

**Design for Testability Techniques and Optimization  
Algorithms for Performance and Functional Testing of  
Multi-Chip Module Interconnections**

A Doctoral Dissertation

Presented to

The Academic Faculty

by

Rajesh Y. Pendurkar

In Partial Fulfillment

of the Requirements for the Degree

Doctor of Philosophy in Electrical & Computer Engineering

School of Electrical and Computer Engineering

Georgia Institute of Technology



March, 1999

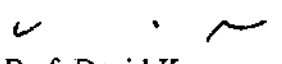
Copyright © 1999 by Rajesh Pendurkar

**Design for Testability Techniques and Optimization  
Algorithms for Performance and Functional Testing of  
Multi-Chip Module Interconnections**

Approved:

  
Prof. Abhijit Chatterjee

  
Prof. Joseph L. A. Hughes

  
Prof. David Keezer

Date Approved: 4/8/1999

**Dedicated to**

**mother Asha (Aai), lovely wife Shakalpi, brother Unmesh**

**&**

**the loving memory of my father Yashavant (Appa)**

## Acknowledgments

I wish to thank my thesis advisor, Dr. Abhijit Chatterjee, for his guidance, inspiration, encouragement and support throughout the course of my graduate studies at Georgia Tech. His knowledge and enthusiasm have provided me with a constant challenge in completing my dissertation. I appreciate his open attitude which enabled me to focus on my research activities with greater freedom and collaborate with Dr. Craig Tovey of School of Industrial and Systems Engineering and Dr. Yervant Zorian of Logic Vision, Inc. Craig has been a source of tremendous inspiration to me and I am indebted to him for giving me an opportunity to work with him on the most celebrated problem in mathematics namely the traveling salesman problem. Throughout my Ph.D. work, he was educational in the process of learning and creativity. I wish to express my gratitude to Yervant for his continuous encouragement and remote technical discussions that laid solid foundations of my thesis. I am grateful to Dr. David Keezer for his insightful suggestions and comments throughout my research. My thanks go to Prof. Joseph Hughes and Dr. Keezer for devoting their precious time to serve on my thesis reading committee.

The establishment of the Packaging Research Center at Georgia Institute of Technology has helped me fulfil my dream of a getting a doctoral degree. I thank Dr. Rao Tum-mala, the PRC director, for this opportunity. His views on technology and interdisciplinary education have profoundly influenced me. I also thank PRC students for electing me as a president of graduate student executive council for 1997, the critical year of NSF review.

My Ph.D. research was greatly influenced by my summer internship at Strategic CAD Laboratory, Intel Corporation, Oregon. I would like to thank Prof. P. Pal Chaudhuri, Ken Stevens and Shai Rotem for giving me an opportunity to work on an industrial design and get hands-on experience. I wish to thank researchers at GTE Laboratories, Massachusetts

who inspired me to pursue my Ph.D. degree. Thanks also go to professors Dr. Mitch Kokar and Dr. Mark Staknis of Northeastern University, Boston for their recommendations. I also wish to thank Dr. Sham Navathe of College of Computing at Georgia Tech for creating an opportunity for me to study at this great institution. It has been a very fruitful experience at Georgia Tech campus and I will always cherish it for the rest of my life. Last, I wish to express my gratitude towards all my teachers, especially Mr. M. Nadkarni, Mr. N. Parab and Mr. B. Abreo.

I have benefited greatly from interacting with the members of dynamic MIST research group and many colleagues and friends at Georgia Tech. I would like to thank Sasidhar, Bruce, Huy, Pramod, Pankaj, Ramki, Sudip, Sasi, Gomes, Sreemala and Vijay.

I have been really fortunate to have a talented younger brother Unmesh and I cherished his endless love throughout my student life. Without his initiative, my doctoral studies would be just a distant dream.

Of course, my dream would not be a reality without Shakalpi, my lovely wife, for her unconditional love and sacrifice. She has been consistently providing me the mental energy necessary to overcome obstacles and to achieve my goal. While renowned physicist Stephen Hawking was my abstract inspiration, my mother has been my concrete inspiration. Likewise, Dr. Sharad Kanetkar has been my abstract motivation and Shakalpi has been my concrete motivation. I am grateful for her understanding, patience and encouragement, and for delivering our wonderful son, Parth just a month before my doctoral defense.

# Contents

<b>Acknowledgments .....</b>	<b>iv</b>
<b>List of Tables.....</b>	<b>x</b>
<b>List of Figures.....</b>	<b>xi</b>
<b>Glossary of Terms .....</b>	<b>xiii</b>
<b>Abstract.....</b>	<b>xiv</b>
<b>1     Introduction.....</b>	<b>1</b>
1.1     Background.....	1
1.2     Economics of MCM Test.....	3
1.3     Problem Statement.....	5
1.4     Dissertation Overview.....	6
<b>2     Overview of MCM Interconnect Testing .....</b>	<b>7</b>
2.1     Interconnect Fault Models .....	7
2.1.1     Stuck-at fault Model .....	8
2.1.2     Bridging Fault Model.....	8
2.1.3     Delay Fault Model .....	9
2.2     Interconnect Issues For MCMs.....	9
2.2.1     Interconnect Delay.....	10
2.2.2     Interconnect Noise .....	11

2.3	Fault Detection and Diagnosis Algorithms for Interconnects.....	12
2.4	Substrate Test Techniques for MCMs.....	14
2.4.1	Flying Probe Testing.....	16
2.5	Structured Testability Techniques for MCMs.....	17
2.5.1	Boundary Scan Technique .....	17
2.5.2	Built-In self-test .....	19
2.6	Interconnect Testing for Yield and Throughput.....	20
<b>3</b>	<b>Single Probe Traversal Optimization .....</b>	<b>22</b>
3.1	Probe Traversal Optimization Problem.....	22
3.2	Definitions, Notation, Assumptions.....	24
3.3	Theoretical Bounds on Test Cost .....	26
3.4	Integer Programming Formulation.....	32
3.5	Complexity Analysis.....	34
3.6	Analysis for Practical MCM Substrate Testing .....	36
<b>4</b>	<b>Heuristic Algorithms for Single Probe Testing .....</b>	<b>39</b>
4.1	Heuristic for Efficient Traversal of Single Probe.....	39
4.2	Init_Tour Procedure .....	42
4.3	Insert Procedure .....	42
4.4	Shuffle Procedure.....	45
4.5	Experimental results.....	47
<b>5</b>	<b>Distributed BIST Technique for Performance Test. ....</b>	<b>54</b>
5.1	BIST Preliminaries.....	54
5.2	Proposed Approach.....	57
5.3	Interconnect Test Model and Assumptions.....	57
5.4	Distributed BIST Architecture.....	59
<b>6</b>	<b>Precharacterized Test Pattern Generator.....</b>	<b>63</b>
6.1	Proposed Approach.....	63

6.2	Precharacterized TPG Architecture .....	64
6.3	P-TPG Design Techniques .....	67
6.4	P-TPG Design Specifications .....	79
6.5	P-TPG Design Optimization .....	81
6.6	Activity Profile Generation Using P-TPG .....	82
6.7	Algorithm Description .....	83
<b>7</b>	<b>Distributed Diagnosis Using MISR Reconfiguration .....</b>	<b>84</b>
7.1	Proposed Approach .....	84
7.2	MISR Reconfiguration Technique .....	90
7.3	MISR Reconfiguration Under Single Fault .....	92
7.4	MISR reconfiguration Under Multiple Faults .....	95
7.5	Discussion .....	99
7.6	Experimental Results .....	102
7.7	Hardware Cost .....	104
<b>8</b>	<b>DFT Tool for MCM Interconnect Test .....</b>	<b>106</b>
8.1	Architectural Framework .....	106
8.2	Discussion .....	117
8.2.1	Interconnect Test Generation for Catastrophic Faults .....	117
8.2.2	Interconnect Test Generation for Performance Faults .....	117
8.2.3	Test Compatibility with Boundary Scan Standard .....	118
8.2.4	Interconnect BIST Overhead Analysis .....	118
8.2.5	Complexity Analysis of MatchProfile Algorithm .....	119
<b>9</b>	<b>Conclusions and Future Work .....</b>	<b>120</b>
9.1	Single Probe Traversal Optimization .....	120
9.1.1	Summary Of Contributions .....	120
9.1.2	Future Research .....	121
9.2	Precharacterized TPG Design .....	121
9.2.1	Summary Of Contributions .....	121



9.2.2	Future Research .....	122
9.3	MISR Reconfiguration .....	122
9.3.1	Summary Of Contributions .....	122
9.3.2	Future Research .....	123
<b>Bibliography .....</b>		<b>124</b>
<b>VITA .....</b>		<b>134</b>

## List of Tables

1.	Technology Trends.....	5
2.	Probe Route Cost Optimization With Different Heuristics.....	48
3.	Performance Comparison of Heuristics.....	50
4.	Effect of Variation of Number of Nets and Terminals on Probe Route Cost .....	50
5.	Results on MCM Benchmark.....	51
6.	Moving Probe Tester Specifications .....	52
7.	Comparison of Test Times: Single Probe Versus Double Probe Test .....	53
8.	Maximum Length Sequence for 4 bit external-XOR type LFSR .....	68
9.	Average of Total Number of Times P-TPG is in Given Transient State.....	76
10.	Simulation Results for Distributed BIST Diagnosis Using R-MISR.....	104
11.	Switching Profile Matching by Cascaded P-TPG Structure .....	110
12.	IUT Profile Matching for Floating Point Multiplier .....	111
13.	IUT Profile Matching for Cache-Processor Interface .....	112
14.	Improved Profile Matching for Cache-Processor Interface Using XOR.....	113
15.	Analytical Computation of Cost Function Using Genetic Algorithm and Markov Solver.....	114
16.	Comparison of Analytical Design and Simulated Design .....	115
17.	Switching Profile Matching of Multiplier IUT Using Analytical and Simulation Design.....	115
18.	Correlation Matrix Matching of Multiplier IUT Using Analytical and Simulation Design.....	116

## List of Figures

1.	Integrated MCM Production Test Flow .....	4
2.	Short and Open Defects in MCM Interconnections.....	9
3.	Interchip Signal Delay [12].....	10
4.	Multichip Module Substrate .....	14
5.	Boundary Scan standard compatible IC.....	18
6.	Top Layer Interconnections With Terminal Pads.....	23
7.	Comparison of SPTP and TSPTP .....	27
8.	SPTP for Unit Square.....	28
9.	Two Point Nets of Infinite Lengths .....	35
10.	An Example Substrate.....	36
11.	Insertion Procedure .....	44
12.	Shuffling of 5 Terminal Tour.....	45
13.	Snapshots of Algorithm .....	49
14.	Generalized BIST Architecture.....	55
15.	Distributed BIST Architecture.....	59
16.	Interconnect BIST Scheme .....	63
17.	Schematic of Internal-XOR LFSR.....	64
18.	Precharacterized TPG Component.....	66
19.	Two LFSR Configurations .....	67
20.	Markov Model of P-TPG Component .....	70

21.	Markov Chain Model of 2-bit P-TPG .....	74
22.	Transition Matrix of 2-bit P-TPG .....	75
23.	Two Weight Circuit Configurations for Increasing Switching Profile .....	77
24.	Types of Switching Profiles .....	80
25.	DFT Architecture for Interconnect BIST .....	81
26.	Cascaded P-TPG Structure and IUT Switching Profile .....	82
27.	Dynamic Reconfiguration of LFSR .....	85
28.	MCM Interconnection Diagnosis Flowchart .....	86
29.	Diagnosis Using R-MISR Reconfiguration .....	91
30.	Logical MISR .....	95
31.	Physical MISR .....	96
32.	Non-optimal MISR Reconfiguration at IC Level .....	98
33.	Diagnosis Using Logical R-MISR v/s Physical R-MISR .....	101
34.	Example MCM Under Test .....	103
35.	Graph Model of MCM Under Test .....	103
36.	Software Architecture of DFT Tool .....	107
37.	Structure of P-TPG Synthesis Tool .....	108
38.	Plots of Matched Switching Profiles .....	112

## **Glossary of Terms**

<b>SPTP</b>	<b>Single Probe Traversal Problem</b>
<b>IUT</b>	<b>Interconnect Under Test</b>
<b>LFSM</b>	<b>Linear Finite State Machine</b>
<b>LFSR</b>	<b>Linear Feedback Shift Register</b>
<b>MISR</b>	<b>Multiple Input Signature Register</b>
<b>MUT</b>	<b>MCM Under Test</b>
<b>ORA</b>	<b>Output Response Analyzer</b>
<b>P-TPG</b>	<b>Precharacterized Test Pattern Generator</b>
<b>R-MISR</b>	<b>Reconfigurable MISR</b>
<b>TPG</b>	<b>Test Pattern Generator</b>

## **Abstract**

The objective of this research is to drive down the cost of functional testing of multi-chip module (MCM) interconnections before assembly of ICs using single probe test technique and to devise a formal design for testability (DFT) strategy for MCM interconnect performance test and diagnosis after assembly. Testing of MCM interconnections has become an activity of critical importance and considerable difficulty in the MCM design and test process in view of the demand for high performance and high density of packaging. First, electrical testing of a bare MCM substrate interconnections using a single probe is considered. A tight bound on single probe testing time is computed using rigorous theoretical analysis. Efficient and practical heuristic algorithm for finding an efficient probe route, to optimize the total test time of an MCM substrate, is presented. Experiments on the benchmark MCM netlist show that a up to 40% reduction in single test probe traversal time can be achieved by using the proposed algorithm. Secondly, a formal DFT methodology for comprehensive performance testing of assembled MCM interconnections is presented. A novel distributed BIST architecture is proposed to test and diagnose key performance issues such as the effects of cross-talk, ground bounce and simultaneous switching noise. The technique consists of specialized and reconfigurable on-chip precharacterized test pattern generators and multiple input signature registers. It is proved that interconnections switching activities can be effectively recreated and an accurate distributed diagnosis can be performed with low area overhead. The algorithms developed in this research are integrated into a CAD tool to automate MCM interconnections test flow.

