupling capacitance, i.e., smaller crosstalk effect, where the amplitude of the induced glitch is not large enough to cause switching of the end inverter. However, in Figure 3.3(b), the output waveform at the inverter, V_{out}, still shows a significant delay compared to V'_{out}, due to the long slowly decreasing tail of the glitch waveform at node *B*. This reveals a fact that, under the same coupling condition of two bus lines, an induced glitch may not cause an erroneous switch but the crosstalk-induced delay may still cause a problem if the induced delay exceeds the specified delay of the bus line. From the above discussion, we see that, to detect crosstalk fault, both glitches and induced delays have to be tested, and among them, induced delays are more important to be tested since they more easily affect the circuit performance.



(a)

(b)

Figure 3.3 Similar simulated crosstalk effects for a smaller coupling capacitance, (a) the induced glitch and, (b) the induced delay.

3.3 Relationship between Crosstalk Glitches and Delay

In this section, we analyze the relationship between the two types of crosstalk effects, glitches and induced delays, and discuss factors that may affect these crosstalk effects.

3.3.1 Glitch vs. Delay

Figure 3.4 shows the SPICE simulation results for the analysis, where there are three (green, blue and red) sets of curves. The green set of curves show the results of 1mm lines with normal coupling capacitance, while the blue and red curves are results of double and quadruple coupling capacitance. In each set of curves, the curve marked with "+" is the response of the victim line when only a rising transition is applied at the input of the line, and the curve marked with "×" is the response of the same victim line but with only the crosstalk glitch effect considered, i.e., a static "0" is applied to the victim line. The curve marked with "✳" is the signal of the line with both excitations considered, i.e., the victim line is affected by the coupling effect and its own applied rising transition input. We can see that all the curves (✳) are exactly superposition of the first (+) and second (×) curves for each case. This means that the crosstalk-induced delay is in fact the crosstalk-induced glitch plus the original response of the victim line. Please note that the rising input of victim line passes an inverter as a driver, therefore, the crosstalk-induced delay in *B* in Figure 3.1 shows the superposition principle of both the inverted falling original signal and the induced glitch.

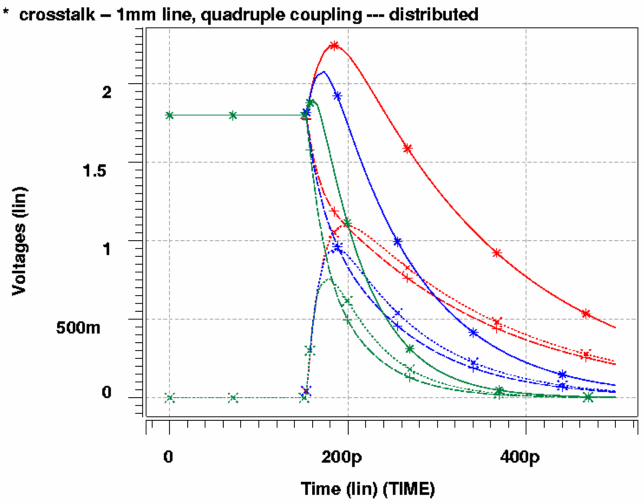


Figure 3.4 Superposition of crosstalk-induced delay.

The above result is obvious since the circuit treated is a linear circuit for which the superposition rule holds. The larger the glitch, the larger the induced delay. There is a monotonic relationship between the induced glitches and the induced delay, which is illustrated in Figure 3.5 where the simulated relationships between the peak of the induced glitches and the delay of the induced delay waveforms are depicted. As the amplitude of the induced glitch increases, the induced delay also increases.

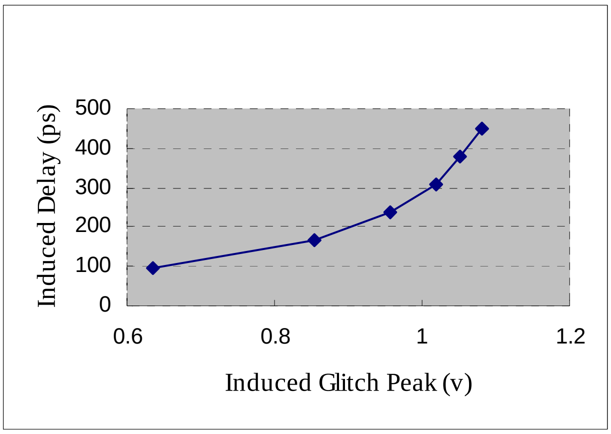


Figure 3.5 Monotonic relationships between the peak of the induced glitch and the induced delay.

The above monotonic relationship suggests a unified approach for crosstalk detection. If the induced glitch is detected, the induced delay can be also detected. For example, in Figure 3.5, if the specified delay of the interconnection line is 200 ps, we can detect if there are induced glitch faults with amplitude higher than 0.91 V. We can devise a detector to detect the induced glitches; and once these induced glitches are detected, the corresponding induced delays are detected. This will simplify testing of crosstalk-induced delay faults since to test an induced delay fault, a two-pattern test is required, not to mention the need of special clocking schemes to feed the patterns and relatively longer test time.

3.4 Pulse Detector with Adjustable Detection Threshold

A glitch is a pulse, and it can be detected by a pulse detector. In order to detect the crosstalk-induced glitch with a given glitch peak, the pulse detector should be able to be adjusted its detection threshold. A pulse detector (PD) with an adjustable detection threshold is shown in Figure 3.6. The detector consists of two major components: an inverter (*INV1*) that is used to adjust the detection threshold, and a pseudo static latch (the remaining part) that is locked to “1” once a pulse is detected. The enabling of the latch comes from V_{DD}, and the input V_{DD} is controlled by a pass transistor. The pass transistor is controlled again by two inverters; one of which, i.e., *INV1*, is able to be adjusted its detection threshold by changing its *W/L* values for pull-down nMOS and/or pull-up pMOS. The latch can be reset by a reset input. Whenever a glitch whose amplitude is high enough to be picked up by *INV1*, the pass transistor will be turned on and the latch is set to “1”, indicating a glitch is detected.

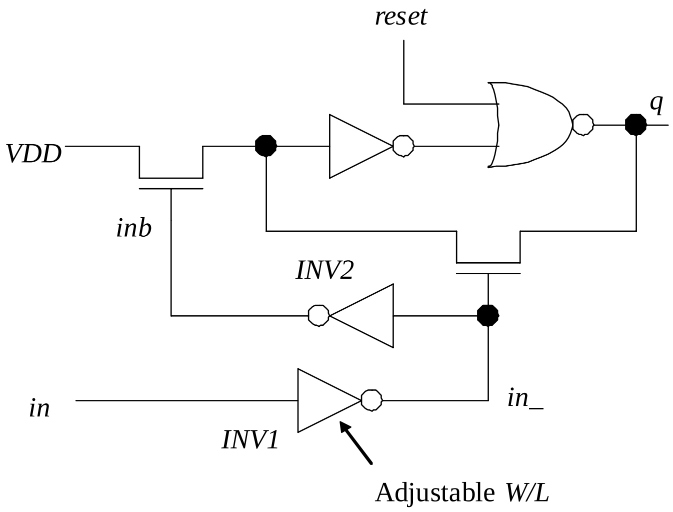


Figure 3.6 A pulse detector (PD) with an adjustable threshold by W/L ratio of INV1.

Figure 3.7 shows the simulated results on the threshold of detected pulse amplitude, V_{th}, versus $(W/L)_{p\_mos}/(W/L)_{n\_mos}$ of *INV1* in the pulse detector with the TSMC 0.18μm technology. The figure clearly shows that the detection threshold of the pulse detector is adjustable by changing the W/L ratio of the pull-up transistors (*pMOS*) and pull-down (*nMOS*) of *INV1*.

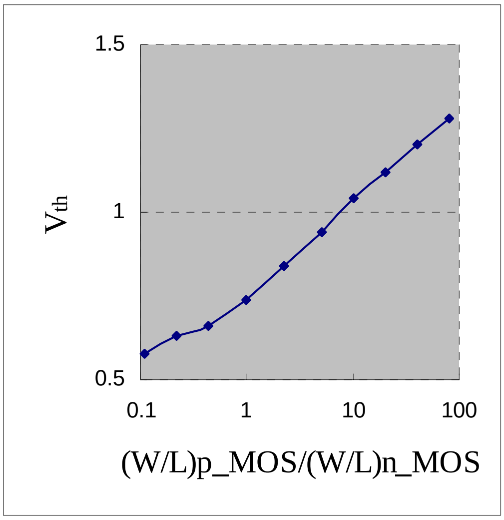


Figure 3.7 The simulated relationships between the threshold of detected pulse amplitude (V_{th}) with respect to the W/L ratio of the pulse detector (PD).

3.5 Some Considerations for Unified Detection Scheme

To make the scheme to be workable, some issues should be considered and investigated:

3.5.1 Glitch Amplitude and Width

In Figure 3.2, 3.3 and 3.4, the glitch pulse shape is seen to be affected by the magnitude of the coupling capacitance. In fact, it is also affected by the resistances of the wires. Figure 3.8 shows the simulated glitches for several different wire resistances and coupling capacitances. In the figure, Case 1 (red curve marked with ×) is for a *1mm line with 10x* unit-length coupling capacitance; Case 2 (purple curve marked with +) is for *1mm line with 6x* unit-length coupling capacitance; Case 3 (green curve marked with ◇) is for a *2mm line with 3x* unit-length coupling capacitance; and Case 4 (blue curve marked with △) is for a *2mm line with 2x* unit-length coupling capacitance. From the figure, we can see that the peak of an induced glitch is mainly affected by the coupling capacitance while the pulse width of the glitch is mainly affected by the wire resistance. In general, the combined *RC* constant value affects the peak and the width of the glitch.

Hence, in plotting the monotonic relationship between the induced glitch peak and the induced delay in Figure 3.5, we should consider the process variation effect of R’s and C’s which may vary due to the manufacture parameter tolerance. Figure 3.9 is a plot of the curve of Figure 3.5 obtained by Monte Carlo simulation where variations on circuit parameters are considered. In simulation, the parameter distribution for each circuit component is assumed to be Gaussian with a variation tolerance of $3\sigma=10\%$ of the nominal value. It can be seen that the relationship, instead of being originally a curve, becomes a band. This means that when an interconnect delay is specified during testing with the process variation, a more stringent value on the peak of the glitch should be chosen for detection.

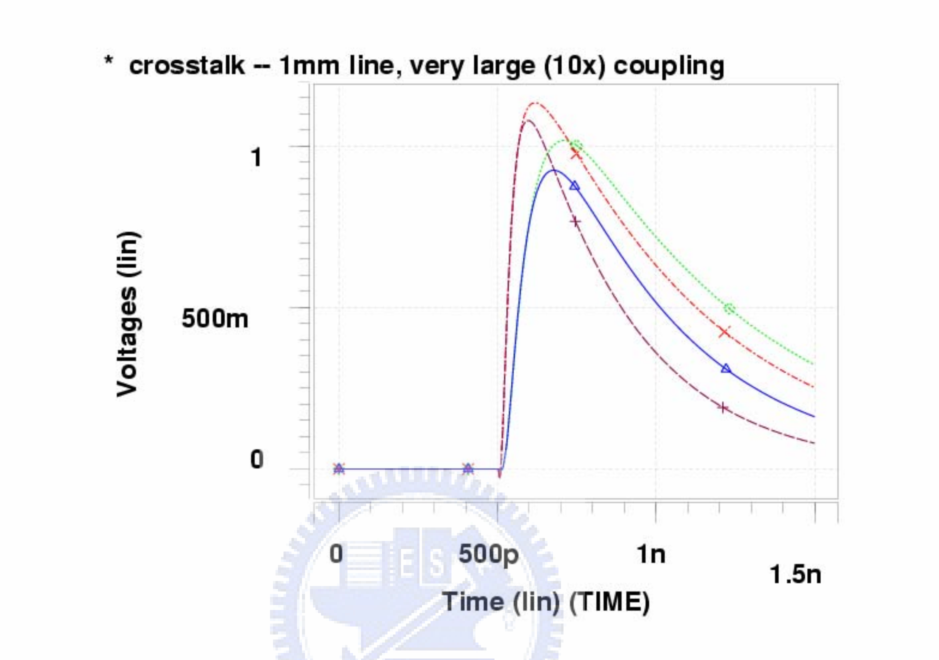


Figure 3.8 Glitch analysis with different resistances and coupling capacitances.

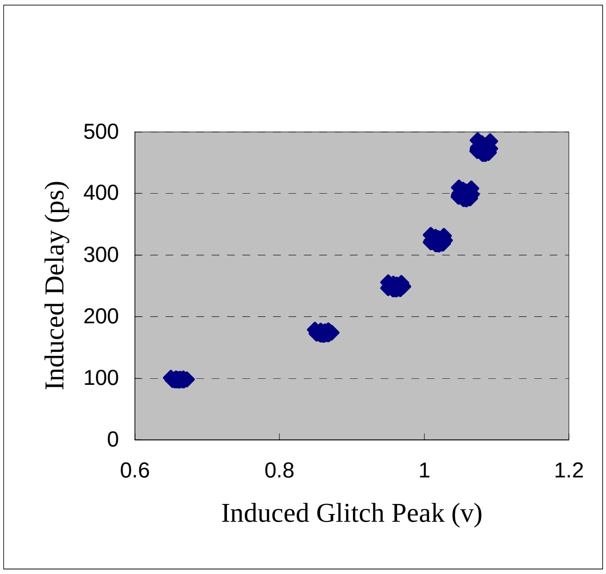


Figure 3.9 Monte Carlo simulation of the induced delay vs. the induced glitch peak (V_p).

3.5.2 Effect of Skew between Aggressor and Victim Signals

In the previous study of Sections 3.2 & 3.3, we assumed that excitations of two aggressor lines were synchronous and the falling excitation signals were in coincidence with the rising edge of the victim line, i.e., there was no skew between signals on the lines. In practice, the signals on aggressor lines and in the victim line may not be in coincidence. According to [36], the maximum induced delay on a victim line caused by an aggressor line occurs not at the coincidence point of the falling edge of the aggressor excitation signal and the rising edge of the victim affected signal, but at a point for which there is a little skew between these two signals. To investigate how the relationship between the peak of the induced glitch and the induced delay is affected by the skew between aggressor lines and the victim line, we did simulation for the three-wire system of Figure 3.1, considering different skews between aggressor lines and the victim line. The results are shown in Figure 3.10 for three different cases: (1) $SK_1 = SK_2 = 0$ (i.e., there is no skew, $SK_1 = 0$, between aggressor 1 and the victim line, and no skew, $SK_2 = 0$, between aggressor 2 and the victim line) (2) $SK_1 = SK_2 = -80$ps; (i.e., both aggressors 1 and 2 have a negative skew of 80 ps with respect to the victim line, and both excitations on aggressors switch before victim line for 80 ps), and (3) $SK_1 = 0$, $SK_2 = 45$ps, (i.e., aggressor 1 has zero skew and aggressor 2 has 45 ps skew with respect to the victim line respectively). For case (1), it is the case that the aggressor signals are in coincidence with the victim signal. For case (2), it is the case of the maximum glitch but a small delay, and for case (3), it is the maximum delay and but a small glitch. It is seen that the relationship between the peak of the induced glitch and the induced delay becomes also spread as a band instead of a single line as the case of $SK_1 = SK_2 =$

0.

The spreading of curves in Figure 3.10, as it is seen, depends on skews between aggressor lines and between aggressor lines and victim line. However, for a interconnect bus system, it could be safely assumed that these skews will not be large since signals on a bus system usually change simultaneously as a set of bits switch their states at the same time. Also, during testing, it could be relatively easy to control every bit switching of a bus line.

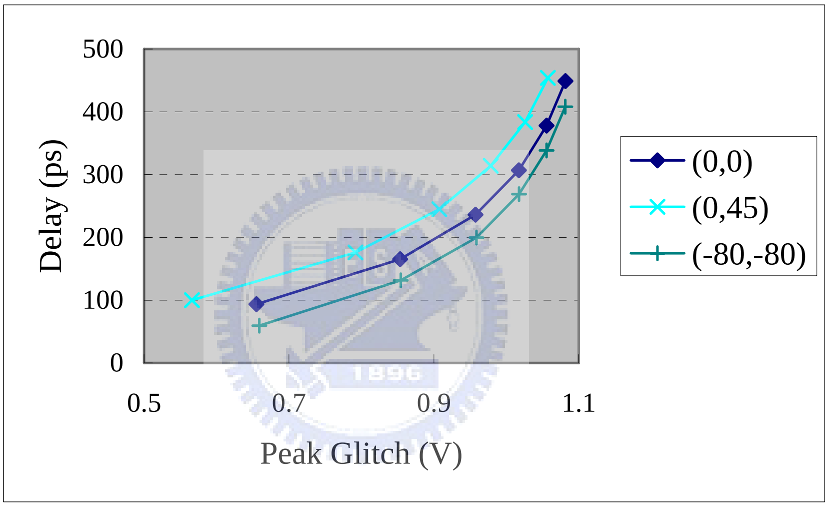


Figure 3.10 The induced delay v.s. the peak of the induced glitch for three different cases: (1) $SK_1 = SK_2 = 0$, (2) $SK_1 = SK_2 = –80$ps, and (3) $SK_1 = 0$, $SK_2 = 45$ps.

3.5.3 Process Variation Effect on Pulse Detector

In section 4, the detection threshold of the proposed pulse detector is adjusted by the W/L ratio of the input inverter. This W/L values, along with other circuit parameters, is easily to be affected by the manufacture process variation. Hence, the relationships of the threshold of detected pulse amplitude (V_{th}) with respect to the $(W/L)_{p\_MOS}/(W/L)_{n\_MOS}$ of the pulse detector of Figure 3.7 was also Monte Carlo simulated with the circuit parameters of the pulse detector allowed to be varied by

10% with respect to the nominal values. The results are shown in Figure 3.11. The relationship also becomes a band.

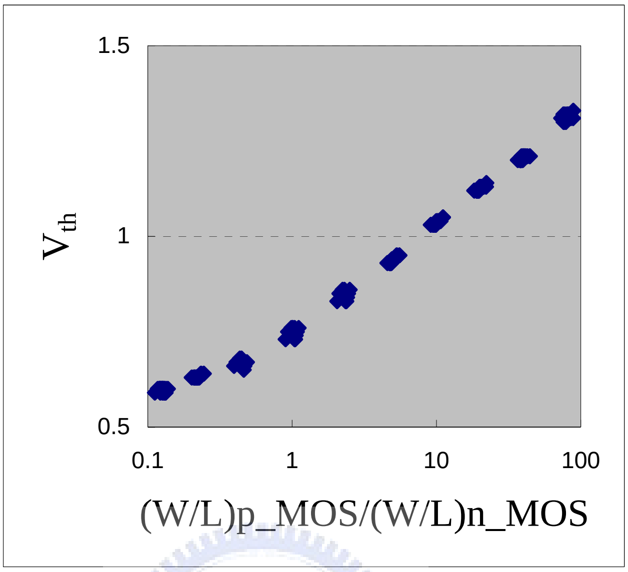


Figure 3.11 Monte Carlo simulation of the threshold of detected pulse amplitude (V_{th}) with respect to the W/L ratio of the pulse detector.

3.6 Experimental Results: Monte Carlo Simulation on Unified Detection Scheme Considering Process Variation

In Figure 3.12, we demonstrate the overkill and escape probability with respect to (W/L) ratio of the pulse detector, which Monte Carlo simulations were done to show the process variation effect on both overkill and escape probability. As revealed in the above discussion, for the detection scheme, once the detection threshold of pulse detector is determined according to a specified induced delay to be detected in testing, uncertainty in testing results will occur due to the above mentioned factors. For example, if the threshold is chosen to be a low value considering the case of the maximum induced delay but with the lowest peak of the induced glitch, i.e., the case (3) of section 5.2, there may exist “***Overkills***”. This is because there may be cases for which the peaks of induced glitches exceed the threshold but their induced delays do not exceed the delay detection threshold when the applied transition signal of victim line is in coincidence, or even has a positive skew, with respect to the inducing transition signals of the aggressor lines, similarly in case (3) of section 5.2 as mentioned before. However, if on the other hand, the threshold is chosen to be a high value considering the case of the minimum induced delay but with the highest peak of the induced glitch, i.e. the case (2) of section 5.2, there may exist "***Escapes***". Figure 3.12 shows simulated probability curves for overkills and escapes with respect to the $(W/L)_{p\_mos}/(W/L)_{n\_mos}$ ratio of the pulse detector. With the ratio set to 1, the detection threshold V_{th} is 0.91V, which corresponds to a delay of about 200ps. When the *(W/L) ratio* decreases, so does the detection threshold V_{th}, and the probability of overkills increases while the probability of escapes decreases. On the other hand, a larger $(W/L)_{p\_mos}/(W/L)_{n\_mos}$ ratio produces a detector with a higher threshold V_{th}, which increases the probability of escapes but decreases the probability of overkill.

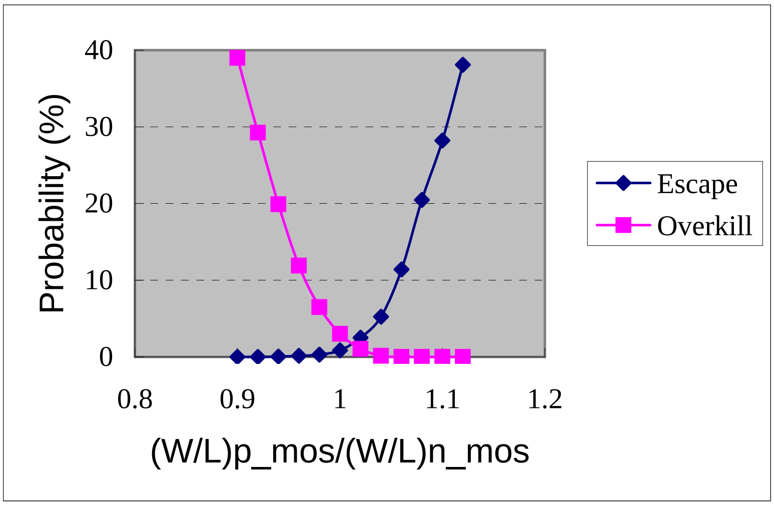


Figure 3.12 Monte Carlo simulations of the Escape Probability and Overkill Probability with respect to (W/L) ratio.

In Figure 3.13, manufacture process variation will also cause the same situation considering the factors discussed in section 5.1 and 5.2. Hence, Monte Carlo simulations were done to simulate the proposed detection scheme by designing the pulse detector and the three-wire bus system with the nominal values on all their circuit parameters but allowing all the circuit parameters to be able to vary by 5, 10, 15 and 20% of their nominal designed values respectively. 2000 samples were simulated for each simulation and the numbers of "***Overkill***" and "***Escape***" were accumulated. It is to see how the above factors will affect the "***Testing Yield***", which is defined to be:

Testing Yield = 1 – (***Overkill Probability*** + ***Escape Probability***)

where

Overkill Probability = ***Number of Overkill*** / ***Total Number of Samples***

Escape Probability = ***Number of Escape*** / ***Total Number of Samples***

The results of these simulations are plotted in Figure 3.13 in terms of the manufacture process variations. In this plot, the detection threshold, V_{th}, was chosen to be 0.91V from the curve of the case of (SK_1 = 0 ps, SK_2 = 0 ps) of Figure 3.10 for the specified induced delay detection of 200 ps. From the figure, it can be seen that, as it is expected, as the process variation increases, the ***Testing Yield*** decreases. For a process variation of (5%, 10%, 15%, 20%), a ***Testing Yield*** of (100%, 99.809%, 97.475%, 92.915%) can be obtained respectively.

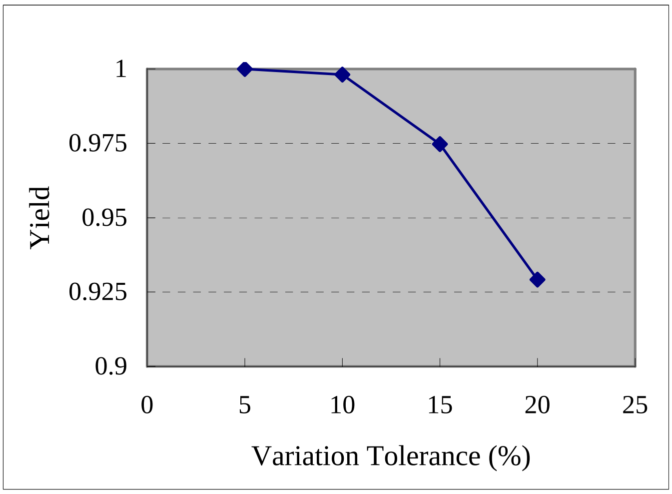


Figure 3.13 Monte Carlo simulation of the “Yield” with respect to process variation on parameter values.

Chapter 4

IEEE Standard 1500 Compatible Oscillation Ring Based Interconnect Delay and Crosstalk Test Methodology

This chapter addresses an interconnect detection problem over general structures. A novel oscillation ring (OR) test scheme and architecture for testing interconnects in SOC is proposed and demonstrated. In addition to stuck-at and open faults, this scheme can also detect delay faults and crosstalk glitches, which are otherwise very difficult to be tested under the traditional test schemes. IEEE Standard 1500 compliant wrapper cells are modified to accommodate the test scheme. An efficient algorithm is proposed to construct ORs for SOC based on a graph model. Experimental results on MCNC benchmark circuits have been included to show the effectiveness of the algorithm. In all experiments, the scheme achieves 100% fault coverage with a small number of tests.

4.1 Introduction

Rapid advance in the VLSI technology has rendered delay caused by interconnects to surpass that caused by transistors [105]. Interconnects have become the key element in determining circuit performance and signal integrity, especially for SOC ICs. In addition, reduced spacing between adjacent interconnects makes crosstalk an important source of anomaly in deep submicron VLSI [17], [24], [54], [65], [99]. It can induce glitches and extra delays for signals propagating along the interconnection lines. Buffer insertion is proposed to alleviate the problems associated with long signal line. As a result, signal lines used for communication consist of not only wire segments, but also logic gates [28], [30], [73].

Traditional test methods are mainly designed for functional check for which the signal integrity issue is usually not the main target. As a result, crosstalk and delay faults are difficult to be detected under conventional test methods. The detection of crosstalk-induced glitches usually involves precise measurement of signals on the victim nets [9], [100], while complex clock control is needed for delay fault detection due to the two-pattern tests [8], [22]. Therefore, much more extra effort has to be devoted to the detection of errors due to these problems.

Cores are usually provided with either predefined test vectors or built-in self-test (BIST) mechanism, so that the SOC system designers only need to consider how to apply test and control the test process. On the other hand, the interconnection structure of an SOC is designed by the system integrator, who is also responsible for defining the test set for interconnect. Interconnect testing occupies an important part in system and chip design [45]. Plenty research works on interconnect testing can be found in the literature. Earlier works in interconnect testing were targeted for board-level testing [7], [45], [61], [117], [122]. These sections described fault models and test generation algorithms for general interconnect structure. However, it is very difficult to apply these interconnect testing methods under SOC environment without design-for-testability (DFT) support. IEEE Standard 1500 [35], [53] provides structural support for core testing as well as interconnect testing in SOC. The IEEE

Standard 1500 compatible SOC test environment consists of a centralized test access mechanism (TAM) and wrappers around cores in the SOC. The TAM defines the test control, while wrappers provide a standardized interface for test data transmission. The proposed SOC test standard IEEE 1500 extends IEEE 1149.1 Boundary Scan test methodology so that interconnect test for SOC can be conducted in a way similar to those used in board-level interconnect test. In this approach, all pins of a core are replaced by wrapper cells, so that a scan path connecting all the pins can be formed during the test mode. In this way, test vectors can be applied to interconnection lines, and test results are captured and observed outside the core, and are propagated to the ATE for inspection. However, the proposed core test standard is designed for traditional test methodology, and the signal integrity issue is not considered under this framework. For example, if we need to apply two-pattern test to detect delay fault, we need to modify not only the wrapper cell structure but also the clock control so as to apply tests and capture responses correctly. The hardware overhead can be significant.

To solve this difficulty, we propose an oscillation-based test scheme and structure for interconnect in SOC ICs. Oscillation ring (OR) test is a useful and efficient method to detect faults in functional circuits [6], [52], [58], [103], [119], [120]. An oscillation ring is a closed loop, which has *odd number of signal inversions*, of the circuit under test. Once the ring is constructed during the test mode, an oscillation signal appears on the ring. For a circuit with stuck-at and open faults, oscillation stops; and for a circuit with gate or path delay faults, the oscillation frequency is different from the fault-free case. By observing the oscillation signal at the output of the circuit, it can tell whether the circuit is faulty or not. There are several important works on the oscillation ring test scheme, e.g., [52], [58], [103],

[119]. Most of the results focus on detecting device faults for analog and/or mixed-signal circuits [52], [58], [103], [119], or on detecting faults on gate-level circuits [6],[120].

In this approach, we construct a ring that goes through a series of interconnect wires (including inserted buffers) and some internal scan paths in core modules of the SOC. Once a ring with odd inversions is constructed, we can decide whether the ring is faulty by observing the oscillation signal on the ring. Various types of interconnect faults are detectable under this scheme, including stuck-at faults, open faults, delay and crosstalk glitch faults. Furthermore, this scheme can be used for circuit parameter measurement since the path delay can be obtained by measuring period of the oscillation signal. Fault diagnosis is also achievable with properly selected rings since fault can be located as multiple different rings passing it all fail.

In order to support the OR test, we modify IEEE Standard 1500 compliant wrapper cell designs. We also develop an efficient algorithm to select interconnects to form the minimal number of oscillation rings to reduce test time. Experimental results on MCNC benchmark circuits show that the algorithm achieves 100% fault coverage with small number of tests. These results show that the proposed method is not only feasible with small hardware overhead, but also efficient in fault detection.

The remainder of this chapter is organized as follows. In Section 4.2, we present the test architecture for oscillation ring based interconnect test, and the modified IEEE Standard 1500 compliant wrapper cell designs are given in Section 4.3. A graph model of interconnect hypernet structure is discussed, and some related theoretical analysis is also given in Section 4.4. An efficient ring-generation algorithm that selects interconnects to form a minimal number of oscillation rings to minimize test time is presented in Section 4.5. Experimental results on MCNC benchmark circuits show that the algorithm achieves 100% fault coverage with a small number of tests in Section 4.6. The results presented in this chapter will be appeared in [79], [80].

4.2 Interconnect Test Architecture for Oscillation Ring Test

In this section, we propose the architecture for the oscillation ring test for interconnects. In order to get the whole understanding, Figure 4.1 is first introduced as Figure 2.4. Figure 4.1 shows the proposed architecture, where C’s are circuit cores implemented with boundary scan cells and a local counter, which is to capture the induced glitches for crosstalk fault detection and to measure delays of oscillation rings for delay measurement. For this architecture, oscillation ring(s) will be formed as shown during the testing mode. If the formed oscillation ring fails to oscillate, it implies that there exists stuck-at or open fault(s) in components of the oscillation ring. If there is a crosstalk fault between a victim interconnect line and the oscillation ring interconnect lines, glitches will be induced on the victim interconnect line.

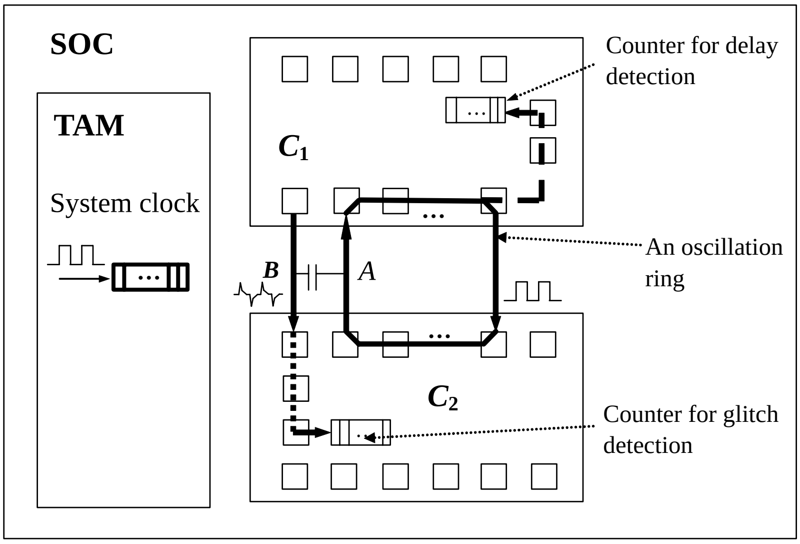


Figure 4.1 Test architecture for interconnect crosstalk detection and delay measurement. (also known as Figure 2.4)

Figure 4.2 shows the oscillation signal at the oscillation ring and the induced glitches at the victim interconnect line. These induced glitches will be captured by the local counter of the core and be shifted out for observation.

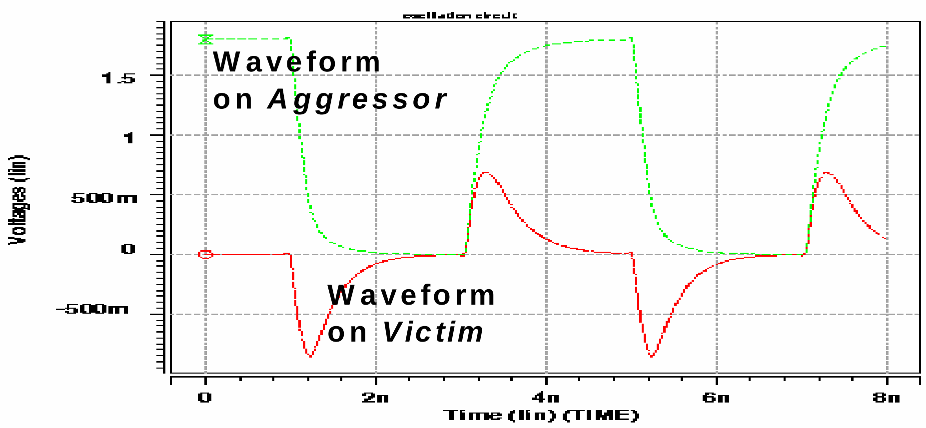


Figure 4.2 The oscillation signals, A, on the oscillation ring and the induced glitches, B, on the victim interconnect.