# **CHAPTER I**

## **INTRODUCTION**

The purpose of this research is to develop a theoretical framework for comprehensive and efficient testing of multi-chip module (MCM) interconnections. This framework allows an efficient functional testing of MCM substrate interconnections before assembly using a single probe test technique. It also enables a comprehensive performance testing of assembled MCM interconnections after assembly using built-in self-test (BIST) techniques. This framework has been validated by software and hardware prototypes.

In this introductory chapter, the motivation and background behind this research are given. This is followed by a discussion on the economics of MCM testing. A high level description of the problem addressed in this research is given. Finally, the organization of the dissertation is outlined.

### **1.1 Background**

Multi-chip module (MCM) technology is making its mark in high-volume markets such as consumer and automobile electronics, as well as in mission-critical applications

such as avionics and communications [1]. By the end of this century, the worldwide MCM market is expected to reach \$2.3 billion out of which \$750 million is for smart cards and rest is for MCM-L, MCM-C and MCM-D [2]. The market in 2002 will reach \$3.3 billion. This increasing global trend of MCM applications is accompanied by increasing challenges resulting from shrinking dimensions of transistors, larger ICs, large I/O count resulting from high integration density and higher clock speeds. Some of the challenges are to increase the yield of MCM assembly, reduce the high cost of MCM manufacture and test, enhance the reliability of the assembled MCM and to overcome the limitations of high-cost external automatic test equipment (ATE).

The MCM technology involves the interconnection and assembly of bare dies on a substrate through metal interconnections. To achieve high volume, cost competitive MCMs, one must start with high yielding substrates and known good die. Due to the high cost of the semiconductor ICs involved, a testing scheme is necessary to ensure the functional integrity and performance of all the package interconnection paths. This promotes the reliability and quality of the end product. The test process consists of two steps. Initial testing of the interconnection paths prior to attachment of the active devices (ICs) must be performed and subsequently the assembled package must be tested. The testing of interconnections prior to die attachment is called *substrate testing* and is used to guarantee a defect-free substrate. The purpose of substrate testing is to test each interconnect for opens and/or shorts defects, thus assuring its functional integrity. This type of testing is essential because assembled MCM costs are high and repair of a substrate interconnect network is not always possible. A summary of electronic test methods for MCMs appears

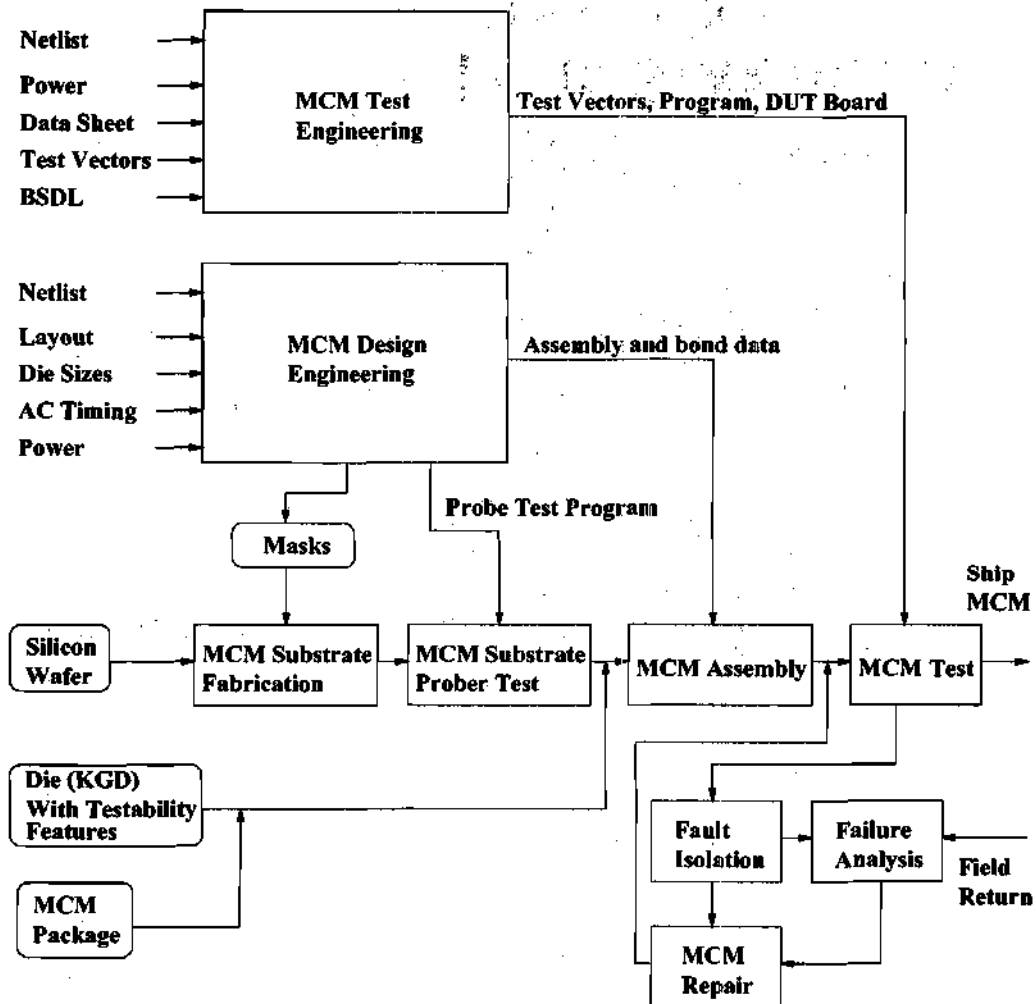


in [3]. *Functional /Performance testing* is a follow-up step to the substrate testing and represents the testing of the package after die attachment. This testing step has two basic goals: 1) provide high quality detection of static and dynamic faults, in order to meet the expected defect rate at the module level, and 2) provide a high quality diagnosis of bad ICs and interconnects to allow potential repair at the module level. The module level testing of MCMs is described in [4].

The testing for dynamic faults is called *performance testing* which verifies that all the ICs on an MCM substrate can properly communicate with each other. Figure 1 shows the integrated MCM test flow. The detailed description of types of MCM test related activities is given in [5].

## **1.2 Economics of MCM Test**

Improvements in the design and test process are mandatory in view of the fact that economic factors, such as reducing the final product cost, are becoming increasingly important. The profitability of high performance MCMs can only be achieved by improving two major components of the global cost namely minimum time to market and maximum end-of-product yield [6]. Therefore, the reduction in the test time for a single substrate has a significant impact over a large production run in terms of minimizing the production testing time. Maximizing the yield directly relates to the test strategies incorporated in the production flow. As a result of this increasing emphasis on testing, it is being considered a part of the design process.



**FIGURE 1. Integrated MCM Production Test Flow**

Test is becoming one of the parameters of a system design and employing design for testability strategies early in the design cycle effectively reduces test cost of the final product,

which is estimated to be about 30% of total production cost for MCMs [7]. Table 1 shows the typical technology and cost trends [8].

**TABLE 1. Technology Trends**

	1970	1980	1990	2000
Complexity	SSI	LSI	ULSI	MCM
Gate Count	10	5k	200k	2000k
Memories	256	16k	16 Mb	10 Gb
Transistors	$10^2$	$10^5$	$10^9$	$10^{14}$
Speed (Hz)	100k	10M	100M	500M
Pins	14	44	356	1000
Test/Total Cost%	5	20	60	60

### 1.3 Problem Statement

Testing MCM substrate interconnections is necessary to assure the reliability and quality of the assembled MCM. Single probe test techniques are becoming increasingly popular for connectivity verification using high frequency test stimulus as well as for the capacitance testing of substrates. The objective of this research is two-fold. First, efficient traversal algorithms to optimize the total distance traveled by a single test probe on an MCM substrate are presented. These heuristic algorithms reduce the substrate testing time and associated test cost. Second, a novel design for testability(DFT) technique to address the issue of performance testing of MCM interconnections is developed. This issue is becoming very important because of the fact that hitherto "second-order" effects, such as ground bounce, crosstalk, and switching noise, play dominant roles in current design methods. Such effects are dominant causes of a low MCM yield. Built-in self-test techniques are devised to address these issues and enhance MCM testability.

## **1.4 Dissertation Overview**

Chapter 2 discusses the interconnect fault models and algorithms used currently for detection and diagnosis of interconnect related faults. A brief overview of structured DFT techniques for MCMs is given. An integer programming formulation of the single probe traversal optimization problem and a theoretical model for lower bound of a single probe test cost is described in Chapter 3. In Chapter 4, a heuristic algorithm designed for an efficient routing of a single test probe is discussed. A novel distributed BIST technique for performance testing of MCM interconnections is proposed in Chapter 5. Chapter 6 discusses in detail the test pattern generation part of the proposed distributed interconnect BIST technique. The distributed BIST diagnosis for performance related interconnect faults is described in Chapter 7. Chapter 8 provides an overview of a CAD tool and a hardware prototype developed based on this research. Chapter 9 summarizes the technical contributions and suggests directions for future research.

# **CHAPTER II**

## **OVERVIEW OF MCM**

### **INTERCONNECT TESTING**

In this chapter, a brief review of the interconnect fault models and the algorithms which are currently used by the test community is given. A general description of MCM substrate interconnections test techniques used in industry is given. Current design for testability (DFT) techniques, including a BIST concept, are discussed. This builds a foundation for the research presented in this dissertation.

#### **2.1 Interconnect Fault Models**

The fault models of interest are based on the likely failures observed in MCM interconnections. The stuck-at fault model and the bridging fault model [9] are widely used in today's test and diagnosis activity for MCM interconnections. The delay fault model is becoming increasingly important in view of emerging deep submicron applications where

the interconnect delays are dominating the gate delays. In the following subsections, each fault model is briefly described.

### **2.1.1 Stuck-at fault Model**

The stuck-at fault model, often referred to as a *classical* fault model [10], assumes that a fault in the interconnect results in logic value on that interconnect being fixed to either a logic 0 (stuck-at-0) or a logic 1 (stuck-at-1). This represents the most common types of failures, for example, short-circuit to ground plane or a power plane, in many technologies. A short results if not enough metal is removed by the photolithography, where as overremoval of metal results in an open circuit which is usually distinguished from a stuck-at fault.

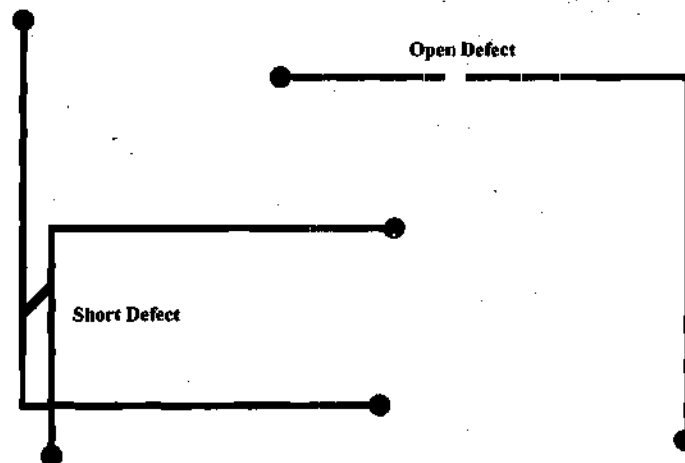
### **2.1.2 Bridging Fault Model**

With the increasing density of ICs that are mounted on MCM substrate, the probability of shorts between two or more interconnects has been significantly increased. Unintended shorts between the interconnects form a class of permanent faults, known as *bridging faults*, which can not be modeled as stuck-at faults. There are two types of bridging faults: AND-bridging fault and OR-bridging fault [10]. In AND type of bridging fault, logic 0 dominates, while in OR type of bridging fault, logic 1 dominates.

### 2.1.3 Delay Fault Model

The MCM interconnections are associated with resistance and capacitance which govern the RC-delay of that interconnect. The failure of the interconnect to meet its timing specification without alteration of its low-frequency functional behavior results in the delay fault. The defects such as near-opens and near-shorts have high probability of occurrence due to the statistical variations in the manufacturing process. These defects result in the violation of its timing specifications.

Examples of typical interconnect defects are shown in Figure 2.



**FIGURE 2. Short and Open Defects in MCM Interconnections**

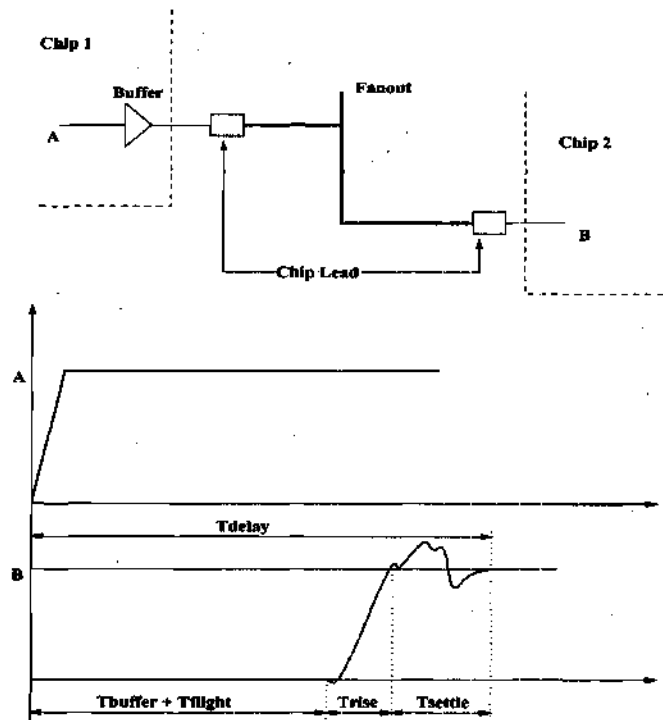
## 2.2 Interconnect Issues For MCMs

Interconnect refers to the medium used to connect any two or more circuit elements. Interconnections include the package pins, lead frames, bonding wires, TAB frames, sol-

der bumps, metal layers inside chips, and wires on MCM substrates. Interconnections differ widely in their electrical performance. Typical interconnect models use lumped resistances and capacitances (R-C model) or distributed resistances and capacitances (Transmission line model) [11]. Two major interconnect performance factors are described below.

### 2.2.1 Interconnect Delay

The total time taken for a signal to travel from chip 1 to chip 2 is given by the sum of the buffer delay, time of flight across interchip interconnection, rise time of the received signal and the settling time of the signal as shown in Figure 3 .



**FIGURE 3. Interchip Signal Delay [12]**



The interconnect capacitance does not scale proportionately with device scaling. As more and more gates are integrated, die size increase and on-chip interconnect capacitance increases. Similarly, interchip interconnection capacitance, which is proportional to the wire length, increases as more and more chips are integrated on a single MCM. As a result of finer line widths and spacings, the resistance of the MCM interconnect becomes comparable to that of a driver circuit and causes increased time of flight for the signal.

### **2.2.2 Interconnect Noise**

The density of MCM substrates is higher compared to typical printed wiring boards. Therefore, capacitive and inductive effects of the interconnects are also nonnegligible and lead to following types of noise [13].

*Reflection noise:* This is caused by unmatched loading and improper terminations of transmission lines of finite lengths. The reflections occur at discontinuities in the transmission lines, such as fanout branches, and can be minimized by interconnect design under controlled impedance environment.

*Crosstalk noise:* The mutual capacitance and inductance between neighboring electrical signal paths cause unwanted coupling between a active line (an interconnect with signal voltage switching from logic level 0 to logic level 1 or vice versa) and a passive line (an interconnect with no signal voltage switching). The detailed analysis and exact formulas for crosstalk noise voltages can be found in [11]. The methods for minimizing crosstalk noise are given in [14].

*Power Distribution noise (Ground Bounce):* Off-chip drivers used to drive package interconnections of an MCM have high drive strength. These high switching currents and transient currents flow through the power supply pin, substrate interconnections and substrate to package bonding and package parasitics. All these components have inductances which lead to voltage drops equal to  $L_{eff}(dI/dt)$  when these drivers switch, where  $L_{eff}$  is an effective inductance between the leads and  $dI/dt$  is the rate of change of current through the component leads. If many of the drivers switch simultaneously, the voltage may bounce below acceptable threshold, resulting in false logic level. This noise is referred to as simultaneous switching noise or ground bounce. Because of large number of drivers and the addition of MCM substrate interconnections, this type of noise becomes a critically important factor for the design and test of an MCM.

The combined effect of delay and noise contributes to performance degradation and is a major cause of lower MCM yield.

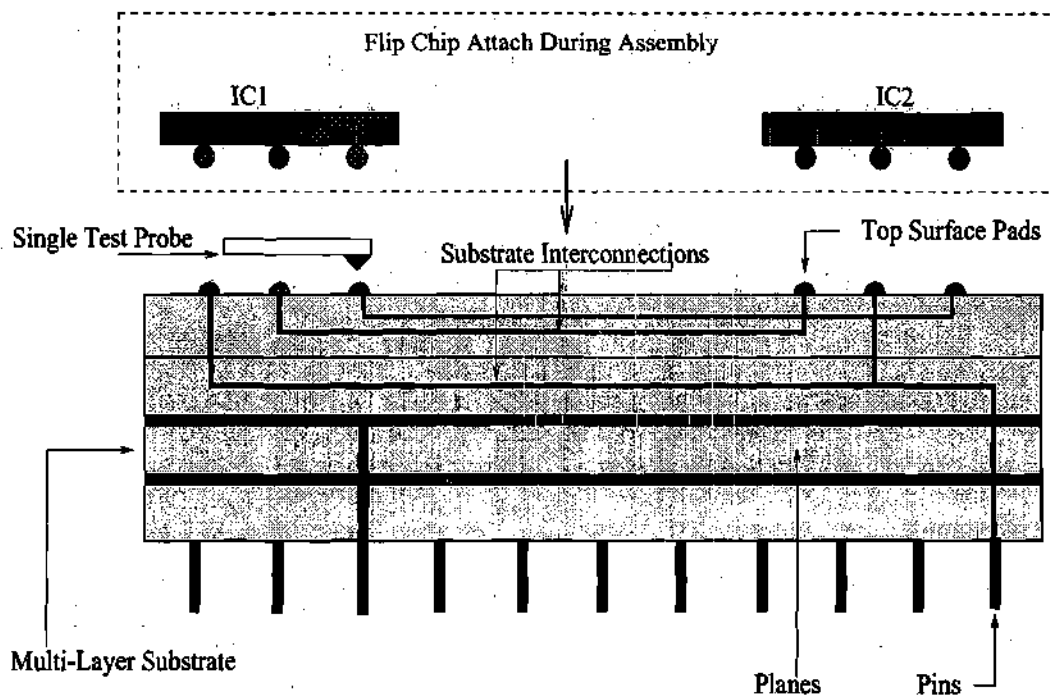
### **2.3 Fault Detection and Diagnosis Algorithms for Interconnects**

The earliest work reported in this area was by Kautz [15]. He showed that  $\lceil \log_2 N \rceil$  parallel test vectors (PTV) are optimal for detecting all shorts in a network of  $N$  unconnected terminals. This method involved applying a unique sequential test vector (STV) to each net and could be generated by a simple counting sequence. This was extended to  $\lceil \log_2(N+2) \rceil$  PTVs by Goel and McMahon [16] to enable detection of all possible stuck-at faults and all shorts. The modified counting sequence suggested allows stuck-at fault test-

ing in addition to detection of a short between any pair of nets. It is also extended to implement a two step diagnosis. Wagner [17] has proposed a diagnosis scheme which requires  $2\lceil \log_2(N+2) \rceil$  vectors. In this scheme,  $\lceil \log_2 N \rceil$  vectors used for shorts and stuck-at faults detection together with its complementary set forms the complete set of vectors for multiple fault diagnosis. A detection and diagnosis scheme which provides a minimal test vector set using the structure independent algorithm has been proposed in [18]. It also discusses a walking sequence scheme for the detection and diagnosis of shorts and stuck-at-faults. Jarwala and Yau [19] have discussed a theoretical framework for analyzing test generation and diagnosis algorithms for wiring interconnects. They define two types of diagnostic resolution. The first identifies, without ambiguity, a list of nets that have a fault. The second type further identifies the sets of nets affected by the same type of short, the nets that are stuck at zero or one, or the nets that are open. There are two types of test and diagnostic techniques. The first is *One Step Test and Diagnosis* where a set of test patterns are applied and the response is analyzed for fault detection and diagnosis. The second type is *Adaptive Test and Diagnosis* where the test is applied, response analyzed, and then one or more additional tests are applied depending on the response to aid diagnostics. The need for minimizing the test size while maintaining its diagnostic resolution is discussed in detail in [20]. It provides generalized, optimal or near-optimal algorithms for interconnect testing, providing the flexibility to trade-off test compactness and diagnostic accuracy.

## 2.4 Substrate Test Techniques for MCMs

Typical MCM substrates contain a large number of interconnections (nets) connecting different terminals (pads) on the topmost substrate layer. The ICs are attached to the substrate after it has been thoroughly tested for shorts and opens of its interconnections. Figure 4 shows the multilayer MCM substrate, buried interconnections, and top layer surface pads required to bond the ICs using a flip-chip attach technology.



**FIGURE 4. Multichip Module Substrate**

The typical interconnection testing process consists of application of the test stimulus to the interconnection under test through a probe which physically touches the top sub-

strate pads. The probe also receives the test response which can be evaluated by a tester [21] to decide whether the interconnection is faulty or fault-free. A moving probe tester physically moves the probe to another substrate pad to test another interconnection. This probe movement significantly impacts the total substrate test time since a typical MCM substrate consists of a large number of interconnections.

As the complexity of multi-chip module increases, the interconnect requirements will continue rising. A wiring density (total wire length per unit area) as well as the substrate size will keep increasing [22]. Before costly ICs are mounted on the package substrate, the substrate themselves need to be verified for electrical integrity and probed for shorts and opens. The current techniques used by the industry are: flying probes for resistance, capacitance [23] and combined capacitance/resistance testing; voltage contrast electron beam [24]; electrical module test (EMT) and time domain network analysis (TDNA) [25]. The key elements that differentiate these test methods are equipment cost, test time, throughput (number of tests per second) and defect resolution and a detailed comparison can be found in [25].

Electron beam technology [24] measures the voltage present at each node of an interconnection network using an electron beam, after injecting a charge in the network at one node. Each node of the network should show the charge voltage present, verifying the electrical continuity. It is a highly accurate and high throughput testing technique and there is no mechanical contact with the substrate, but high cost and longer setup times limit its usefulness. Also, this technique is ideally suitable for shorts testing. TDNA has

high test application time and high equipment cost. Although EMT has the capability of detecting near-opens, it can not detect near-shorts. Membrane probe cards [26] can detect open, short and high resistance faults. However, it is difficult to apply them to test cofired ceramic MCM substrates. Although parallel test vectors could be applied to the inputs, as would occur during the normal operation, the clock speed during probe card testing ( $TCK < 6$  MHz) is usually much slower than that during normal operation [27]. A massively parallel test of large area substrate is presented in [28] as a low cost, high throughput method for substrate test. This approach uses digital signatures and claims an improved sensitivity to near failures such as resistive opens.

#### **2.4.1 Flying Probe Testing**

Flying probe testers employ two or more single needle probes which are mounted on the mechanical systems moved by linear motors, positioning each probe at a programmed position and contacting the substrate at a bond pad for connection to an IC. By moving the probes to each bond pad in an interconnect network, this type of tester verifies the continuity of each interconnect network in a substrate. After a continuity check for the network, one of the probes may be moved to a ground connection, and then to the power connections(s), to check for isolation from ground and power planes.

Capacitance probing involves a single probe to contact each bond pad on the surface of the substrate and measure the capacitance with respect to the ground reference plane. This system is mechanically less complicated than the multiple flying probe systems. In addition, as the number of interconnection pads increases to even a moderate degree, this sin-

gle probe technique becomes essential. This is because the number of two-probe measurements required to fully test a substrate for net to net shorts grows as the square of the number of pads while the number of single probe tests only grows linearly [23].

Recently, a new technique has been developed to detect interconnect faults by measuring the attenuation of the test stimulus applied through a tuned load using a single probe [21]. This technique reduces the total hardware cost of the test equipment and has a higher diagnostic resolution. It also has the potential to be faster and, therefore, cheaper compared with the multiple probe techniques. It also has the additional advantage of being able to detect latent defects such as near-open and near-shorts.

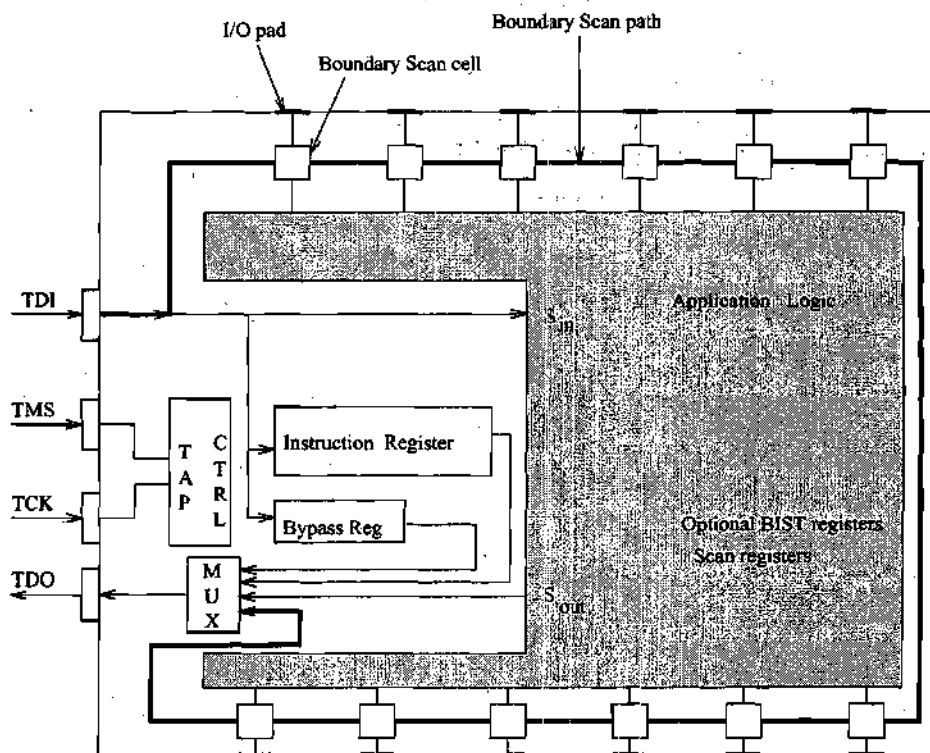
## **2.5 Structured Testability Techniques for MCMs**

Structured testability techniques improve controllability and observability of a circuit under design by making it more testable. The requirement of these techniques stems from two major problems faced by the MCM technology. First, the MCM pinouts allow limited access to internal nodes (isolation problem) and, second, the ICs mounted on an MCM do not undergo full functional testing after packaging (incomplete die testing problem) [29].

### **2.5.1 Boundary Scan Technique**

A boundary scan technique, which is standardized by IEEE 1149.1-1990, is a collection of design rules applied principally at the IC level [Figure 5] that allows test software to address these problems in a structured way. The architecture permits associating memory cells with each input and output pin of every chip so that known signals may be sent

across interconnections and captured for observation. It provides a single test access port (TAP) to each chip, through which all the test related instructions and data can be transferred. It is possible to execute an on-chip BIST using the "RUNBIST" instruction and test electrical interconnections functionally by transmitting data from a boundary scan cell (BSC) of one IC to the boundary scan cell of its neighboring IC using the "EXTEST" instruction. The application of the boundary scan technique [20] has opened up possibilities to access internal device test structures, such as internal scan, even though the device is now loaded on a board. This has caused an increased interest in adding BIST structures to devices and then re-using those structures at board or system level and particularly for field service.