To test the delay fault, the delay of the oscillation ring will be measured through using the local counter and the central counter of TAM of the SOC. At this time, the central counter is enabled by signal *OscTest* and triggered by the system clock, and a local counter is connected to one wrapper cell of the oscillation ring so that the oscillation signal is fed to the local counter. When the oscillation test session starts (*OscTest* = 1), the central counter as well as all local counters in cores are enabled. After the counter in TAM counts to a specific number *n*, the oscillation test session terminates and all local counters are disabled (*OscTest* = 0). The counter contents are shifted out to an ATE for inspection.

Assume that the frequency of the system clock to be f and the local counter content of the ring to be *di*. The ring's oscillation frequency, *fi*, is:

$$f_i = f \times d_i / n \qquad (4.1)$$

According to the timing specification, for a good oscillation ring connected by interconnect lines and boundary scan cells, $f_{min} \le f_i \le f_{max}$. That is: $d_{min} \le d_i \le d_{max}$.

4.3 IEEE Standard 1500 Compatible Modified Wrapper Cell Design

An oscillation ring for interconnect test consists of interconnect wires and part of the scan path in each core where the ring passes. Therefore, a wrapper cell must provide a path between input/output ports and scan in/scan out ports. If oscillation test is used to test wires attached to/from pads, the boundary scan cells also have to be modified in a similar way. In order to facilitate the scheme, IEEE Standard 1500 compliant boundary wrapper cells need to be modified. In this section, the modified wrapper cell design is presented.

A normal wrapper cell provides two types of paths: a scan path connecting all wrapper cells into a shift register, and an interface buffering between core internal and the wire connected to the pin. Whenever oscillation test is applied, a third combination path must be provided. For an input pin, the wrapper cell must connect the pin input (IN) to scan output (SO), while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session. Examples of these connections are the four “corners” of the ring in Figure 4.1.

The modified wrapper cell design is shown in Figure 4.3 for input and output cells. In each cell, two MUXs are added for path selection. For an input wrapper cell, the extra paths are SI→SO and IN→SO, while for an output wrapper cell the extra paths are SI→SO and SI→OUT. The added inverting and non-inverting buffers in output cells are used to provide odd inversions on the oscillation ring path to generate oscillation signals for the OR test. *OscTest* is a global control signal, while *sel* is used in the input wrapper cell and *inv* is used in the output wrapper cell. Signals *sel* and *inv* are individually set and are scanned into the wrapper cells before an OR test session.

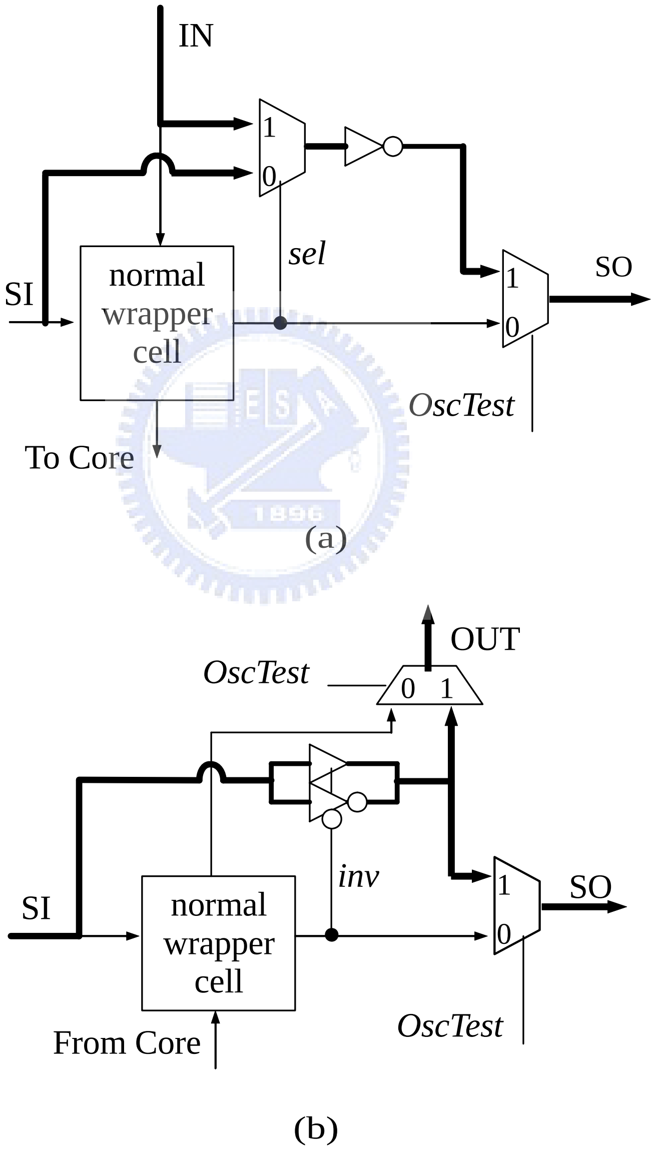


Figure 4.3 Modified wrapper cells: (a) input cell (b) output cell.

In either the normal mode or the IEEE Std.1500 test mode (*OscTest* =0), modified cells act as normal wrapper cells. In the OR test mode (*OscTest* =1), the part of “normal wrapper cell” is bypassed. For an input cell, *sel* is used to select either SI→SO or IN→SO, depending on the position of the input wrapper. If the cell connects an external interconnect to the internal scan path, it is configured as IN→SO. Otherwise, it is configured as SI→SO. For an output cell, the bit information stored in the cell is used for inversion control *inv*, which decides whether the passing signal should be complemented. This is also applied to buffered interconnects where inverters are used for timing closure and signal amplification.

A summary of control signals for the modified wrapper cells shown in Figure 4.3 is given in Tables 4.1 and 4.2, respectively

Table 4.1 Control signals for the modified input wrapper cell.

OscTest	*Sel*	Comments
1	1	~IN → SO (OscTest Mode)
1	0	~SI → SO (OscTest Mode)
0	–	normal or IEEE Std. 1500 test mode

Table 4.2 Control signals for the modified output wrapper cell.

OscTest	*inv*	Comments
1	1	SI → SO and SI → OUT (OscTest Mode)
1	0	~SI → SO and ~SI → OUT (OscTest Mode)
0	–	normal or IEEE Std. 1500 test mode

4.4 Oscillation Ring Construction: Model and Analysis

To apply the OR testing to an SOC with many cores connected with interconnect lines, it needs to form oscillation rings which can cover all interconnects in order to completely test all interconnects of the SOC. In the two sections which follow, we will present the model and analysis and the algorithm to construct oscillation rings respectively.

As mentioned previously, the proposed methodology is targeted for SOC with IEEE Std. 1500. A more detailed example of the test mechanism is illustrated in Figure 4.4, where there are three cores, C_1, C_2, and C_3. All pins in a core are connected into a scan path during the test mode, which is indicated by the broken line in Figure 4.4(a). Oscillation rings can be constructed with the help of scan paths provided by wrapper cells. The interconnect wires connecting cores are also shown by heavy lines in Figure 4.4(a), where an arrow indicates the direction of signal transmission. There are three nets in the figure, in which net N_1 connects three terminals (pins), while N_2 and N_3 connect two pins each. Only the heavy lines are the target of interconnect test.

For example, there are two rings in Figure 4.4. The first ring consists of nets N_1 (and its right-hand side branch), N_2, and N_3, and it passes all three cores. The second ring consists of N_1 (and its left-hand side branch) and N_3, and scan paths in C_1 and C_3. In order to make the signal on a ring oscillate, we must ensure that the number of inversions on a ring is *odd*, and this includes the inversions on wire segments as well as those in wrapper cells.

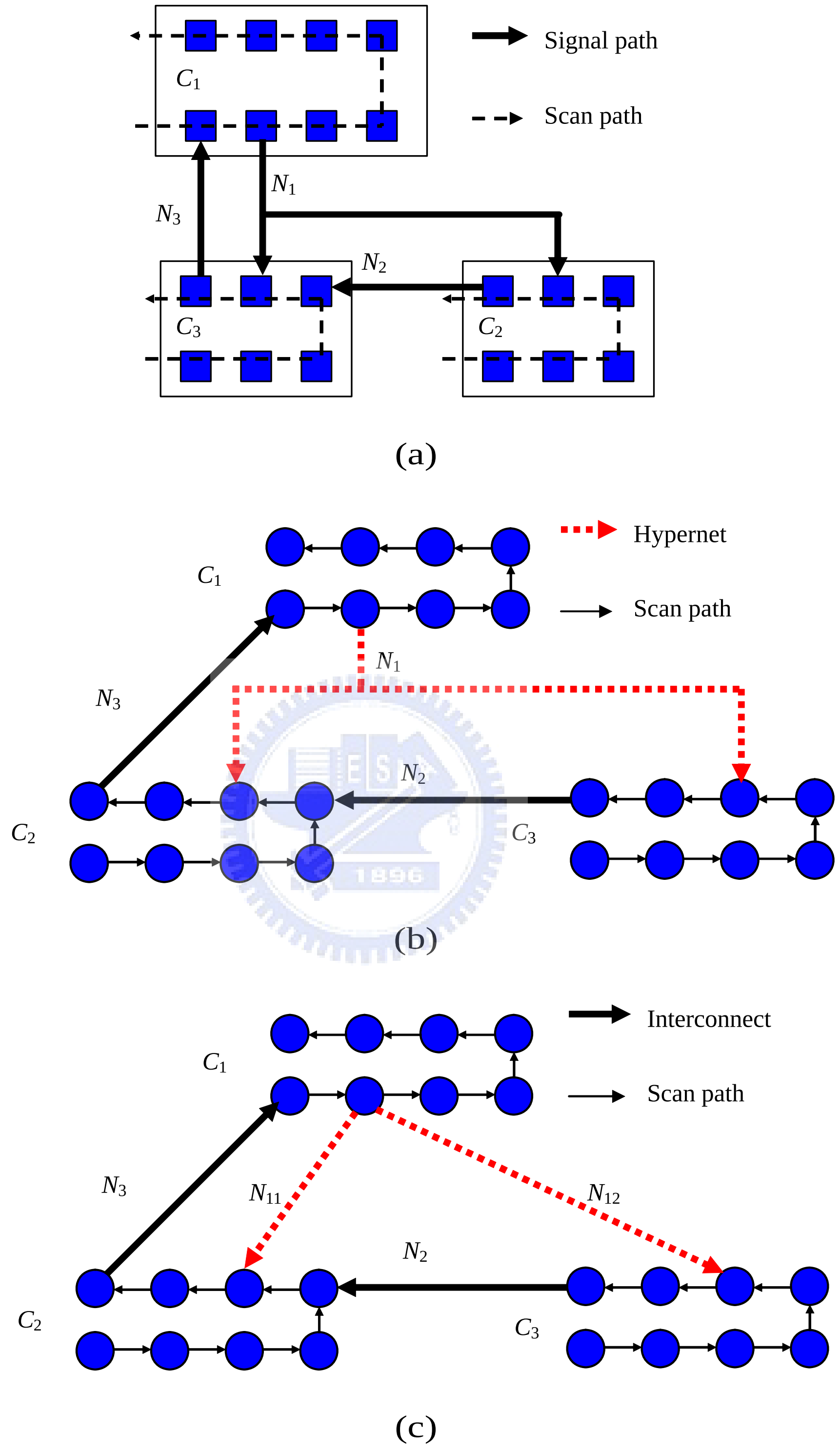


Figure 4.4 (a) The interconnect diagram, (b) hypernet graph, (c) graph model with 2-pin nets. (Also same as Figure 2.1)

4.4.1 Graph Model for Oscillation Ring Tests

In order to simplify the problem under investigation, we represent the circuit interconnection by using an abstract *hypergraph*.

Definition 1: A *hypergraph* $G' = (V, L)$ consists of a vertex set V and an edge set L, which consists of multi-terminal edges connecting a set of vertices $V_i \subseteq V$ and $|V_i| \geq 2$. Such an edge is referred to as a *hypernet*.

For example, N_1 in Figure 4.4(b) is a hypernet. Furthermore, we assume that in an n-terminal hypernet, one terminal is the source node (i.e., sending signal) while the others n–1 are the sink nodes (i.e., receiving signals).

The circuit structure of an SOC can be directly transformed into a hypergraph, in which each pin is a vertex while each signal net is a hypernet. However, this graph model is not good enough for our problem. Consider net N_1 in Figure 4.4(a) again. It is obvious that the two branches of N_1 should belong to two different rings, and they cannot be tested simultaneously. Therefore, it would be better to consider each branch of a hypernet individually or separately instead of treating them all as a whole. Each branch of a hypernet is thus a 2-pin net. For example, nets N_{11} and N_{12} in Figure 4.4(c) are two 2-pin nets, which correspond to hypernet net N_1 in Figure 4.4(b), and each 2-pin net connects the source vertex to one of its sink vertices. An n-terminal hypernet is thus broken into (n–1) 2-pin nets. The result is a normal graph $G = (V, E)$, where E is the set of 2-pin nets. There are two rings in Figure 4.4(c): $R_1 = \{N_{11}, N_3\}$, $R_2 = \{N_{12}, N_2, N_3\}$.

Definition 2: A *weighted graph* $G = (V, E)$ consists of a vertex set V and an edge set E, in which each edge, $e_i \in E$, is an ordered pair (u, v), where $u, v \in V$, and has a *weight* w_i.

A complete test for stuck-at faults and open faults for all interconnections is thus reduced to a problem of finding a set of rings that cover all edges corresponding to interconnection structure in the graph G. This is equivalent to find a set of sub-circuits (rings) $R = \{G_1, G_2, \ldots, G_n\}$, such that:

- $\forall G_i$, $G_i \subseteq G$, $G_i = (V_i, E_i)$, G_i is a ring.
- $\bigcup_{i=1}^{n} E_i = E$

A minimum test is thus the set of rings with minimum cardinality.

For the delay fault testing, signal delay on each net along the ring is considered. To deal with the delay fault, a weight w_i, which is the timing specification, on a 2-pin net e_i by a 2-tuple $w_i = (l_i, u_i)$, where l_i and u_i are lower and upper bound on the distribution of normal path delay respectively, is defined. The graph model for Figure 4.4(c) with aforementioned weights is shown in Figure 4.5.

Let t_i be the actual propagation delay on net e_i, and the variance of delay on net e_i be $\delta_i = u_i - l_i$. The following lemma gives a sufficient condition under which the delay fault t_i is detectable by applying oscillation test.

Lemma 1: Consider a ring of n edges, $e_1, e_2, \ldots, e_n$. The delay fault on edge e_i, $1 \le i \le n$, is detectable if the following condition holds:

$$t_i - u_i \ge \sum_{j \in \{1..n\} - \{i\}} \delta_j \tag{3}$$

Proof: In a fault-free circuit, the maximum delay in a ring will be the summation of the upper bounds of individual nets. A large delay t_i will not be masked if:

$$t_i + \sum_{j \in \{1..n\} - \{i\}} l_j \ge \sum_{j=1}^{n} u_j \tag{4}$$

Eq. (3) can be obtained by rearranging Eq. (4).

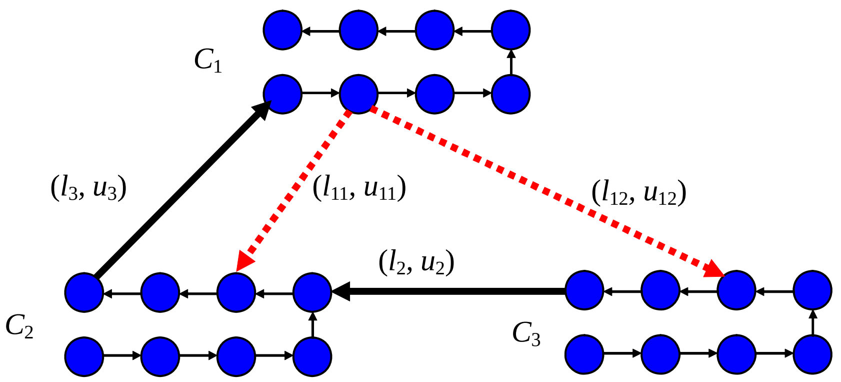


Figure 4.5 Graph model for delay faults. (Also same as Figure 2.2)

From Lemma 1, it can be seen that a delay fault may be masked when ti > ui but Eq . (3) is not satisfied. In order to reduce the probability of undetectable faults, we shall try to construct short rings, so that the accumulation of delay variance will not mask delay faults.

4.4.2 Analysis of Rings and Test Cost

Test cost is dominated by test application time. In the case of oscillation ring test, the single factor affecting test application time is the number of rings required to cover all nets. The number of rings is closely related to interconnect structure. We can analyze it by using the hypernet model.

For a hypernet, at most one of its fanout branches can be tested at a time. The reason is that no two branches belong to the same ring, and any two rings containing the two branches under consideration share the stem of the hypernet before fanout point. If both rings are formed simultaneously, the two oscillation signals will interfere with each other. Therefore, at most one of the fanout branches can be tested at a time. This condition is illustrated in Figure 4.6.

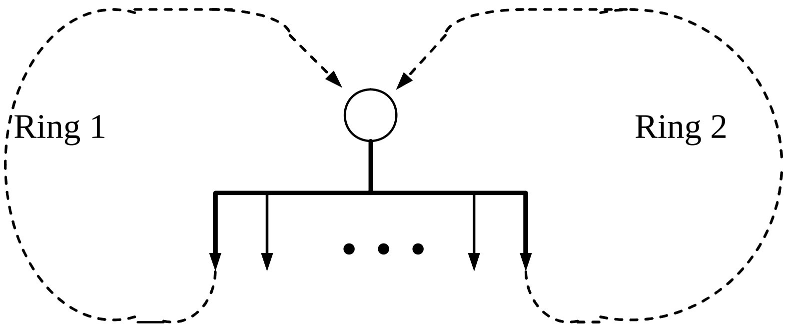


Figure 4.6 Hypernet branches and rings.

Since two interconnect wires are connected by a scan path in a core, the *positions of pins* (i.e. relative position between input pin and output pin) in a core will also affect the interconnection structure. The following lemma explains why *pin location* will affect the number of rings.

Lemma 2: Any two 2-pin nets driven by adjacent pins of a core must belong to two different rings for the oscillation ring test.

Proof: The graph model for two adjacent output pins is shown in Figure 4.7. In this figure, a vertex represents a modified wrapper cell for a pin.

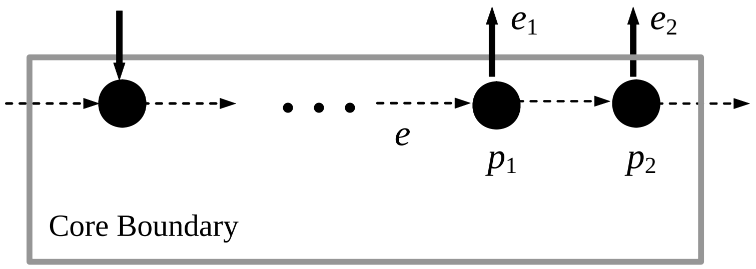


Figure 4.7 Rings for adjacent output pins.

Any oscillation ring going out of a core from a given output pin p_i must enter the same core via an input pin whose wrapper cell is a predecessor to p_i in the graph. Thus, any ring going through either edge e_1 or e_2 must pass edge e, which is located on the scan path. Since a ring cannot go through an edge e twice, no rings contain edges e_1 and e_2 simultaneously.

The following theorem gives the minimum number of rings required for a core module.

Theorem 1: Assume that k output pins p_1, p_2, …, p_k are adjacent to each other in a core. Let the number of fanout branches of the hypernet connected to pin p_i be n_i. The minimum number of oscillation ring tests for interconnect wires attached to theses k output pins is $\sum_{i=1}^{k} n_i$.

The proof of Theorem 1 follows the results of Lemma 2. A lower bound on the number of oscillation rings required to test an SOC can be establishes as follows:

Corollary 1: A lower bound on the number of rings required for interconnect test is equal to the maximum number of 2-pin nets connected to a sequence of core output pins in all SOC cores.

4.5 Oscillation Ring Construction Algorithm

In this section, we discuss how to generate rings for OR test. We analyze the complexity of ring generation algorithms, and propose a heuristic approach for ring generation.

4.5.1 Exact Algorithm

Since our goal is to reduce test time for the interconnect test, we should find out a set of rings with minimum cardinality. A very naive algorithm for the generation of minimum set of rings can be performed as follows. First, we find all possible rings in an SOC, and the minimum set of rings that cover all 2-pin nets. This approach, however, is not feasible for any reasonably sized SOC due to its high time complexity.

In our graph model, any ring is a sub-graph, which contains a subset of all 2-pin nets. The number of all possible rings grows exponentially as the number of 2-pin nets increases, and thus the first part of the naïve algorithm is already intractable. The problem of finding out minimum number of rings covering all 2-pin nets can be mapped to a column-covering problem, in which each column is a ring and each row represents a 2-pin net. However, it is well known that the column-covering problem in NP-complete.

A more refined exact algorithm can be conducted similar to Quine-McCluskey algorithm for two-level logic optimization. A ring is redundant if it is contained in other rings. In order to reduce the search space, we shall start with “prime” rings that are not contained in other rings. However, the problem of finding out “prime” rings is still difficult. For example, if there exists a Hamiltonian cycle in a graph, the cycle must be a prime ring. Unfortunately, searching for a Hamiltonian cycle is also NP-complete.

From the above discussion, it is obvious that any exact algorithm for searching a minimum set of rings can only work for small circuits. For larger SOC ICs, heuristic solutions must be applied.

4.5.2 Ring Generation Algorithm: A Heuristic Algorithm

We propose a heuristic algorithm to find a minimum set of rings that cover all 2-pin nets under test. The algorithm is a modified depth-first search which works as follows:

The SOC under test is first modeled as a hypergraph G'. This graph is then transformed into graph $G = (V, E)$ with 2-pin nets only. The vertex set V consists of pins in all cores. The edge set E is partitioned into two disjoint subsets E_i and E_e, where E_i is the set of internal edges (i.e. those edges in the scan paths within modules/IP cores) and E_e is the set of external interconnect wires (i.e. interconnects). Our goal is to generate rings that cover E_e.

We generate a ring containing a 2-pin net $(u, v) \in E_e$ by starting from vertex v, which is an input pin. Then we try to find an output pin w that locates in the same core as v, and w is connected to a 2-pin net that is not yet covered by any other ring. If no such unvisited 2-pin net (w, x) exists, we just select the first available output net from any available set of output pins. This process is repeated until a ring is found. The procedure then goes over again and again until all 2-pin nets are covered.

The above heuristic works as follows: Whenever we start looking for a new ring, we explore paths containing 2-pin nets that are not yet covered. In this way, each new ring may cover as many other uncovered nets as possible. After all rings having been generated, a simple reverse order simulation is conducted to remove redundant rings. A net is *oscillation ring testable* if there exists at least one ring covering this net. The algorithm is outlined below.

This algorithm can take into account the variation of delay on each individual wire. Long rings with very large variance in path delay will not be constructed since delay faults may be masked in these rings. The extra restriction increases the complexity of the ring searching algorithm, but it reduces the probability of error masking caused by process variation.

Algorithm: ***Ring Construction under IORT Algorithm***

Input: A hypergraph *G'* = (*V*, *L*) representing a circuit

Output: A list of rings *R*

```
Transform $G'$ to a new graph $G = (V, E)$ with hypernets into equivalent 2-pin nets
    only, in which all nets are oscillation ring testable;
$R = \emptyset$;
for every $e = (u, v) \in E$ and $e$ is not visited
    $R = R \cup$ find_ring($G$, $e$);
reverse-order simulation for rings in $R$, end program.

function find_ring($G$, $e$)
Let $e = (u, v)$ and $v$ is an input pin in core $C$;
if $v$ is a pin in the starting core
    return the ring and mark all nets as visited;
for every output pin $w$ in $C$
    if there is an unvisited edge $(w, x)$
        find_ring($G$, $(w,x)$);
    else if there is an untried output net $(w, x)$
        find_ring($G$, $(w,x)$);
    else
        return $\emptyset$;
end function
```

Figure 4.8 Interconnect Oscillation Ring Test (IORT) Algorithm.

4.6 Experimental Results

We implement the proposed algorithm, and evaluate its performance with some MCNC benchmark circuits. The results are presented in this section.

4.6.1 Simulated Results on HP benchmark circuit

To validate the proposed OR test methodology, an MCNC benchmark “hp” consisting of 11 cores connected by 195 interconnects, was placed and routed as shown in Figure 4.9 and then implemented using the TSMC 0.18 μm technology to simulate its oscillation condition.

The simulation results are shown in Figure 4.10, where Figure 4.10(a) shows the oscillation signal of the longest ring; and Figure 4.10(b) shows the oscillation signal of the shortest rings. The cycle time of the longest rings (with nine interconnects) is about 38ns and that of the shortest rings is about 2.8ns. Thus, the oscillating frequency ranges from 26 MHz to 357 MHz, and this shows that this oscillation detection scheme is feasible.

Figure 4.11 shows the glitches induced by the oscillation signals shown in Figure 4.10(a). In the figure, the blue curve is simulated under normal circuit parameters, while the red curve shows the glitches when a very large coupling capacitance (10 times of the normal value) existing between a wire in the ring and a neighboring net. A crosstalk fault with about 0.65V amplitude is induced and it can be detected by a carefully designed detector.

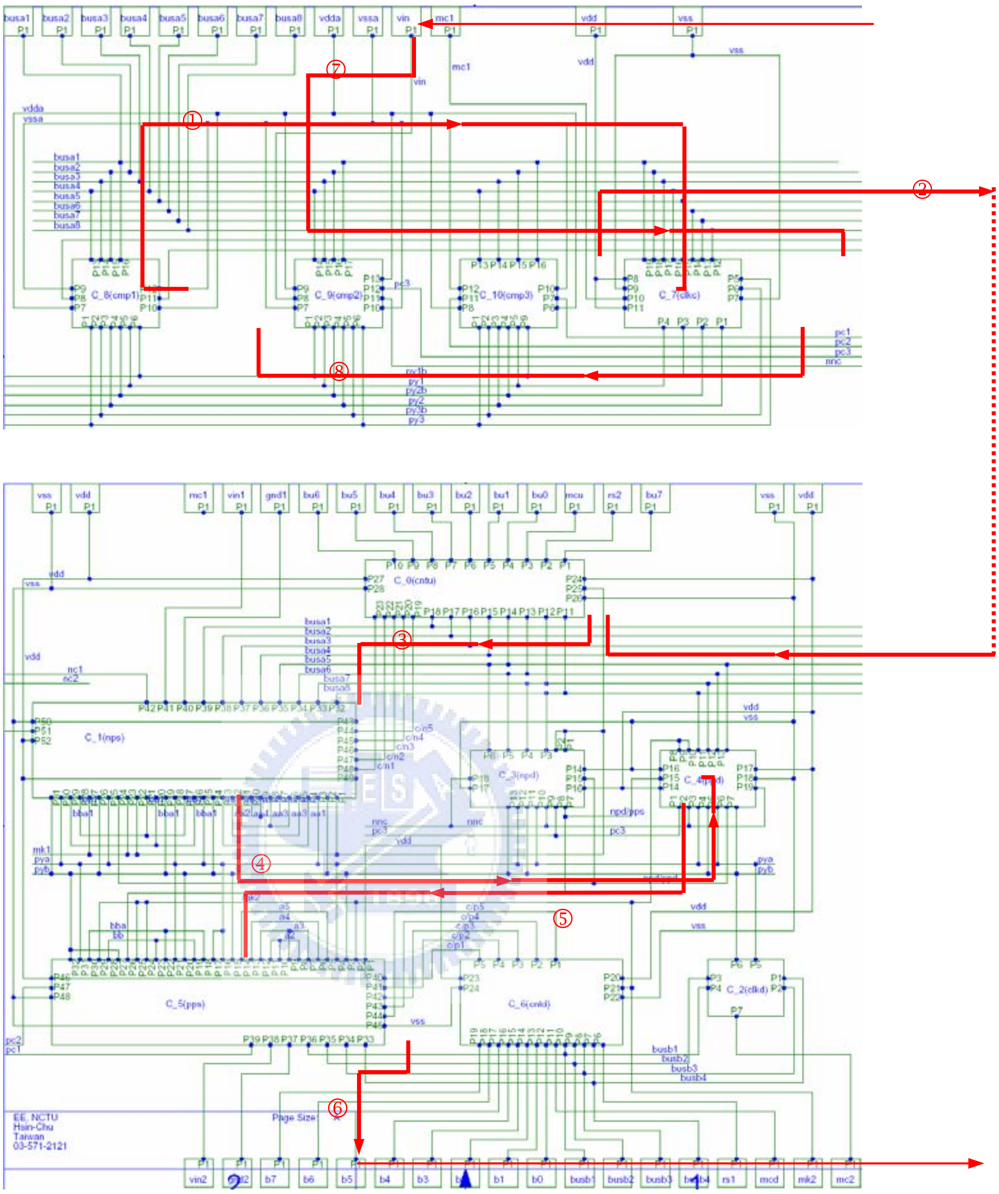


Figure 4.9 The placement and routing of an illustrative example of the OR testing for a benchmark circuit *hp*.

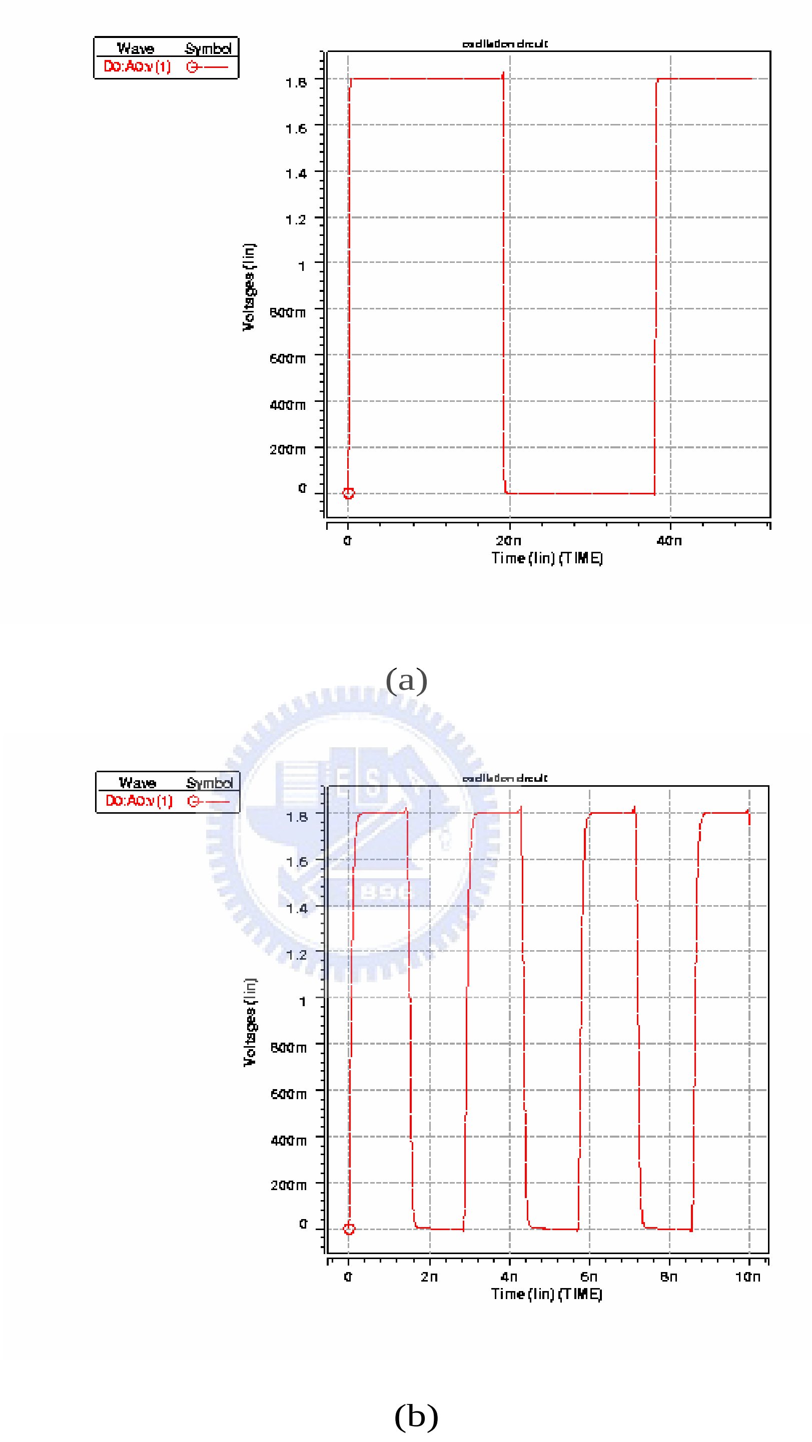


Figure 4.10 Simulated waveforms of the longest (a) and shortest rings (b) of benchmark circuit *hp*. (also same as Figure 2.5)

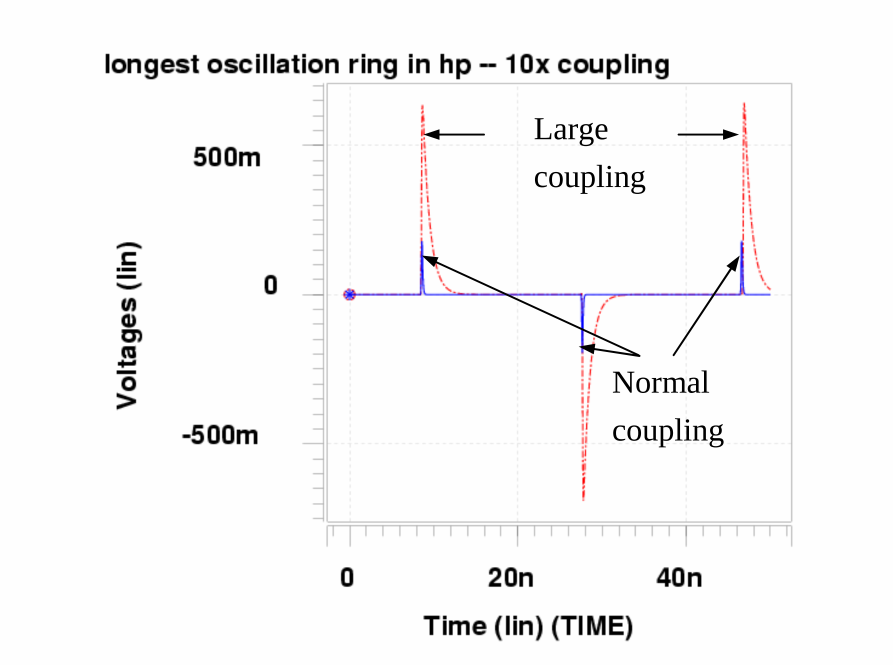


Figure 4.11 Simulated waveforms of glitches induced by oscillation signals in the longest ring of "*hp*". (also same as Figure 2.6)

4.6.2 Oscillation Ring Generation for Interconnect Detection Algorithm

We experiment the ring-generating algorithm (Oscillation Ring Generation for Interconnect Detection Algorithm, IORT) with six MCNC benchmark circuits [28],[30]. The circuit statistics and results are all shown in Table 4.3, where the first column gives the circuit names, while the next four columns are circuit statistics, including number of cores (#core), number of pads (#pads), number of hypernets (#hypernets), and number of 2-pin nets (#2-pin), respectively. The next two columns are experimental results. The sixth column gives the number of rings formed to cover all 2-pin nets (#rings) for complete detection of 100% fault coverage. In column 7, we give the estimated testing time (given in msec), which is obtained by assuming a 4 MHz measuring period. The time needed to set up the rings should be roughly proportional to the testing time. The last column (L.B.) gives the *lower bound* on the number of rings required for complete detection, calculated according to Corollary 1, for each circuit. From the last two columns, it can be seen that the lower bound is not very tight for some cases. If we assume that the ratio for the test setup time and the test application time is 1:1, then the longest time required among all MCNC benchmarks to finish the OR test is around 100 ms, which is for the circuit ***xerox***.