**FIGURE 5. Boundary Scan standard compatible IC**



### 2.5.2 Built-In self-test

Built-in self-test (BIST) [30][31] represents one of the embedded test (Design-For-Test) technologies that has finally been accepted by industry. Internal scan, boundary scan, level sensitive scan design (LSSD), random access scan, scan path, scan-set logic are some of the structured DFT techniques used in industry[32]. Built-in logic block observation(BILBO), syndrome testing, testing by verifying Walsh coefficients and autonomous testing are some of the self testing technologies[32]. Internal scan, was created to overcome fundamental implementation problems of Automatic Test-Pattern Generation based on the 1959 stuck-at model and path-sensitizing theory[10]. The motivation behind boundary scan, was to provide a virtual bed-of-nails to access boards containing surface-mount devices and MCMs. Typically, BIST is used at a device level to solve a variety of device-level test problems such as lack of direct pin access (especially for multiple instances of embedded RAMs), ability to carry out tests "at-speed", improved diagnostics and use for device burn-in. One of the problems encountered in the past was that BIST was seen to be an extra investment, i.e., once it had been used and the device had passed the tests, it would not be used any further. This makes an economic justification of BIST difficult. Boundary scan has changed this view. Access to the BIST resources through boundary scan means that the BIST can now be re-run at all stages of the product life cycle, particularly for the system-level diagnostic purposes such as MCM diagnostics [33].

## 2.6 Interconnect Testing for Yield and Throughput

Important issues of ATE capabilities, test/repair environment and board yield are discussed in [19]. Traditionally, manufacturers use two techniques to test boards: In-circuit test and functional test. The onset of deep sub-micron (now currently eluded to as nanometer) technology is changing the way chips are being designed and manufactured. New problems are arising that are driving design automation to integrate all the tools that are needed to successfully take a design from concept to reality. Test is one part of this process that is getting significant attention. An area once classified as a "back end" process in the design flow is moving closer to the "front end". Design methodologies are incorporating test related structures in the beginning of the design cycle. Manufacturability of the complex designs caused by the excess silicon available is a significant issue. The following is a quote from a leading technology magazine EE Times [34].

"But the industry has hit the wall in the quarter-micron generation. There are serious, serious problems with yields at quarter-micron..... The power signal-integrity and metal-migration challenges are exceeding the capability of the tools. Instead of 80% yields, crosstalk is cutting yields for quarter-micron designs down to 30 or 40 percent at some companies."

As yield becomes more of a problem, test and testability issues will get more focus. We are already observing a trend towards full scan designs. Deep sub-micron is challenging test in a number of areas. Capacity, performance, Iddq, pinouts, cores, power, and heat are issues that are stressing testing. The SIA roadmap [35] is predicting problems in the

area of test. The roadmaps focus areas for test are: Testability, Testing and Testers, and Known Good Die. The roadmap sees "integrated circuit testing to be a major cost factor and must be done in prepackaged and packaged form, including future multi-chip packaging. Design productivity and performance improvement, cost-effective tools covering design verification from system design down to physical transistor, and interconnect design at clock frequencies greater than 1 GHz and 0.5 volts power supplies is a major challenge."

# **CHAPTER III**

## **SINGLE PROBE TRAVERSAL**

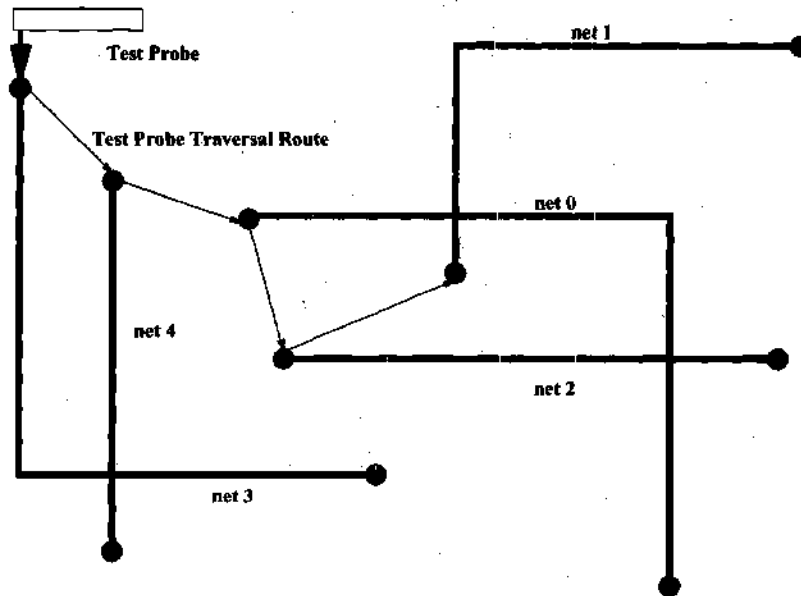
### **OPTIMIZATION**

In this chapter, a problem of finding an optimal route for a single probe for testing an MCM substrate, is formulated. An integer programming formulation of this optimization problem is given. A theoretical framework to analyze this problem is developed and a tight bound on the single probe traversal cost is computed.

#### **3.1 Probe Traversal Optimization Problem**

An example of MCM interconnects, each having two terminals, as viewed from the top layer of an MCM substrate is shown in Figure 6. Each net may connect more than two terminals on the topmost substrate layer and is tested by probing only one of the terminals on the layer. The probe then traverses to a terminal of another net on the substrate and the test procedure is repeated. It is desired to find the optimal route for the probe so that the

total distance traveled, and thereby total test cost, is minimized. The generalized problem can be stated as follows.



**FIGURE 6. Top Layer Interconnections With Terminal Pads**

**Generalized  $k$ -probe traversal problem:** A  $k$ -probe tester contacts  $k$  terminal pads and simultaneously verifies the connectivity among  $\binom{k}{2}$  pairs of the  $k$  terminal pads. Given the netlist of a finite number of nets with fixed multiple terminals, find efficient and collision-free routes for the  $k$ -probes assuming that the probes move simultaneously. An algorithm for finding efficient routes is given in [36].

**Single probe traversal problem:** Given the netlist of a finite number of nets with fixed multiple terminals, identify the sequence of terminals of each net (one per net) such that the total traversal cost of the probe is minimized, thereby reducing the overall test time while ensuring that every net is probed once.

### 3.2 Definitions, Notation, Assumptions

For the sake of theoretical proof, it is assumed that the MCM substrate is a unit square. A net is a set of points on the substrate. Distances between points are either Euclidean (also called  $L_2$  norm) or the maximum of the absolute differences in x or y coordinates (called sup or  $L_\infty$  norm). Some machines used in MCM substrate testing can move probes simultaneously and independently in the x and y directions. The sup norm distances apply to this type of movement. Since the great majority of nets have 2 nodes, we focus on these cases. We analyze a substrate containing  $n_i$  nets of  $i$  nodes,  $i = 2$  with  $n = n_2$ . Extensions could be easily carried out for more nodes per net but the results would be qualitatively the same.

The SPTP, or single probe traversal problem, is to visit one point from each net, so as to minimize the total distance traveled. Note that SPTP requires deciding which point from each net is to be visited, and in what sequence to visit the nets.

The two probe traversal problem, DPTP, involves a test procedure that uses a pair of probes. A probe is placed at each of two nodes of the net to test connectivity between the nodes. One such test is required for a 2-node net; depending on the internal wiring, either one or two such tests may be required for a 3-node net. After a pair of nodes is tested, the probes can move simultaneously to new locations for the next test. The effective travel distance between tests is therefore the maximum of the two distances that must be traveled by the probes. In DPTP one must decide on the sequence of tests, and decide which probe

will travel to which node, to minimize the total effective distance traveled. One efficient technique to find such a conflict-free route of a two probe tester is described in [37]. A 2-probe tester for substrate testing utilizing a bandsort algorithm to determine the route of one probe is discussed in [38]. In an optimal solution, it may be that two tests for the same net do not occur consecutively.

The SPTP is similar to the traveling salesman problem (TSP) which can be defined conceptually as follows: Given a set of  $N$  terminals, determine the shortest complete circuit that connects all terminals, so that every terminal is visited exactly once. There are various modifications and extensions to the basic TSP definition. These include the multiple traveling salesman problem [39], the vehicle routing problem [39], the capacitated arc routing problem [40] and the multiple tour maximum collection problem [41]. The SPTP can be considered as a multiple choice TSP since a probe can contact any one of the terminals of the selected interconnect. Thus it has an added complexity that *the set of terminals to be visited is not predetermined*.

It is well known [39] that for every instance of TSP, the optimal solution has length  $\leq 2\sqrt{n}$ . This is for the Euclidean metric, so it also holds for the sup norm metric. This result scales: for example if the points are in a square of side length  $1/2$ , the optimal solution has length  $\leq \sqrt{n}$ .

For the average case analysis, it is assumed that the location of every point in every net is identically independently distributed uniformly on the substrate. Under this distributional assumption, the Euclidean TSP is known to have expected optimal value tending to

$\alpha\sqrt{n} + o(\sqrt{n})$ . For the Euclidean norm,  $0.3 \leq \alpha \leq 2$  and the constant  $\alpha$  has been experimentally determined to have value approximately 0.71. This result scales just as the deterministic result discussed previously. Some of the results will involve the constant  $\alpha$ .

Let  $N$  denote the set of  $n$  net locations, which comprise an instance of SPTP or DPTP.  $N$  implicitly gives value of  $n_2$ . Let  $SPTP(N)$  (respectively  $DPTP(N)$ ) denote the value of the optimum solution of SPTP (respectively DPTP) on the instance  $N$ . When the instance  $N$  is random,  $E[SPTP(N)]$  denotes the expected value of  $SPTP(N)$ . For random  $N$ , the values  $n_2$  are treated as parameters rather than as data following some probabilistic distribution.

### 3.3 Theoretical Bounds on Test Cost

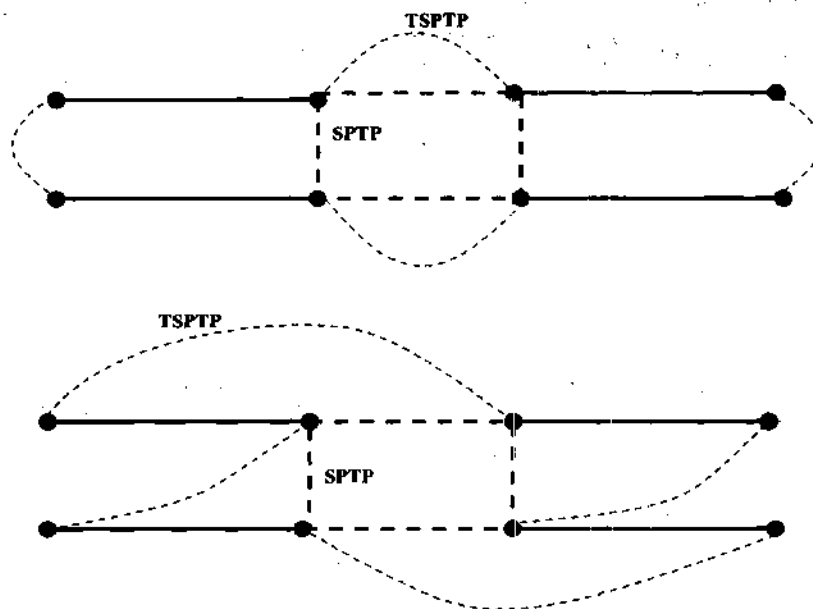
SPTP solution consists of  $n$  nodes (1 node per net) in sequenced tour. This problem can be related to the classical TSP by seeking an optimal solution to the related TSP instance on  $2n$  nodes without any option of visiting one node per net. The optimal solution for tsp tour of all  $2n$  nodes in the instance of  $N$  is denoted by  $TSPTP(N)$ .

**Theorem 1:**  $SPTP(N) \leq TSPTP(N)$

**Proof:** The triangle inequality for an euclidean TSP is used. By triangle inequality, the shortest distance between any pair of cities is the direct route. For an instance of TSP over  $2n$  nodes, all nodes of all nets are to be visited. There are  $2n$  edges in the tour as shown in Figure 7. For the instance of SPTP, however there are only  $n$  edges in the tour. A step by



step solution to SPTP from the TSP tour over  $2n$  nodes is constructed, starting from an arbitrary node say  $i$ . The next node  $j$  of the tsp tour is checked. If  $j$  belongs to the net different from that of node  $i$ , it is required to check whether the net to which node  $j$  belongs has been already visited. If not, the edge  $i-j$  in subtour leading to step by step construction of complete SPTP tour is preserved. Otherwise  $j$  is skipped and the next node  $k$  which belongs to the unvisited net in TSP tour, is considered. A new edge  $i-k$  is added to construct a subtour as a part of SPTP tour. Obviously by triangle inequality, length of the edge  $i-k$  is less than the path  $i-j-k$  in the TSP tour. Thus for any arbitrary new edge  $l-m$  added to construct SPTP tour, length edge  $(l-m)$  is always less than path  $(l-m)$  through intermediate nodes. Thus the optimal SPTP tour length can be no longer than optimal TSP tour length over  $2n$  nodes. Q.E.D.

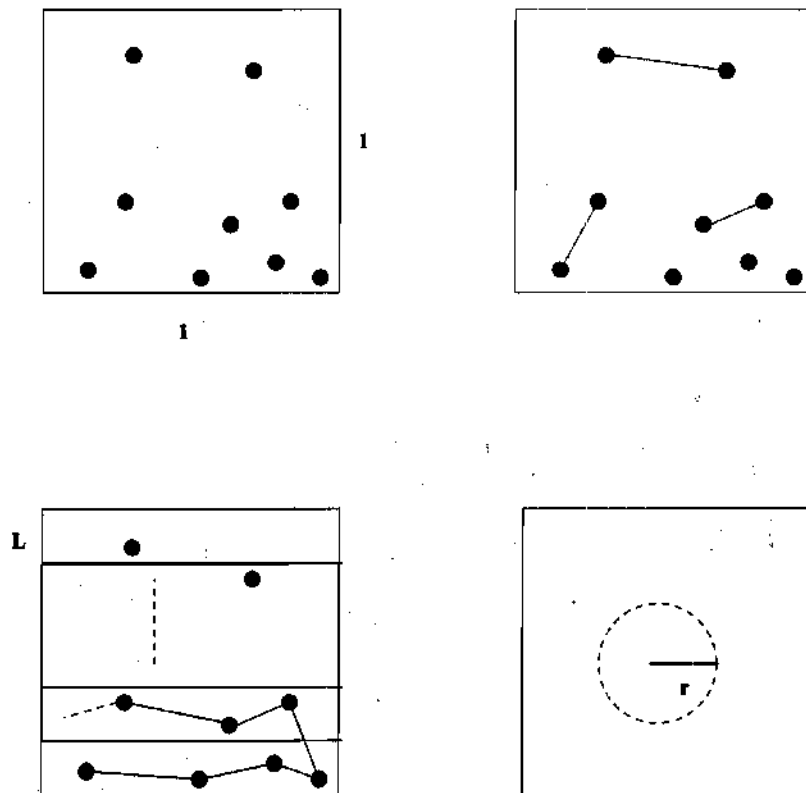


**FIGURE 7. Comparison of SPTP and TSPTP**

It is always interesting to find out the performance guarantee of the heuristic to solve computationally hard problem such as SPTP. In the following section we provide rigorous mathematical analysis to find out the lower bound on SPTP instance.

**Theorem 2:** Expected value of  $SPTP(N)$  is  $\theta(\sqrt{n})$

**Proof:** For the purpose of the proof, we restrict our attention in which all the nodes of the given instance of SPTP lie on the MCM substrate which is square of unit width as shown in Figure 8.



**FIGURE 8. SPTP for Unit Square**

**Theoretical Analysis:** We assume that all nodes of  $n$  nets on the substrate are uniformly distributed within the square. For the case of 2 nodes per net ( $n = n_2$ ), we randomly choose any 2 points from the given distribution of nodes and construct one net. Thus, we construct  $n$  2-point nets out of given nodes in the unit square substrate. We compute theoretical bound on  $SPTP(N)$  in terms of general value of  $n$  as follows.

**Upper Bound Analysis:** We divide the unit square substrate into horizontal stripes  $S_1, S_2, \dots, S_{1/L}$ . Clearly total number of stripes are  $1/L$ . The probe traverses within odd numbered stripes from left to right and then visits nodes in the even numbered stripes from right to left, finally returning to the initial node from the last node of the final strip completely testing the unit square substrate. The worst case length of SPTP tour under the above constraints is  $SPTP(N) \leq \sqrt{2} + 1/L + 2 + nL$ . The first component represents the diagonal jump to starting position. Second component is the number of bands of the square constituting to length due to X direction moment. Third component is  $2L \times 1/L$  where  $2L$  denotes worst case moment in Y direction from one band to the other. The last component denotes the Y moment within the band. This bound from above is smallest for  $L = 1/(\sqrt{n})$ . Thus the  $SPTP(N) < \sqrt{2} + 2 + 2\sqrt{n}$ . This proves that  $SPTP(N)$  is bounded from above by  $\sqrt{n}$  as constant factors are ignored.

**Lower Bound Analysis:** Now we give rigorous proof for bound from below. This is estimated with probabilistic approach due to obvious bound of zero. We define  $d_{min}$  as the minimum distance between two nodes each belonging to separate nets. To bound  $SPTP(N)$  from below, we must find expected value of sum of  $d_{min}$  over all nets ( $n = n_2$  for

2-point nets) in the unit square substrate. For this purpose we introduce probabilistic approach as shown in figure. The probability of random node falling in the circle of radius  $r$  is  $\pi r^2$ . Let  $d_{ij}$  be the distance from node  $i$  to node  $j$ . Let  $r = 1/(c\sqrt{n})$  for some constant  $c$ .

Let  $E[\text{Min}(d_{ij}) | i \neq j] = E[D(i)]$ ,  $D(i)$  are i.i.d. (identical and independent distribution) in the unit square. By definition of expectation, we have

$$E[D(i)] = \text{Prob}[D(i) \leq r](E[D(i) | D(i) \leq r]) + \text{Prob}[D(i) > r](E[D(i) | D(i) > r]) \quad (1)$$

First term is bounded by 0. We bound the second term using probability theory as follows.

$$D(i) > r = \bigcap_{i \neq j} (d_{ij} > r) \text{ for all } i;$$

$$\text{Prob}[D(i) > r] \geq 1 - \text{prob}\left[\overline{\bigcap_{i \neq j} (d_{ij} > r)}\right] \text{ since } p(A) = 1 - p(\bar{A})$$

By De'Morgan's Law, complement of intersection is union of the complement, hence for independent events, we have

$$\text{Prob}[D(i) > r] \geq 1 - \sum_{i=1}^n \text{prob}[d_{ij} \leq r], \text{ where } N \text{ is number of nets.}$$

$$\text{Prob}[D(i) > r] \geq 1 - \sum_{i=1}^n (\pi r^2) = 1 - n(\pi r^2) \quad (2)$$

Since  $E[D(i) | D(i) > r]$  is bounded by  $Nr$ , equation (1) becomes

$$E[D(i)] \geq 0 + [1 - n(\pi r^2)](nr) = nr - n^2 r^3 = (n)(1/(c\sqrt{n})) - (n^2)(1/(c^3 n^{3/2})) = c_1 \sqrt{n} + c_2 \sqrt{n}$$

This proves that  $SPTP(N)$  is bounded from below by  $\sqrt{n}$  (ignoring constants) since  $SPTP(N)$  has to be at least equal to or greater than expected value of  $D(i)$  over all  $i$  where  $i$  is node of probed net.

Since for some constants  $C_1$  and  $C_2$ , from upper bound and lower bound analysis, we have

$$c_1 \sqrt{n} \leq SPTP(N) \leq c_2 \sqrt{n}$$

$SPTP(N) = \theta(\sqrt{n})$ . Q.E.D.

**Lemma 1:** Given  $\rho\sqrt{2n}$  and  $\alpha\sqrt{n}$  as bounds on expected value of  $TSPTP(N)$  and  $SPTP(N)$  respectively,  $\alpha < \rho\sqrt{2}$ .

**Proof:** Obvious from upper bound analysis of theorem 2.

**Theorem 3:** Let  $\rho\sqrt{2n}$  be the expected value of  $TSPTP(S)$  and Let  $\alpha\sqrt{n}$  be expected value of  $SPTP(S)$ . Then  $\rho\sqrt{2n} \leq 2\alpha\sqrt{n}$

**Proof:** Consider  $TSPTP(N)$  and  $SPTP(N)$  for nets with 2 nodes each. The average saving per edge in constructing solution  $SPTP(N)$  from  $TSPTP(N)$  is  $(\rho\sqrt{2n})/(2n)$ . Since we remove  $n$  edges to construct solution  $SPTP(N)$ , bound on expected value of  $SPTP(N)$  can

be obtained easily as  $\rho\sqrt{2n} - n(\rho\sqrt{2n})/(2n) = (\rho\sqrt{2n})/2$ . Since SPTP( $N$ ) must be at least  $((\rho\sqrt{2n})/2)$ , we have  $((\rho\sqrt{2n})/2) \leq \alpha\sqrt{n}$ . Hence  $\rho\sqrt{2n} \leq 2\alpha\sqrt{n}$ . Q.E.D.

The worst-case and average case behavior of SPTP and DPTP has been analyzed rigorously in [42] and it is shown that for substrates with 2 to 3 terminal pads in each of  $n$  nets, the expected travel time for a single test probe is shorter by a factor of order  $n^{1/4}$ .

### 3.4 Integer Programming Formulation

The mathematical formulation of the single probe routing problem corresponding to the problem statement in Section 3.1 is described in [43] and restated below for completeness.

An iteration  $t$  refers to the movements made by the single probe in the  $t^{th}$  step of the probe sequence, with  $t = 1$  corresponding to the terminal from which the probe tour is started.

In this formulation the variables are defined as follows:

$Z_{ijt} = 1$  , if terminal  $j$  of interconnect  $i$  is probed during iteration  $t$

$Z_{ijt} = 0$  , otherwise

$X_{ijkl} = 1$  , if probe moves from terminal  $j$  of interconnect  $i$  to terminal  $l$  of  
interconnect  $k$

$X_{ijkl} = 0$  , otherwise

and the problem data are given by

$n_i$  = Number of terminals of interconnect  $i$

$N$  = Number of interconnects of given substrate

$C_{ijkl}$  = Cost of moving from terminal  $j$  of interconnect  $i$  to terminal  $l$  of

interconnect  $k$

$$\text{minimize} \quad \sum_{ijkl} C_{ijkl} X_{ijkl} \quad (1)$$

subject to

$$\sum_{t=1}^N \sum_{j=1}^{n_i} Z_{ijt} = 1, \forall i \quad (2)$$

$$\sum_j \sum_i Z_{ijt} = 1, (t = 1, 2, \dots, N) \quad (3)$$

$$X_{ijkl} \geq Z_{ijt} + Z_{kl(t+1)} - 1, (t = 1, 2, \dots, (N-1)) \quad (4)$$

All the variables are binary. Equation 1 states that the total cost of traveling from a terminal of an net to the terminal of the next net is to be minimized over all the terminals.

Equation 2 restricts each single probe to contact each net at one and only one terminal.

Equation 3 ensures that two nets can not be probed at the same time. Equation 4 links auxiliary variable  $X$  to main variable  $Z$ .

This formulation gives an exact solution technique for solving very small instances of

SPTP. However since SPTP is similar to the TSP (a well-known NP complete problem), SPTP must also be NP-Hard. We prove this in the following section.

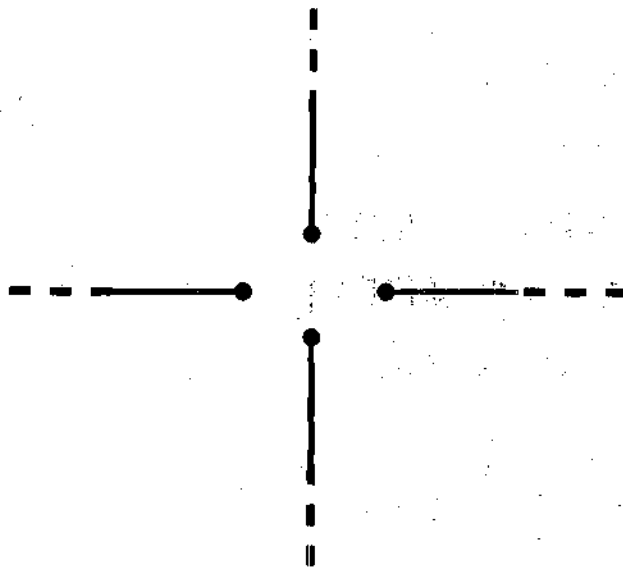
### 3.5 Complexity Analysis

Following theorem proves that SPTP and DPTP are computationally hard to solve.

**Theorem 4:** SPTP and DPTP are NP-Hard.

**Proof:** We assume that we have a deterministic polynomial time algorithm SPTP\_Solver to solve SPTP. Assume SPTP instance has  $N$  nets of 2 nodes each. By applying SPTP\_Solver, we get optimal tour of  $N$  nodes (one from each net). We shall use SPTP\_Solver to solve related euclidean TSP. Let  $X_1, X_2, \dots, X_n$  be an instance of TSP where each  $X_i$  is a node selected uniformly within a square unit of width. Let  $X_{n+i} = X_i$  for  $i = 1..n$ . Let an instance of SPTP be defined as net  $i \{X_i, X_{n+i}\}$  where  $i = 1..n$ . We give above stated TSP instance to the SPTP\_Solver. Obviously since for each net, both nodes are in the same location, an optimal solution to the instance of SPTP yields an optimal solution to the instance of TSP. It follows therefore that we can use SPTP\_Solver to solve TSP on  $N$  nodes. Alternatively if we consider each net of infinite length as shown in the Figure 9, we have a polynomial time algorithm for  $N$  node optimization version of TSP. But since optimization version of TSP has the same claim to intractability as any NP-Complete problem, there is no polynomial time algorithm to solve decision version TSP optimally unless  $P = NP$ . It follows consequently that we can not have deterministic polynomial time algorithm to solve SPTP unless  $P = NP$ . Q.E.D.





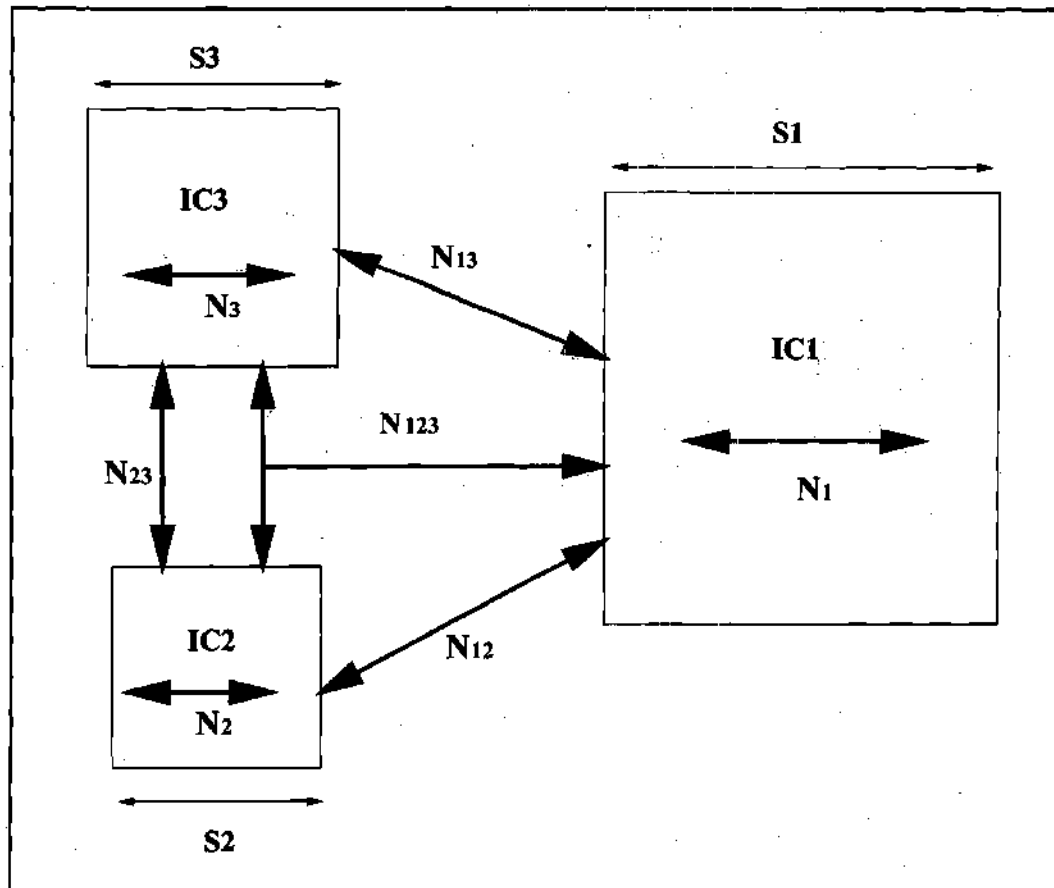
**FIGURE 9. Two Point Nets of Infinite Lengths**

For 2PTP, again consider instances with 2 nodes per net, where the first node of each net is located in the small square  $[0, 2]^2$ . Let  $\epsilon$  denote the least distance between first nodes. Place all the second nodes in a circle of diameter  $\epsilon$  centered at  $\{0.8, 0.8\}$ . Then one probe will visit all the first nodes, and its travel times will dominate the second probe's times. Solving 2PTP on these instances is equivalent to solving the TSP instance defined by the first nodes.