During experiments, since in these benchmark circuits the net directions are not known, we assume that: (1) Cores are listed in an order. For a hypernet formed, the pin corresponding to a core first in the core list is assumed to be the source while others are sinks. (2) Since the order on internal scan paths is not known, all output pins are conservatively assumed to be placed in consecutive positions. (3) All pads are connected through the boundary scan path, while positions of the pads are unknown. The above assumptions represent the worst-case scenario. Under assumption (2), none of the output 2-pin nets of a core can be tested in a single ring, and thus each ring may pass any core only once. The assumption (3) makes the boundary scan path appear only once in each ring. Thus, the results obtained could be treated as the *upper bound* on the rings required; the actual number of rings required would be smaller.

Table 4.3: Comparison between the number of rings generated for experimental and theoretical results of Lower Bounds.

Circuit	#cores	#pads	#hypernet	#2-pin	#ring	Time (ms)	L.B.
ac3	27	75	211	416	133	33.3	69
ami33	33	42	117	343	242	60.5	214
ami49	49	22	361	475	154	38.5	35
apte	9	73	92	136	73	18.3	38
hp	11	45	72	195	82	20.5	68
xerox	10	2	161	356	218	54.5	174

If test rings form a partition of all the 2-pin nets, a 2-pin net will appear in only one ring. In this case, the average ring length is minimum: (#2-pin net)/(#ring), and this number is also given in Table 4.4. However, such a partition is usually not possible, and most 2-pin nets appear in many different rings in circuits. Thus, the average ring length in our experiments is larger than (#2-pin net)/(#ring), which is shown in column 4 and column 5 in Table 4.4.

Table 4.4: Analysis of Ring Lengths

Circuit	#2-pin	#ring	Average Ring Length	#2-pin/#ring
ac3	416	133	6.99	3.13
ami33	343	242	8.93	1.33
ami49	475	154	16.54	3.08
apte	136	73	3.75	1.86
hp	195	82	4.7	2.38
xerox	356	218	3.73	1.63

In Figure 4.12, the distributions of ring lengths for each simulated benchmark are also shown. These distributions are quite different because of the different circuit structure of interconnects for each circuit.

The relationship between the number of OR testing rings and the achieved interconnect coverage, which consequently reflects the stuck-at fault, open fault, and delay fault coverage, is shown in Figure 4.13 for all the simulated circuits. As to crosstalk fault, it can be observed by scanning out the contents of all the local counters to check whether it exist any crosstalk faults between the target nets in the oscillation ring and all the adjacent victim nets. It can be seen that the fault coverage in Figure 4.13 increases roughly *linearly* with the number of test rings applied. This is in contrast with logic testing, in which a small number of test vectors usually account for the detection of most of the faults. There are some observations for the OR test methodology. First, the difficulty for the fault detection is almost the same for all interconnects in one circuit since the relationship between fault coverage and number of rings is approximately a straight line. Second, the difficulties which are in detecting faults are approximately proportional to the circuit size and interconnect structure, and thus determine the number of rings.

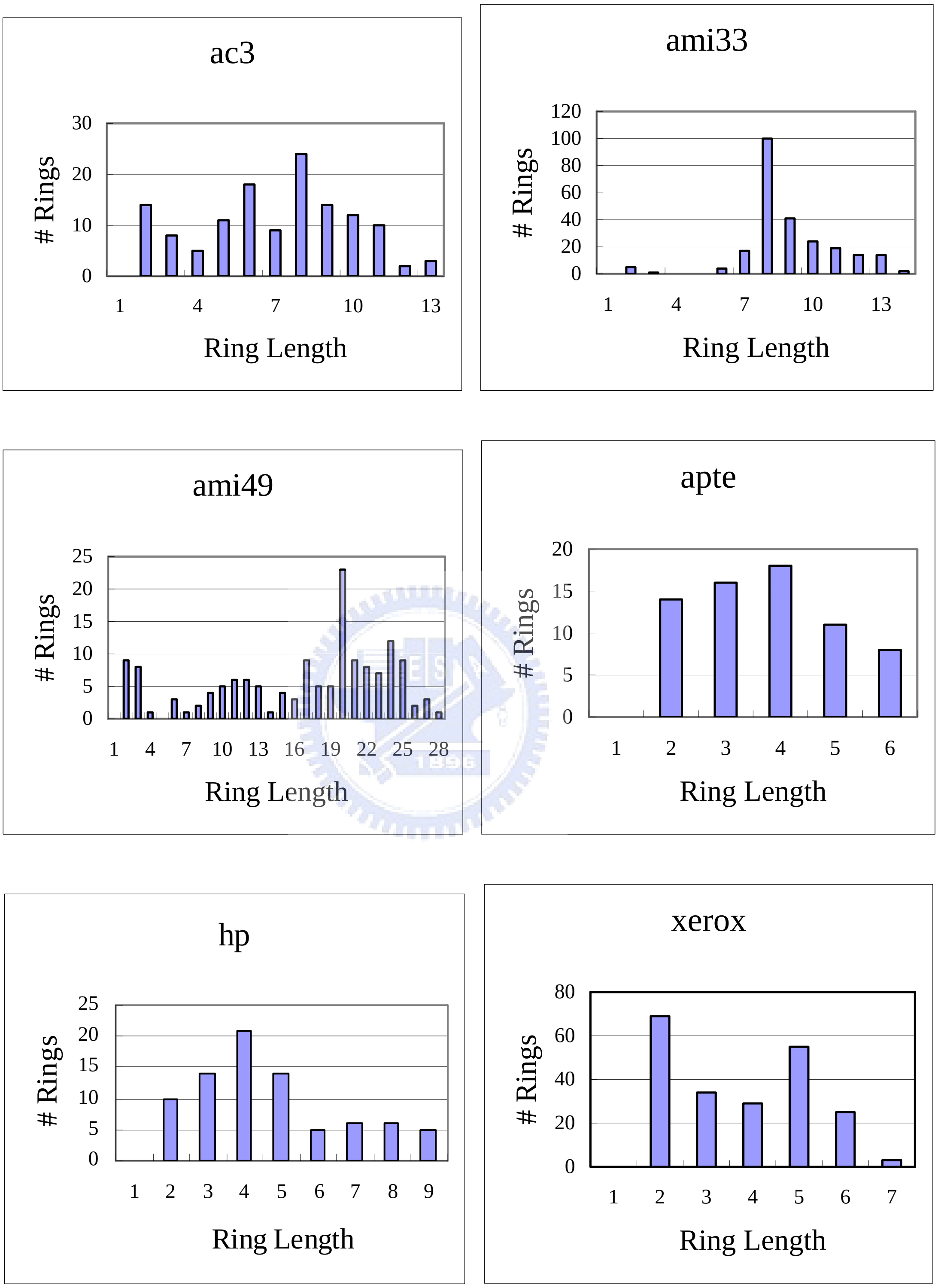


Figure 4.12 Distribution of the ring lengths for the benchmark circuits by applying OR testing.

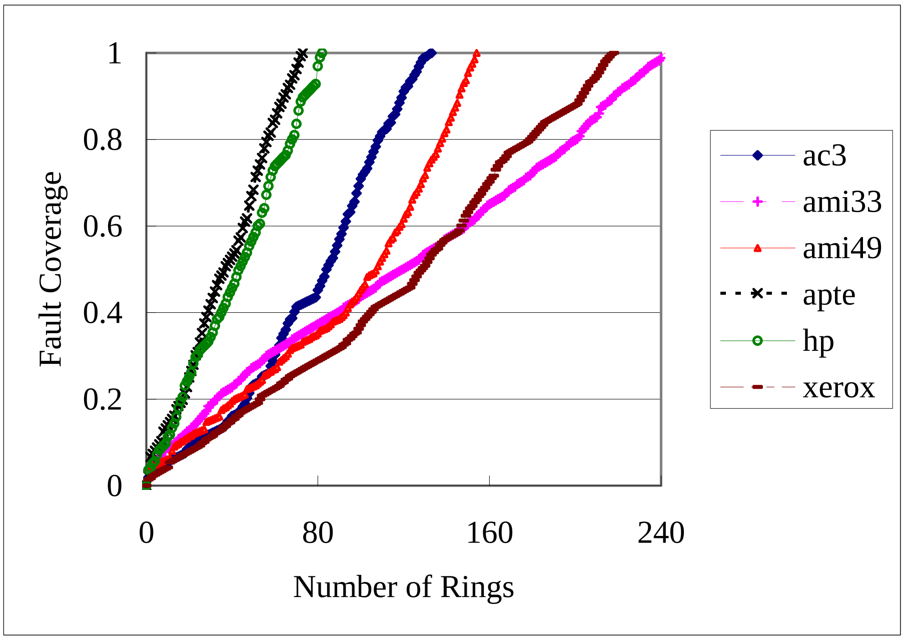


Figure 4.13 Relationship between fault coverage versus number of rings.

Chapter 5

IEEE Std. 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults

In this chapter, we consider the interconnect diagnosis problem. We propose an interconnect diagnosis scheme based on the oscillation ring test methodology for SOC design with heterogeneous cores. In addition to traditional stuck-at and open faults, the oscillation ring test can also detect and diagnose important interconnect faults such as delay faults and crosstalk glitches. The large number of test rings in the SOC design, however, significantly complicates the interconnect diagnosis problem. In this chapter, we first analyze the diagnosability of an interconnect structure, and then propose a fast diagnosability checking algorithm and an efficient diagnosis ring generation algorithm. We show that the generation algorithm achieves the *maximum diagnosability* for any interconnect. We also propose two optimization techniques, an adaptive and a concurrent diagnosis method, to improve the efficiency and effectiveness of interconnect diagnosis. Experiments on the MCNC benchmark circuits show the effectiveness of the proposed diagnosis algorithms. In all experiments, our method achieves 100% fault detection coverage and the *optimal* interconnect diagnosis resolution.

5.1 Introduction

Interconnect delays, rather than gate delays, dominate overall circuit performance in the nanometer era [105], [111] especially for systems-on-chip (SOC) designs. The *SOC design methodology* has become a reality in IC industry. Integrating reusable cores from multiple sources is essential in SOC designs, and different design-for-testability methodologies are usually required to test different cores. In particular, the long and parallel global interconnects incur significant delay and crosstalk glitch faults and thus special interconnect detection and diagnosis schemes are desirable.

Interconnect diagnosis, including the detection and location of faulty nets, plays a key role in enhancing circuit reliability and yield. It is not easy to directly apply those existing interconnect diagnosis techniques to SOC designs, and the diagnosis costs significantly increase for manufacturing and yield enhancement. Therefore, it is desired to develop an effective test scheme to reduce the costs of interconnect diagnosis.

Interconnect test and diagnosis for various applications, such as printed circuit board (PCB), multi-chip module (MCM), and systems in package (SiP), have been studied extensively in the literature [1], [20], [21], [26], [43], [46], [61], [62], [63], [69], [89], [106], [108], [114], [117], [122]. Previous works on interconnect test or diagnosis mainly focus on traditional fault models including stuck-at and bridging faults. Those diagnosis algorithms include counting sequence, walking-0 and walking-1 sequence, maximal independent test set [20], [21], [43], [61], [62], [69], [89], [106], [122], etc. An efficient way to apply these tests is to exploit the boundary-scan architecture [63], [117]. Many diagnosis algorithms presented in previous works focus mainly on special interconnect structures, especially for bus-oriented systems [108], sparsely interconnected systems [26], or FPGA designs [1], [46], [114]. The diagnosis of wire delay and crosstalk faults, often considered the most important segments of interconnect diagnosis, has attracted increasing attention since the process technology enters the deep submicron era. Much work has been done in these areas, including the development of fault models, test generation algorithms, and test methodology for delay tests [63] and BIST schemes for crosstalk faults [12], [34], [95].

Oscillation ring based test is an efficient and effective method to detect faults in a circuit or a device [6], [58]. An oscillation ring is a closed loop with an odd number of signal inversions. Once the ring is constructed, the oscillation signal appears on the ring. For a circuit with faults, some rings will not oscillate correctly. Once a set of oscillation tests have been conducted, we can locate some or all of the faults according to the test outcome [77], [78]. Whether each fault can be correctly identified, or *diagnosed*, depends on the interconnect structure and the test rings applied.

Although there is much work in the literature on oscillation test for faulty devices in analog or mixed-signal circuits [6], [58] The advantage of applying oscillation ring based diagnosis for the interconnect structure [77], [78] is that, in addition to functional faults like stuck-at and open faults, it is also capable of identifying delay faults and crosstalk glitch faults, the main sources for the loss of signal integrity [34], [95], [105]. Therefore, the oscillation ring based technique is an ideal approach to interconnect diagnosis.

The problem with the oscillation ring test, however, lies in how to achieve the *maximum testability and diagnosability* with the minimum test application time in the system-level interconnects. The main difficulty is the large number of test rings to be processed, as the number of rings can be exponential to the number of nets and is usually very large in an SOC. Therefore, it is very time-consuming, and often infeasible, to find out a minimum set of test rings to achieve the maximum diagnosability.

In this chapter, we propose an oscillation ring based scheme to diagnose interconnect faults for SOC designs. Unlike previous works, our scheme has the following features: (1) It is applicable to the general interconnect structure, (2) it is compatible with the IEEE Std. 1500 core test standard for SOC [35], [37], [53] by providing enhanced IEEE Std. 1500-compliant wrapper designs, and (3) in addition to traditional fault models (stuck-at, open, and bridging faults), delay and crosstalk glitch faults can also be handled with this approach.

We summarize our main contributions as follows:

- We give a theoretic analysis of the diagnosability of any general interconnect structure, and propose a fast diagnosability checking algorithm to greatly reduce the time complexity.
- We propose an efficient ring generation algorithm for interconnect diagnosis to minimize the number of required test rings. It exploits the fast diagnosability check to accelerate the ring generation process.
- We propose two optimization techniques to further improve the test time**:** (1) An *adaptive diagnosis method* dynamically applies the next test diagnosis ring according to the result of the previous test, and it reduces the test time by 1.54X-2.67X compared to the predetermined diagnosis method. (2) A *concurrent*

diagnosis method, in which multiple compatible rings are applied simultaneously, improves test effectiveness by up to 9.66%. In particular, neither of the two techniques incurs any hardware overhead.

- Experiments on the MCNC benchmark circuits show the effectiveness of the proposed diagnosis algorithm. In all experiments, our method achieves 100% fault detection coverage and the highest diagnosis resolution. Here, the diagnosis resolution is defined as the maximum number of nets with the same syndrome under a given set of test diagnosis rings and the single fault assumption, and the optimal diagnosis resolution is defined as that a test can diagnose all interconnects and the maximum net number is 1.

Experiments on the MCNC benchmark circuits show the effectiveness of the proposed diagnosis algorithm. In all experiments, our method achieves 100% fault detection coverage and the optimal diagnosis resolution. Here, the diagnosis resolution is defined as under single fault assumption the maximum number of nets with the same syndrome under a given set of test diagnosis rings, and the optimal diagnosis resolution is defined that a test can diagnose all interconnects and the maximum net number is 1.

The remainder of this section is organized as follows. Subsection 5.2 presents some preliminary information on the oscillation ring test scheme for the interconnect detection and diagnosis, and gives the problem formulation for the problem addressed in this paper. In Subsection 5.3, we first analyze the diagnosability of an interconnect structure, and then present an algorithm for quick diagnosability check. A theoretical analysis of the lower and upper bounds for the interconnect detection and diagnosis test scheme is then given. Subsection 5.4 presents an efficient integrated interconnect diagnosis algorithm. Subsection 5.5 presents two optimization mechanisms for interconnect diagnosis. Experimental results are reported in Subsection 5.6, and finally a brief conclusion is given in Subsection 5.7.

5.2 Oscillation Ring for Interconnect Testing and Diagnosis

In this subsection, we give some preliminary knowledge of the oscillation ring (OR) test scheme for interconnect testing, including the global test structure, basic test operations, the frequency measurement formula, the modified wrapper cell designs, and the interconnect model for diagnosis.

5.2.1 Oscillation Ring Test Architecture

In this subsection, we discuss the *interconnect oscillation ring test* (*IORT* for short) for SOC interconnects [77], [78]. Figure 2.4 (or Figure 4.1), test architecture for wire delay and crosstalk detection, and wire delay measurement, illustrates the global counter-based test architecture for both wire delay and crosstalk glitch detection for SOC IC interconnects. This test architecture implements the IEEE IEEE Std. 1500 compliant core test standard. In IEEE Std. 1500, each input/output pin of a core is attached with a *wrapper cell*, and a centralized *test access mechanism* (*TAM*) is provided to coordinate all test processes. In addition to the normal input/output connections, all wrapper cells in a core can also be connected with a shift register, usually referred to as a *scan path*, to facilitate test access. An enhanced wrapper cell design has been proposed to provide extra connections and inversion control so that the oscillation rings can be constructed through the wires and the boundary scan paths in cores [77], [78]. For example, the oscillation ring test architecture shown in Figure 2.4 consists of one oscillation ring and two neighboring nets, and the scan paths in cores C_1, C_2, C_3 and C_4 are part of the oscillation ring.

The target fault models of this test architecture are stuck-at, open, wire delay and crosstalk glitch faults. In addition to fault detection, measuring the wire delay fault can also be achieved. If an oscillation ring fails to oscillate, there exists stuck-at or open fault(s) in the components of the oscillation ring. The period of the oscillation signal is measured by using a delay counter in a local core to test wire delay faults, and a similar scheme is also applied for crosstalk glitch detection.

A local counter is included in each core, and a central counter is in the TAM of an SOC. The central counter in the TAM is enabled by the signal *OscTest*, and triggered by the system clock. A local counter is connected to one wrapper cell in each core; however, it can be accessed by every wrapper cell through the wrapper cell chain. When an oscillation ring passes a core, an internal scan path is formed to connect the oscillation signal to the local counter. For example, consider core C_1, passed by the oscillation ring in Figure 2.4. The oscillation signal is fed to the local counter through a series of modified wrapper cells. When an oscillation test session starts (*OscTest* = 1), the TAM enables its own central counter as well as all local counters in cores. After the central counter in the TAM counts to a specific number n, the oscillation test session terminates and all local counters are disabled (*OscTest* = 0). Then all the local counter contents can then be scanned out to ATE for inspection.

Assume that m oscillation rings are tested. Let the frequency of the system clock be f, and the delay counter contents of the rings be n_1, n_2, …, n_m, respectively. An estimation of the i-th ring's oscillation frequency f_i can be approximated by

$$f_i = f \times n_i / n \qquad (5.1)$$

Since the frequency of each ring is predetermined during the design phase, a wire delay fault is detected and measured by inspecting the contents of the delay counters. Let the oscillation frequency of the rings, according to the timing specification, be $f_{min} \le f_i \le f_{max}$, with the unit of measuring T_0 (= n/f). Thus, we have $n_{min} \le n_i \le n_{max}$, where $n_{min}= f_{min}\times T_0$ and $n_{max}= f_{max}\times T_0$. Let ξ be the resolution of delay measurement, and ε be the maximum measurement error. Since a counter's maximum measurement error is ±1, the requirement for ε should be the reciprocal of f_{min} and T_0.

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \le \zeta \qquad (5.2)$$

An example for delay measurement is given as follows. Let the frequency specification of the oscillation rings be 4 MHz to 400 MHz and ξ be 0.001, implying the counter content n_{min} is at least 1000. From Equation (5.2), we have the required T_0 to be 250μs. This example illustrates the feasibility of the oscillation test scheme from a measurement prospect, and this frequency specification is actually compliant with ATE specifications.

In order to detect the crosstalk glitch in Figure 2.4, consider wire b_1 and b_2 in Figure 2.4 and assume that there is a coupling crosstalk effect between *victim nets* b_1 (or b_2) and the *aggressor nets* a_1 (or a_2) of the oscillation ring. Interconnects (b_1 and b_2) adjacent to an oscillation ring are affected by the oscillation signal if there is an excessive coupling capacitance between these two lines (a_1 and b_1, a_2 and b_2) [75], [76]. When the oscillation signal occurs, crosstalk-induced glitches appear on the victim nets b_1 and b_2. Similar to the case for wire delay detection, the glitches on net b_1 (b_2) are sent to local counters in core C_4 (C_3) through a series of modified wrapper cells. Since there is an inverter per modified wrapper cell in the OR test mode, the induced glitches are amplified when the glitches pass through the wrapper cells, and the amplified glitches are used to trigger the local counters in core C_3 and C_4 for glitch detection and diagnosis.

5.2.2 Enhanced IEEE Std. 1500-Compliant Wrapper Cell Design

An oscillation ring for interconnect test consists of interconnect wires and parts of the scan path in each core where the ring passes. Therefore, an IEEE Std. 1500-compliant wrapper cell must provide necessary paths between input/output ports and scan in/scan out ports. If an oscillation test is used to test wires attached to/from pads, the IEEE Std. 1500-compliant boundary scan cells also have to be modified in a similar way in order to facilitate the scheme. In this subsection, we present the enhanced wrapper cell designs.

A normal wrapper cell provides two types of paths: a scan path connecting all wrapper cells into a shift register, and an interface buffering between internal core and the wire connected to the pin. Whenever an oscillation test is applied, a third combination path must be provided. For an input pin, the wrapper cell must connect the pin input (IN) to the scan output (SO); while for an output pin, it should connect scan in (SI) to pin output (OUT) during an oscillation test session.

The enhanced wrapper cell design is shown in Figure 2.9 (also Figure 4.3) for input and output cells. In each cell, two MUXs are added for path selection. For an input wrapper cell, the extra paths are SI→SO and IN→SO; while for an output wrapper cell, the extra paths are SI→SO and SI→OUT. The added inverting and non-inverting buffers are used to generate oscillation signals for the OR test; however, in an input wrapper cell, only one type of buffer is provided due to the limited control signals. We assume that an inverter is used in an input cell. Two control signals are needed in each enhanced wrapper: signal *OscTest* is a global control signal; while the signal *sel* is only used in the input wrapper cell, and the signal *inv* is only used in the output wrapper cell to ensure the odd parity of each ring. Signals *sel* and *inv* are set individually and scanned into the wrapper cells before an oscillation ring test session starts.

5.2.3 Crosstalk Glitch Fault Detection

In order to verify that the proposed architecture can be applied to detect crosstalk-induced glitches, we conduct HSPICE simulation with TSMC 0.18μm technology. An oscillation signal is generated on a ring as shown in Figure 2.4, and a 1 mm wire with three times of normal coupling capacitance is assumed. The results are shown in Figures 5.3 and 5.4. Figure 2.6 shows the oscillation signal on the ring, the induced glitches on the victim net, and the output of the counter. The crosstalk-induced glitch shown in Figure 2.6 can be detected and verified since the counter changes the state on every positive glitch.

Figure 2.7 gives an illustration on how to detect the glitches. The oscillation signal is shown in top of Figure 2.7, and the induced positive glitch, whose peak value is about 0.8V, is shown in the middle set of figures. This glitch is amplified by a detector, which is a specially designed inverter in our IEEE Std. 1500-compliant input wrapper cell. We may adjust the *W*/*L* ratio of the detector's transistors to determine the detection threshold of glitches [75], [76]. For example, in our experiment we set $(W/L)_{pu}/(W/L)_{pd}$ to be 1/4. In other words, the width of the pull-down nMOS is four times that of the pull-up pMOS, while the channel lengths of both transistors are set to the minimum. Since the positive crosstalk glitch and the negative glitch are symmetric, we only need a design to detect either a positive glitch or a negative glitch. Here we just give the basic detection principles for the positive glitch detection shown in Figures 5.3 and 5.4.

The detector's output is passed through a chain of wrapper cells. In our experiment, there are five wrapper cells in the chain, and it can be seen that a near rectangular pulse is formed. This pulse is used to trigger a two-port T-type flip-flop (2P-TFF) successfully without causing any setup/hold time violation. The 2P-TFF can be triggered by two different signals, one port is triggered by the crosstalk glitch signal and the second port is trigger by the system clock to scan out the counter contents. In the oscillation test mode, this 2P-TFF is triggered by the amplified glitches and acts as a counter. When we need to scan out the counter contents, it is triggered by the system clock. All the transistors, except for the detector, are minimum-sized.

The crosstalk is caused by excessive coupling capacitance between adjacent wires, and it can incur two types of errors: glitch and delay [75], [76], [108]. When there is a signal transition in the aggressor while the victim signal is stable, a crosstalk-induced glitch appears in the victim net. On the other hand, a crosstalk-induced delay occurs when the victim net makes a signal transition opposite to the direction of the aggressor net's signal at roughly the same time. The crosstalk-induced delay is just a superposition of the original signal in the victim and the glitch induced by the aggressor [75], [76]. Therefore, it is possible to detect crosstalk-induced delay simply by detecting induced glitches [75], [76].

5.2.4 Interconnect Graph Models

Further to Section 2.1, a more formal and detailed discussion on the issue of modeling interconnects for diagnosis is discussed as follows. A circuit example consisting of three cores (C_1, C_2, and C_3) and four nets (n_1, n_2, n_3, and n_4) is illustrated in Figure 5.1(a). There are three rings in Figure 5.1. The first ring consists of nets n_1 (and its right-hand side branch), n_2, and n_3, and it passes all three cores. The second ring consists of n_1 (and its left-hand side branch) and n_3, and scan paths in C_1 and C_3. The third ring consists of nets n_1 (and its right-hand side branch) and n_4, and scan paths in C_1 and C_2.

Since the internal scan paths can be separately tested and diagnosed, we shall assume that they are fault-free. The diagnosis problem is restricted to interconnect wires among modules in this paper. The goal for interconnect diagnosis is to diagnose any single fault for each net segment to achieve the highest diagnosis resolution. For diagnosis purpose, all wire segments of a net are different since they are may be passed by different rings. Therefore, we shall assign distinct labels to them. For example, in Figure 5.1(b), the seven net segments are labeled as edges e_1 to e_7, and our goal is to diagnose any single fault on every edge or net segment to achieve the optimal diagnosis resolution.

To perform the interconnect test for detection or diagnosis with an oscillation ring, we must find rings to cover all nets to be tested. In order to simplify the interconnect diagnosis problem, we model the SOC circuit by a *hypergraph*, and model interconnects by a *hypernet* in Figure 5.2.

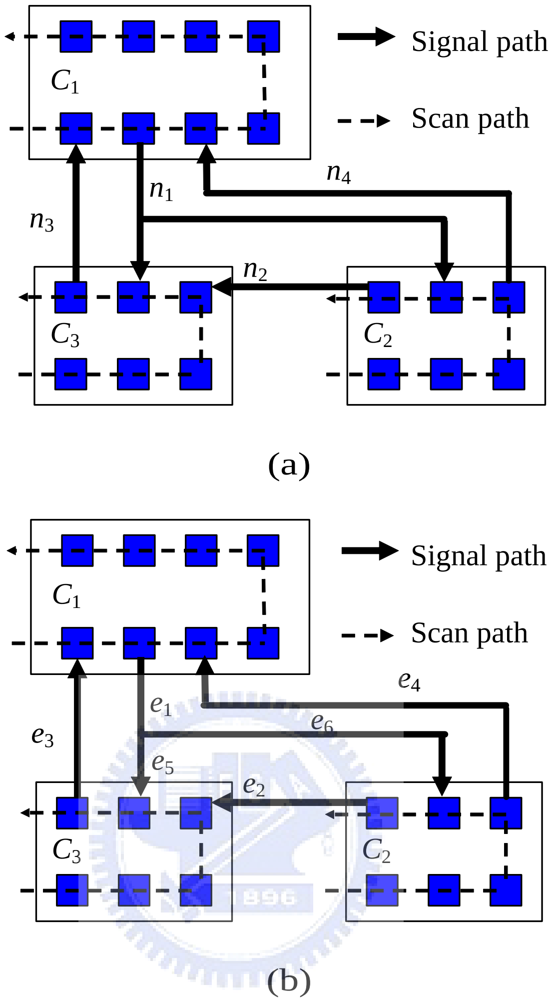
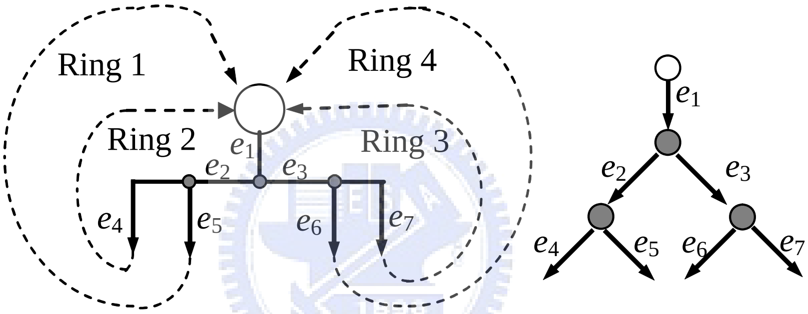


Figure 5.1 An example SOC circuit: (a) a hypergragh and 3 hypernets in the interconnect structure, (b) labelling all net segments or edges.

Definition 5.1: A *hypergraph* $H = (V, L)$ consists of a vertex set V and an edge set L consists of multi-terminal edges connecting a set of vertices $V_i \subseteq V$, $| V_i | \geq 2$. Such an edge is referred to as a *hypernet*.

For example, n_1 in Figure 5.1 is a hypernet connecting three terminals (pins). Furthermore, we assume that in an n-terminal hypernet, one terminal is the source node (i.e., sending signal) while the others n–1 are the sink nodes (i.e., receiving signals).

In Figure 5.1, the circuit structure of an SOC can be directly transformed into a hypergraph, and each pin is a vertex while each signal net is a hypernet. However, this graph model is not good enough for diagnosis, since different parts of the same net (i.e. different net segments) affect different rings. Consider the 5-terminal hypernet shown in Figure 5.2(a), divided into seven net segments e_1 to e_7. If edge e_1 is faulty, all four rings will not oscillate correctly. A faulty e_2 affects rings 1 and 2, a faulty e_3 affects rings 3 and 4, and faults on edges e_4, e_5, e_6 and e_7 affect rings 1, 2, 3, and 4, respectively. For diagnosis purpose, all these seven segments are different.



(a) (b)

Figure 5.2 (a) a *hypernet*, and (b) the corresponding *interconnect diagnosis graph model* (Similar to Figure 2.3).

From the above discussion, the hypernet cannot be used for diagnosis. Therefore, we transform an interconnect structure into an interconnect diagnosis graph model as follows. The scan path and wrapper cells in a core are lumped into a single *terminal node*, as we assume that they are fault-free. The fanout points of a hypernet form dummy *intermediate nodes*, and a net segment connecting two nodes is modeled as an edge. For example, the graph model for the hypernet in Figure 5.2(a) is transformed into an *interconnect diagnosis graph model* in Figure 5.2(b), where the white node is a terminal node and gray nodes are intermediate nodes. An *edge* is the smallest unit of net segments that can be uniquely diagnosed. For diagnosis observation, any stem edge affects all its downstream nodes and edges in Figure 5.2(b).

Definition 5.2: A directed *graph* $G = (V, E)$ consists of a vertex set V and an edge set E, and each edge in E is an ordered pair (u, v), where $u, v \in V$.

The interconnect structure in an SOC can thus be transformed into a graph G, and the vertex set includes all cores (terminal nodes) and fanout nodes (intermediate nodes). A ring r is a subgraph $r \subseteq G$ such that all the edges in r form a cycle. Since our goal is to diagnose the interconnect structure, we shall concern only the *edges* in the following discussion. Thus, a ring can be treated as a set of edges.

5.2.5 Problem Formulation

This subsection explores the problem complexity and gives the formal problem formulation and related constraints.

5.2.5.1 Problem Complexity

The goal of this work is to find a set of test rings that achieve the highest diagnosis resolution in the shortest time. To achieve this goal, we need to find a minimum number of rings so that all faults can be correctly identified. The main difficulty, however, is the high complexity with the huge problem size. An SOC usually contains a large number of interconnect wires, and the possible number of rings is likely to be exponential to the number of nets, although the exact number of rings depends on the interconnect structure. Consider the simple example shown in Figure 5.3, in which m cores are connected by a bus of width n, denoted by n-bus. For simplicity, we shall assume that each core is passed by a ring only once. For example, the ring shown in Figure 5.3 is of length 2.

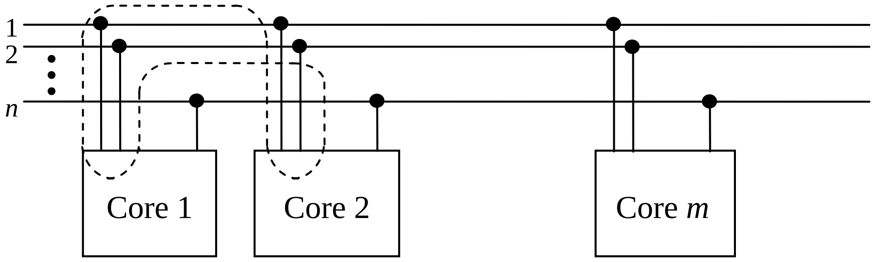


Figure 5.3 Interconnect diagnosis for a bus-structure.

In general, given an *n*-bus and a set of *i* modules, there are C_i^n different ways to connect these modules into a ring of length *i*. Therefore, the total number of all possible rings in this system of *m* cores and an *n*-bus is:

$$\sum_{i=2}^{\min(m,n)} C_i^m C_i^n .$$

Obviously, this number is at least exponential to min(*m*, *n*), and thus it is computationally intractable to search all possible rings and find a minimum subset of them for complete fault diagnosis. Even if we restrict the problem of the diagnosis check to a given set of rings, a brute-force exact algorithm is still very expensive. Therefore, it is desirable to find an efficient algorithm to achieve the optimal diagnosis resolution with a small number of test rings.

5.2.5.2 Problem Constraints and Constraints

In this paper, we aim to develop an algorithm to find (1) a small set of detection rings for 100% fault detection (IORT), and (2) an extra set of diagnosis rings for the highest diagnosis resolution (IORD). Alternatively, if the highest resolution is not necessary, the algorithm shall find the smallest set of test rings corresponding to the required resolution

As defined earlier, a ring is a set of net segments forming a closed loop. However, the following constraints must be satisfied when a ring is generated.

(1) *Hypernet constraint*: If two edges e_i and e_j belong to the same net or hypernet and e_i is neither a downstream nor an upstream edge of e_j, they cannot belong to the same ring. In other words, if two edges e_i and e_j belong to the same ring, e_i is either a downstream or an upstream of e_j (see Figure 5.2).

(2) *Frequency (period) constraint*: Let the wire delay of an edge e be $d(e)$, and the delay of a wrapper cell w under oscillation test be $d(w)$. The wire delay of an edge in a ring r can be detected if the following condition holds: $1/f_{max} \le \sum_{e \in r} d(e) + \sum_{w \in r} d(w) \le 1/f_{min}$.

(3) *Core constraint*: Let the total number of cores be m and the number of rings constructed at the same time in a test session T be $|T|$. We need at least one counter for each ring to measure whether the oscillation frequency is correct; therefore, at least $|T|$ local counters are required in this session for delay measurement. Let the number crosstalk fault detectable in this session be $n_{xtalk}(T)$. Since each target crosstalk fault must be checked by a local counter and each module is assumed to contain one local counter, the following condition holds: $|T| + n_{xtalk}(T) \le m$.

Since the number of rings is usually too large to be checked exhaustively, it is difficult to find the minimum set of rings for fault detection and diagnosis. In order to handle the problem efficiently, we shall develop fast algorithms for diagnosability check and ring generation.

5.3 Interconnect Diagnosability

In this section we present a theoretic framework for interconnect diagnosability analysis. We provide some diagnosability conditions for net segments (edges). With such conditions, we develop an algorithm for fast diagnosability check. We also derive the lower and upper bounds for the interconnect detection and diagnosis test scheme.

5.3.1 Diagnosability Analysis

Given a circuit consisting of n edges $E = \{e_1, e_2, \ldots, e_n\}$ and a set of m oscillation rings $R = \{r_1, r_2, \ldots, r_m\}$. Once a ring is constructed, the test outcome is either "*pass*" (P) or "*fail*" (F). When an edge e_i is faulty, the test outcome of applying the m rings is said to be the *syndrome* of faulty e_i.

Definition 5.3: A fault on edge e_i and a fault on edge e_j are *distinguishable* under the test set R if the syndrome of faulty e_i and faulty e_j are different.

Definition 5.4: An edge is said to be *single-fault diagnosable* under the test set R if a fault on the edge can be correctly identified, given that there is at most one fault in the interconnect structure.

Edge e_i is single-fault diagnosable if and only if its syndrome is different from all the other edges' syndromes. The diagnosability problem is to determine whether edge e_i is single-fault diagnosable under the test set R. Assume that edge e_i belongs to a set of l different rings $R_i = \{r \mid r \in R, e_i \in r\}$. In other words, R_i is a subset of R with cardinality l ($|R_i|=l$). Let $E_i = \bigcap_{r \in R_i} r$ be the set of edges appearing in all rings of R_i, obviously, $e_i \in E_i$. An example is illustrated in Figure 5.4, where e_i belongs to four different rings $R_i = \{r_1, r_2, r_3, r_4\}$, and thus $|R_i|=4$. $E_i = r_1 \cap r_2 \cap r_3 \cap r_4 = \{e_i, e_j, e_k\}$ contains edges appearing in all rings in R_i.

Lemma 5.1: A fault on edge e_i and a fault on edge e_j are *distinguishable* under the test set $R \Leftrightarrow R_i \neq R_j$.

Proof: $\Leftarrow$ The fact $R_i \neq R_j$ implies that there exists a ring r such that either (1) $r \in R_i \wedge r \notin R_j$, or (2) $r \in R_j \wedge r \notin R_i$. Thus, the syndromes of faulty e_i and faulty e_j are different.

$\Rightarrow$ When $R_i = R_j$, both faulty e_i and faulty e_j fail the same set of rings, and thus they have the same syndrome. □

Theorem 5.2: Edge e_i is *single-fault diagnosable* $\Leftrightarrow R_i \neq R_j$ for all $1 \leq j \leq n$ and $j \neq i$.

The correctness of Theorem 5.2 follows the result of Lemma 5.1. It takes $\mathbf{O}(n^2m)$ time to verify Theorem 5.2 (where n is the number of nodes and m is the number of edges), since each pair of edges have to be compared. In order to reduce the complexity for diagnosability check, the following theorems can be used.

Theorem 5.3: Edge e_i is single-fault diagnosable if $|E_i| = 1$.

Proof: Assume that edge e_i is not single-fault diagnosable. From Theorem 2, there must exist an edge e_j such that $j \neq i$ and $R_i = R_j$. Therefore, both e_i and e_j belong to E_i and thus $|E_i| > 1$.

The application of Theorem 5.3 can greatly reduce the time complexity for the diagnosability check of an edge if the edge is single-fault diagnosable. However, the reverse of Theorem 5.3 is not true, since $|E_i| = 1$ is only a sufficient condition for single-fault diagnosability.

Note that, when the sufficient condition given in Theorem 5.3 is true, we must have $l \geq 2$. When R_i has only one ring (i.e., l=1), E_i is the set of all edges in this ring.

Since a ring consists of at least two edges, $|E_i|$ must be greater than 1.

When the intersection of l rings consists of multiple edges, it is still possible to diagnose the faults as outlined in the following theorem.

Theorem 5.4: Let R_i' be any non-empty subset of R_i for an edge e_i, and $E_i^{'} = \bigcap_{r \in R_i^{'}} r$.

Edge e_i is single-fault diagnosable ⇔ $\forall\ e_k \in E_i\text{'}-\{e_i\}$, e_i and e_k are distinguishable.

Proof: ⇐ When at least one ring in R_i' oscillates correctly, e_i must be fault-free. On the other hand, when no rings in R_i' oscillate correctly, at least one edge in E_i' is faulty. Since all edges in $E_i\text{'}-\{e_i\}$ are distinguishable from e_i, we know whether e_i is faulty. Therefore, e_i is also single-fault diagnosable.

⇒ Assume that there is an $e_k \in E_i\text{'}-\{e_i\}$ and e_k is not distinguishable from e_i. When every ring in R_i' fails, it may be attributed to either e_k or e_i. Thus, e_i is not single-fault diagnosable. □

Theorem 5.4 shows that not all rings in R_i are necessary to diagnose e_i, and a subset R_i' is informative enough if and only if e_i is distinguishable with other edges in E_i'. The following corollary is a natural extension of Theorem 4.

Corollary 5.5: Let R_i' be any non-empty subset of R_i for an edge e_i, and $E_i^{'} = \bigcap_{r \in R_i^{'}} r$.

If for each $e_k \in E_i\text{'}-\{e_i\}$, e_k is single-fault diagnosable, then edge e_i is also single-fault diagnosable.

An example for the above definitions, theorems and corollaries is shown in Figure 5.4. Let the edge under consideration be e_i, then $R_i = \{r_1, r_2, r_3, r_4\}$, and $E_i = \{e_i, e_j, e_k\}$. Since R_i' can be any non-empty subset of R_i, we may choose $R_i\text{'} = \{r_2, r_3\}$, and thus $E_i\text{'} = \{e_i, e_j, e_k\}$. It is not necessary to have both e_j and e_k diagnosable to make e_i diagnosable. For example, let faults on e_j and e_k be indistinguishable; if a fault on e_i

is distinguishable with $\{e_j, e_k\}$, then e_i is diagnosable according to Theorem 5.4.

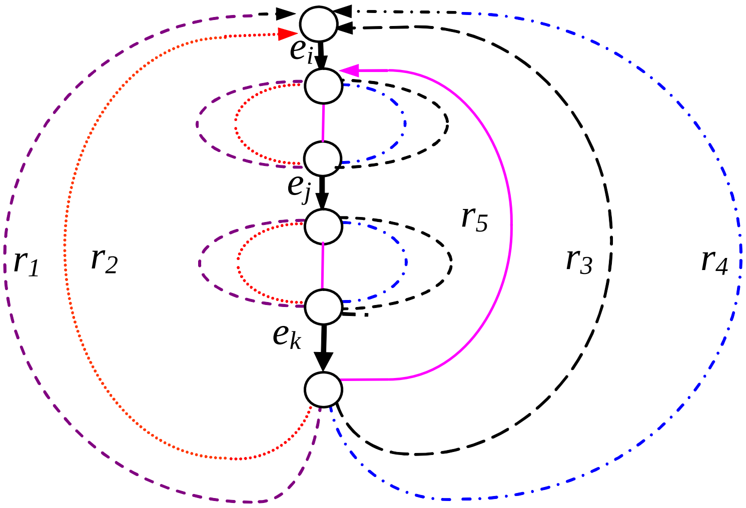


Figure 5.4 An interconnect diagnosis graph example.

Note that the above analysis applies to all types of faults except crosstalk glitches since they can be located directly from the test results of each ring. For example, consider the example shown in Figure 2.4. If there are detectable glitches due to the crosstalk fault between wires a (a_1 or a_2) and b (b_1 or b_2), they will be observable through the counter in core C_3 and C_4, and hence the fault is located.

5.3.2 Heuristic Diagnosability Check

In order to accelerate the process of diagnosability analysis, we propose a heuristic for diagnosability check in this section. Consider two edges e_i and e_j. According to Lemma 5.1, faults on these two edges are distinguishable if $|R_i| \neq |R_j|$. Conversely, if faults on e_i and e_j and are indistinguishable, then we must have $|R_i| = |R_j|$. Thus, as the first step, we sort and partition all edges according to the number of rings passing them (i.e., $|R_i|$ for edge e_i). Edges e_i and e_j are put into the same group when $|R_i| = |R_j|$. Obviously, faults on two edges will be distinguishable if the two edges are in two different groups. Therefore, we only need to check whether the fault on an edge is distinguishable from faults on the edges that are in the same group as the target edge. The diagnosability analysis should start with the group with the highest $|R_i|$. For example, in Figure 5.4, e_j and e_k are in the same group as $|R_j|=|R_k|=5$, distinguishable from $|R_i|=4$.

The second heuristic is to apply Theorem 5.3 first to check the diagnosability of an edge since it is much easier. Since the condition of Theorem 5.3, $|E_i|=1$, is only sufficient but not necessary to guarantee that e_i be single-fault diagnosable, it is still possible that e_i is single-fault diagnosable when $|E_i|\neq 1$. In this case, we need to compare R_i with R_j for each e_j in the same group as e_i.

To avoid the aforementioned problem, a third heuristic is used. The most likely reason for diagnosable e_i with $|E_i|\neq 1$ is that there exists an e_j such that $R_j \supset R_i$. When the edge e_j has been checked and removed from the check list before edge e_i is processed, we shall not run into this problem. To further simplify the diagnosability check, whenever edge e_i is found to be single-fault diagnosable, it should be removed from all rings in R_i, as suggested by Corollary 5.5. The flowchart of the diagnosis checking heuristic is shown in Figure 5.5.

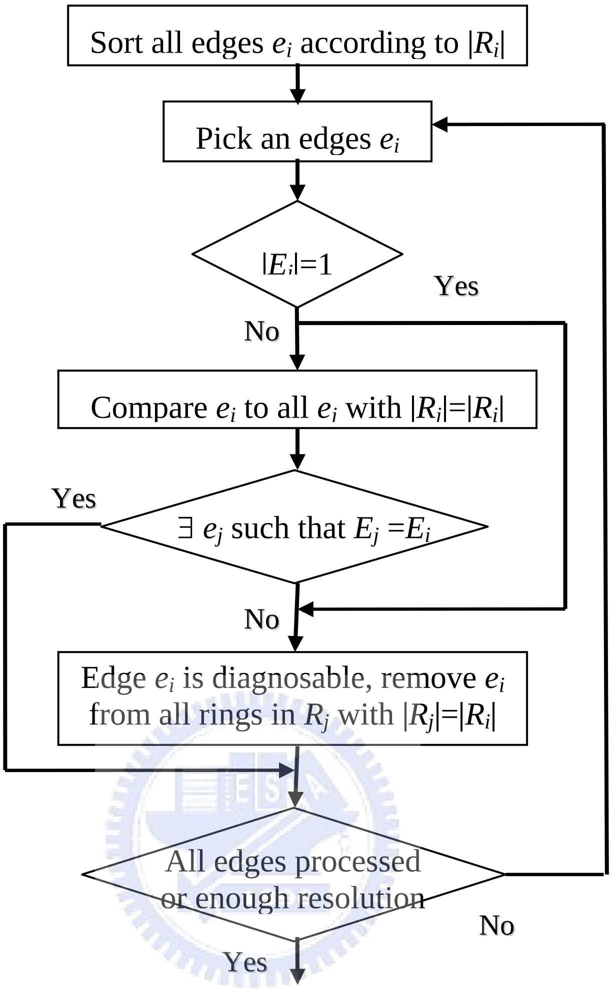


Figure 5.5 Flow chart of the heuristic for diagnosability checking.

Finally, when two faults are indistinguishable, they should be collapsed into the same *equivalent class* so as not to be compared twice.

The interconnect diagnosis heuristic algorithm is illustrated as follows. Consider the graph shown in Figure 5.6, which is the graph model for Figure 5.1(b). There are three rings in the figure: $r_1 = \{e_1, e_2, e_3, e_6\}$ (ordered by e_1, e_6, e_2, and e_3), $r_2 = \{e_1, e_3, e_5\}$, and $r_3 = \{e_1, e_4, e_6\}$.

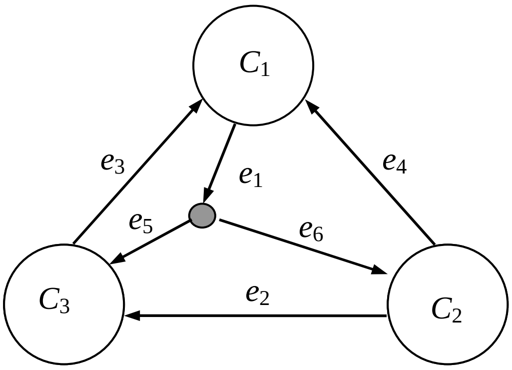


Figure 5.6 A diagnosability example for Figure 5.1(b).

A straightforward way to represent the diagnosability information is to use a matrix. The matrix representation for the example of Figure 5.6 is illustrated in Figure 5.7(a), where each column represents an edge and each row represents a ring. The entry (i,j) is "1" if ring i contains edge j. Note that the edges are sorted and partitioned into three groups that are separated by the dashed line. The first group consists of the edge e_1, which is contained in all three rings (i.e., $|R_1|=3$). The second group consists of edges e_3 and e_6, with each of them being contained in two rings (i.e., $|R_3|=|R_6|=2$). The third group consists of the remaining three edges, with each of them being contained in only one ring (i.e., $|R_2|=|R_4|=|R_5|=1$).

The syndrome of e_3 = {110} indicates that the test results of r_1 and r_2 are incorrect and r_3 is correct when e_3 is faulty; the syndrome of e_6 = {101} indicates that r_1 and r_3 are incorrect and r_2 is correct when e_6 is faulty. The diagnosability analysis is applied to the groups in the non-increasing order of $|R_i|$. We start with the group with $|R_i|$ = 3 (i.e., {e_1}), followed by the group with $|R_i|$ = 2 ({e_3, e_6}), and finally the group with $|R_i|$ = 1 ({e_2, e_4, e_5}).

The diagnosability checking proceeds as follows. First, since the first group contains only one edge and the syndrome of e_1 = {111}, e_1 is single-fault diagnosable. Then we check edges in the next group e_3 and e_6. Edge e_3 is contained in rings r_1 and r_2, and the intersection of these two rings is {e_1, e_3}. Since edge e_1 is diagnosable, edge e_3 is single-fault diagnosable according to Corollary 5. Similarly, edge e_6 is also diagnosable for the same reason.

$$\begin{array}{c|c:cc:ccc}
 & e_1 & e_3 & e_6 & e_2 & e_4 & e_5 \\
\hline
r_1 & 1 & 1 & 1 & 1 & & \\
r_2 & 1 & 1 & & & & 1 \\
r_3 & 1 & & 1 & & 1 &
\end{array}$$

(a)

$$\begin{array}{c|c:cc:ccc}
 & \surd & \surd & \surd & & & \\
 & e_1 & e_3 & e_6 & e_2 & e_4 & e_5 \\
\hline
r_1 & 1 & 1 & 1 & 1 & & \\
r_2 & 1 & 1 & & & & 1 \\
r_3 & 1 & & 1 & & 1 &
\end{array}$$

(b)

Figure 5.7 Matrices for the heuristic diagnosability checking.

Edges e_3 and e_6 are then marked and removed from the rings, as shown in Figure 5.7(b). The reason is that they are already known to be diagnosable, which means they can be distinguished with any other faults. As a result, they do not need to be considered in the following process. There is only one edge remained in each ring, and thus edges e_2, e_4, and e_5 are single-fault diagnosable, again due to Corollary 5.

5.3.3 Number of Tests

In this subsection, we analyze the lower and upper bounds on the test time or the number of tests in terms of the number of rings for our IORT scheme and *interconnect oscillation ring diagnosis* (IORD) scheme.

The test time required for both detection and diagnosis is proportional to the number of rings. Therefore, it is important to minimize the number of rings required for either detection or diagnosis. In general, it requires more tests for fault location than fault detection.

For the IORT scheme, in an *n*-edge system, the lower bound on fault detection test is 1 if all the edges form a single large ring. This lower bound, however, is usually not achievable. A more realistic bound on fault detection tests is obtained by considering the pin order in cores. Thus, a smaller number of rings may be achievable through pin reordering. The upper bound of fault detection is *n*.

For IORD, to estimate the minimum number of tests required for diagnosis, we shall examine the theorems given in the previous section. In order to ensure that an edge is single-fault diagnosable regardless of other edges, it must belong to at least two rings and it is the only common shared edge of these two or more rings according to Theorem 5.3. A minimum set of rings satisfying this condition consist of $\lfloor n/2 \rfloor$ distinct 2-edge rings for the set of *n* edges. An illustrative example of this situation is shown in Figures 5.10 and 5.11, for which edges e_1 is single fault diagnosable for $|R_1|=3$, e_3 and e_6 are diagnosable according to Theorem 3, and all the other edges are diagnosable according to Corollary 5.5.

Another interesting special case is the bidirectional bus. An *n*-line bus can be diagnosed with $n–1$ rings, where a ring is constructed for every pair of adjacent nets. It can be verified that the internal $n–2$ lines are diagnosable due to Theorem 5.3, while the other three nets are diagnosable due to Corollary 5.5.

For a random interconnect structure, the number of diagnostic rings may be difficult to find. We estimate the number of rings required for diagnosis as follows. Let the number of rings required for fault detection be $|R_t|$. In the worst case, we need to a new ring for each edge to satisfy Theorem 5.3, total m edges. Therefore, $|R_d|=|R_t|+m$ predetermined rings should be enough for fault diagnosis if such rings exist. In general, if we can find a distinct ring for each net segment, we should be able to diagnose all net segments with m rings. These m rings are the extra test rings (or the number of tests) to achieve the highest diagnosis resolution for each net segment (i.e., total $|R_d|$), in addition to the original test time for the interconnect detection scheme ($|R_t|$). We will show the details on how to get $|R_d|$ and $|R_t|$ in the following interconnect diagnosis algorithm.

Also, we will show that the adaptive diagnosis can further reduce the number of tests from the predetermined approach of linear complexity to logarithmic complexity for a relatively balanced adaptive approach to be presented in Section 5.5.2.

5.4 Interconnect Diagnosis Algorithm

In order to uniquely identify the faulty net segment, we need to ensure the maximuml diagnosability. The diagnosis resolution is defined as the maximum number of nets with the same syndrome under a given set of test rings. A higher resolution implies a smaller number of edges in each indistinguishable set. In general, we need more rings to achieve a higher level of diagnosis resolution. Our target is to diagnose every fault on every net segment, defined as the maximum diagnosis resolution.

5.4.1 Fast Heuristic Diagnosability Check

We propose a heuristic to find a small set of rings for single fault diagnosis. The SOC under test is modeled as a hypergraph H. This graph is then transformed into graph $G = (V', E)$ as outlined in Section 2.4. The vertex set V' consists of cores and fanout points (intermediate nodes). The edge set E consists of net segments partitioned from the original hypernets as explained in Figure 5.2(b). Our goal is to generate a predetermined set of rings to diagnose all edges in E.

Since we need to detect the interconnect structure before diagnosis, the set of fault-detection test rings R_t should be applied first. In order to find R_t, we propose a heuristic algorithm to find a minimum set of rings that cover all 2-pin nets under test. The algorithm is a modified depth-first search and works as follows. The SOC under test is first modeled as a hypergraph H, and then transformed into graph $G = (V', E)$ with 2-pin nets only. We generate a ring containing a 2-pin net $(u, v) \in E$ by starting from vertex v, an input pin. Then we try to find an output pin w that locates in the same core as v, and w is connected to a 2-pin net that is not yet covered by any other ring. If no such unvisited 2-pin net (w, x) exists, we just select the first available output net from any available set of output pins. This process is repeated until a ring is found. The procedure then goes over again until all 2-pin nets are covered.

The above heuristic works as follows. Whenever we start looking for a new ring, we explore the paths containing 2-pin nets that are not yet covered. In this way, each new ring may cover as many other uncovered nets as possible. After all rings having been generated, a simple reverse order simulation is conducted to remove redundant rings. A net is oscillation ring testable if there exists at least one ring containing this net. The algorithm is shown in Figure 5.8.

5.4.2 Interconnect Oscillation Ring Construction for Fault Detection (IORT)

As a preprocessed unit to interconnect diagnosis, we first introduce IORT which details are referred in Chapter 4.

```
Algorithm: IORT (Interconnect Oscillation Ring for Fault Detection)
Input: A hypergraph $H = (V, L)$ representing a circuit
Output: A list of rings $R_t$

Transform hypergraph $H$ into a new graph $G = (V', E)$ with equivalent 2-pin nets;
$R_t$; = $\varnothing$;
for every $e = (u, v) \in E$ and $e$ is not visited
    $R_t = R_t \cup$ find_ring($G$, $e$);
reverse-order simulation for rings in $R_t$.

function find_ring($G$, $e$)
Let $e = (u, v)$ and $v$ is an input pin in core $C$;
if $v$ is a pin in the starting core
    return the ring and mark all nets as visited;
for every output pin $w$ in $C$
    if there is an unvisited edge $(w, x)$
        find_ring($G$, $(w, x)$);
    else if there is an untried output net $(w, x)$
        find_ring($G$, $(w, x)$);
    else
        return $\varnothing$;
end function
```

Figure 5.8 The ring generation for interconnect fault detection algorithm (IORT).

5.4.3 Interconnect Oscillation Ring Construction for Fault Diagnosis (IORD)

Our goal for the interconnect diagnosis is to find a small set of rings R_d that can uniquely identify the faulty *edge* or *net segment* if it exists. The set R_d is obtained by augmenting R_t as follows. We first apply the diagnosability checking techniques discussed in Section 5.3 to R_t to find out the net segments that are not diagnosable. For an edge *e* that is not single fault diagnosable, we try to find a new ring passing it without going through the edges that are indistinguishable to *e*. If such a ring exists, it will be included in R_d. The diagnosability checking should be conducted for each added ring so that other edges that become diagnosable with the new ring will be found.

In this diagnosis algorithm, we can achieve the highest diagnosis resolution when every net segment is diagnosable under R_d. With the reduced diagnosis resolution, the number of diagnosis rings can be reduced accordingly. Thus, this algorithm can be adjusted to the required diagnosis resolution to reduce the number of diagnostic rings. The algorithm for the generation of diagnosis rings is given below.

Algorithm: IORD (Interconnect Oscillation Ring Generation for Fault Diagnosis)

Input: A hypergraph $H = (V, L)$ representing a circuit

Output: A set of rings R_d

```
Transform hypergraph $H$ into a new graph $G = (V', E)$ with equivalent 2-pin net segments;
Generate a set of rings $R_t$ for fault detection;
$R_d = R_t$;
Conduct diagnosability check;
for every $e \in E$ {
    if ($e$ is not single-fault diagnosable)
        Find a ring $r$ to make $e$ diagnosable;
    $R_d = R_d \cup \{r\}$;
    Modify the diagnosability of all edges in $E$;
}
return $R_d$;
```

Figure 5.9 The ring generation for interconnect fault diagnosis algorithm.

The flowchart illustrating the process of diagnosis ring generation is given in Figure 5.10.

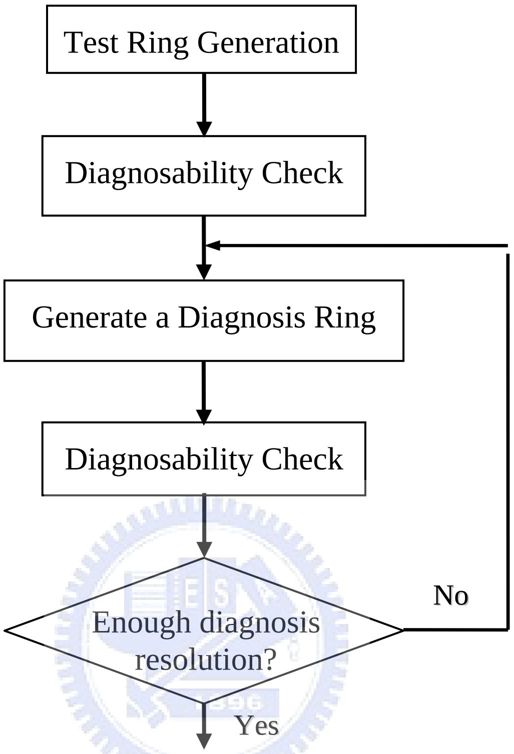


Figure 5.10 Diagnosis ring generation procedure.

5.5 Optimization Techniques for Interconnect Diagnosis

In this section, we discuss two techniques for test time minimization for the IORD scheme.

5.5.1 Concurrent Test

Multiple oscillation rings can be applied simultaneously as long as they do not interfere with each other. Two rings can not be applied concurrently if they share some net segment, or they go through the same scan path in a core. The condition is illustrated in Figure 5.11 for scan path conflict. Assume that two rings with no common net segments pass the same core. The first ring contains edges e_1 and e_3, while the second ring includes edges e_2 and e_4. Although these two rings do not share common net segments, they can not be applied at the same time due to the same scan path they go through.

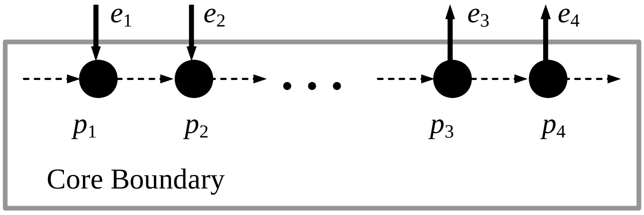
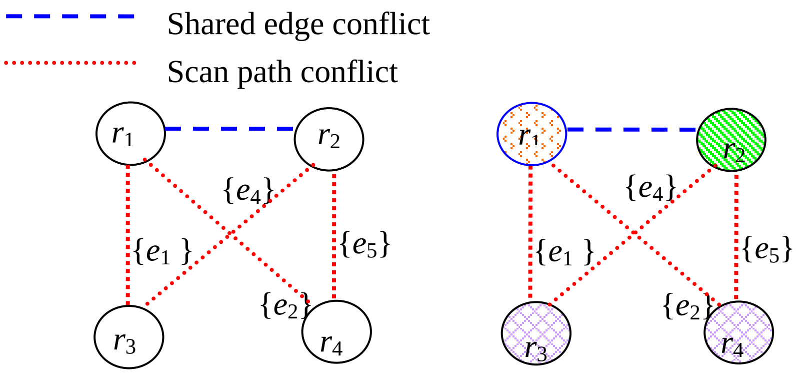


Figure 5.11 Scan chain constraint.

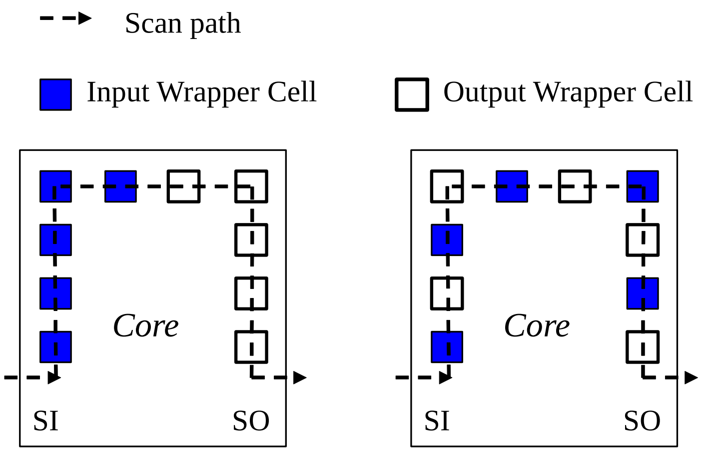
In order to achieve the maximum concurrency or parallelism test, we model all the constraints by a *conflict graph* [44] as shown in Figure 5.12(a). Each ring is represented by a node, and two nodes are connected by an edge if they interfere with each other for a scan-path constraint in Figure 5.11 or a common-edge constraint. The problem of finding the maximum concurrency tests can thus be reduced to the well-known *graph coloring problem* [44], as shown in Figure 5.12(b). Those rings/nodes colored with the same color can be tested concurrently, e.g., r_3 and r_4 in Figure 5.12(b), and we need at least three colors for the four nodes of Figure 5.12. The coloring problem in the general graph has known to be NP-complete. (Nevertheless, in our experiment, we focus on the interconnect tree structure, for which the coloring problem can be solved in polynomial time [44].)



(a) (b)

Figure 5.12 (a) Conflict Graph (b) Graph coloring.

One possible way to reduce the number of “mutually exclusive” rings (i.e., rings that cannot be tested concurrently) is to reorder the pin positions. Consider the core illustrated in Figure 5.13(a). The five nets connecting to the five input wrapper cells belong to different rings, and none of the rings can be tested at the same time due to the shared scan path constraint. However, if we reorder the pin positions such that input cells and output cells appear alternately, at most five rings can be formed simultaneously, with each ring passing two adjacent pins, as shown in Figure 5.13(b).



(a) (b)

Figure 5.13 Pin reordering for interleaving configuration.

5.5.2 Adaptive Diagnosis

The number of test patterns can be greatly reduced whenever adaptive diagnosis is possible. In the adaptive diagnosis, a test pattern is selected according to the result of previous tests. An adaptive diagnosis tree, typically a binary tree, can be constructed according to the test patterns. For example, the adaptive diagnosis tree for the diagnosis example given in Figures 5.10 and 5.11 is illustrated in Figure 5.14.

For an n-net system, initially there are n+1 possible diagnosis results, namely fault-free ($\varnothing$) and a single fault on net e_i (f_{ei}) for $1 \le i \le n$. Each node in the tree represents a test pattern (ring), and the test outcome can be either pass (P) or fail (F). According to the test outcome of applying a ring, the indistinguishable set of edges can be divided into two groups. If the tree is balanced, the minimum number of diagnosis patterns required is $\lceil \log_2(n+1) \rceil$.

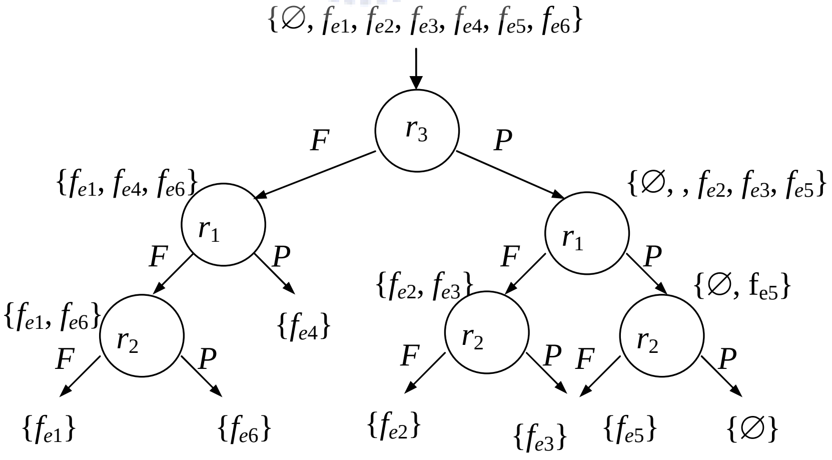


Figure 5.14 An adaptive diagnosis tree.