**Q.E.D.**

### 3.6 Analysis for Practical MCM Substrate Testing

Application of the theoretical analysis to optimize test cost for connectivity verification of MCM Substrates is discussed in detail in this section. Typical MCM substrate is shown in Figure 10.



**FIGURE 10. An Example Substrate**

Our example substrate consists of three chips namely, IC1, IC2 and IC3, representing a typical microprocessor, a peripheral controller IC and a memory chip such as DRAM. We denote respective sizes of these chips by  $S_1$ ,  $S_2$  and  $S_3$ . Let  $N_i$  denote number of nets

completely within Chip  $i$ , Let  $N_{ij}$  denote number of nets between chips  $i$  and  $j$  and  $N_{ijk}$  denote number of nets that connect all three chips. where  $i, j$  belongs to  $\{1, 2, 3\}$ . Obviously  $N_{ij} = N_{ji}$ . We claim that the expected value of SPTP tour cost for net integrity test for a chip mounted on MCM substrate is proportional to the size of the chip. For our model we have all the chips with square footprints. Thus without loss of generality we can use the theorem 2 developed on unit square substrate to calculate bound on SPTP tour cost for a particular chip. Total SPTP tour cost for testing net integrity of chip  $i$  of size  $S_i$  with  $N_i$  nets is thus bounded by the product  $S_i \times \sqrt{N_i}$ . Total MCM substrate test cost can be modeled in many different ways depending on the choice of nodes selected for probing. For our example substrate, two options are described out of many different options available.

Option 1: Single probe testing is started at IC1. Testing of nets  $N_{123}$ ,  $N_{12}$ ,  $N_{13}$  and  $N_1$  is performed by probing the nodes terminating at IC1. Then the probe is moved to IC2 to test the nets  $N_2$  and finally the probe traverses to the IC3 and the nets  $N_{23}$  and  $N_3$  are tested. Then the probe traversal cost for the example substrate of is bounded by  $S_1 \times \sqrt{N_{123} + N_1 + N_{12} + N_{13}} + S_2 \times \sqrt{N_2} + S_3 \times \sqrt{N_3 + N_{23}}$ .

Option 2: Single probe testing of the example substrate can also be done by first probing nets  $N_1$  at IC1. Then the probe is moved to IC2 and nets  $N_{12}$  and  $N_2$  are tested by probing nodes terminating at IC2. Finally the probe traverses to IC3 and nets  $N_{123}$ ,  $N_{13}$ ,  $N_{23}$  and

$N_3$  are tested to complete the substrate test. With this option, the overall probe traversal cost for the example substrate of is bounded by

$$S_1 \times \sqrt{N_1} + S_2 \times \sqrt{N_2 + N_{12}} + S_3 \times \sqrt{N_3 + N_{23} + N_{13} + N_{123}}.$$

It is clear that these bounds give a way to select the best option from the physical netlist, so that substrate test time using the single probe can be optimized. The analysis shows that the traversal cost depends on the footprint sizes of the constituent ICs of an MCM as well as the die level netlist of the substrate which defines the interchip interconnections. For a substrate with an IC having maximum percentage of connectivity with other ICs, testing all the nets belonging to that IC at the location of that IC itself, might reduce the overall test time for single probe technique. Thus, the computation of theoretical bounds on the test cost proves to be effective tool in the CAD environment, especially for bigger MCM substrates.

# **CHAPTER IV**

## **HEURISTIC ALGORITHMS FOR**

### **SINGLE PROBE TESTING**

In this chapter, two practical heuristic algorithms to optimize traversal cost of a single probe for testing MCM substrate interconnects, are discussed in detail. Experimental results confirm the validity of these algorithms. The comparison of test times using single and double probe technique on a benchmark MCM netlist is presented.

#### **4.1 Heuristic for Efficient Traversal of Single Probe**

Since SPTP is NP-hard, solving even a moderate size problem with the mathematical formulation is apt to be impractical. Considering the large number of interconnections on an MCM substrate any attempt at using an exact solution procedure will prove to be less practical. In view of the complexity of the problem, a heuristic procedure to solve the SPTP is proposed and is presented in the following sections. Below, some important definitions related to the problem are given.

- A net represents a set of interconnected layers of substrate corresponding to a single electrical node. Each net has more than one terminal on the topmost layer of the substrate and is identified uniquely by a net identifier. A netlist is a list of all nets with their associated net identifiers and terminals each net connects.
- A terminal is a pad on the topmost layer of substrate that is probed during substrate testing. Each terminal belongs to only one net. The terminals are used to provide electrical connections to the ICs that are bonded to the substrate during MCM assembly.
- A terminal tour is the sequence of terminals to be probed. It consists of one terminal from each net in the netlist.
- A subtour is defined to be a sequence of net terminals connected by edges (probe traversal paths), but not necessarily including a terminal from each net.
- A net is unprobed if none of its terminals is in the current subtour.
- The cost  $C(i, j)$  of an edge of a subtour is the Euclidean distance between the two terminals  $i$  and  $j$  on which that edge is incident.
- The tour cost is the sum of costs of all the edges in the terminal tour.

A heuristic procedure for efficient single probe tour construction is described below. This procedure takes MCM netlist and Insert\_Mode as an input parameters along with other optional parameters such as Shuffle\_Mode and Improve\_Flag. The Insert\_Mode selects either an arbitrary insertion or a farthest insertion heuristic explained in Section 4.3. The Shuffle\_Mode activates or deactivates the shuffle procedure used for further optimization of the tour cost. Improve\_Flag is of type boolean which, when TRUE,

enables improved version of Insert and Shuffle procedures explained in Section 4.3 and Section 4.4 respectively.

**Procedure Construct\_Terminal\_Tour**

**Input :** Die level netlist of an MCM substrate, Insert\_Mode, Shuffle\_Mode, Improve\_Flag

**Output :** terminal tour to be probed using a single probe and corresponding tour cost.

**Construct\_Terminal\_Tour {**

    subtour = Init\_Tour(Insert\_Mode);

    While ( (there are unprobed nets) and (subtour is NOT equal to a terminal tour)) {

        if (Insert\_Mode == Arbitrary\_Insertion)

            Select an unprobed net randomly;

        else /\* Insert\_Mode == Farthest\_Insertion \*/

            Select an unprobed net with maximum distance to the current subtour;

        Insert (subtour, selected unprobed net, Improve\_Flag);

    } /\* end while \*/

    if (Shuffle\_Mode is activated)

        Shuffle(terminal tour, Improve\_Flag);

    Compute terminal tour cost.

    Return terminal tour;

**}**

## 4.2 Init\_Tour Procedure

This procedure initializes the sequence of terminals to be probed by selecting the first terminal of the tour. The method of initialization depends on the Insert\_Mode.

```
Init_Tour (Insert_Mode){  
    if (Insert_Mode == Arbitrary_Insertion) {  
        Randomly select a terminal of the arbitrarily selected net and initialize the terminal tour;  
        Find the nearest neighbor of the start terminal and construct a subtour;  
    }  
    else { /* Insert_Mode == Farthest_Insertion */  
        For every possible pair of nets  $S$  and  $T$ , compute  
        the distance  $D(S,T)$  = minimum over all nodes  $s$  in  $S$ , nodes  $t$  in  $T$ ;  
        Select a pair of nets which are farthest from each other i.e. maximize  $D(S,T)$  to construct  
        a subtour;  
    }  
    return subtour;  
}
```

## 4.3 Insert Procedure

This procedure is based on the using either the arbitrary insertion or the farthest insertion heuristic to construct a TSP tour. For arbitrary insertion, an unprobed net is selected randomly. For farthest insertion, a distance of the net to a subtour is defined as the mini-

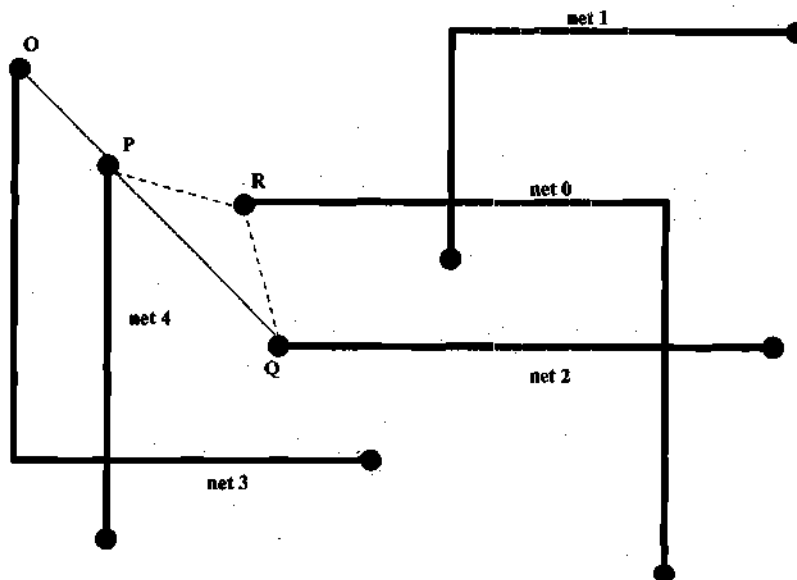


imum of the distances of every node of the net from every member of the current subtour.

For example suppose the current subtour has nodes  $a, b, c$ . Suppose net  $S$  contains nodes  $s1$  and  $s2$ . Then the distance from net  $S$  to the subtour is given by

$$\min(d(a,s1),d(b,s1),d(c,s1),d(a,s2),d(b,s2),d(c,s2))$$

For this step, the unprobed net with maximum distance to the subtour is selected [44]. For each terminal  $r$  of the selected unprobed net using either of the insertion heuristic described above, the edge  $pq$  of the subtour is found such that  $C_{pr} + C_{rq} - C_{pq}$  is minimized [45]. The terminal  $r$  is inserted between  $p$  and  $q$ . As an example in Figure 11, the current subtour consists of the terminals  $O, P$ , and  $Q$  of net 3, net 4, and net 2, respectively. Terminal  $R$  is to be inserted between  $P$  and  $Q$  to form a new subtour  $O-P-R-Q$  and compute the corresponding subtour cost. The *Improved Insertion* procedure consists of selecting the best terminal of the net with the best subtour cost and inserting that terminal and form a new subtour. This improvement procedure is repeated till an optimized terminal tour is constructed.



**FIGURE 11. Insertion Procedure**

The pseudo code for improved insertion is given below.

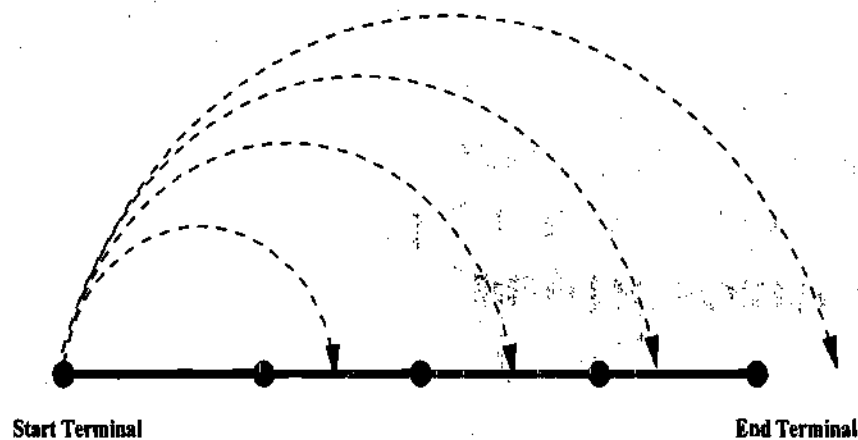
```

Insert (subtour, selected unprobed net, Improve_Flag) {
    if (Improve_Flag is TRUE) { /* improved insert */
        While (cost computations of all the terminals of the selected net are not complete) {
            Select a terminal of the net.
            Find minimum edge cost for insertion and insert the terminal in the current subtour.
            Find the new subtour cost.
            Select the terminal with minimum cost and insert in the current subtour.
        }
    }
    else
        Find minimum edge cost for insertion and insert the terminal in the current subtour.
}

```

The improved insertion gives a terminal tour with an improved tour cost. To further improve the tour cost, the shuffle procedure, below, is invoked.

#### 4.4 Shuffle Procedure



**FIGURE 12. Shuffling of 5 Terminal Tour**

Given a terminal tour, sequentially drop every terminal in the tour out of its present position and insert it at every alternative position within the tour sequence and compute the total tour cost. This simple shuffling procedure is illustrated in Figure 12 . The pseudo code for the procedure is also given.