In order to construct a balanced adaptive diagnosis tree, in each internal tree node we need to select the test pattern (i.e. test ring) that evenly partitions the possible outcomes into two groups: Fail (F) and Pass (P). For example, in Figure 5.14, we choose the test pattern r_3 as the first test, since it evenly partitions the six possible outcomes into Fail (f_{e1}, f_{e4}, f_{e6}) and Pass ($\varnothing$,f_{e2}, f_{e3}, f_{e5}). It can be seen that, in Figure 5.14, each test partitions possible outcomes into two groups whose cardinalities differ by at most 1 in each level.

The upper bound on the number of adaptive diagnosis test sessions needed in our method can be computed as follows. Let the number of test rings (without diagnosis) be $|R_t|$, and the length of the longest test ring be L_h. In the worst case, we need to apply $|R_t|$ rings to find out that there is a faulty net, and the last ring contains L_h net segments that are all passed by the ring only. It takes up to L_h–1 rings to distinguish these L_h possible faults, and thus the maximum number of diagnosis rings is $|R_t|+ L_h-1$.

5.6 Experimental Results

There are two parts in this experiment. The first part covers the subsections of the above-mentioned two optimization methods. First of all, we compare the experimental results for interconnect diagnosis between Predetermined and Adaptive methods. Second, we compare Predetermined and Concurrent methods. Then, the second part covers the comparison between theoretical bounds and experimental results.

5.6.1 Comparison between Predetermined and Adaptive Methods

We tested the diagnosis algorithm based on six commonly used MCNC benchmark circuits. The results are listed in Table 5.1, where the first column gives the circuit names, and the next four columns give the circuit statistics ("Statistics"), including the number of cores (#core), the number of pads (#pads), the number of hypernets (#hyp), and the number of net segments (#net_seg). The 5th column, #net_seg, lists the number of net segments, modeled as shown in Figure 5.2(b), to be diagnosed in each benchmark. The next three columns ("Predetermined") give the experimental results for predetermined diagnosis, including the number of rings required to detect all *2-pin nets* ($|R_t|$) and to diagnose all single faults ($|R_d|$). In each benchmark, all *net segments* are single-fault diagnosable. The last column, $|R_d|/|R_t|$, gives the ratio of rings from 1.25X to 2.81X for the maximum diagnosis vs. rings for fault detection. This ratio means that we need extra test time of 1.25X to 2.81X to diagnose the single fault in each net segment under the predetermined diagnosis method, compared to the IORT scheme.

In each case, we also give the estimated testing time (given in parenthesis), obtained by assuming only 4 MHz measuring period as discussed in Section 5.2.1 to estimate the *longest* test application time for each ring. The time needed to set up the rings should be roughly proportional to the testing time.

Since in these benchmark circuits the net directions are not given, we make the following assumptions in our experiment: (1) All cores are listed in a given order. For a hypernet, the pin corresponding to the first core in the core list is assumed the source, while the others are sinks. (2) Since the order on internal scan paths is not known, we conservatively assume that all output pins are placed in consecutive positions, and thus each ring may pass any core only once. (3) All I/O pads are connected through the boundary scan path, while the positions of the pads are unknown, and thus the boundary scan path appears only once in each ring. Under

Assumption (2), each ring may pass any core only once, corresponding to the worst-case scan-path constraint that makes concurrent test impossible. Thus, no concurrent tests are assumed in Table 5.1. Assumption (3) makes the boundary scan path appear only once in each ring due to unknown pad positions. In summary, the above three assumptions give the worst-case scenario. The results are an upper bound on both test and diagnosis rings, and the actual number of rings should be smaller.

The next four columns ("analysis") give the diagnosis related information after applying R_t rings. The column #OneRing gives the number of nets passed by only one ring. It can be seen that most nets are passed by one ring only when compared with the number of net segments (#net_segment). Since the purpose of R_t is to detect faults with the minimum number of rings, it is not surprising that most nets are passed by one ring only. Most nets that are not diagnosable at this stage fall into this category. Columns "#NoDiag" and "#EquClass" give the number of nets that are <u>no</u>t <u>diag</u>nosable and the number of <u>equ</u>ivalence <u>class</u>es after applying R_t, respectively, and they are the targets for further diagnosis. Two faults are in the same equivalence class if their syndromes for the tests are identical. The last column in this group ("$|R_d|–|R_t|$") gives the number of extra diagnosis rings required in each case to make all nets single-fault diagnosable. Note that, in an equivalence class of size s, we need no more than s–1 extra rings to distinguish these s nets. Assume that there are m equivalence classes whose sizes are s_1, s_2, …, s_m, respectively. The upper bound on the number of additional diagnosis rings "$|R_d|–|R_t|$" can be expressed as follows:

$$\sum_{i=1}^{m}(S_i - 1) = \sum_{i=1}^{m} S_i - m = \#NoDiag - \#EquClass \qquad (5.3)$$

The last three columns ("adaptive") compare the number of rings required in both predetermined and adaptive diagnosis. The number of rings in a predetermined diagnosis is $|R_d|$. After applying R_t rings, the size of the largest equivalence class for each benchmark is given in the column “max. EC”. In the worst case, the adaptive diagnosis needs to apply $|R_t|$ rings, and then (max. EC)–1 rings for diagnosis. The number of the worst-case adaptive diagnosis rings is given in column “$|R_a|$”. The last column ($|R_d|/|R_a|$) shows the ratio of rings for the predetermined vs. adaptive diagnosis schemes. For the results shown in the column, the adaptive algorithm obtains 1.23X to 2.67X improvements over the predetermined diagnosis scheme. Also, from the normalized $|R_a|$ and $|R_t|$, the test time of adaptive diagnosis is approximately equal to that for detection alone, which further reveals the effectiveness of adaptive diagnosis.

In summary, the oscillation ring scheme can detect and diagnose delay faults and crosstalk glitches very efficiently and effectively. In conventional schemes, the detection of crosstalk-induced glitches usually involves precise measurement of signals on the victim nets, for which complex clock control is needed for the delay fault detection due to the two-pattern tests. Therefore, more areas have to be devoted to the detection of errors due to these problems. In contrast, our scheme only slightly modifies IEEE Std. 1500 wrapper cells, and the area overhead is small as shown in Section 5.2.1. Further, by applying the adaptive diagnosis technique, the time needed for diagnosis is approximately equal to that of detection alone. In other words, diagnosis can be accomplished with very small extra cost.

5.6.2 Comparison between Predetermined and Concurrent Methods

The experimental results for the concurrent test are given in Table 5.2. The 3rd column ($|R_c|$) lists the number of test sessions after applying the concurrency test.

When a set of rings are applied concurrently, we refer to these rings as a test session. The 4th column ($|R_d|–|R_c|$) gives the percentage of improvements in the number of test sessions based on the worst-case scenario of the interconnect structure. We note that the improvement can be even better for general interconnect structures. The reduction in test time due to the concurrent test ranges from 0.27% to 9.66% *with no hardware overhead*. Notice that the numbers give the lower bounds of empirical improvements by using the concurrency optimization technique.

The lack of concurrency is mainly a structure issue; however, it can also be attributed to several reasons. First, in the ring generation algorithm, we try to generate long rings so that the number of rings can be reduced. The longer rings tend to conflict with each other, and thus they cannot be applied concurrently. Second, since we do not know the pin order in any core, we conservatively assume that each core can be passed by only one ring in a test session in order to avoid scan chain conflict. This may lead to an over pessimistic estimation on the scan path constraints and the number of test sessions. Third, the ring generation algorithm might not be perfect. The nets are searched according to their ordering in the data structure, and thus some net segments are used more often than others, reducing the possibility of the concurrent test. Our future work should handle this problem.

5.6.3 Comparison between Theoretical Bounds and Experimental Results

Also in Table 5.3, the upper bound on the required number of extra rings ($|R_d|–|R_t|$) is "(#NoDiag)–(#EquClass)", and it can be seen that these two numbers "(#NoDiag)–(#EquClass)" and "$|R_d|–|R_t|$" are pretty close in all cases. Specifically, the empirical results "$|R_d|–|R_t|$" differs from the theoretical results

“(#NoDiag)–(#EquClass)” given in Equation (5.3) by small differences of only up to 6.64%.

Table 5.1 Experimental results for Interconnect Diagnosis both for Predetermined and Adaptive Methods.

Circuit	Statistics				Predetermined			Analysis				Adaptive		
	#core	#pad	#hyp	#net_ seg.	$\|R_t\|$	$\|R_d\|$	$\|R_d\|/\|R_t\|$	#One Ring	#No Diag	#Equ Class	$\|R_d\|–\|R_t\|$	max. EC	$\|R_a\|$	$\|R_d\|/\|R_a\|$
ac3	27	75	211	416	133 (33.3ms)	374 (93.5ms)	2.81	389	323	68	241	8	140 (35ms)	2.67
ami33	33	42	117	343	242 (60.5ms)	303 (75.8ms)	1.25	309	126	59	61	5	246 (61.5ms)	1.23
ami49	49	22	361	475	156 (39ms)	386 (96.5ms)	2.47	406	337	88	230	9	162 (40.5ms)	2.38
apte	9	73	92	136	73 (18.3ms)	122 (30.5ms)	1.67	127	94	40	49	4	76 (19ms)	1.61
hp	11	45	72	195	81 (20.3ms)	164 (41ms)	2.02	176	145	51	82	7	87 (21.8ms)	1.89
xerox	10	2	161	356	218 (54.5ms)	342 (85.5ms)	1.57	346	214	86	124	5	222 (55.5ms)	1.54
Comp.					0.9679								1	

Table 5.2 Concurrent Test Sessions.

Circuit	$\|R_d\|$	$\|R_c\|$ (worst case)	$\|R_d\|$-$\|R_c\|$
ac3	374	373	1 (0.27%)
ami33	303	290	17 (5.86%)
ami49	386	352	34 (9.66%)
apte	122	119	3 (2.52%)
hp	164	160	4 (2.50%)
xerox	342	327	15 (4.59%)
Comparison	1.0432	1	4.57%

Table 5.3 Comparison between Theoretical Bounds and Experimental Results.

Circuit	#NoDiag	#Equ Class	Equation (3) (#NoDiag-#Equ Class)	Extra Rings $(\lvert R_d\rvert - \lvert R_t\rvert)$	(#NoDiag-#EquClass) and $(\lvert R_d\rvert - \lvert R_t\rvert)$
ac3	323	68	255	241	14 (5.49%)
ami33	126	59	67	61	6 (8.96%)
ami49	337	88	249	230	19 (7.63%)
apte	94	40	50	49	1 (2.00%)
hp	145	51	94	82	12 (12.77%)
xerox	214	86	128	124	4 (3.13%)
Comparison			1.0712	1	6.64%

Chapter 6

Oscillation Ring Test for Synchronous Sequential Circuits

In this chapter, we propose an *oscillation-based* test methodology for sequential testing. This approach provides many advantages over traditional methods. (1) It is *at-speed* testing, which makes *delay-inducing defects* detectable. (2) The ATPG is much easier, and the *test set* is usually smaller. (3) There is no need to store output responses, which greatly reduces the communication bandwidth between the Automatic Test Equipment (ATE) and Circuit under Test (CUT). We provide a register design that supports the oscillation test, and give an effective algorithm for oscillation test generation. Experimental results on LGSyn91 and ISCAS89 benchmarks show that the proposed test method achieves high fault coverage with a smaller number of test vectors.

6.1 Introduction

Decreasing feature sizes and increasing clock speeds have combined to alter the defect effects dramatically. Recent evidence indicates that delay defects can no longer be ignored nor go untested [18], [66], [85], [90], [94], [113]. For circuits designed with 130nm or more advanced technologies, the transition fault is considered essential to achieve the acceptable defect level. The detection of delay faults requires at-speed test techniques, which creates signal transitions to be captured at normal speed. In the past, it was typically accomplished with functional patterns, but it was undesirable mainly due to the cost consideration. Scan-based test techniques [68], [112] offer a viable alternative for at-speed testing. However, there are many complicating factors when moving from relatively slow scan-based tests for stuck-at faults to test delay faults. Design methodologies such as multiple clock domains, mixed negative and positive edge clocking, and so on, all pose challenges to the implementation of successful, high coverage delay tests. The costs associated with such design methodologies are also ever increasingly important issues. Thus, it is desirable to have a cost-effective test methodology for at-speed test, and it is the motivation of this chapter.

The sequential testing is difficult due to the lack of controllability and observability in internal storage elements. Sequential ATPG is much more complicated [32], [96], [118], and it may not be able to achieve high fault coverage with sequential testing. Many design-for-testability methods for sequential circuits have been proposed [16], [38], [41], [121] in which the scan-based designs are the most popular. However, scan tests are usually carried out in lower speed, and thus it is mainly targeted for stuck-at faults.

We propose an *oscillation-based* test methodology for sequential testing in this chapter. An oscillation ring is a closed loop with an odd number of signal inversions. If the CUT is fault-free, an oscillation signal will appear on the ring. Otherwise, the CUT is deemed faulty.

This approach provides three major advantages over traditional scan-based approaches. (1) In this architecture, testing is conducted at-speed, which makes delay-inducing defects detectable. This is due to that the oscillation test is triggered by system clock and thus operates at normal speed. (2) Test vectors can be derived directly from the finite-state machine (FSM) model in our Oscillation Test Pattern Generation (OTPG) algorithm, and it greatly simplifies the ATPG process accordingly. (3) Our method does not need complex test clocks, which is required for two-pattern tests used in transitional delay tests. (4) The correctness of CUT is determined by simply observing whether there are oscillation signals in the outputs, and there is no need to store and analyze output responses. Besides, the number of vectors is roughly the same as scan tests. Thus, the communication bandwidth between the ATE and CUT is greatly reduced, which partly solves the problem of test data compression in SOC testing.

Oscillation based test is an efficient and effective method to detect faults in a circuit or a device [6], [58]. There are many works on oscillation tests. For example, Kaneko and Sakaguchi proposed an oscillation fault diagnosis method for analog circuits based on boundary search with perturbation model [58]. Arabi and Kaminska proposed an oscillation-based test strategy for analog and mixed-signal integrated circuits [6]. Recently, oscillation ring test is applied for system-level interconnects for delay faults and crosstalk glitch faults [77], [78]. The proposed oscillation test methodology attacks the testing problem from a different perspective. It modifies the storage elements such that oscillation signals can be generates according to the functional specifications of a given circuit.

In order to conduct the oscillation test, the state-holding elements must be modified to generate oscillation signals in test mode. In this chapter, we develop a *Modified State Register* (MSR) cell for this purpose, and give an algorithm to generate tests with the help of MSR cells. The proposed MSR design required extra silicon area. However, in deep submicron designs, silicon area is no longer the major issue. Other issues, including *delay fault* and *soft fault testability*, *low-power testing*, and etc., become the more important concerns. For example, Intel proposes a scan-cell design targeted for soft faults [91]. This cell-level design uses 1.08X-1.24X area with power overhead of 2.02X-2.26X, while chip-level design suffers from power overhead by 4.0X-5.0X [91]. The proposed MSR cell can be combined with other register designs to achieve highly testable and reliable systems.

Experimental results on LGSyn91 and ISCAS’89 benchmark circuits show that the proposed oscillation test method achieves high fault coverage and with smaller number of test vectors.

The remaining sections are organized as follows. In Section 6.2, we introduce the proposed Oscillation Test Architecture and MSR cell design for both asynchronous and synchronous circuits. Section 6.3 gives Oscillation Test Pattern Generation (OTPG) algorithm. Experimental results are show in Section 6.4, and some brief conclusions are in Section 6.5.

6.2 Oscillation Test for Sequential Circuits

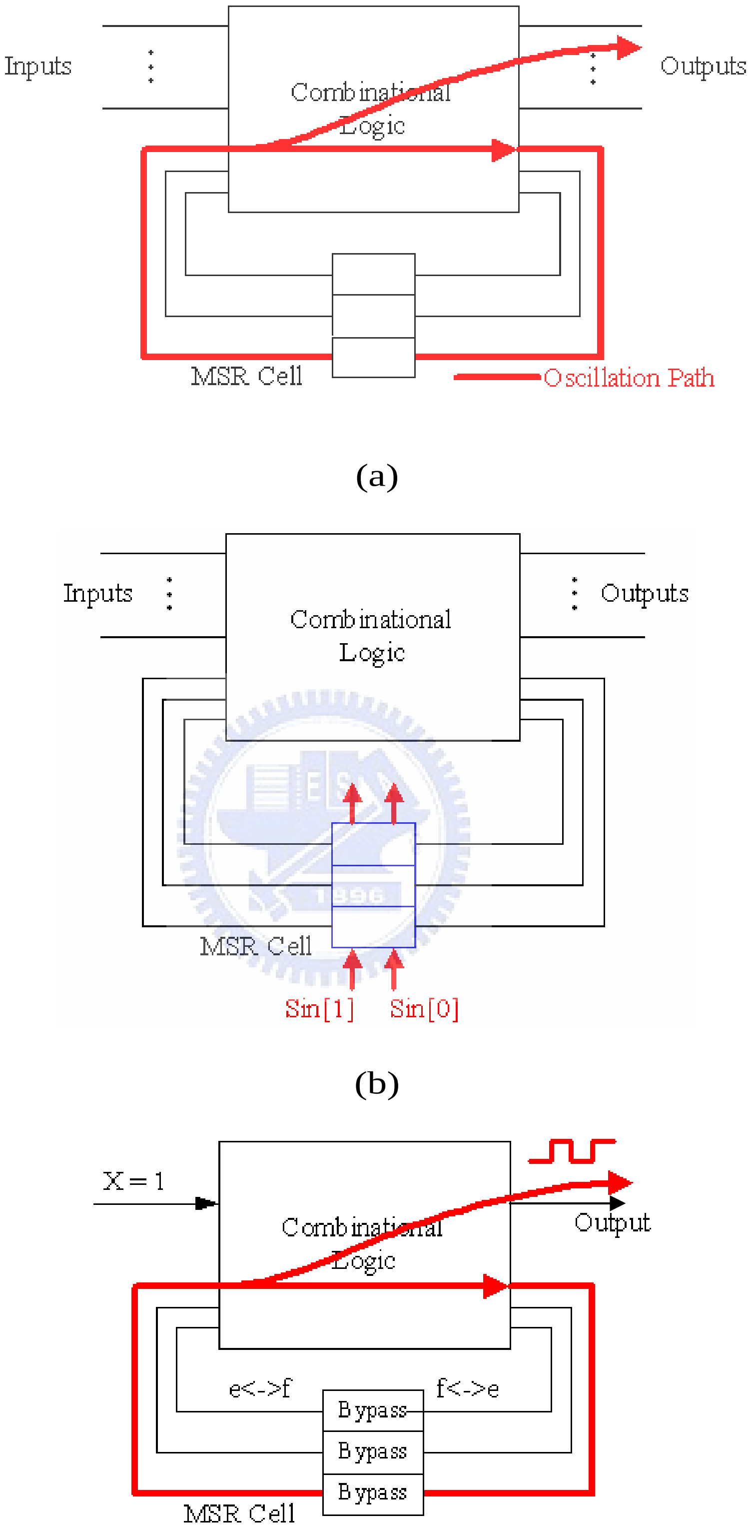
6.2.1 Oscillation Ring Test Architecture

The oscillation ring test architecture for sequential circuits is shown in Figure 6.1. In this architecture, we replace the flip-flops by MSR cells. In the normal mode operation, the MSR cells work as state-holding elements. In the oscillation test mode, MSR cells transform the target sequential circuits into asynchronous circuits with odd-inversion feedback paths, and oscillation signals show on these loops (rings) accordingly.

With appropriate inputs, oscillation signals can be propagated to at least one primary output through some sensitized paths in Figure 6.1(a). Stuck-at faults on wires passed by oscillation signals will stop these oscillation signals; while delay (transition) faults will change the oscillation frequency. The faults are detected by observing the oscillation signals in the primary outputs.

In order to construct oscillation rings in sequential circuits, we need to set up appropriate connections in MSR cells. Figure 6.1(b) shows how to set the states in MSR cells. The control signals for each MSR cell are fed to the cell through the scan paths.

It is usually difficult to implement the asynchronous test architecture. Whenever there are multiple oscillation signals, there is always a race problem. To solve this problem, we may use the system clock to control the feedback paths, which makes the design synchronous, as shown in Figure 6.1(c). The circuit is forced to move between states *e* and *f*, whose outputs are 0 and 1, respectively, when the input *X* is held at *X*=1. As a result, we can see that the output changes every cycle. Whenever there are faults, either stuck-at or delay faults, on the signal paths, the corresponding outputs will stop oscillating.



(a)

(b)

(c)

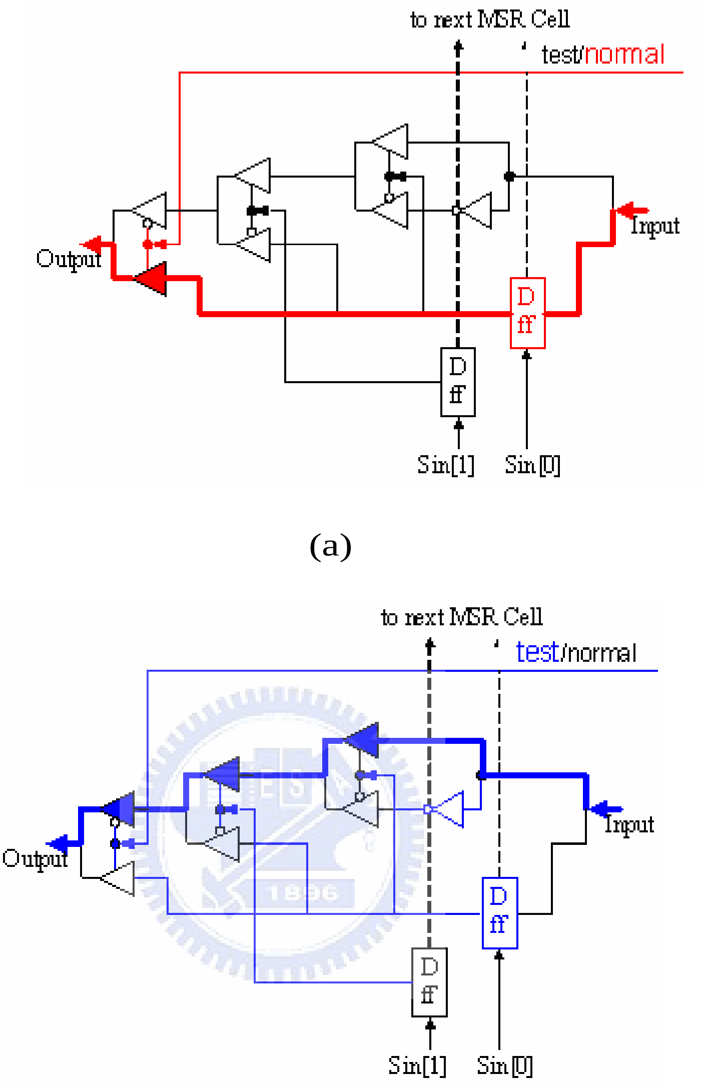
Figure 6.1 Oscillation test architecture for sequential circuits: (a) Oscillation Rings; (b) MSR states are controlled through scans, and (c) Oscillation Test is controlled by system clock.

6.2.2 Modified State Register (MSR) Design

6.2.2.1 MSR Design for Asynchronous Circuits

The MSR cell design for asynchronous oscillation test is shown in Figure 6.2, and the control states for the MSR cells are shown in Figure 6.3.

Under normal operation, the D-Type Flip Flops (DFFs) connected to Sin[0] are used as state-holding elements. In the oscillation test mode, an MSR cells operate in four states: hold 0, hold 1, INV and bypass. Hold 0 and 1 provide steady output values of 0 and 1, respectively. INV and bypass are used to set up odd-inversion loops to generate oscillation signal. A loop (ring) consists of two paths: one forward path in the combinational circuit, and a feedback path passing an MSR cell. If the number of signal inversion in the forward path is odd, the MSR cell is set to the “bypass” state; otherwise, it is set to the “INV” state.



(a)

(b)

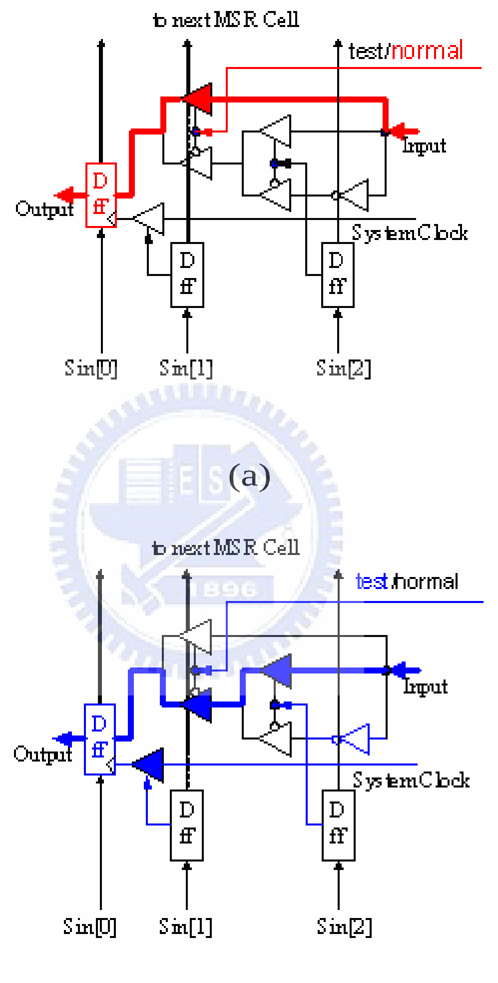
Figure 6.2 MSR cell (a) normal mode, and (b) oscillation test mode.

$S_{in}[1]$	$S_{in}[0]$	Operation
0	0	Hold 0
0	1	Hold 1
1	0	INV
1	1	Bypass

Figure 6.3 Control state table of an MSR cell for asynchronous sequential circuit test.

6.2.2.2 MSR Design for Synchronous Sequential Circuits

In order to avoid the race conditions caused by the asynchronous test in Figure 6.1(c), we use system clock to sample the oscillation signals, as shown in Figure 6.4.



(a)

(b)

Figure 6.4 MSR cell for synchronous oscillation test: (a) normal mode, (b) oscillation test mode.

An MSR cell design and its operations for the synchronous oscillation test are illustrated in Figure 6.4, and the control states for MSR cells are given in Figure 6.5.

$S_{in}[2]$	$S_{in}[1]$	$S_{in}[0]$	Operation
-	0	0	Hold 0
-	0	1	Hold 1
0	1	-	INV
1	1	-	Bypass

Figure 6.5 Control state table of MSR Cell for synchronous sequential circuits test.

6.3 Synchronous Oscillation Ring Test

The synchronous oscillation ring test is preferred for several reasons. The most important advantage is that it avoids race problems, which is very difficult to handle in the asynchronous approach. Secondly, it also simplifies the ATPG process. The test patterns can be obtained directly from the FSM model. The drawback of this approach is that an MSR cell is significant larger. This large hardware overhead can be partly offset if we can restrict the number of operations required in an MSR cell, and this can be achieved through state assignment for the given FSM. In the remaining part of the paper, we shall concentrate on the synchronous oscillation test.

6.3.1 Constructing Oscillation Signals from FSM

An example on how to find test patterns from an FSM model is given in Figure 6.6, which shows the state transition and output table of an FSM. The output table gives the candidates for oscillating outputs. For example, when the primary input X is held at 1 (X=1) and the FSM is in either state e or f, the FSM moves back and forth between these two states. Since the outputs corresponding to states e and f with X=1 are 0 and 1, respectively, we shall see oscillating signals at the output. This oscillation condition is shown in Figure 6.1(c), in which the least significant bit (LSB) of the state vector is an oscillating signal.

In the above example, an oscillation signal is generated without using MSR cells. This is achievable when both the next states and outputs of a state pair are alternating. Unfortunately, no other state pairs satisfy the oscillation condition. We shall use the MSR to force state pairs to alternating if only their corresponding outputs are different. All the candidate state pairs are listed below. With X = 0, the following state pairs generate the opposite output values: (*a*, *d*), (*a*, *e*), (*b*, *d*), (*b*, *e*), (*c*, *d*), (*c*, *e*), (*f*, *d*), (*f*, *e*). With X=1, the possible choices include: (*a*, *c*), (*a*, *d*), (*a*, *f*), (*b*, *c*), (*b*, *d*), (*b*, *f*), (*e*, *c*), (*e*, *d*), (*e*, *f*).

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a 000	a 000	c 010	1	0
b 001	d 011	b 001	1	0
c 010	f 101	d 011	1	1
d 011	c 010	a 000	0	1
e 100	e 100	f 101	0	0
f 101	b 001	e 100	1	1

Figure 6.6 State transition and output table of an FSM.

6.3.2 MSR State Transition Algorithm (MSR STA Algorithm)

In order to generate oscillation signals in the test mode, we need to change the next state functions for the selected state pair with the help of MSR cells. An example is shown in Figure 6.7, which modifies the state transition table of Figure 6.6 in the test mode to produce oscillation signals.

For example, since state pair (*b*, *e*) in Figure 6.6 produces different outputs when

X=0, it is a candidate for generating oscillation signals. To do this, in the test mode we need to change the next states of (*b*, *e*) from (*d*, *e*) to (*e*, *b*) when *X*=0, as indicated in Figure 6.7.

Present State	Next State		Output	
	X=0	X=1	X=0	X=1
a 000	a 001	c 010	1	0
b 001	d → e 011	b 001	1	0
c 010	f 101	d 011	1	1
d 011	c 010	a 000	0	1
e 100	e → b 100	f 101	0	0
f 101	b 001	e 100	1	1

Figure 6.7 Modified State Transition Table.

The modification of next state functions in the test mode can be achieved by setting MSR cells to appropriate states. We present an algorithm to select the MSR states in this section. Two tables are used in this algorithm: (1) Truth Table of State Bit Transition (Figure 6.8(a)), and (2) Operation Table of MSR Cell State (Figure 6.8(b)).

In Figure 6.8(a), Change of Bit shows the bit change between current state and next state. There are four operation definitions: (1) When both current and next states are “0”, the operation value is “Low”; (2) When both current and next states are changed from “0” to “1”, the operation value is “Rising”; (3) When both current and next states are changed from “1” to “0”, the operation value is “Falling”; (4) When both current and next states are “1”, the operation value is “High”.

In Figure 6.8(b), Operation Table of MSR Cell State defines the state transition relationship between normal next state in normal mode (i.e. operation value 1) and alternate next state in test mode (i.e. operation value 2). “Fail” state is not defined in MSR Cell due to conflicts between two operation values. Please note this table is symmetric due to binary commutative characteristic, and the inverse diagonal is full of “Fail” entries. As to how the operation entries are derived, we show in the following paragraphs.

Bit Transition	OP Value
0 -> 0	Low
0 -> 1	Rising
1 -> 0	Falling
1 -> 1	High

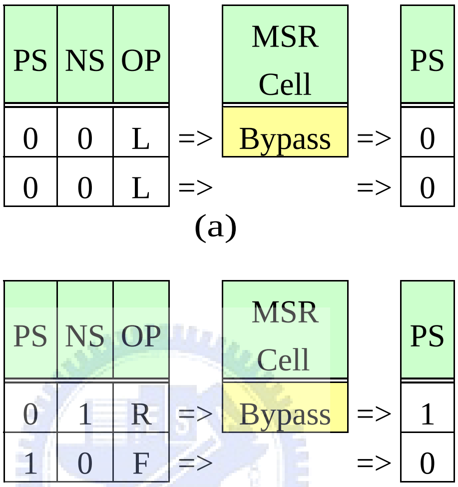
(a)

OP Value		2nd Operand			
		L	**H**	**R**	**F**
1st Operand	**L**	Bypass	INV	Hold 0	Fail
	H	INV	Bypass	Fail	Hold 1
	R	Hold 0	Fail	INV	Bypass
	F	Fail	Hold 1	Bypass	INV

(b)

Figure 6.8 (a) Truth Table of a state bit, (b) Operation Table of the MSR cell state.

In Figure 6.8(b), there are four types of operation definitions in MSR Operation Table. The first type in Figure 6.9 is the “Bypass” State in MSR Cell. It means that MSR Cell’s output is the same as its input. As shown in Figure 6.9(a), two state bits in Present State (PS) are “0”, and two state bits in Next State (NS) are also “0” in the alternate state pair. According to Truth Table of State Bit, both operation values are “Low”, which leads to “Bypass” state. The “Bypass” state satisfies the state transition condition that present and next states are the same. Another example for “Bypass” is in Figure 6.9(b). The difference between Figure 6.9(a) and 6.9(b) is that there is oscillation signals in Figure 6.9(b) since one PS bit is “0” while the other is ”1”. In summary, when two operation values are {L, L}, {H, H}, or {R, F}, the MSR Cell State is set to “Bypass”.



(b)

Figure 6.9 Operation values of (a) {L, L}, (b) {R, F}.

The second type in Figure 6.9 is the “INV” State in MSR Cell. It means that MSR Cell’s output is the complement of its input. As in Figure 6.9(c), two state bits in Present State (PS) are “0” and two state bits in Next State (NS) are “1” in the alternate state pair. To achieve this, the MSR Cell State must be “INV”. According to Truth Table of State Bit, both operation values are “Rising”. Another example for “INV” is in Figure 6.9(d). The difference between Figure 6.9(c) and 6.9(d) is that there is oscillation signals in Figure 6.9(d) since current bits (PS) are “0” and ”1” alternated. In summary, when two operation values are three types of {R, R}, {F, F} or {H, L}, the MSR Cell State is set to “INV”.

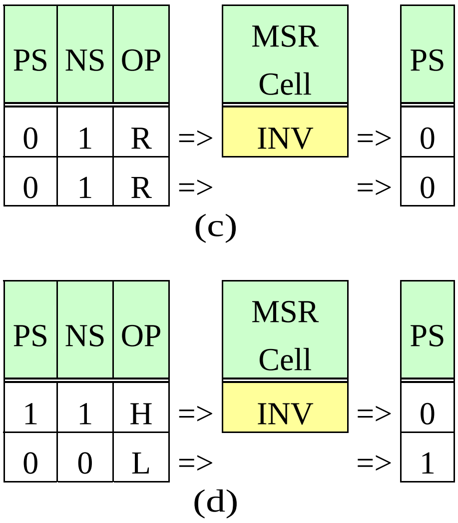


Figure 6.9 Operation values of (c) {R, R}, (d) {H, L}.

The third type in Figure 6.9 is the “Hold” State (either “Hold 0” or “Hold 1”). As in Figure 6.9(e), two state bits in Present State (PS) are static “0” while two state bits in Next State (NS) are “1” and “0”. This requires MSR Cell State in “Hold 0”. Figure 6.9(f) is for “Hold 1” since current states have two static “1” and next states are “1” and “0”. In summary, when two operation values are {R, L}, MSR Cell State is set to “Hold 0”; and {F, H} corresponds to State “Hold 1”.

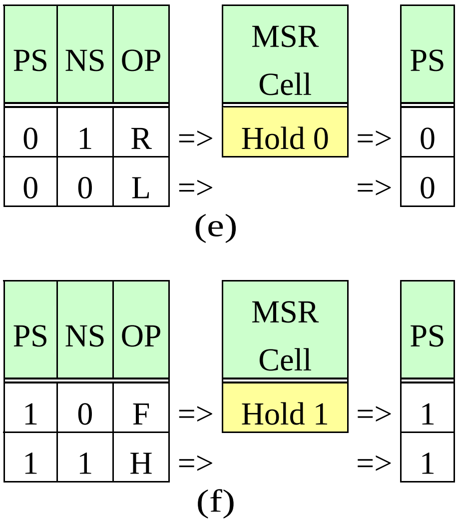
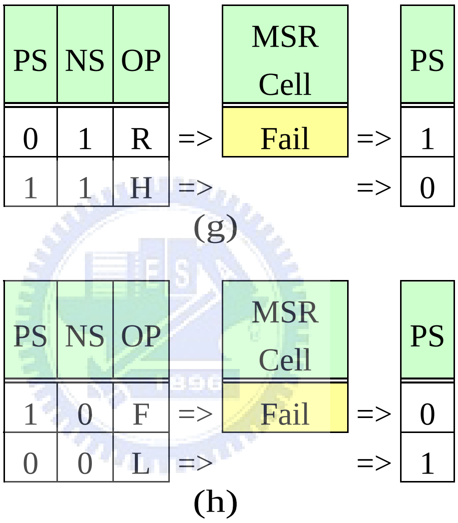


Figure 6.9 Operation values of (e) {R, L}, (f) {F, H}.

The forth type in Figure 6.9 is the “Fail” State, which means that MSR Cell can not satisfy given circuit conditions. As in Figure 6.9(g), the two Present States are “1” and ”0” and two Next State bits are static “1”, which is not possible. Another example for “Fail” is in Figure 6.9(h). In summary, when two operation values are types of {R, H} or {F, L}, the MSR Cell State is set to “Fail”.



(g)

(h)

Figure 6.9 Operation values of (g) {R, H}, (h) {F, L}.

6.3.3 Test Pattern Generation Algorithm for Oscillation Test (OTPG Algorithm)

The test generation algorithm is outlined below. The input of the algorithm is an FSM model. Each state transition is a four-tupple (x, p, n, y), which represents input vector, present state, next state, and output vector, respectively. Let the distance between two vectors $d(v_1, v_2)$ be the number of bit differences where the two vectors are different (i.e., one is 0 and the other is 1). For example, $d(-00, 11-) = 1$, where – indicates a don’t-care bit. This $d(v_1, v_2)$ is also known as Hamming distance.

For example, consider state pair (a, e) under X=0 in Figure 6.6. After the MSR State Transition Algorithm, we get the MSR Cell State [2..0]=[INV, Bypass, Bypass] in Figure 6.10.

```
Algorithm: Oscillation Test Pattern Generation (OTPG)
Input: a set of state transition function $T$
Output: a set of test vectors
for each ($t_i$, $t_j \in T$)
        if ($d(y_i, y_j) > 0$ && $d(x_i, x_j) > 0$) {
        $x \leftarrow x_i \cap x_j$;
        calculate MSR states from the Operation Table;
        if (no "Fail" state)
        record valid state pairs $p_i$, $p_j$ with input $x$;
}
```

Figure 6.10 Oscillation Test Pattern Generation (OTPG) Algorithm.

		state	$b_2b_1b_0$	OP value		
1st	Present state	*a*	0 0 0	L	L	L
	Next state	*a*	0 0 0			
2nd	Present state	*e*	1 0 0	H	L	L
	Next state	*e*	1 0 0			
				INV	Bypass	Bypass

Figure 6.11 MSR cell state for state pair (a, e).

6.4 Experimental Results

We have conducted experiments on LGSyn91 of MCNC benchmark circuits whose statistics are shown in Table 6.1. In order to make the proposed method effective, we should have enough oscillation signals in the outputs. Therefore, circuits with very few outputs and output signal transitions are not included in the experiment.

In our experiment, the symbolic states of the first 16 LGSynth91 benchmark circuits are assigned by NOVA [115]. In the remaining 5 ISCAS’89 circuits, the binary codes of the states are known. The proposed oscillation TPG is called to generate oscillation test vectors. The FSMs are then synthesized and the test vectors are evaluated for stuck-at test efficiency. The results are shown in Table 6.2. Columns 2 to 4 give the results of the proposed method. The column under #*t (osc)* indicates the number of oscillation tests generated by our algorithm, while *TE* is the test efficiency achieved by this set of tests. The forth column (#*t* (*scan*)) gives the number of extra scan test vectors required to achieve 100% test efficiency. The last column (#stv) indicates the number of test vectors to achieve 100% test efficiency if only scan test is used. The proposed oscillation test (#*t (osc)*) achieves average test efficiency of 90.11%, and 100% test efficiency can be obtained by adding extra scan test (#*t* (*scan*)). Under the same 100% test efficiency, the proposed method (#*t* (*osc*)+#*t* (*scan*)) requires almost same total/average numbers of test patterns compared to the pure scan test. The proposed method requires approximately the same amount of test vectors as traditional scan tests to achieve 100% test efficiency for stuck-at faults, and it outperforms the pure scan tests at the average ratio of 2:1 in the test cases.

The proposed method has three major advantages over the scan test. (1) It enables at-speed test, since oscillation test is triggered by system clock and thus operates at normal speed. (2) Faults are detected if outputs fail to oscillate, thus it is not necessary to store and analyze output response. Thus, the communication bandwidth between the automatic test equipment (ATE) and CUT is greatly reduced, which partly solves the problem of test data compression in SOC testing. (3) Our method does not need complex test clocks, which is required for two-pattern tests used in transitional delay tests. Test vectors can be derived directly from the finite-state machine (FSM) model in our OTPG algorithm, and it greatly simplifies the ATPG process accordingly.

Table 6.1. Statistics of benchmark circuits

Circuit	#input	#output	#states
bbsse	7	7	16
cse	7	7	16
dk14	3	5	7
dk15	3	5	4
dk16	2	3	27
dk17	2	3	8
dk27	1	2	7
dk512	1	3	15
lion	2	1	4
mc	3	5	4
planet	7	19	48
s1	8	6	20
sand	11	9	32
sse	7	7	16
styr	9	10	30
tbk	6	3	32
s27	4	1	6
s298	3	6	218
s386	7	7	13
s1488	8	19	48
s1494	8	19	48

Table 6.2. Experimental comparison between our proposed oscillation test generation (OTPG) and pure scan methods.

Circuit	**Proposed**			***#stv* (scan only)**
	#t (osc)	*TE* (%)	*#t(scan)*	
bbsse	28	83.87	21	52
cse	50	87.35	23	76
dk14	20	96.09	6	36
dk15	5	60.61	15	24
dk16	60	98.66	5	65
dk17	15	98.15	1	21
dk27	5	90.74	2	11
dk512	17	98.31	1	24
lion	4	90.00	3	8
mc	7	100.00	0	10
planet	120	97.08	20	128
s1	88	92.48	20	105
sand	152	99.60	3	140
sse	28	83.87	21	52
styr	154	98.35	9	157
tbk	87	57.81	119	189
s27	6	97.37	1	11
s298	42	98.87	7	30
s386	26	75.12	21	42
s1488	115	95.34	17	135
s1494	116	92.58	27	154
Average		**90.11**		

Chapter 7

Multilevel Full-Chip Routing with Testability and Yield Enhancement

We propose in this chapter a multilevel full-chip routing algorithm that improves testability and diagnosability, manufacturability, and signal integrity for yield enhancement. Two major issues are addressed. (1) The oscillation ring (OR) test and its diagnosis scheme for interconnect based on the popular IEEE Std. 1500 are integrated into the multilevel routing framework to achieve testability enhancement. We augment the traditional multilevel framework of coarsening followed by uncoarsening by introducing a preprocessing stage that analyzes the oscillation ring structure for better resource estimation before the coarsening stage, and a final stage after uncoarsening that improves testability to achieve 100% interconnect fault coverage and maximal diagnosability. (2) We present a heuristic to balance routing congestion to optimize the multiple-fault probability, chemical mechanic polishing (CMP) and optical proximity correction (OPC) induced manufacturability, and crosstalk effects, for yield improvement. Experimental results on the MCNC benchmark circuits show that the proposed OR method achieves 100% fault coverage and the maximal diagnosis resolution for interconnects, and the multilevel routing algorithm effectively balances the routing density to achieve 100% routing completion. Experimental results show that our method significantly improves routing quality for testability and yield enhancement.

7.1 Introduction

With ever decreasing feature sizes and increasing chip dimensions, the integration complexity in system-on-a-chip (SOC) designs grows dramatically [105]. The high integration complexity is not only caused by the huge number of transistors and interconnects fabricated in a single chip, but also the modern SOC design issues in testability, manufacturability, and signal integrity. In particular, it is well known that interconnect delay dominates the circuit performance for nanometer IC designs. Therefore, it is desirable to handle the large-scale interconnect integration considering testability and diagnosability (defect reduction, yield enhancement, etc), manufacturability (process variation control, optical proximity correction [OPC], etc), and signal integrity (crosstalk minimization, etc) simultaneously.

Testability and diagnosability are very important issues for interconnect design in SOC ICs. Plenty of research works on interconnect testing can be found in the literature. Earlier works on interconnect testing were targeted for board-level testing. However, it is very difficult to apply these interconnect testing methods under the SOC environment without design-for-testability (DFT) support. The popular IEEE Std. 1500 [53] provides a structural support for core testing as well as interconnect testing in SOC. The IEEE Std. 1500 SOC test environment consists of a centralized test access mechanism (TAM) and wrappers around cores. The TAM defines the test control, while the wrappers provide a standardized interface for test data transmission. An oscillation ring test (ORT) [77], [78] method for interconnect test was proposed to detect not only stuck-at and open faults, but also delay and crosstalk glitch faults.

Many testing and diagnosis problems are incurred by particular interconnect structures, which can be partly solved by carefully determining the interconnect test structures. Further, to reduce the probability of multiple faults, it is desirable to reduce wiring congestion in a specific area. This approach is specifically important as the probability of back-end-of-line (BEOL) defects (i.e., high-resistance via and interconnect defects) increases [57]. Therefore, many issues with testability and diagnosability should be addressed during routing.

As technology advances, the manufacturing process increasingly constrains physical layout design and verification [86]. The chemical-mechanical polishing (CMP) technology [19], [93] is widely used to increase the metal layers integrated in a single chip. CMP induced variation is kept within acceptable limits by controlling local feature (interconnect) density, relative to a process-specific “window size,” to achieve global planarization for manufacturability and performance. Thus, balancing interconnect density minimizes the CMP induced variation, and thus routing plays an important role in determining the variation.

Optical proximity correction (OPC) is one of the most effective methods adopted to compensate for the light diffraction effect, typically used as a post layout process to improve manufacturability [51]. Recently, Huang and Wong proposed an algorithm that considers the OPC effect during routing by utilizing a symmetrical property. However, the process is time-consuming, and its results are still limited by the original layout quality. Again, balancing interconnect density can improve the OPC effects efficiently and effectively since the effects are also influenced by neighboring structures and shapes.

Signal integrity is an important factor that affects yield in nanometer IC technology [53]. Crosstalk affects the signal integrity in nanometer IC technology. Two adjacent wires form a coupling capacitor, and a signal changes on an aggressor net can interfere with the signal on a victim net. There are two types of crosstalk effects. One is glitch, which might induce malfunctioning in the logic values of circuit nodes and differ from what we design; the other is crosstalk-induced delay, caused by opposite switching signals in adjacent wires that slow down both signals. Crosstalk is also a crucial issue in modern router design [75], [76].

In this chapter, we handle the modern SOC design issues of testability and diagnosability, manufacturability, and signal integrity simultaneously in the routing stage for yield improvement (see Figure 7.1(a)). Traditionally, those issues are tackled at the post-layout stage. With the increasing design complexity, it is very difficult and even infeasible to handle those issues at the post-layout stage when most interconnect layouts are fixed and not flexible to be changed. In particular, those design issues can all be improved through balancing the routing congestion (see Figure 7.1(b)). Therefore, we shall present a congestion-driven routing algorithm for yield improvement.

We shall first review some important routing work. Traditionally, the complex routing problem is often solved by using the two-stage approach of global routing followed by detailed routing. Global routing first partitions the routing area into tiles and decides tile-to-tile paths for all nets while detailed routing assigns actual tracks and vias for nets. Many routing algorithms adopt a flat framework of finding paths for all nets. Those algorithms can be classified into sequential and concurrent approaches. Sequential routing algorithms include maze-searching approaches [48] and line-searching approaches, which route net-by-net. Most concurrent algorithms apply network-flow or linear-assignment formulation [15], [88] to route a set of nets at one time.

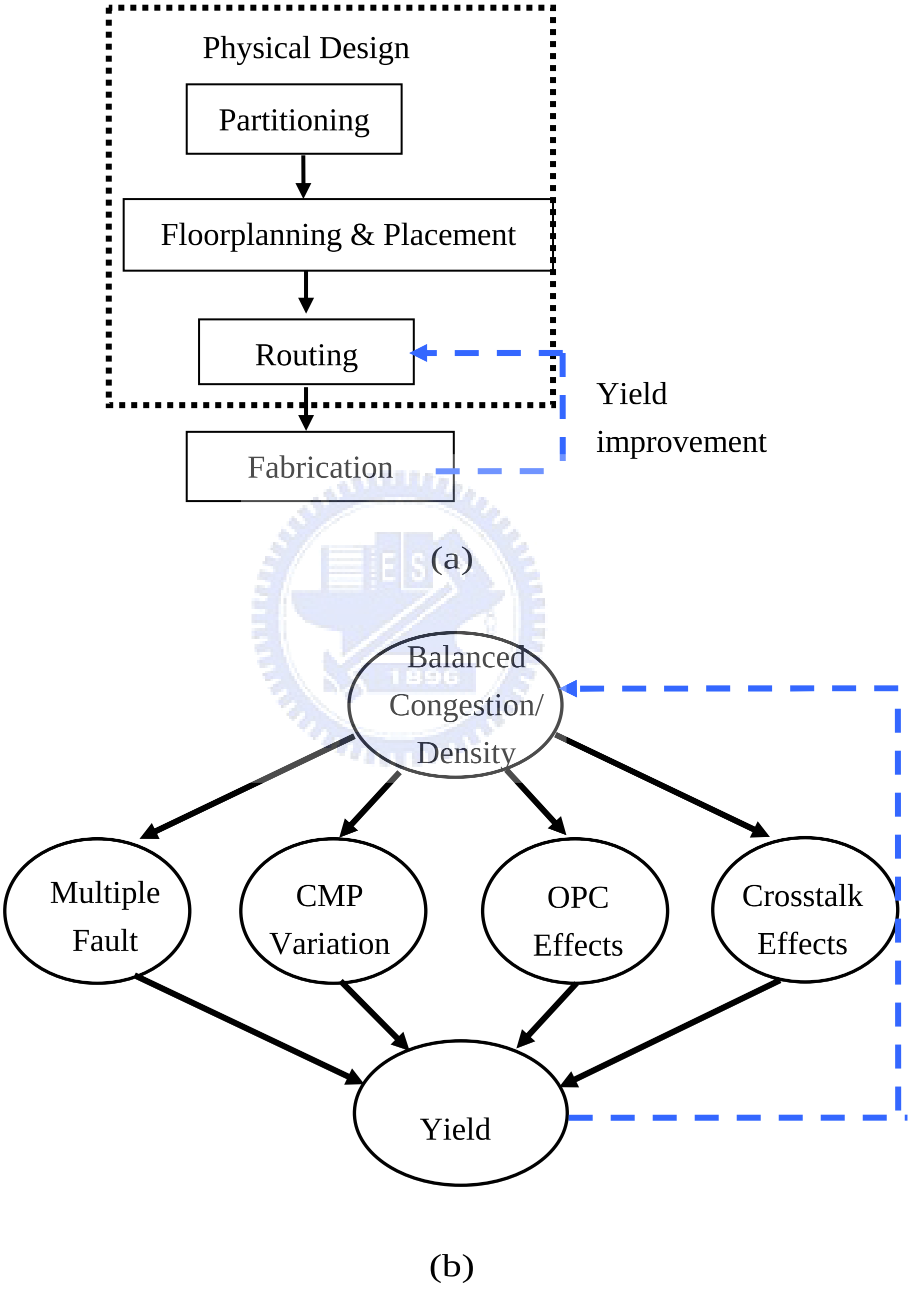


Figure 7.1 (a) Yield enhancement in routing stage. (b) Balancing routing congestion reduces multiple fault probability, CMP induced variation, OPC and crosstalk effects, all of which improve yield.

The major problem of the flat framework lies in their scalability for handling larger designs. As technology advances, technology nodes are getting smaller and circuit sizes are getting larger. To cope with the increasing complexity, researchers proposed to use hierarchical approaches to handle the problem by dividing a routing region into subregions and routing each subregion independently. Marek-Sadowska [88] proposed a hierarchical global router based on linear assignment. Chang, Zhu, and Wong [15] applied linear assignment to develop a hierarchical, concurrent global and detailed router for FPGA's.

The two-level, hierarchical routing framework, however, lacks information for the interactions among the subregions and is thus still insufficient in handling the dramatically growing complexity in current and future IC designs [27]. Therefore, it is desired to employ more levels of routing for very large-scale IC designs. The multilevel framework has attracted much attention in the literature recently. It employs a two-stage technique: coarsening followed by uncoarsening. The coarsening stage iteratively groups a set of circuit components (e.g., circuit nodes, cells, modules, routing tiles, etc) based on a predefined cost metric until the number of components being considered is smaller than a threshold. Then, the uncoarsening stage iteratively ungroups a set of previously clustered circuit components and refines the solution by using a combinatorial optimization technique (e.g., simulated annealing, local refinement, etc). The multilevel framework has been successfully applied to VLSI physical design. For example, the famous multilevel partitioners, *ML* [4], and *hMETIS* [59] the multilevel placer, *mPL* [13], and the multilevel floorplanner/placer, *MB*-tree* [64], all show the promise of the multilevel framework for large-scale circuit partitioning, placement, and floorplanning. A framework similar to multilevel routing was presented in [47], [71]. Lin, Hsu, and Tsai in [71] and Hayashi and Tsukiyama in [47] presented hybrid hierarchical *global* routers for multi-layer VLSI's, in which both the bottom-up (coarsening) and top-down (uncoarsening) techniques were used for global routing. Recently, Cong, Fang, and Zhang proposed a pioneering multilevel global-routing approach for large-scale, full-chip, routability-driven routing [27]. Cong, Xie, and Zhang later proposed an enhanced multilevel routing system, named MARS [31], which incorporates resource reservation, a graph-based Steiner tree heuristic and a history-based multi-iteration scheme to improve the quality of the multilevel global routing algorithm in [27]. The final tile-to-tile paths for all the nets are then fed into a detailed router to find the exact connection for each net. Lin and Chang also proposed a novel multilevel framework for full-chip routing, which considers both routability and performance [70]. This framework integrates global routing, detailed routing, and resource estimation together at each level, leading to more accurate routing resource estimation during coarsening and thus facilitating the solution refinement during uncoarsening. Their experimental results show the best routability among the previous works. Recently, Ho, et al. proposed yet another multilevel framework by introducing an intermediate layer and track assignment stage between coarsening and uncoarsening to handle crosstalk minimization [49]. A multilevel routing considering antenna effects was recently presented by Ho, Chang, and Chen in [50].

In this chapter, we propose a multilevel full-chip routing framework considering testability and diagnosability, manufacturability, and signal integrity simultaneously. Different from the previous works, our approach has the following distinguished features:

- Consider testability and diagnosability, manufacturability, and signal integrity simultaneously in the multilevel routing framework.
- Propose a new testability-driven multilevel routing framework, consisting of a preprocessing stage for oscillation ring test (ORT) generation for interconnect, a coarsening stage, an intermediate stage for optimization, an uncoarsening stage, and a postprocessing stage to process diagnosis patterns for ORT (i.e. oscillation ring diagnosis, ORD).
- Provide testability and yield enhancement solutions in the routing stage to both diagnose interconnects and improve density flexibility.
- Present heuristics to balance and reduce congestion in routing for yield improvement (by reducing multiple fault probability, CMP variation, OPC effects, and crosstalk).

Experimental results on the MCNC benchmark circuits show that the proposed OR method achieves 100% fault detection coverage and maximal diagnosis resolution for interconnects, and the multilevel routing algorithm effectively balances the routing density to achieve 100% routing completion. Experimental results show that our method significantly improves routing quality for testability and yield enhancement. Compared with [70], the experimental results show that our router improves the maximal congestion by 1.24X--6.11X in runtime speedup by 1.08X--7.66X, and improves the average congestion by 1.00X--4.52X with the improved congestion deviation by 1.37X--5.55X. Compared with [49], the experimental results also show that our router improves the maximal congestion by 1.54X--1.84X and the average congestion by 1.17X--1.34X with the congestion deviation being improved by 1.13X--1.63X.

This chapter is organized as follows. Section 7.2 gives a brief review of oscillation ring test and diagnosis. Section 7.3 presents the multilevel routing framework. Experimental results are reported in Section 7.4.

7.2 Preliminaries

In preliminaries, we study OR test architecture for interconnect detection and diagnosis, process variation on oscillation signals, the interconnect model for detection and further proposed interconnect model for diagnosis, CMP Models, Signal Integrity detection issues.

7.2.1 OR Test Architecture for Interconnects

In this section, we discuss the oscillation ring test for interconnects. Oscillation ring test (ORT) is a useful and efficient method to detect faults in SOC interconnects [77], [78]. An oscillation ring (OR) is a closed loop of a circuit under test in which has an *odd number of signal inversions*. Once the ring is constructed during test mode, oscillation signal appears on the ring. Figure 7.2 illustrates a global counter-based test architecture for both delay and crosstalk glitch detection for SOC ICs. This test architecture implements the IEEE Std. 1500 core test standard, in which each input/output pin of a core is attached with a *wrapper cell*, and a centralized test access mechanism (TAM) is provided to coordinate all test process. In additional to the normal input/output connections, all wrapper cells in a core can also be connected with a shift register, which is usually referred to as a scan path, to facilitate test access. A modified wrapper cell design has been proposed to provide extra connections and inversion control so that the oscillation rings can be constructed through the wires and the boundary scan paths in cores [77],[78]. For example, the oscillation ring test architecture in Figure 7.2 consists of one oscillation ring and a neighboring net, and two scan paths in cores C_1 and C_2 are part of the oscillation ring.

This test architecture can detect stuck-at, open, delay and crosstalk glitch faults. If an oscillation ring fails to oscillate, it implies that there exists stuck-at or open fault(s) in the oscillation ring. The period of the oscillation signal can also be measured by using a delay counter in a core to test delay faults, and a similar approach can be used for crosstalk glitch detection.

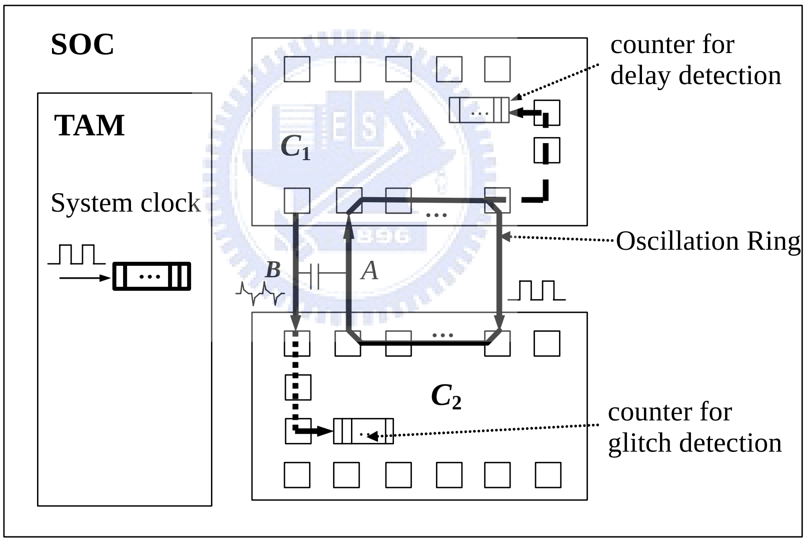


Figure 7.2 Test architecture among IPs for delay and crosstalk detection, and delay measurement (same as Figure 2.4).

A local counter is included in each core, and a central counter is in the TAM of the chip. The central counter in the TAM is enabled by signal *OscTest* and triggered by the system clock. A local counter is connected to one wrapper cell in each core; however, it can be accessed by every wrapper cell through the wrapper cell chain. When an oscillation ring passes a core, an internal scan path is formed to connect the oscillation signal to the local counter. For example, consider core C_1, in which the oscillation ring pass by (see Figure 7.2). The oscillation signal is fed to the local counter through a series of modified wrapper cells that are configured as SI→SO. When an oscillation test session starts ($OscTest = 1$), the TAM enables its own central counter as well as all local counters in cores. After the central counter in the TAM counts to a specific number n, the oscillation test session terminates and all local counters are disabled ($OscTest = 0$). Then all the local counter contents can then be scanned out to ATE for inspection.

Assume that m oscillation rings are tested. Let the frequency of the system clock be f, and the delay counter contents of the rings be n_1, n_2, …, n_m, respectively. An estimation of the i-th ring's oscillation frequency f_i can be approximated by

$$f_i = f \times n_i / n \tag{7.1}$$

Since the frequency of each ring is predetermined during the design phase, a delay fault can thus be detected and measured as compared with the result of the counters.

7.2.2 Process Variation Effects on Oscillation Signals

In order to consider process variation effect on this proposed OR scheme, we conducted experiment for a ring consisting of 7 inverters (plus transmission gates) and 20μm lines. The Monte Carlo simulation is conducted by changing the W/L ratio of all transistors and the R, C parameters of the nets. The mean is the nominal value, while the distribution is Gaussian with $3\sigma = 20\%$ of the nominal value. In all, 30 simulation runs are performed, and the simulation results are shown in Figure 7.3, in which all oscillation signals start at time 0. At the end of the first cycle, there is a small variation in the cycle period, and the variations are less than 0.9% of the nominal cycle period of the oscillation signal. The simulation results show that (1) this scheme can oscillate with an odd number of inversions, and (2) the process variation effects with 20% variance contribute to less than 0.9% in the frequency and oscillation period.

7.2.3 Interconnect Models in Oscillation Ring Test

A multi-terminal net is usually modeled by a *hypergraph*. The circuit structure of an SOC can be directly transformed into a hypergraph, in which each vertex denotes a pin while each *hypernet* represents a signal net. However, this graph model is not good enough for the OR test problem, as two branches of a net should belong to two different rings, and they cannot be tested simultaneously [77], [78]. Therefore, it would be better to consider each branch of a hypernet separately, instead of treating them as a whole. Each branch of a hypernet thus corresponds to a 2-pin net, which connects the source vertex to one of its sink vertices. An n-terminal hypernet is thus broken into (n–1) 2-pin nets. The result is a normal graph $G = (V, E)$, where E is the set of 2-pin nets.

A complete test for all interconnections is thus reduced to the problem of finding a set of rings that cover all edges corresponding to the interconnection structure in the graph G. This is equivalent to finding a set of sub-circuits (rings) $R = \{G_1, G_2, \dots, G_n\}$, such that

- $\forall G_i, G_i \subseteq G$, $G_i = (V_i, E_i)$, G_i is a ring, and
- $\bigcup_{i=1}^{n} E_i = E$

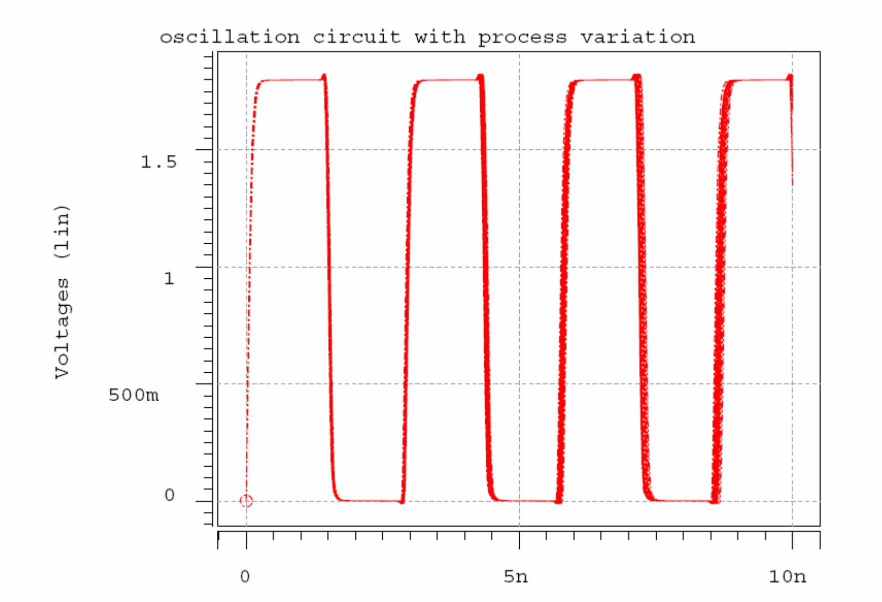


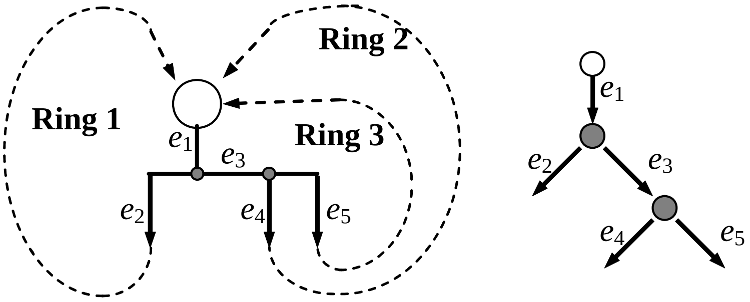
Figure 7.3 Simulation waveform with process variation effects on the oscillation ring test scheme.

If delay fault is considered, signal delay on each net along the ring should also be considered. The period of the oscillation signal is thus the summation of the path delay on all wires and scan paths. A large delay on an interconnect wire can be detected by observing the frequency of an oscillation signal that passes the wire under consideration. The detection can be masked by the variation of delays on other wires in the same ring, and thus the control of process variation is crucial for the correct detection.

7.2.4 Interconnect Diagnosis Model with Oscillation Ring Tests

Diagnosis is the process of locating the exact fault site. The oscillation ring test can also be used for interconnect diagnosis. For interconnect diagnosis, the two-pin net model is also not sufficient. Consider the 4-terminal net shown in Figure 7.4(a), which is divided into five edge segments e_1 to e_5. If edge e_1 is faulty, all three rings will not oscillate correctly. A faulty e_3 affects rings 2 and 3, while faults on edges e_2, e_4, and e_5 affect rings 1, 2, and 3, respectively. For diagnosis purpose, all these five segments are different.

From the above discussion, it is obvious that hypernets cannot be used for diagnosis. Therefore, the interconnect structure is transformed into a diagnosis graph model as follows. The scan path and wrapper cells in a core are lumped into a single *terminal node*, as we assume that they are fault-free. The fanout points of a hypernet form dummy *intermediate nodes*, and a wire segment connecting two nodes is an edge. For example, the diagnosis graph model for the hypernet of Figure 7.4(a) is shown in Figure 7.4(b), in which the white node is a terminal node and gray nodes are intermediate nodes. An edge is the smallest unit of a wire segment that can be uniquely diagnosed. From the above discussion, it can be seen that any stem affects all the downstream nodes and edges.



(a) (b)

Figure 7.4 (a) hypernet, and (b) interconnect diagnosis graph model
(Same as Figure 2.1).

7.2.5 CMP Model

One major source of process variations is the CMP process. In order to improve manufacturability, the variation induced by CMP should be kept within acceptable limits.

Several models for oxide planarization via CMP have been proposed in [93]. Among them, the model of [107] is neither computationally expensive nor difficult to calibrate. In this model, the interlevel dielectric (ILD) thickness z at location (x, y) is calculated as follows:

$$z = \begin{cases} z_0 - \left(\dfrac{K_i t}{\rho(x, y)}\right) & t < (\rho_0 z_1)/K_i \\ z_0 - z_1 - K_i t + \rho_0(x, y) z_1 & t > (\rho_0 z_1)/K_i \end{cases} \qquad (7.2)$$

In this model, the most important factor that determines the value of z is the effective pattern density $\rho(x, y)$. In other words, we can reduce the variation of dielectric thickness z by keeping the effective pattern density $\rho(x, y)$ relatively constant across the routing surface. Balancing the wiring congestion can effectively achieve this goal.