When every terminal of the tour is selected, the shuffling procedure is repeated for all the terminals of the net to which the selected terminal belongs. In each cycle, we select the

best terminal of the respective net so that the total tour cost is minimized. This is an *Improved Shuffling* procedure.

```

Shuffle (terminal tour, selected unprobed net) {
    for each terminal of the terminal tour {
        find the corresponding net.
        if (Improve_Flag is TRUE) { /* improved shuffle */
            While (all the terminals of selected net are not checked for minimum cost) {
                Select a terminal of the net.
                Delete the terminal from the tour and insert it at every alternative position in the tour.
                Find out the terminal tour with minimum cost.
                Select the terminal for which terminal tour cost is minimum and return terminal tour.
            }
        }
        else
        {
            Delete the terminal from the tour and insert it at every alternative position in the tour.
            Find out the terminal tour with minimum cost.
            Select the terminal tour minimum cost and return that terminal tour.
        }
    }
}

```

Any heuristic offers a trade-off between computational complexity and performance. The performance of the probe routing optimization can be improved by including higher level shuffling (2 terminals at a time) but only at the expense of a substantial increase in computing cost. If there are  $m$  terminals per net and the total number of nets are  $n$ , then the complexity of the improved insertion procedure is  $O(m \cdot n)$  and the complexity of the improved shuffling procedure is  $O(m^2 \cdot n)$ . If there are  $m$  terminals per interconnection and the total number of interconnections is  $n$ , then the complexity of the selecting farthest

net is  $O(n^2 \cdot m)$ . This time complexity is justified since, to find the optimal route of a single probe, this algorithm is executed only once and then program the solution into the probe controller scheme.

## 4.5 Experimental results

The proposed heuristic algorithm was implemented in 'C' and tested on different test problems on a SUN Ultra-2 SPARCstation hardware platform. Table 2 shows the step by step improvement of the traversal cost by applying different heuristic procedures discussed in the previous section. First column lists the heuristic procedures applied. For improved insert heuristic, Construct\_Terminal\_Tour procedure is called with Improve\_Flag set to boolean value TRUE. For improved shuffle, Shuffle\_Mode in Construct\_Terminal\_Tour procedure is activated. Second column lists number of nets in the netlist. Corresponding terminals per net are shown in the third column. The tour costs which are obtained using different heuristics are noted in the fourth column. These tour costs are computed using Euclidean norm. The CPU time for executing Construct\_Terminal\_Tour procedure is computed using gprof utility in Unix and noted in the fifth column. The last column shows percentage improvement in the terminal tour cost over the tour cost obtained using insert procedure without improvement and shuffling. This tour cost improvement was calculated as follows. Let  $A$  be the tour cost for a given netlist with insert procedure without improvement. Let  $B$  be the tour cost for the same netlist with improved insert procedure. Then percentage improvement in the tour cost for

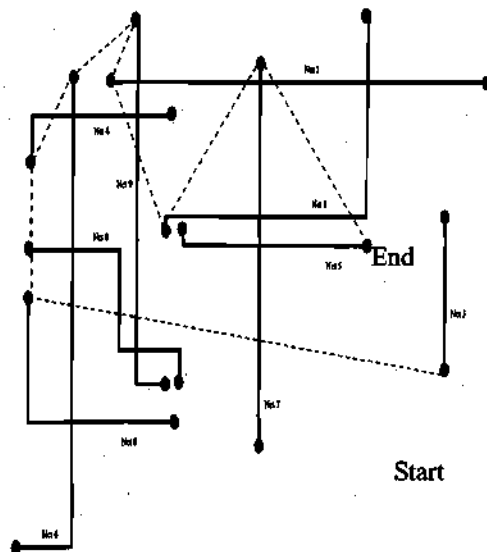
that netlist is given by  $\frac{A-B}{A} \times 100$ . A netlist consisting of 576 nets with four terminals per

**TABLE 2. Probe Route Cost Optimization With Different Heuristics**

Heuristic	Size of netlist	terminals per net	Cost	CPU Time Sec.	% Improvement	Heuristic	Size of netlist	terminals per net	Cost	CPU Time Sec.	% Improvement
Insert	10	2	23.09	0.0	-	Shuffle	10	2	15.49	0.01	32.91
	25	2	39.25	0.0	-		25	2	33.19	0.01	15.44
	24	4	43.10	0.01	-		24	4	22.11	0.07	48.7
	48	4	65.13	0.02	-		48	4	23.29	0.46	64.2
	576	4	697.32	3.18	-		576	4	336.71	783.56	53.93
Improved Insert	10	2	18.18	0.0	21.26	Improved Shuffle	10	2	14.25	0.01	38.28
	25	2	34.38	0.0	12.40		25	2	31.80	0.01	18.98
	24	4	25.88	0.03	39.9		24	4	22.11	0.21	48.7
	48	4	23.75	0.05	63.5		48	4	23.22	1.93	64.3
	576	4	337.57	11.0	53.93		576	4	336.31	3096.6	54.10

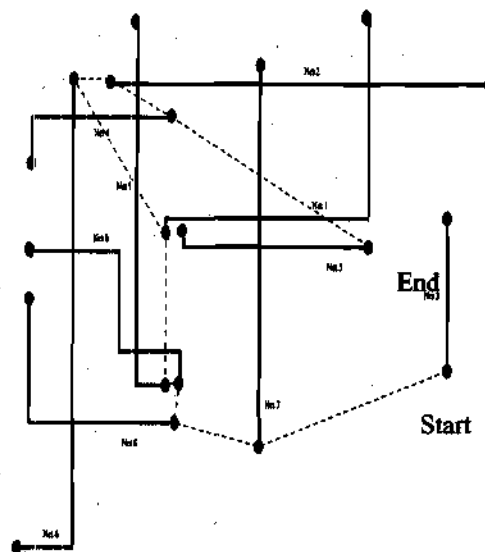
net represents a realistic example and shows 54.1% improvement in the tour cost with 3096 seconds of CPU time. Table 3 compares terminal tour cost computed using the arbitrary insertion and farthest insertion heuristics. In addition to the cost of the best solution, the size of the netlist and the CPU time were used as performance evaluation factors. Column 4 of this table shows the tour costs. For a realistic netlist of 576 nets with five nodes per net, the farthest insertion heuristic gave 74.08% improvement in the terminal tour cost as compared to 69.76% improvement achieved using the arbitrary insertion heuristic. However this improvement was achieved at the cost of 691.22 (5913.28 - 5222.06) seconds of additional computational time. It follows that Construct\_Terminal\_Tour procedure implemented using farthest insertion heuristic constructs more efficient terminal tour at the cost of CPU time which is shown in column 5. Figure 13 shows the snapshots of application of Construct\_Terminal\_Tour procedure to a two terminal netlist of ten inter-

connections. The arbitrary insertion heuristic was used in this case. The total terminal



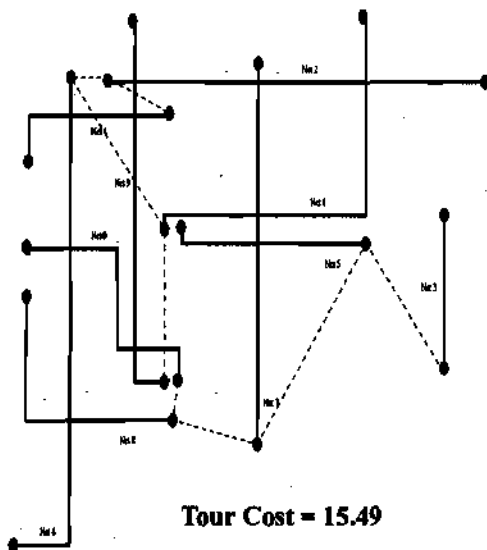
**Tour Cost = 23.09**

(a) Terminal Tour with Arbitrary Insertion



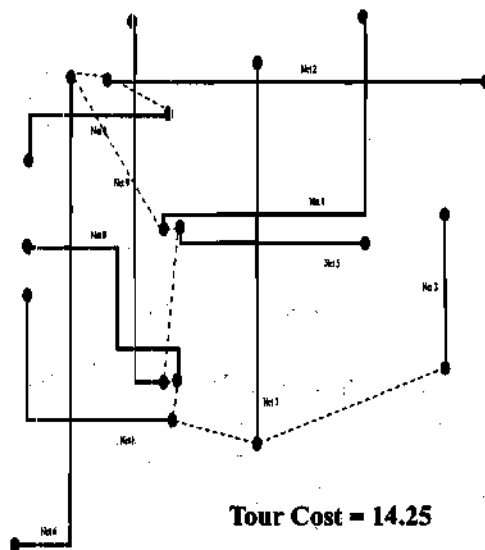
**Tour Cost = 18.18**

(b) Terminal Tour with Improved Insertion



**Tour Cost = 15.49**

(c) Shuffled Terminal Tour



**Tour Cost = 14.25**

(d) Improved Shuffled Terminal Tour

**FIGURE 13. Snapshots of Algorithm**

tour cost improved from 23.09 to 14.25 after improved insertion and shuffling of the initial

tour. We tested netlists with various numbers of interconnections with different terminals

**TABLE 3. Performance Comparison of Heuristics**

Heuristic	Netlist Size	terminals per net	Cost	CPU Time Sec.	% Improvement Over Insertion
Insertion Procedure	10	2	23.09	0.0	-
	25	2	39.25	0.0	-
	48	4	65.13	0.02	-
	567	5	2731.18	4.16	-
Arbitrary Insert with Improved Shuffle	10	2	18.18	0.0	21.26
	25	2	34.38	0.0	12.40
	48	4	29.67	0.05	53.93
	576	5	825.87	5222.06	69.76
Farthest Insert with Improved Shuffle	10	2	17.35	0.01	24.85
	25	2	31.00	0.01	21.01
	48	4	24.57	0.07	62.28
	576	5	707.74	5913.28	74.08

per interconnections. The results are shown in Table 4.

**TABLE 4. Effect of Variation of Number of Nets and Terminals on Probe Route Cost**

Netlist Size	terminals per net	Arbitrary Insert Procedure	Arbitrary Insert with Improved Shuffle	Farthest Insert	Farthest Insert with Improved Shuffle
Constant number of terminals per net with varying number of nets					
96	4	99.85	63.91	51.47	50.20
192	4	191.79	101.13	97.91	96.84
288	4	628.42	428.12	337.76	313.25
576	4	1227.33	694.58	578.36	569.19
Constant number of nets with varying terminals per net					
288	4	628.42	428.12	337.76	313.25
288	5	543.03	232.50	270.48	175.19
288	6	1002.23	331.54	247.76	246.35
288	7	1019.29	397.19	361.09	353.56

The table shows that in most cases the farthest insertion heuristic gives better results compared to heuristic based on the arbitrary insertion. Also both heuristics work efficiently



irrespective of distribution of number of terminals per net or the size of netlist. This fact is important particularly when it comes to the practical application of these heuristics.

The proposed heuristics were applied on benchmark MCM netlist to evaluate their practical use. The netlist mcc1-75.net from the PDWorkshop93 public domain benchmarks for MCM routing was used as a test case. The design consisted of 6 chips, 765 I/O pins and contains 799 signal nets. There are numerous three to seven pin nets. Euclidean norm was used to compute the route cost in this experiment.

Table 5 shows that the heuristic procedure finds efficient probe routes at the cost of CPU time. It achieves significant improvement in the probe traversal cost. The single probe

**TABLE 5. Results on MCM Benchmark**

Netlist: mcc1-75.net, 799 Signal nets, 2 to 7 nodes per net			
Heuristic	Cost	CPU Time Sec.	% Improvement Over Insertion
Arbitrary Insertion			
Insertion Procedure	6569.22		-
Improved Insert	3913.30	18.75	40.42
Shuffle Procedure	3910.30	2366.92	40.48
Improved Shuffle	3886.85	5881.81	40.83
Farthest Insertion			
Insertion Procedure	4229.24	846.63	35.62
Shuffle Procedure	4206.59	2833.49	35.96
Improved Shuffle	4123.82	6820.07	37.22

test technique was compared with double probe test which is currently used in industry.

Today's commercial flying probe testers move all probes independently in X and Y directions with programmable control over Z axis movement and positioning. The typical probe movement speeds of 300 mm per second in both X and Y directions and that of 5

mm per millisecond in Z direction are integrated into our algorithm. The probe is assumed to be positioned 10 mm from the surface of the MCM substrate under test. The specifications of moving probe substrate testers from product catalogs of three leading manufacturers are noted in Table 6. Experiments are performed with total test application time of 200 milliseconds (resistance/capacitance and high frequency probing) and 20 milliseconds (electron beam probing) [25]. The test times include the time to lower the probe, induce the test stimulus, apply the test, raise the probe and move it to the next pad to be probed. The double probe traversal heuristic used the route cost optimization techniques similar to those used for the single probe technique. The test times are computed using the  $L_\infty$  norm distance and corresponding probe speed.

For a double probe technique, it is assumed that both probes are moved simultaneously

**TABLE 6. Moving Probe Tester Specifications**

Speed of Linear Motors	Manufacturer		
	Probot Inc.	Bath Scientific	SPEA
X axis speed	304 mm/second	1143 mm/second	Not Specified
Y axis speed	406 mm/second	1143 mm/second	Not Specified
Z axis speed	Not Specified	Not Specified	5 mm/ms
Z axis positioning	programmable	programmable (20mm)	programmable (40 mm)

and therefore the test time is dominated by a probe which travels a longer distance (either in X or Y direction). The comparison results for the real MCM netlist as well as two randomly generated MCM netlists are summarized in Table 7. As shown in column 1, mcm-600.net and mcm-288.net are two randomly generated netlists with 600 and 288 nets respectively. This validates that our heuristic procedure is effective irrespective of the netlist size. Second and fifth columns denote probe route costs in terms of total time taken

to test a given netlist for a single and a double probe technique respectively. The ratio of the test times in these columns gives the test time reduction achieved by using single probe testing over double probe testing and it is noted in the last column. Taking the netlist with 288 nets case as a baseline, we observe that the improvement in test time reduction is  $2.06/1.80 = 1.14$  and  $2.54/1.80 = 1.41$  for the netlists with 600 and 799 nets, respectively. These ratios are within 5 to 9 percent of the theoretical prediction of  $\left(\frac{600}{288}\right)^{1/4} = 1.20$  and  $\left(\frac{799}{288}\right)^{1/4} = 1.29$  obtained from the result noted in Section 3.3 and proved in [42]. As the number of nets on the MCM substrate increases, the advantage of single probe testing over two probe testing in terms of total test time becomes obvious.