7.2.6 Signal Integrity

As the process technology advances, interconnect plays a dominant role in determining circuit performance and signal integrity [105]. Crosstalk-induced noise significantly affects delay and reduces signal integrity when technology improves, spacing diminishes, and coupling capacitance/ inductance increases [75], [76].

Figure 7.5 shows the noise model. The noise χ on the victim net is induced by a rising transition on the aggressor net through the coupling capacitance cc. The coupling capacitance is proportional to the fringing capacitance (cf) and the coupling length (*lc*), and it is inversely proportional to the distance (*d*) between the aggressor and the victim nets:

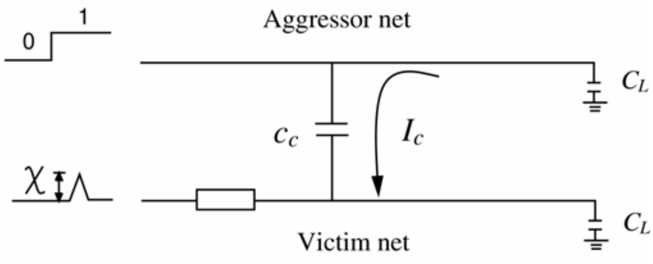


Figure 7.5 Noise due to crosstalk-induced current.

$$c_c = \frac{l_c c_f}{d} \tag{7.3}$$

Consider a wire *e* = (*u*, *v*), where *u* and *v* are two nodes in a routing tree. Let the length of the wire segment *e* be *le*, and *T*(*v*) be the subtree rooted at *v*. $IT_{(v)}$ is the total downstream current seen at *v* and is the current induced by aggressor nets on downstream wires of *v*. The current on a unit-length wire induced by aggressor nets is $i_0 = \lambda p c_0$ [3], where λ is the fixed ratio of coupling to total wire capacitance, *p* is the slope (i.e., power supply voltage over input rise time) of the aggressor net's signal, and c_0 is the unit-length wire capacitance. In deep submicron process, a major part of the wire capacitance is attributed to the coupling capacitance if the wire spacing is kept minimum (e.g., λ = 0.7 in [3]). The resulting noise $\chi(u, v)$ induced from the coupling current is the voltage pulse coupled from aggressor nets in the victim net for a wire segment *e* = (*u*, *v*). The induced noise can be expressed as

$$\chi(u,v) = r_b I_{T(v)} + r_0 l_e \left(\frac{i_0 l_e}{2} + I_{T(v)} \right) \tag{7.4}$$

The crosstalk effect can be effectively reduced by increasing wire spacing, which decreases the unit-length coupling capacitance according to Equation (7.3). To achieve this goal, a router should constrain or limit the coupled number and length of adjacent wires in any area, i.e., the maximum routing congestion, should be minimized.

7.3 Multilevel Routing Framework

We propose in this section a new multilevel routing framework, which takes routability, performance, testability, diagnosability, congestion, process variation, and crosstalk into account. The oscillation rings for test are based on circuit connectivity, and thus they can be constructed before routing. However, when delay fault is considered, the routing structure must also be considered, since the wire delay is mainly decided by the wire length. On the other hand, the diagnosis process has to consider the actual net layout, and they must be considered after the routing process.

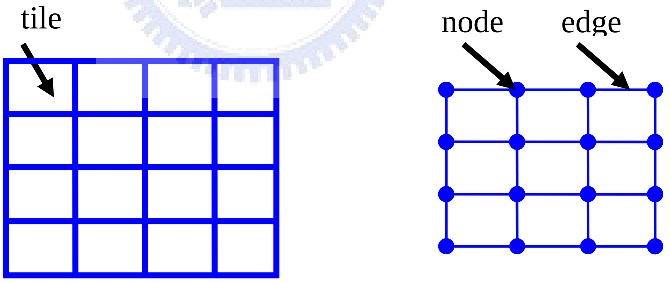
7.3.1 Routing Model

Our global routing algorithm is based on a graph search technique guided by the congestion information associated with routing regions. The router assigns higher costs to route nets through congested areas (or those of higher delay and/or crosstalk costs) to balance the net distribution among routing regions. Before we apply the graph search technique to multilevel routing, we first model the routing architecture as a graph such that the graph topology represents the chip structure. Figure 7.6 illustrates the routing graph model.

For the modeling, we first partition a chip into an array of rectangular subregions.

These subregions are called *global cells* (GC). A node in the graph represents a GC in the chip, and an edge denotes the boundary between two adjacent GCs. Each edge is assigned a weight/capacity according to the physical area or the number of tracks of a GC. The graph is used to represent the routing area and is called a *multilevel routing graph*, denoted by *Gk*, where *k* is the level ID. A global router finds GC-to-GC paths for all nets on a routing graph to guide the detailed routing. The goal of global routing is to route as many nets as possible while meeting the capacity constraint of each edge and any other constraints, if specified.

As the process technology advances, multiple routing layers are possible. The number of layers in a modern chip can be more than eight. Wires in each layer can run either horizontally (H) or vertically (V) in a grid style.



(a) the partitioned layout (b) the routing graph

Figure 7.6 The routing graph.

As illustrated in Figure 7.7, G_o corresponds to the routing graph of the level 0 of the multilevel coarsening stage. At each level, our global router first finds routing paths for the *local nets* (or *local 2-pin connections*) (those nets that entirely sit inside a GC). After the global routing is performed, we merge 2×2 of GC into a larger *Gi* and at the same time perform resource estimation for use at the next level (i.e., level 1 here). Coarsening continues until the number of GCs at a level, say the *k*-th level, is below a threshold. The uncoarsening stage tries to refine the routing solution of the unassigned segments of the level *k*. During uncoarsening, the unroutable nets are performed by point-to-path maze routing and rip-up and re-route to refine the routing solution. Then we proceed to the next level (level *k*–1) of uncoarsening by expanding each *Gk* to four finer Gk_{-1}'s. The process continues until we reach level 0 when the final routing solution is obtained.

7.3.2 Testability-Aware Multilevel Routing

In the coarsening stage of multilevel routing, shorter nets are routed first, and a congestion-driven heuristic is used to guide a pattern router. For all the nets that can be successfully routed, both global route and detailed route are conducted. All the nets that fail to complete will be handled at the uncoarsening stage. At the uncoarsening stage, the failed nets are routed by a global router with a different cost function to avoid heavily congested area, and a detailed maze router is used to determine the final routing paths. In addition to the traditional multilevel framework, we incorporate an oscillation ring test in the preprocessing stage to guide the resource estimation for interconnect and 100% fault detection coverage, an intermediate stage for interconnect optimization, and an oscillation ring diagnosis (ORD) in the postprocessing stage to guarantee maximal interconnect diagnosability (see Figure 7.7).

7.3.3 Diagnosability-Aware Routing Structure

The minimum spanning tree (MST) topology leads to the minimum total wire length, and thus congestion is often easier to be controlled for MST than other topologies. This topology may result in longer critical paths and thus degrade circuit performance. In contrast, a shortest path tree (SPT) may result in the best performance, but its total wire length (and congestion) may be significantly larger than that constructed by the MST algorithm.

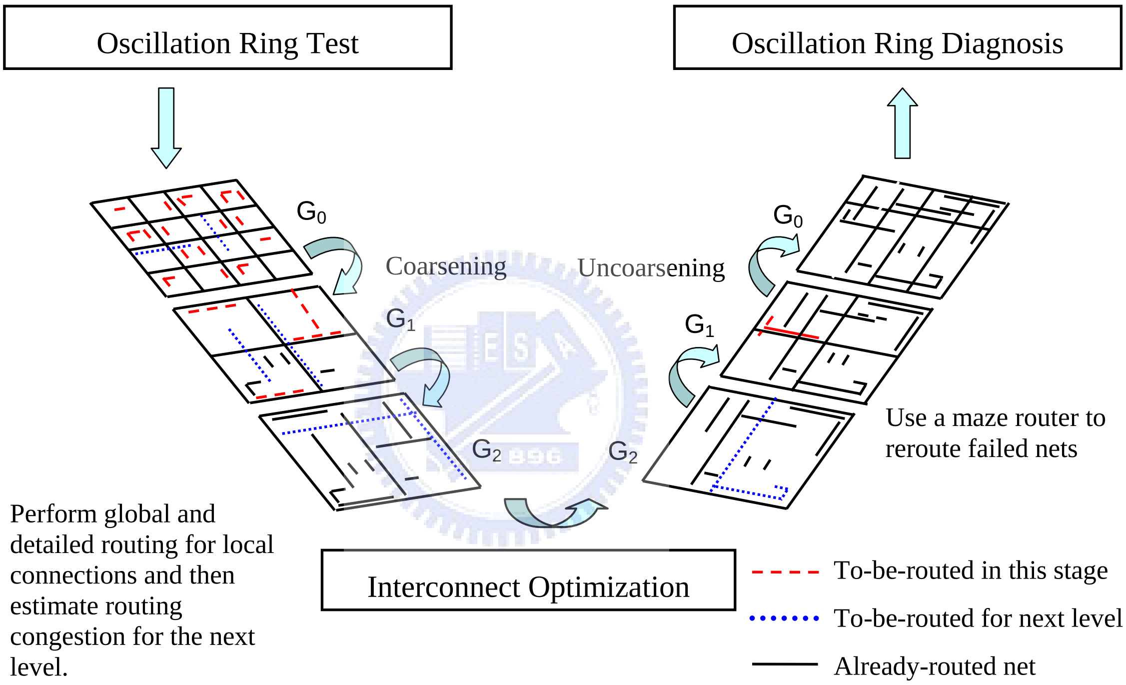


Figure 7.7 Integrated multilevel routing framework.

The diagnosis problem also affects the routing structure. For instance, consider the 4-terminal net example shown in Figure 7.8. With the *spanning tree* connection given in Figure 7.8(a), there are three different net segments to be diagnosed. On the other hand, as the diagnosis graph model shown in Figure 7.4(b), for the *Steiner tree* connection given in Figure 7.8(b), there are two intermediate nodes (indicated by the two dotted circles) and thus five net segments to be diagnosed. In general, a spanning tree connection employed fewer wire segments to be diagnosed, and thus it is favored in our router. Our algorithm first constructs the minimum spanning tree (MST) structure whenever possible, which is best for diagnosability. Otherwise, it will find a routing tree with the least number of intermediate nodes.


(a) (b)

Figure 7.8 Two routing trees: (a) a spanning tree with three segments (b) a Steiner tree with the minimum number of intermediate nodes, resulting in five segments.

In order to route a net with the minimum number of intermediate branch nodes and the shortest path, we apply the algorithm shown in Figure 7.9(a) for the routing tree construction. The algorithm, which is based on Dijkstra's shortest path algorithm, finds a shortest path with the minimum number of intermediate nodes. It associates each basic detailed routing region u with two labels: $d(u)$ and $n(u)$, where $d(u)$ is the distance of the shortest path from source s to u, and $n(u)$ is the minimum number of intermediate nodes along the shortest path from s to u. Initially, $d(u) = \infty$, $n(u) = \infty$, $\forall u \neq s$, $d(s) = 0$, and $n(s) = 0$. The computation of label d's is the same as the original Dijkstra's algorithm. The computation of $n(v)$ is shown in Figure 7.9(b), where $dist(u, v)$ and $node(u, v)$ are the distance and the number of intermediate nodes between nodes u and v, respectively.

```
SPA($G$, $dist$, $s$)
    Initialize_Source($G$, $s$);
    $S \leftarrow \emptyset$;
    $Q \leftarrow V[G]$;
    while $Q \neq \emptyset$
        $u \leftarrow$ Extract_Min($Q$);
        $S \leftarrow S \cup \{u\}$;
        for each $v \in Adj[u]$
            Count_Nodes($u$, $v$, $dist$);
```

```
Count_Nodes($u$, $v$, $dist$)
if $d(v) \geq d(u) + dist(u, v)$
    $d(v) \leftarrow d(v) + dist(u, v)$;
    if $n(v) \geq n(u) + node(u, v)$
        $n(v) \leftarrow n(u) + node(u, v)$;
        record $u$ as the
```

(a) (b)

Figure 7.9 (a) Shortest path algorithm, (b) $n(v)$ computation.

7.3.4 Cost Metric for Routing Density Control

A router that incurs imbalanced routing density may degrade system performance in many ways.

- Crosstalk effects are the results of signal coupling between adjacent wires, and the coupling capacitance is usually inversely proportional to the distance between wires. In a heavily congested area, the distance between adjacent wires is small and thus the probability of crosstalk faults is increased.
- Physical defects in a congested area may create multiple faults, which are difficult to be detected and diagnosed.
- Process variation due to CMP effects is usually caused by unbalanced routing congestion or density.

Therefore, it is desirable to balance routing congestion/density in all areas for router design. Given a netlist, we first run the minimum spanning tree (MST) algorithm to construct the topology for each net, and then decompose each net into 2-pin connections, with each connection corresponding to an edge of the minimum spanning tree. Our multilevel framework starts from coarsening the finest tiles of level 0. At each level, tiles are processed one by one, and only local nets (connections) are routed. At each level, the two-stage routing approach of global routing followed by detailed routing is applied. The global routing is based on the approach used in the pattern router [60] and first routes local nets on the tiles of level 0. Let the multilevel routing graph of level i be $Gi = (Vi, Ei)$. Let $Re = \{e \in Ei \mid e$ is the edge chosen for routing$\}$. In order to balance the routing density, we use the cost function : $Ei \rightarrow R$ to guide the routing:

$$\alpha(R_e) = \sum_{e \in R_e} c_e \tag{7.5}$$

where c_e is the congestion of edge and it is defined as

$$c_e = \begin{cases} \dfrac{1}{2^{[(p_e/t)-d_e]}} & d_e < (p_e/t) \\ 1 & d_e \geq (p_e/t) \end{cases}$$

where p_e and d_e are the capacity (p_e) and the number of nets assigned to edge e (d_e), respectively. The parameter t is used to define the target level of the maximum density, and it can be determined either by the user or by averaging over all routing areas. For example, if the goal is to make the average routing density to be half of the maximum acceptable density, then t is set to 2.

After the global routing is completed, we perform detailed routing with the guidance of the global-routing results and find a real path in the chip. Our detailed router is based on the maze-searching algorithm. Pattern routing uses an L-shaped or a Z-shaped route to make the connection, which gives the shortest path length between two points. Therefore, the wire length is minimized, and we do not include wire length in the cost function at this stage. We measure the routing congestion based on the commonly used channel density. After the detailed routing finishes routing a net, the channel density associated with an edge of a multilevel graph is updated accordingly.

Our global router first tries L-shaped pattern routing. If the routing fails, we try Z-shaped pattern routing. If both pattern routes fail, we give up routing the connection, and an overflow occurs. We refer to a *failed net* (*failed connection*) as that causes an overflow. The failed nets (connections) will be reconsidered (refined) at the uncoarsening stage.

The uncoarsening stage starts to refine each local failed net (connection), left from the coarsening stage. The global router is now changed to the maze router with the following cost function : $Ei \rightarrow R$:

$$\beta(R_e) = \sum_{e \in R_e} \left(a \cdot c_e + b \cdot o_e \right) \tag{7.6}$$

where a, b, are user-defined parameters, and $o_e \in \{0,1\}$. If an overflow happens, o_e is set to 1; otherwise, it is set to 0.

There is a trade-off between minimizing congestion and overflow. At the uncoarsening stage, we intend to resolve the overflow in a tile. Therefore, we make b much larger than a. Also, a detailed maze routing is performed after the global maze routing. Iterative refinement of a failed net is stopped when a route is found or several tries have been made. Uncoarsening continues until the first level G_0 is reached and the final solution is found.

7.4 Experimental Results

The multilevel routing system is implemented in the C programming language on a 900 MHz SUN Blade 2500 workstation with 1GB memory. We conduct two sets of experiments: (1) testability enhancement, and (2) congestion control for routing considering multiple faults, manufacturability, and crosstalk. Three types of benchmarks are used in our experiments: the first type is for inter-module interconnects only (see Table 7.1); the second is the full-chip benchmarks (only mcc1 and mcc2), which include both inter-module interconnections and intra-module interconnections; the third type contains only intra-module interconnections which are local interconnections within standard-cell modules. The statistics of type-2 and -3 benchmarks are given in Table 7.2.

7.4.1 Testability Enhancement

For testability enhancement, the experimental results of the embedded OR scheme in the proposed multilevel routing framework are reported in Table 7.1. We have presented both a detection (the preprocessing stage) and a diagnosis schemes (the postprocessing stage) as shown in Figure 7.7 for oscillation ring based interconnect testing in SOC in a predetermined design flow. Thus, $fmin \leq fi \leq fmax$ gives the timing specification for this scheme, where fi is the estimated oscillation frequency for the i-th ring. Since our target of this OR scheme is for interconnect among modules, our experiments are conducted based on the MCNC benchmark circuits with inter-module connections.

Table 7.1: Experimental results based on the MCNC benchmarks for testability enhancement of interconnect detection and diagnosis.

Circuit	Statistics				#rings constructed for testability $\lvert R_t \rvert$ & diagnosis $\lvert R_d \rvert$	
	#core	#pad	#hyp	#2-pin	$\lvert R_t \rvert$	$\lvert R_d \rvert$
ac3	27	75	211	416	133(33.3ms)	374(93.5ms)
ami33	33	42	117	343	242(60.5ms)	303(75.8ms)
ami49	49	22	361	475	156(39ms)	386(96.5ms)
apte	9	73	92	136	73(18.3ms)	122(30.5ms)
hp	11	45	72	195	81(20.3ms)	164(41ms)
xerox	10	2	161	356	218(54.5ms)	342(85.5ms)

Table 7.1 gives the names of the circuits, the statistics for the circuits (the number of cores, #core; the number of pads, #pad; the number of hyperedges, #hyp; the number of 2-pin nets), the number of rings constructed for detection, $\lvert Rt \rvert$, and the number of rings constructed for diagnosis, $\lvert Rd \rvert$. Thus, $\lvert Rt \rvert$ is the testability-driven cost in the preprocessing stage, and $\lvert Rd \rvert - \lvert Rt \rvert$ is the additional cost for the postprocessing stage. In addition to the 100% fault coverage of the oscillation ring detection scheme, we also obtain 100% net segment diagnosability.

To show the feasibility of this scheme, we include the actual estimated ATE measurement time in the parentheses in Table 7.1. Since the frequency of each ring is predetermined during the design phase, a delay fault can thus be detected and measured by inspecting the contents of the local core counters (see Figure 7.2). Let the oscillation frequency of the rings, according to the timing specification, be $fmin \leq fi \leq fmax$, with the unit time of measuring T_0 (= n/f). Thus, we have delay the counter contents of $nmin \leq ni \leq nmax$, where $nmin = fmin \times T_0$ and $nmax = fmax \times T_0$. Let ξ be the resolution of delay measurement, and ε be the maximum measurement error. Since a counter's maximum measurement error is ±1, the requirement for ε should be the reciprocal of $fmin$ times T_0.

$$\varepsilon = \frac{1}{f_{\min} \times T_0} \leq \zeta \tag{7.7}$$

We show an example of the delay measurement. Let the frequency specification of the oscillation rings be 4 MHz to 400 MHz, and ξ is 0.001, which implies that the counter content *nmin* is at least 1000. From Equation (7), we have the required T_0 250μs. Thus, we get the estimated detection and diagnosis time in the parentheses. For example, for the ac3 circuit, we need 133 rings to detection and 374 rings to diagnose; therefore 133 x 250μs = 33.25 ms for interconnect detection, and 374 x 250μs = 93.5 ms for interconnect diagnosis. This shows the effectiveness and efficiency of the testability enhancement.

Table 7.2:Routing benchmark circuits.

Circuit	Size (μm)	#Layers	#Nets	#Pins
Mcc1	39000×45000	4	1694	3101
Mcc2	152400×152400	4	7541	25024
Struct	4903x4904	3	3551	5717
Primary1	7552x4988	3	2037	2941
Primary2	10438x6468	3	8197	11226
S5378	4330x2370	3	3124	4734
S9234	4020x2230	3	2774	4185
S13207	6590x3640	3	6995	10562
S15850	7040x3880	3	8321	12566
S38417	111430x6180	3	21035	32210
S38584	12940x6710	3	28177	42589

7.4.2 Congestion Control for Multi-objective Optimization

As mentioned in Section 2, the CMP variation is controlled by reducing the variation of pattern density ρ (see Section 2.5), while the signal integrity problem can be alleviated by reducing the routing congestion (see Section 2.6). It will be clear later that our router effectively reduces both parameters, (i.e., the congestion variation for CMP and the maximum congestion for the crosstalk effect).

Table 7.3 reports the results for multilevel routing considering multiple faults, manufacturability, and crosstalk. We compare three different routing algorithms: (A) performance-driven MR [70], (B) routability-driven MR [70], and (C) our proposed method (with MST routing and balanced density).

In each case, we give the maximum (critical path) delay d_{max}, average delay d_{avg}, and the maximum number of nets crossing a level-0 tile $\#Net_{PEAK}$, which is a good estimate for the maximum routing density. In our experiment, we set the parameter $t = 4$ for the ISCAS89 circuits, while for other benchmarks are set to $t = 2$. The completion rate is 100% for all cases. It can be seen that the proposed method achieves about the same level of performance as the routability-driven method does by up to 0.2% increase in d_{max} and d_{avg},, but the maximum density is much smaller. Compared with [70], the experimental results show that our router improves the maximal congestion ($\#Net_{PEAK}$) by 1.24X--6.11X in runtime speedup by 1.08X--7.66X.

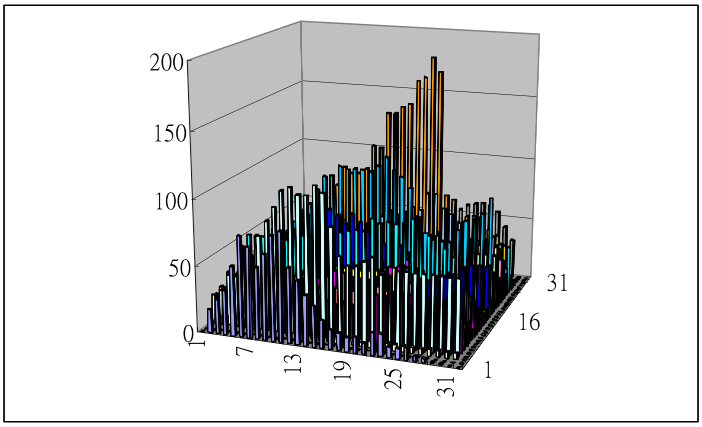
In Table 7.3, we show some statistical density results. The average number of nets crossing a level-0 tile is denoted by $\#Net_{avg}$, and we also list those of vertical tiles and horizontal tiles $\#Net_{avg\_v}$ and $\#Net_{avg\_h}$ respectively. Also, σ_v is denoted for the standard deviation from the vertical tile prospect and σ_h for that of the horizontal tile prospect. The results show that our scheme is more effective for the full-chip benchmarks mcc1 and mcc2. For other intra-module routing, our scheme also improve the results for most cases. Compared with [70], the experimental results show that our router improves the average congestion by about 1.00X--4.52X, and improves the balanced congestion (σ_v and σ_h, standard deviation respective for vertical and horizontal tiles) by 1.37X-- 5.55X.

To demonstrate the effectiveness of the proposed algorithm in balancing the routing density, the number of horizontal wires crossing each level-0 tile for benchmark mcc1 is shown in Figure 7.10 for the three algorithms. It can be seen that the performance-driven MR results are the least balanced routing; and the peak congestion is 181 (*#NetPEAK*) in mcc1. The routability-driven MR tries to avoid congested area to improve the probability of successful routing, and thus reduces the maximum density; and its peak congestion is still 61. With the proposed algorithm, the maximum density is further reduced to 45; and thus the manufacturability effects, the probability of multiple faults, and crosstalk effects are reduced accordingly.

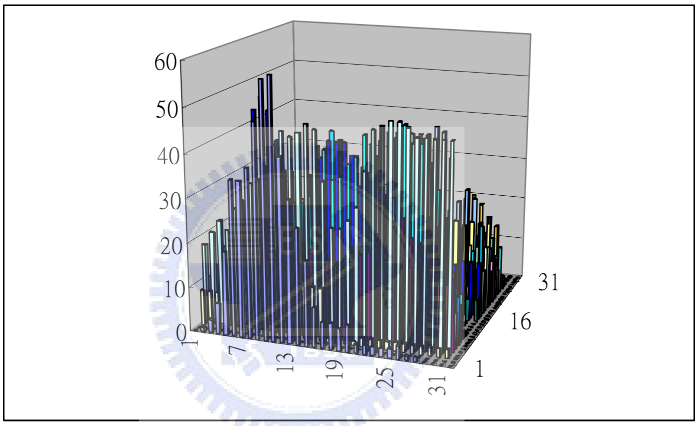
Mcc1 shows the maximal congestion improvement in our proposed algorithm by 1.36X compared to the routablility-driven MR and by 4.02X compared to the performance-driven MR. For mcc1, our proposed algorithm improves the average congestion by 1.01X--1.02X compared to the routablility-driven MR and by 2.81X--2.85X compared to the performance-driven MR. For balanced congestion on mcc1, our proposed algorithm improves the result by 1.38X--1.48X compared to the routablility-driven MR and by 2.72X--3.32X compared to the performance-driven MR. For runtime speedup, our approach improves by 1.06X compared to routabillity-drive MR and by 3.08X compared to performance-driven MR.

Further, the interconnection congestion, as evident in the inter-module connections in mcc1 and mcc2, demonstrates the respective maximal and average congestion improvements by 1.39X--3.23X and 1.03X--2.36X with the congestion balance improvement (σ_v and σ_h, standard deviation respective for vertical and horizontal tiles) by 1.37X--2.76X.

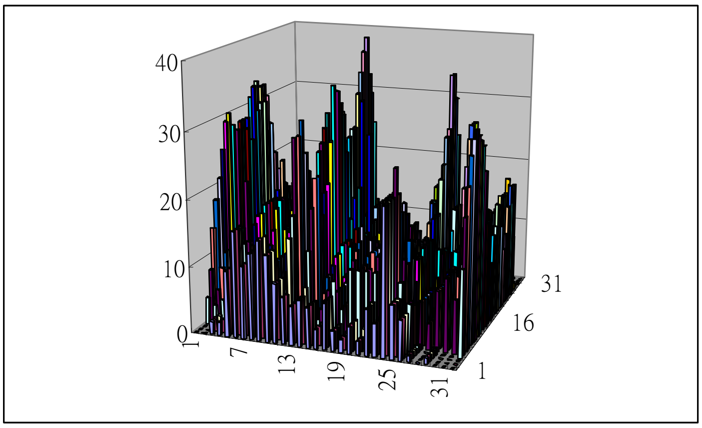
Finally, in Table 7.5 we compare the results of our work with the crosstalk- and performance-driven router presented in [49]. We can see that our router improves the respective maximal and average congestion improvements by 1.54X--1.84X and 1.17X--1.34X with the congestion balance improvement (σ_v and σ_h, respective standard deviations for vertical and horizontal tiles) by 1.13X--1.63X. Please note that we do not compare the maximum (critical path) delay d_{max}, average delay d_{avg} with [49] since our congestion-guided router achieves 100% routing completion while the work [49] does not.



(a)



(b)



(c)

Figure 7.10 Routing density distribution for mcc1 for (a) the performance-driven MR, (b) the routability-driven MR, (c) and the proposed algorithm.

Table 7.3. Comparison of routing results of maximum density with both maximum delay and average delay.

Circuit	(A) Performance-Driven [70]				(B) Routability-Driven [70]				(C) Proposed Balanced Density with 100% routability			
	d_{max}	d_{avg}	#Net_{PEAK}	CPU	d_{max}	d_{avg}	#Net_{PEAK}	CPU	d_{max}	d_{avg}	#Net_{PEAK}	CPU
Mcc1	4.65e+7	1.08e+7	181	223.68	2.03e+8	3.32e+7	61	77.11	2.03e+8	3.33e+7	40	72.63
Mcc2	7.26e+7	5.07e+6	274	5964.2	8.46e+7	5.12e+6	135	2855.5	8.51e+7	5.11e+6	96	2592.34
Compar.	0.41	0.41	3.35	2.32	1	1	1.44	1.10	1	1	1	1
Struct	1.13e+6	6.93e+4	32	307.91	1.52e+6	7.13e+4	9	56.33	1.52e+6	7.13e+4	7	56.53
Primary1	3.01e+5	3.33e+4	51	241.96	7.00e+5	5.51e+4	17	63.9	6.99e+5	5.50e+4	15	64.36
Primary2	3.91e+6	2.08e+5	91	1808.56	3.92e+6	2.09e+5	28	298.17	3.91e+6	2.09e+5	25	295.32
S5378	8.89e+4	6.38e+3	49	23.28	8.91e+4	6.39e+3	17	4.13	8.94e+4	6.41e+3	15	4.29
S9234	1.02e+5	9.4e+3	61	16.78	2.53e+5	1.19e+4	15	2.91	2.53e+5	1.19e+4	14	2.9
S13207	3.96e+5	2.04e+4	114	65.45	4.64e+5	2.04e+4	30	14.44	4.64e+5	2.03e+4	27	14.57
S15850	6.03e+5	2.89e+4	140	181.82	2.66e+6	6.68e+4	30	22.04	2.66e+6	6.67e+4	26	21.77
S38417	5.22e+5	2.93e+4	272	741.53	8.52e+6	3.94e+5	27	50.02	8.52e+6	3.94e+5	23	50.08
S38584	1.64e+6	5.83e+4	295	1453.8	1.76e+8	1.25e+7	31	127.8	1.76e+8	1.25e+7	29	122.5
Compar.	0.45	0.35	6.11	7.66	1	1	1.24	1.08	1	1	1	1

Table 7.4. Comparison of routing results of statistical density with [70].

Circuit	(A) Performance-Driven [70]				(B) Routability-Driven [70]				(C) Proposed Balanced Density with 100% routability			
	#Net_{avg_v}	#Net_{avg_h}	σ_v	σ_h	#Net_{avg_v}	#Net_{avg_h}	σ_v	σ_h	#Net_{avg_v}	#Net_{avg_h}	σ_v	σ_h
Mcc1	28.19	31.78	20.59	24.35	10.03	11.50	10.45	10.82	9.91	11.33	7.58	7.33
Mcc2	39.35	44.05	37.26	46.98	19.39	21.65	23.53	25.80	18.74	20.88	17.30	18.54
Comparison	2.36	2.35	2.33	2.76	1.03	1.03	1.37	1.42	1	1	1	1
Struct	4.97	4.86	4.62	5.03	1.42	1.41	1.24	1.67	1.42	1.41	1.07	1.59
Primary1	2.29	1.74	3.00	5.67	0.70	0.60	1.05	1.95	0.70	0.60	1.20	1.80
Primary2	7.22	7.49	5.56	18.23	2.05	1.85	1.59	4.57	2.05	1.85	1.56	4.45
S5378	12.53	13.46	9.16	8.40	4.38	3.44	3.45	2.13	4.40	3.46	3.44	2.10
S9234	14.16	9.99	12.91	7.04	3.95	2.56	3.25	1.62	3.95	2.56	3.24	1.60
S13207	28.43	20.49	18.40	11.08	9.30	5.93	5.77	2.76	9.29	5.92	5.23	2.81
S15850	36.61	34.48	23.89	20.42	10.29	7.41	5.63	2.92	10.31	7.41	5.39	2.91
S38417	44.58	27.38	37.36	27.94	7.31	4.27	4.75	2.17	7.3	4.27	4.44	2.18
S38584	43.99	30.53	35.93	20.12	9.06	5.80	5.74	2.86	9.05	5.79	5.43	2.88
Comparison	4.02	4.52	4.87	5.55	1.00	1.00	1.17	1.23	1	1	1	1

Table 7.5. Comparison of routing results of statistical density with [49].

Circuit	(A) crosstalk- and performance-driven [49]						(B) Proposed Balanced Density with 100% routability					
	$\#Net_{PEAK\_v}$	$\#Net_{PEAK\_h}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h	$\#Net_{PEAK\_v}$	$\#Net_{PEAK\_h}$	$\#Net_{avg\_v}$	$\#Net_{avg\_h}$	σ_v	σ_h
Mcc1	75	66	17.96	10.98	6.44	9.46	40	36	9.91	11.33	7.58	7.33
Mcc2	146	172	20.53	28.62	21.64	32.54	80	96	18.74	20.88	17.30	18.54
Compar.	1.84	1.80	1.34	1.23	1.13	1.63	1	1	1	1	1	1
Struct	31	28	4.59	4.85	4.49	5.07	7	7	1.42	1.41	1.07	1.59
Primary1	27	29	2.03	0.94	3.58	3.30	7	15	0.70	0.60	1.20	1.80
Primary2	32	37	2.90	2.65	2.95	5.20	10	25	2.05	1.85	1.56	4.45
S5378	17	13	5.11	3.59	3.62	2.43	15	10	4.40	3.46	3.44	2.10
S9234	16	9	4.79	2.56	3.36	1.77	14	7	3.95	2.56	3.24	1.60
S13207	33	17	9.39	6.35	6.14	3.42	27	15	9.29	5.92	5.23	2.81
S15850	32	15	9.94	7.43	6.33	2.96	26	16	10.31	7.41	5.39	2.91
S38417	29	13	7.41	4.19	4.75	2.23	23	12	7.3	4.27	4.44	2.18
S38584	56	28	10.83	6.45	8.36	3.51	29	16	9.05	5.79	5.43	2.88
Compar.	1.73	1.54	1.18	1.17	1.41	1.34	1	1	1	1	1	1

Chapter 8

Conclusions

We have addressed in this thesis two closely related issues on Interconnect-centric architectures and algorithms. Specifically, we have discussed the analysis, design and test of Oscillation Ring Based for system-level interconnect detection and diagnosis, gate-level Oscillation Ring Based synthesis. SoC global routing based on a novel metric for congestion guided routing-tree construction for yield and testability enhancement. In this chapter, we give concluding remarks on the study and discuss directions for future research.

8.1 A unified approach to detecting and optimizing

In this chapter, we have analyzed the crosstalk effects on the interconnect bus in the very deep sub-micron SOC VLSI and have proposed and demonstrated a unified detection scheme to test glitches and the crosstalk induced delay at the same time. For the crosstalk analysis, we have found that the crosstalk induced fault in fact is superposition of the crosstalk induced glitch fault and the original applied signal on the victim line and the induced delay is more important in affecting the timing performance of the circuit and hence is more important to be considered in testing. For the detection scheme, we have proposed a pulse detector with adjustable detection threshold to detect induced glitches, thus detect the induced delay. Several issues regarding the detection scheme such as the amplitude and width of the induced glitch, the skews between the applied excitation signals on the aggressor line and the victim line, and manufacture process variation effect have been discussed. Experiments based on the Monte Carlo simulation, which simulates the real testing environment under the manufacture process variation, have been done on the detection scheme. The results show that the scheme, under the bus interconnect topology for which skews between each bus interconnect approach zero, offer a testing yield of 92.915%, which is an acceptable testing yield, for testing crosstalk faults.

8.2 IEEE Standard 1500 Compatible Oscillation Ring Based Interconnect Delay and Crosstalk Test Methodology

We have presented a novel oscillation ring (OR) test methodology as an effective DFT technology for interconnects among intellectual property (IP) modules in an SOC under the IEEE IEEE Std. 1500 environment. The target interconnect fault models include stuck-at, open, delay faults, and crosstalk glitches, which are difficult to test under traditional test architectures. Simulation on an MCNC benchmark circuit implemented with the TSMC 0.18 μm technology has shown the feasibility of the methodology. The path delay of the interconnection wires can also obtained by measuring period of the oscillation signals, thus the proposed scheme can be used for the circuit parameter measurement. The IEEE Std. 1500 wrapper cells and boundary scan wrapper cells are modified to accommodate the test scheme. We adopted a graph model approach to propose an efficient ring-construction algorithm to construct to achieve 100% interconnect coverage, consequently fault coverage. The OR scheme as an oscillation DFT gives high testability, and can be viewed as a variance of BIST with the IEEE Std. 1500 design. The scheme can also be extended to diagnosis the interconnect faults since the fault on an edge common to two or more rings will fail all oscillation signals passing through the shared common edge(s). This will be the topic of the followed study for this scheme.

8.3 IEEE Standard 1500 Compliant Interconnect Diagnosis for Delay and Crosstalk Faults

In this chapter, we have presented an IORD scheme for interconnect faults in SOC. In addition to the *100% fault detection coverage for each net* achieved by the IORT scheme, we have shown that fault location or fault diagnosis can also be done by including some extra test rings to achieve the *maximum diagnosis for each net segment*. We have also presented two heuristics, diagnosability check and diagnosis ring generation, with theoretical study and integrated them into the IORD algorithm. We have further proposed the predetermined and adaptive diagnosis schemes and analyzed their costs. Finally, two optimization techniques for improving interconnect diagnosability are proposed and showed to be effective. Experimental results have justified the efficiency and effectiveness of the proposed methods.

8.4 Oscillation Ring Test for Synchronous Sequential Circuits (also known as Finite State Machine Synthesis for At-Speed Oscillation Testability)

In this chapter we presented a novel oscillation test architecture for sequential circuits, in which at-speed testing is possible. As a result, delay related faults are detectable. We developed MSR cells for this architecture, and proposed an efficient algorithm for test generation. The proposed method requires approximately the same amount of test vectors as scan tests to achieve 100% test efficiency for stuck-at faults, and it outperforms the pure scan tests at the average ratio of 2:1 in the test cases.

In the future, we shall also consider state assignment method that makes 100% test efficiency possible with the oscillation test only.

8.5 Multilevel Full-Chip Routing with Testability and Yield Enhancement

We have shown that the embedded oscillation ring test and diagnosis scheme is feasible based on the simulation results with TSMC .18 μm process technology. Also, this OR scheme achieves 100% fault detection coverage and maximal diagnosability. We have also presented an effective multilevel routing framework that applies a congestion-driven routing algorithm to reduce the multiple-fault probability, CMP and OPC induced effects, and crosstalk effects for yield enhancement.

8.6 Future Work

Future researches in Interconnect-Centric oscillation ring based design and test includes:

- By applying the embedded oscillation rings as an on-chip measurement unit for process variation and thermal effects to layout-related delay effects.
- Application of DSP techniques to analyze the syndrome of mutual interleaving oscillation rings' frequency to study crosstalk effects, especially for 2nd and other parasitic effects based on RLC distributed wire models.

- Development of the 3-D maximum concurrency algorithm with a tighter performance bound for system-level interconnects.
- Unification of the statistical timing-driven non-tree routing-tree formulation and the new congestion metric presented in Chapter 7 for yield enhancement verified in inverse-V ML X-Structure Router.
- Study of multiple sinks of interconnects. Unification of multiple-sink net model with the minimum radium with blockage consideration in interconnect-driven X-Structure based on multi-weighted graphs.

Bibliography

[1] M. Abramovici, C. Stroud, M. Emmert, “Designing SoCs for yield improvement: Using embedded FPGAs for SoC yield improvement,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 713-724, 2002.

[2] C. J. Alpert and A. Devgan, “Wire Segmenting for Improved buffer insertion,” in *Proc. ACM/IEEE Design Automation Conf.*, 1997, pp. 588–593

[3] C. J. Alpert, A. Devgan, and S. T. Quay, “Buffer Insertion for Noise and Delay Optimization,” *IEEE Trans. on CAD*, vol. 18, no. 11, pp. 1633–1645, 1999.

[4] C. J. Alpert, J.-H. Huang, and A. B. Kahng, “Multilevel circuit partitioning,” *IEEE Trans. on CAD*, vol. 17, no. 8, pp. 655–667, Aug. 1998.

[5] C. J. Alpert, J. H. Hu, S. S. Sapatnekar, and P. G. Villarrubia, “A practical methodology for early buffer and wire resource allocation,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 189–194, 2001.

[6] K. Arabi and B. Kaminska, “Oscillation-based test strategy for analog and mixed-signal integrated circuits,” in *Proc. IEEE VLSI Test Symp.*, pp. 476-482, 1996.

[7] H. B. Bakoglu, Circuit, Interconnections, and Packaging for VLSI, Addison-Wesley, 1990.

[8] S. Bose, P. Agrawal, and V. D. Agrawal, “A Rated-Clock Test Method for Path Delay Faults,” *IEEE Trans. Very Large Scale Integration Systems*, pp. 323-331, 1998.

[9] F. Caignet, S. D.-B. Dhia, and E. Sicard, “On the measurement of crosstalk in integrated circuits,” *IEEE Trans. Very Large Scale Integration Systems*, vol. 8, no. 5, pp. 606-609, Oct. 2000.

[10] F. Caignet, S. D.-B. Dhia, and E. Sicard, “The challenge of signal integrity in deep-submicrometer CMOS Technology,” *Proc. IEEE* Mobile Nachrichtensysteme (Mobile Communications Systems), vol. 89, no. 4, pp. 556-573, Apr. 2001.

[11] I. Catt, “Crosstalk (noise) in digital systems,” *IEEE Trans. Computer*, pp. 743-763, 1967.

[12] D. Chai, A. Kondratyev, Y. Ran, K. H. Tseng, Y. Watanabe, M. Marek-Sadowska, “Novel design methodologies and signal integrity: Temporofunctional crosstalk noise analysis,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 860-863, 2003.

[13] T. F. Chan, J. Cong, T. Kong, J. R. Shinnerl, “Multilevel optimization for large-scale circuit placement,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 171-176, Nov. 2000.

[14] K. J. Chang, N. H. Chang, S. Y. Oh, and K. Lee, “Parameterized SPICE subcircuits for multilevel interconnect modeling and simulation,” *IEEE Trans. Circuits and Systems*, vol. 39, pp. 779-789, Nov. 1992.

[15] Y.-W. Chang, K. Zhu and D. F. Wong, “Timing-driven routing for symmetrical-array based FPGAs,” *Trans. on Design Automation of Electronic Systems*, vol. 5, no. 3, pp. 433-450, July 2000.

[16] M. Chatterjee, D.-K. Pradhan, and W. Kunz, “LOT: Logic Optimization with Testability—New Transformations for Logic Synthesis”, *IEEE Trans. Computer-Aided Design*, vol. 17, no. 5, May 1998.

[17] W. Chen, S. K. Gupta, M. A. Breuer, “Analytic Model for Crosstalk Delay and Pulse Analysis under Non-Ideal Inputs,” in *Proc. Int’l Test Conf.*, pp. 809-818, 1997.

[18] L.-C. Chen, S.K. Gupta, and M.A. Breuer, “A new gate delay model for simultaneous switching and its applications,” in P*roc. Design Automation Conf.*, pp. 289-294, 2001.

[19] Y. Chen, A. B. Kahng, G. Robins, and A. Zelikovsky, “Practical Iterated Fill Synthesis for CMP Uniformity,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 671-674, 2000.

[20] W.-T. Chen, J.-L. Lewandowski, and E. Wu, “Optimal diagnostic methods for wiring interconnects,” *IEEE Trans. Computer-Aided Design*, vol. 11, no. 9, pp. 1161-1166, Sep. 1992.

[21] X.-T. Chen, F. J. Meyer, and F. Lombardi, “Structural diagnosis of interconnects by coloring,” *ACM Trans. Design Automation Electronic Systems*, vol. 3, no. 2, pp. 249-271, Apr. 1998.

[22] K.-T. Cheng and H.-C. Chen, “Classification and Identification of Nonrobust Untestable Path Delay Faults,” *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, pp. 845-853, 1996.

[23] W. Chen, S. Gupta, and M. A. Breuer, “Analytic models for crosstalk delay and pulse analysis under non-ideal in Puts,” in P*roc. Int’l Test Conf.*, pp. 809-818, 1997.

[24] K.-T. Cheng, S. Dey, M. Rdgers, and K. Roy, “Test challenges for deep sub-micron technologies”, In Proc. 37th Design Automation Conf., pp. 142-149, 2000.

[25] V.K.R. Chiluvuri, “Yield optimization in physical design: a review,” In Proc. of the Fifth ACM/SIGDA Physical Design Workshop, pages 198–206, 1996.

[26] Y.-H. Choi, T. Jung, “System-level self-diagnosis in sparsely interconnected systems,” *IEEE Trans. on Reliability*, vol.41, No. 3, Sept., pp. 433-439, 1992.

[27] J. Cong, J. Fang and Y. Zhang, “Multilevel approach to full-chip gridless routing,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 396-403, Nov. 2001.

[28] J. Cong, T. Kong and Z. D. Pan, “Buffer Block Planning for Interconnect-Driven Floorplanning,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 358-363, 1999.

[29] J. Cong, T. Kong, and D. Z. Pan,, “Buffer block planning for interconnect planning and prediction,” *IEEE Trans. Very Large Scale Integration Systems*, vol.9, no. 6, pp. 929–937, 2001.

[30] J. Cong, D.Z. Pan, P.V. Srinivas, “Improved Crosstalk Modeling for Noise Constrained interconnect Optimization”, in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, pp. 373--378, 2001.

[31] J. Cong, M. Xie and Y. Zhang, “An enhanced multilevel routing system,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 51-58, Nov. 2002.

[32] F. Corno, P. Prinetto, M. Rebaudengo, M. Sonza Reorda, “GATTO: a genetic algorithm for automatic test pattern generation for large synchronous sequential circuits,” *IEEE Trans. Computer-Aided Design*, vol. 15, no. 8, pp. 991-1000, Aug. 1996.

[33] Kenneth Crow, Design for Manufacturability, http://www.npd-solutions.com/dfm.html.

[34] M. Cuviello, S. Dey, X. Bai, Y. Zhao, “Fault modeling and simulation for crosstalk in system-on-chip interconnects”, in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 297-303,1999.

[35] F. DaSilva, Y. Zorian, L. Whetsel, K. Arabi, R. Kapur, “Overview of the IEEE P1500 standard,” in *Proc. IEEE Int’l Test Conf.*, pp. 988-997, 2003.

[36] S. Delmas-Bendhia, F. Caignet, E. Sicard and M. Roca, “On-chip sampling in CMOS integrated circuits,” *IEEE Transactions on Eletromagnetic Compatibility*, vol. 41, Issue 4, pp. 403-406, Nov. 1999.

[37] B. I. Dervisoglu, “A unified DFT architecture for use with IEEE 1149.1 and VSIA/IEEE P1500 compliant test access controllers,” in *Proc. ACM/IEEE Design Automation Conf.*, 2001.

[38] S. Devadas, A. R. Newton, “Decomposition and factorization of sequential finite state machines”, *IEEE Trans. Computer-Aided Design*, vol 8, no. 11, pp.1206-1217, 1989.

[39] A. Devgan, “Efficient coupled noise estimation for on-chip interconnects,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 147–151, 1997.

[40] F. F. Dragan, A. B. Kahng, I. Mandoiu, S. Mraddu, and A. Zelikovsky, “Provably good global buffering using an available buffer block plan,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 104–109, 2000.

[41] K.L. Einspahr, S.K. Mehta, and S.C. Seth, “Synthesis of sequential circuits with clock control to improve testability,” in P*roc. Asian Test Symp.*, pp. 472-477, 1998.

[42] L. Gal, “On-chip crosstalk–The new signal integrity challenge,” in *Proc. Custom Integrated Circuits Conf.*, pp. 12.1.1–12.1.4, 1995.

[43] M. Garey, D. Johnson, and H. Ho, “An application of graph coloring to printed circuit testing,” *IEEE Trans. Circuits and Systems*, vol. CAS-23, no. 10, pp. 591-599, Oct. 1976.

[44] M. R. Garey, and D. S. Johnson, Computer and Intractability: A Guide to the Theory of NP-Completeness, W. H. Freeman and Company, 1979.

[45] P. Goel and M.T. McMahon, “Electronic chip in place test,” in P*roc. Int’l Test Conf.*, pp. 83-90, 1982.

[46] I. Harris and R. Tessier, “Diagnosis of interconnect faults in cluster-based FPGA,” *in Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, 2000.

[47] M. Hayashi and S. Tsukiyama, “A hybrid hierarchical global router for multi-layer VLSI’s,” *IEICE Trans. Fundamentals*, Vol. E78-A, No. 3, pp. 337–344, 1995.

[48] D. Hightower, “A solution to line routing problems on the continuous plane,” in P*roc. DAW*, pp. 1-24, 1969.

[49] T.-Y. Ho, Y.-W. Chang, S.-J. Chen, and D.-T. Lee "A fast crosstalk- and performance-driven multilevel routing system," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 382-387, 2003.

[50] T.-Y. Ho, Y.-W. Chang, and S.-J. Chen, "Multilevel routing with antenna avoidance," in *Proc. ISPD-2004*, pp.34-40, 2004.

[51] L.-D Huang, D. F. Wong, “Optical Proximity Correction (OPC)-Friendly Maze Routing,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 812-817, Jun. 2003.

[52] G. Huertas, D. Vazquez, E.J. Peralias, A. Rueda, and J.L. Huertas, “Practical Oscillation-Based Test of Integrated Filters”, *IEEE Design & Test Computers*, vol. 19, no.6, pp. 64-72, 2002.

[53] IEEE P1500 Website, http://grouper.ieee.org/groups/1500/.

[54] S. Irajpour, S. Nazarian, L. Wang, S.K. Gupta, and M. A. Breuer, “Analyzing Crosstalk in the Presence of Weak Bridge Defects”, *in Proc. IEEE VLSI Test Symp*., pp.385-392, 2003.

[55] International Technology Roadmap for Semiconductors, 1999, 2001, 2003, http://www.itri.org/.

[56] H.-R. Jiang, Y.-W. Chang and J.-Y. Jou, ”Crosstalk-driven interconnect optimization by simultaneous gate and wire sizing,” *IEEE Trans. CAD*-19, (9), pp. 999–1010, 2000.

[57] A. B. Kahng, B. Liu, and I. I. Mandoiu, “Non-Tree Routing for Reliability and Yield Improvement,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 260-266, November, 2002.

[58] M. Kaneko and K. Sakaguchi, “Oscillation fault diagnosis for analog circuits based on boundary search with perturbation model,” in *Proc. IEEE Int’l Symp. Circuits and Systems*, pp. 93-96, 1994.

[59] G. Karypis, R. Aggarwal, V. Kumar, and S. shekhar, “Multilevel hypergraph partitioning: Application in VLSI domain,” *IEEE Trans. Very Large Scale Integration Systems*, Vol. 7, pp. 69–79, Mar. 1999.

[60] R. Kastner, E. Bozorgzadeh and M. Sarrafzadeh, “Predictable routing,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 110-114, 2000.

[61] W. K. Kautz, “Testing of faults in wiring interconnects,” *IEEE Trans. Computers*, vol. C-23, no. 4, pp. 358-363, Apr. 1974.

[62] Y. Kim, H.-D. Kim, and S. Kang, “A new maximal diagnosis algorithm for interconnect test,” *IEEE Trans. Very Large Scale Integration Systems*, vol. 12, no. 5, pp. 532-537, May 2004.

[63] A. Krstic, L.-C. Wang, K.-T. Cheng, J.-J. Liou, T. M. Mak, “Test and diagnosis for complex designs: Enhancing diagnosis resolution for delay defects based upon statistical timing and statistical fault models”, in *Proc. ACM/IEEE Design Automation Conf.,* 2003.

[64] H.-C. Lee, Y.-W. Chang, J.-M. Hsu, and H. Yang, “Multilevel large-scale module floorplanning/placement using B*-trees,” in *Proc. ACM/IEEE Design Automation Conf.*, pp. 812-817, Jun. 2003.

[65] K. T. Lee, C. Nordquist, and J. A. Abraham, “Automatic test pattern generation for crosstalk glitches in digital circuits,” in *Proc. VLSI Test Symp.* , pp. 34-39, 1998.

[66] B. N. Lee, L.-C. Wang, and M. S. Abadir, “Reducing pattern delay variations for screening frequency dependent defects,” in P*roc. VLSI Test Symp.*, pp. 153-160, May 2005.

[67] C. Lee, “An algorithm for path connection and its application,” *IRE Trans. Electronic Computer*, VEC-10, pp. 346-365, Sept.1961.

[68] C.C. Liaw, S.Y. Su, and Y.K. Malaiya, “Test generation for delay faults using stuck-at-fault test set,” in P*roc. IEEE Int’l Test Conf.*, pp. 167-175, 1980.

[69] J.-C. Lien and M. A. Breuer, “Maximal diagnosis for wiring networks,” in *Proc. Int’l Test C,* pp. 71-77, 1991.

[70] S.-P. Lin and Y.-W. Chang, “A novel framework for multilevel routing considering routability and performance,” in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, pp. 44-50, Nov. 2002.

[71] Y.-L. Lin, Y.-C. Hsu, and F.-S. Tsai, “Hybrid routing,” *IEEE Trans. Computer Aided-Design*, Vol. 9, No. 2, pp. 151–157, Feb. 1990.

[72] B.-S. Liu, C.-L. Lee, K.S.-M. Li, and J.-E Chen, "Crosstalk fault testing for SOC interconnection lines by using oscillation ring testing methodology," *IEEE International Test Synthesis Workshop* 2004.

[73] K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, “Noise-aware buffer planning for interconnect-driven floorplanning,” in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, pp. 423-426, Kitakyushu, Japan, Jan. 2003.

[74] K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, and C. -L Lee, “Noise-aware buffer planning for interconnect-driven floorplanning,” submitted to *IEE Proceedings Circuits, Devices, and Systems,* May 2005.

[75] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI,” in *Proc. IEEE Asia Test Symposium*, pp. 145-150, Nov. 2004.

[76] K.S.-M. Li i, C.-L. Lee, C. Su and J.-E Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI,” submitted to *IEEE Trans. Instrumentation and Measurement*, Nov. 2004.

[77] K.S.-M. Li, C.-L. Lee, C. Su, and J.-E Chen, “Oscillation ring based interconnect test scheme for SOC,” in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, pp. 184-187, Jan. 2005.

[78] K.S.-M. Li, C.-L. Lee, C. Su, and J.-E Chen, "Oscillation ring based interconnect test scheme for SOC," submitted to *Journal of Electronic Testing: Theory and Applications*, Jun. 2005.

[79] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, "IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults," to appear in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, 2006.

[80] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, "IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults," *IEEE Trans. Computer-Aided Design*, accepted on Nov. 2005.

[81] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, "Multilevel full-chip routing with testability and yield enhancement," in *Proc. ACM/IEEE System Level Interconnect Prediction*, pp. 29-36, Apr. 2005.

[82] K.S.-M. Li, C.-L. Lee, Y.-M. Chang, C. Su, and J.-E Chen, "Multilevel full-chip routing with testability and yield enhancement," submitted to *IEEE Trans. Computer-Aided Design*, under minor revision, June 2005.

[83] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, "Finite State Machine Synthesis for At-Speed Oscillation Testability," in *Proc. IEEE Asia Test Symposium*, pp. 360-365, Dec. 2005.

[84] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, "Finite State Machine Synthesis for At-Speed Oscillation Testability," submitted to *IEE Electronics Letters*, Aug. 2005.

[85] S. Majumder, B.B., Bhattacharya, V.D. Agrawal, and M.L. Bushnell, "A complete characterization of path delay faults through stuck-at faults," in *Proc. Int'l Conf. VLSI Design*, pp. 492-497, 1999.

[86] W. Maly, "Moore's Law and physical design of ICs," (special address), in P*roc. ISPD*, 1998.

[87] S. L. Manney, M. S. Nakhla, and Q. Zhang, "Analysis of non-uniform, frequency dependent high-speed interconnects using numerical inversion of Laplace transform," *IEEE Trans. CAD*, vol. 13, pp. 1513-1525, Dec. 1994.

[88] M. Marek-Sadowska, "Router planner for custom chip design," in *Proc. IEEE/ACM Int. Conf. Computer-Aided Design*, Nov. 1986.

[89] E.J. Marinissen, B. Vermeulen, H. Hollmann, R.G. Bennetts, "Minimizing pattern count for interconnect test under a ground bounce constraint," *IEEE Design &. Computers*, Vol. 20, No. 2, pp. 8-18, Mar-April, 2003.

[90] P. Maxwell, I. Hartanto, and L. Bentz, "Comparing functional and structural tests," in *Proc. IEEE Int'l Test Conf.*, pp. 400-407, 2000.

[91] S. Mitra, N. Seifert, M. Zhang, Q. Shi, K.S. Kim, "Robust system design with built-in soft-error resilience," *IEEE Computers*, Vol. 38, No. 2, pp. 43-52, Feb. 2005.

[92] F. Moll and A. Rubio, "Spurious signals in digital CMOS VLSI circuits: a propagation analysis," *IEEE Trans Circuits and Systems*, vol. 39, pp. 749-752, Oct. 1992.

[93] G. Nanz and L. E. Camilletti, “Modeling of chemical-mechanical polishing: a review,” *IEEE Trans. Semiconductor Manufacturing*, vol. 8, no. 4, pp. 382-389, 1995.

[94] P. Nigh, W. Needhan, K. Butler, P. Maxwell, and R. Aitken, “An experimental study comparing the relative effectiveness of functional, scan, IDDQ, and delay-fault testing,” in *Proc. IEEE VLSI Test Symp.*, pp. 459-464, 1997.

[95] M. Nourani, A. Attarha, “Build-in self-test for signal integrity,” in *Proc. ACM/IEEE Design Automation Conf.*, 2001.

[96] I. Pomeranz and S.M. Reddy, “On the use of fully specified initial states for testing of synchronous sequential circuits,” *IEEE Trans. Computers*, Vol. 49, No. 2, pp. 175-182, Feb. 2000.

[97] M. Roca, F. Moll, and A. Rubio, “Crosstalk effects between metal and polysilicon lines in CMOS integrated circuits,” *IEEE Trans. Electromagnetic Compatibility*, vol. 36, pp. 250-253, Aug. 1994.

[98] A. Rubio, N. Itazaki, X. Xu, and K. Kinoshita, “An approach to the analysis and detection of crosstalk faults in digital VSLI circuits,” IEEE Trans. CAD, vol. 13, pp. 387-394, Aug. 1994.

[99] P. B. Sabet, F. Ilponse, “A Model for Crosstalk Noise Evaluation in Deep Submicron Processes,” in Proc. Int’l Symposium on Quality Electronic Design, pp. 139-144, 2001.

[100] J. A. Sainz, M. Roca, R. Munoz, J. A. Maiz, and L. A. Aguado, “A crosstalk sensor implementation for measuring interferences in digital CMOS VLSI circuits,” in Proc. On-Line Testing Workshop, pp. 45-51, 2000.

[101] P. Sarkar, and C.K. Koh, “Routability-Driven Repeater Block Planning for Interconnect-Centric Floorplanning,” IEEE Trans. Computer-Aided Design, vol.20, no. 5, pp. 660–671, 2001.

[102] T. Sakurai, “Closed-form expressions for interconnection delay, coupling, and crosstalk in VLSI’s,” IEEE Trans. Electron Devices, vol. 40, no. 1, pp. 118-124, Jan. 1993.

[103] Z.M. Santo, F. Novak, and S. Macek, “Design of Oscillation-Based Test Structures of Active RC Filters,” In Proc. Circuits Devices System, vol. 147, no. 5, pp. 295-302, 2000.

[104] M. Schrader, R. McConnell, “SoC Design and Test Considerations,” in Proc. Design, Automation and Test in Europe Conference and Exhibition, (DATE), pp. 202 - 207, 2003.

[105] Semiconductor Industry Association, 1999, 2001, 2003, http://www.sia-online.org/, International Technology Roadmap for Semiconductors (ITRS), 2001, 2003.

[106] W. Shi and W. K. Fuchs, “Optimal interconnect diagnosis of wiring networks,” IEEE Trans. Very Large Scale Integration Systems, vol. 3, no. 3, pp. 430-436, September, 1995.

[107] B. Stine, “A Closed-Form Analytical Model for ILD Thickness Variation in CMP Processes,” in Proc. CMP-MIC, 1997.

[108] C. Su, Y.-T. Chen, M.-J. Huang, G.-N. Chen, and C.-L. Lee, “All digital built-in delay and crosstalk measurement for on-chip buses”, In Proc. Design, Automation and Test in Europe, pp. 527-531, Mar. 2000.

[109] C. Su and W. Tseng, “Configuration free SoC interconnect BIST methodology”, In Proc. Int’l Test Conf., pp. 1033-1038, 30 Oct.-1 Nov. 2001.

[110] X. Tang, and D. F. Wong, “Planning buffer locations by network flows,” in Proc. ACM/SIGDA Int’l Symp. on Physical Design, pp. 180–185, 2000.

[111] M. Tehranipour, N. Ahmed, M. Nourani, “Testing SoC Interconnects for signal integrity using boundary scan”, in Proc. IEEE VLSI Test Symp., 2003.

[112] N. Tendolkar, R. Raina, R. Woltenberg, X. Lin, B. Swanson, and G. Aldrich, “Novel techniques for achieving high at-speed transition fault test coverage for Motorola’s microprocessors based on PowerPCTM instruction set architecture,” in Proc. IEEE VLSI Test Symp., pp. 3-8, 2002.

[113] H.-C. Tsai, K.-T. Cheng, and V.D. Agrawal, “A testability metric for path delay faults and its application,” in Proc. Asia and South Pacific Design Automation Conf., pp. 593 – 598, 2000.

[114] V. Verma, S. Dutt, V. Suthar, “Efficient on-line testing of FPGAs with provable diagnosabilities,” in Proc. ACM/IEEE Design Automation Conf., pp. 498-503, 2004.

[115] T. Villa and A.-L. Sangiovanni-Vincentelli, “NOVA: State assignment of finite state machine of optimal two-level logic implementation,” IEEE Trans. Computer-Aided Design, vol. 9, no. 9, pp. 905-924, 1990.

[116] A. Vittal and M. Marek-Sadowska, “Crosstalk reduction for VLSI,” in Proc. IEEE/ACM Computer-Aided Design, vol. 16, pp. 290–298, 1997.

[117] P.T. Wagner, “Interconnect testing with boundary scan,” in Proc. IEEE Int’l Test Conf., pp. 52-57, 1987.

[118] Q. Wu, M.S. Hsiao; “Efficient sequential ATPG based on partitioned finite-state-machine traversal,” in Proc. Int’l test Conf., pp. 281-289, 2003.

[119] M.-S. Wu, C. L. Lee, C.P. Chang and J.E. Chen, “A Testing Scheme for Crosstalk Faults Based on the Oscillation Test Signal”, In Proc. IEEE Asian Test Symposium, pp. 170-175, 2002.

[120] W.-C. Wu, C.-L. Lee, M.-S. Wu, J.-E. Chen, and M. Abadir, “Oscillation Ring Delay Test for High Performance Microprocessor”, Journal of Electronic Testing: Theory and Applications, pp. 147-155, 2000.

[121] D. Xiang, Y. Xu, and H. Fijiwara, “Non-scan design for testability for synchronous sequential circuits based on conflict analysis,” in Proc. Int’l test Conf., pp. 520-529, 2000.

[122] C.-W. Yau and N. Jarwala, “A unified theory for designing optimal test generation and diagnosis algorithms for board interconnects,” in Proc. Int’l Test Conf.,, pp. 71-77, 1989.

[123] H. You and M. Soma, “Crosstalk and transient analysis of high-speed interconnects and packages,” IEEE Trans. Solid State Circuits, vol. 26, pp. 319-320, Mar. 1991.

[124] H. You and M. Soma, “Crosstalk analysis of interconnect lines and packages in high-speed integrated circuits,” IEEE Trans. Circuits and Systems, vol. 37, pp. 1019-1026, Aug. 1990.

[125] A. E. Zain and S. Chowdhury, “An analytical method for finding the maximum crosstalk in lossless-coupled transmission lines,” in Proc. Int’l Conf. CAD, pp. 443-448, 1992.

[126] Y. Zhao and S. Dey, “Analysis of interconnect crosstalk defect coverage of test sets,” in Proc. Int’l Test Conf., pp. 492-501, 2000.

Vita

Katherine Shu-Min Li was born in Taipei, Taiwan, the daughter of Yoan-Jian Lee and Ru-Huai Xue. She received the Bachelor of Science degree in Department of Computer Science of Rutgers University, New Jersey, U.S.A. and the master degree in Department of Computer Information Science of National Chiao Tung University, Hsinchu, Taiwan in 2000. She entered the Ph.D. program in Department of Electronics Engineering of National Chiao Tung Univserity in 2000, became a Candidate of Philosophy in February, 2001 and served as a Research and Teaching Assistant from 2000 to 2005.

Permanent address: 20F-3, No. 177, Shin-Jing Road, Taichung, Taiwan 40450.

Publication List

3 points

International Journal:

[1] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults,” submitted to *IEEE Trans. Computer-Aided Design*, accepted on Nov. 21, 2005.

[2] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “Multilevel full-chip routing with testability and yield enhancement,” submitted to *IEEE Trans. Computer-Aided Design*, under minor revision, June 2005.

[3] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI,” submitted to *IEEE Trans. Instrumentation and Measurement*, in Nov. 2004.

[4] K.S.-M. Li, C.-L. Lee, C. Su, and J.-E Chen, “Oscillation ring based interconnect test scheme for SOC,” submitted to *Journal of Electronic Testing: Theory and Applications*, in Jun. 2005.

[5] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “Oscillation Test for Synchronous Sequential Circuits,” submitted to *Design & Test of Computers*, in Oct. 2005.

[6] K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, and C. -L Lee, “Noise-aware buffer planning for interconnect-driven floorplanning,” submitted to *IEE Proceedings Circuits, Devices, and Systems*, under revision, May 2005.

International Conference:

[1] B.-S. Liu, C.-L. Lee, K.S.-M. Li, and J.-E Chen, "Crosstalk fault testing for SOC interconnection lines by using oscillation ring testing methodology," *IEEE International Test Synthesis Workshop* 2004.

[2] K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, “Noise-aware buffer planning for interconnect-driven floorplanning,” in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, pp. 423-426, Kitakyushu, Japan, Jan. 2003. (not in PhD study duration)

[3] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI,” in *Proc. IEEE Asia Test Symposium*, pp. 145-150, Nov. 2004.

[4] K.S.-M. Li, C.-L. Lee, C. Su, and J.-E Chen, “Oscillation ring based interconnect test scheme for SOC,” in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, pp. 184-187, Jan. 2005.

[5] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “IEEE Standard 1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults,” to be appeared in *Proc. ACM/IEEE Asia South Pacific Design Automation Conf.*, 2006.

[6] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “Multilevel full-chip routing with testability and yield enhancement,” in *Proc. ACM/IEEE System Level Interconnect Prediction*, pp. 29-36, Apr. 2005.

[7] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “Finite State Machine Synthesis for At-Speed Oscillation Testability,” in *Proc. IEEE Asia Test Symposium*, pp. 360-365, Dec. 2005.

Domestic Conference:

[1] K.S.-M. Li, Y.-H. Cherng, and Y.-W. Chang, “Noise-aware buffer planning for interconnect-driven floorplanning,” The 12th VLSI Design/CAD Symposium, Aug. 2001. (not in PhD study duration)

[2] K.S.-M. Li, C.-L. Lee, C. Su and J.-E Chen, “A unified approach to detecting crosstalk faults of interconnects in deep sub-micron VLSI,” The 15th VLSI Design/CAD Symposium, Aug. 2004.

[3] K.S.-M. Li, C.-L. Lee, C. Su, and J.-E Chen, “Oscillation ring based interconnect test scheme for SOC,” The 15th VLSI Design/CAD Symposium, Aug. 2004.

[4] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “P1500 Compatible Interconnect Diagnosis for Delay and Crosstalk Faults,” The 16th VLSI Design/CAD Symposium, Aug. 2005.

[5] K.S.-M. Li, C.-L. Lee, Y.-W. Chang, C. Su, and J.-E Chen, “Multilevel full-chip routing with testability and yield enhancement,” The 16th VLSI Design/CAD Symposium, Aug. 2005.