**TABLE 7. Comparison of Test Times: Single Probe Versus Double Probe Test**

MCM Netlist (2 nodes per net)	#of Nets	Single Probe Technique  P1:Time (Seconds)	CPU Time (Seconds)	Double Probe Technique  P2:Time (Seconds)	CPU Time (Seconds)	Test Time Reduction Factor (P2/P1)
Test Application time = 20 ms						
mcc1-75.net	799	33.07	5399.86	83.99	543.33	2.54
mcm-600.net	600	18.13	1908.37	37.36	217.81	2.06
mcm-288.net	288	8.50	250.01	15.29	22.63	1.80
Test Application Time = 200 ms						
mcc1-75.net	799	176.89	5399.86	227.82	543.33	1.29
mcm-600.net	600	126.13	1908.37	145.36	217.81	1.15
mcm-288.net	288	60.35	250.01	67.13	22.63	1.11

# **CHAPTER V**

## **DISTRIBUTED BIST TECHNIQUE**

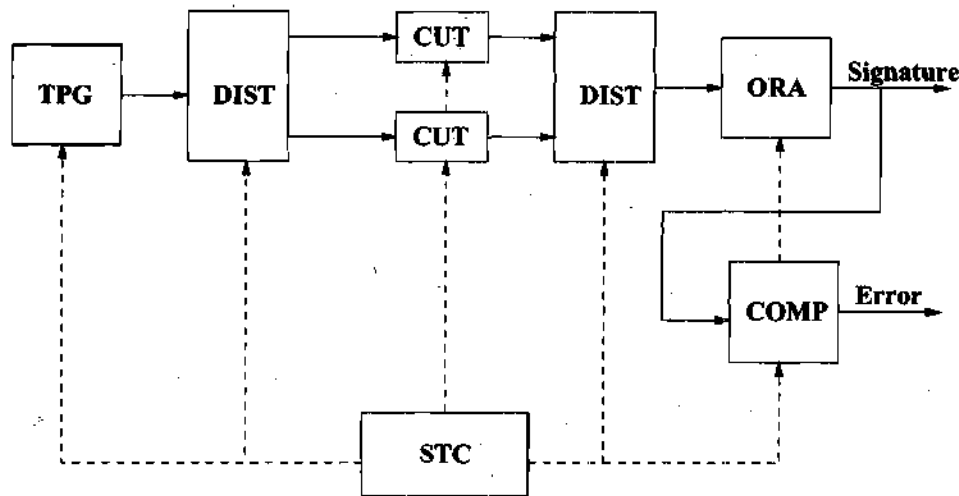
### **FOR PERFORMANCE TEST**

In this chapter, a novel technique of distributed interconnect BIST is presented. A distributed BIST architecture is outlined which can be used to design completely self-testable MCMs. This architecture consists of a specialized test pattern generator structure termed as a P-TPG and a specialized response analyzer denoted as R-MISR.

#### **5.1 BIST Preliminaries**

The issue of interconnect performance testing has become very important due to the increased package densities of multi-chips modules. In view of the speed and the accuracy limitations of the commercial ATEs, it is essential to develop effective built-in self-test strategies to test assembled modules. An on-chip BIST technique can provide at-speed testing of individual ICs of MCMs.

A general BIST architecture is shown in Figure 14.



**FIGURE 14. Generalized BIST Architecture**

Typical built-in self-test (BIST) builds the key functions of an external tester directly into a silicon device. These functions include a source of test stimuli (test pattern generator TPG), distribution system (DIST) for transmitting and receiving test data from circuit under test (CUT), a means of compacting the circuit output response and analyzing it (Output Response Analyzer, ORA), a knowledge of the correct response and comparator circuit (COMP), a pass/fail output signal, and a self-test controller (STC). The boundary scan techniques provide complete diagnosis and detection of static faults located between ICs given that all the constituent ICs conform to the IEEE 1149.1 boundary scan standard [46]. A methodology for testing high performance ICs that are mounted on multichip silicon substrates is presented in [47]. In [48], the author describes an extensive effort to achieve embedded at-speed test (EAST) and results are shown to be better than conventional test methods. The importance of developing a sound at-speed test strategy for

MCMs with an understanding of system design issues such as cost and projected volume is described in [49]. High yield MCM test strategies are discussed in [50]. The Boundary Scan Master [51] was developed at Bell Labs to solve small-board/large device problem. Board Level BIST was achieved but no solution was provided for performance testing of interconnections [52]. A universal testability strategy for MCMs based on BIST and boundary scan is suggested in [53]. It is essential that an MCM as a single component meets its performance specifications. This performance test must verify that all the ICs communicate properly with each other. The performance test must characterize propagation delays including die delays, interconnect delays, and substrate routing delays and also second order effects such as simultaneous switching noise, cross talks and ground bounce.

In the following section, a novel distributed BIST technique that enables at-speed MCM interconnect performance testing with increased diagnostic resolution, in a cost effective way, is proposed. This technique consists of distributed TPGs and distributed MISRs. The distributed TPGs are characterized by an excellent control over the switching behavior of the interchip interconnections they drive. The distributed MISRs are characterized by their modular and reconfigurable structure which leads to increased diagnostic resolution for detecting performance faults, such as delay, cross-talk, and the ground bounce on interchip nets of the MCM. The distributed diagnosis strategy is discussed with a MISR reconfiguration algorithm. A *completely self testable MCM* can be defined as an MCM combining Dual-BIST [54] and Distributed BIST capabilities.

## 5.2 Proposed Approach

The concept of distributed BIST is realized by integrating distribution system functionality into the TPG and ORA functionality. The distributed BIST facility consists of low overhead cascadable and reconfigurable linear finite state machines (LFSM). The specialized test pattern generators are superimposed on the internal boundary scan output nodes of the ICs that comprise the MCM under test. These TPGs produce random test patterns at-speed to drive the internal outputs of the ICs and generate MCM interconnect switching activities that resemble real-life interconnect switching profiles. Internal boundary scan input nodes of every IC act as a MISR or an ORA for compacting the responses sent over output nodes. A high fault coverage and more controlled diagnosis of interconnect faults can be achieved since each LFSM (TPG or MISR) can be used either individually or in a cascaded fashion and these LFSMs can be selectively activated using *Incremental Performance Testing* (IPT) discussed in chapter 7.

## 5.3 Interconnect Test Model and Assumptions

It is assumed that the MCM consists of ICs that conform to level sensitive scan design (LSSD) technique. Therefore, all the latches in the design become scannable. The ICs also have the IEEE 1149.1 standard compatible boundary scan chain. It is also assumed that these boundary scan cells can be converted to a LFSR or a MISR by adding extra XOR gates [55]. This overhead is acceptable considering the return of investment in the testability enhancement. The switching probabilities of i/o nodes of all the constituent ICs

of an MCM are precomputed from its high level model. The correlation between the pair of switching nodes is the absolute probability that these nets switch simultaneously. This probability is computed using a simulation model. For  $N$  number of i/o nodes, the correlation values between nodes are stored in  $N \times N$  correlation matrix. Conventionally, the boundary scan cells on the  $N$  i/o nodes are configured as  $N$  stage LFSR and the pseudo-random patterns generated by it are allowed to propagate on the nets driven by these nodes. In the case of the system with non-boundary-scannable ICs, the BIST test pattern generators are embedded on the boundary nodes. This can be realized easily with LSSD cells on ICs i/o nodes. Important notations related to our Distributed BIST scheme are given below.

*IUT*: Interconnect Under Test ( $N$  bit)

*P-TPG*: Precharacterized Test Pattern Generator

*Switching Profile (P)*: An array  $P$  of size  $N$  where each element  $P[i]$  denotes the absolute switching probability of net  $i$ .

*Correlation Matrix (C)*:  $N \times N$  matrix in which entry  $C[i, j]$  denote simultaneous switching probability of net  $i$  and net  $j$ .

*Activity Profile (A) of IUT*: Switching profile and correlation matrix data of the given IUT constitute activity profile of the IUT.

*MUT*: MCM Under Test.

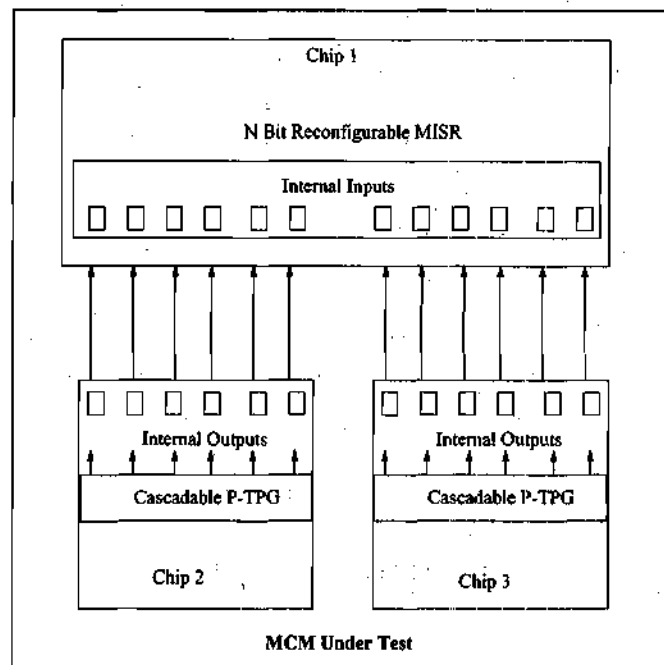
*R-MISR*: Reconfigurable MISR. ( $N$  Bit)

*LFSM*: Linear Finite State Machine.

*FC*: Faulty IC.



## 5.4 Distributed BIST Architecture



**FIGURE 15. Distributed BIST Architecture**

The architecture of the proposed technique is shown in Figure 15. To test the internal interchip interconnections, as well as pad-to-substrate interconnections, the internal outputs of constituent ICs are stimulated with cascadable P-TPGs. To characterize simultaneous switching activity in a realistic fashion, the internal interconnections between all ICs are simultaneously activated. This places the MCM under test into an interconnect self-test mode. These P-TPGs are distributed over the constituent ICs. The extensibility of these P-TPGs is achieved by cascading them to build P-TPGs of higher length. The architecture also consists of a BIST controller to control the interconnect self-test. The MISRs superimposed on the internal input boundary scan nodes of constituent ICs of the

MCM are used to compress the interconnect test responses. A comprehensive description of the theory and design of specialized distributed Precharacterized Test Pattern Generators (P-TPGs) for interconnect switching activity profile generation to achieve performance test has been presented in [56][57] and discussed in chapter VI. The detailed distributed BIST diagnosis based on programmable partitioning of MISR is described in [58][59] and is discussed in chapter 7.

# **CHAPTER VI**

## **PRECHARACTERIZED**

### **TEST PATTERN GENERATOR**

In this chapter, a design and implementation of a precharacterized test pattern generator component of a distributed BIST is discussed in detail. Various P-TPG design techniques are developed and theoretical justification of the design methodology is given. A Markov model of a P-TPG component is developed and analyzed. The P-TPG design optimization problem is formulated and an algorithm to match a given interconnect profile is presented.

Design for testability (DFT) schemes for system level interconnects have been an area of extensive research during recent years. A comprehensive interconnect BIST architecture has been developed by modifying boundary scan cells to generate test vectors on chip in [60]. This technique is suitable for a static interconnect test using deterministic walking 0 and walking 1 algorithms. Random testing that uses specialized random vectors to avoid activating multiple drivers with different values is presented in [61]. A universal BIST

methodology for boundary scan based interconnects is presented in [62]. Dual BIST architecture has been shown to be an effective self-test strategy for MCMs [54]. This strategy can detect interconnect delay faults between ICs running at-speed. It, however, does not address other performance issues such as cross talk, simultaneous switching noise and ground bounce. Enhanced boundary scan design proposed in [63] is required for detection of such dynamic failures. The skin model for IBM S-390 is an effective one for AC interconnect testing, but is not a BIST approach and relies on external ATE channels to apply test patterns [64].

DFT schemes for IC level performance testing typically include various schemes based on BIST [65] and modified boundary scan architectures for level sensitive scan design (LSSD) based ASICs [66]. Hybrid schemes for designing maximum length sequence generators are discussed in [67]. Two pattern test capabilities of autonomous TPG circuits are investigated in [68]. BIST TPGs for interconnect testing have been discussed in [69][70].

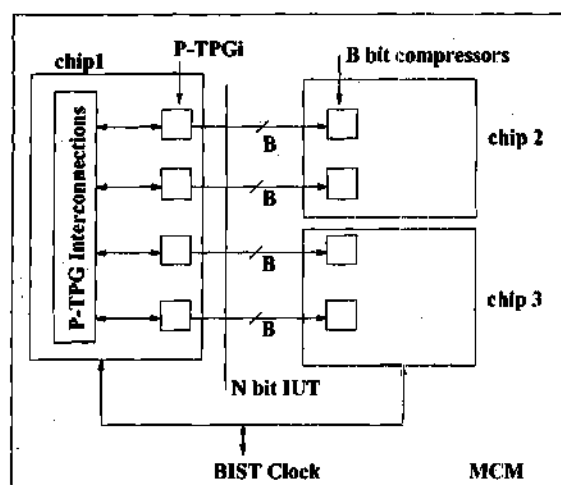
In this section, we propose a novel scheme for synthesizing nonlinear feedback shift register structures called precharacterized test pattern generators(P-TPGs), that can be superimposed on the boundary scan cells of typical ICs to generate MCM interconnect switching activities that resemble real life interconnect switching profiles. The goal is to perform an at-speed MCM interconnect test while simultaneously capturing the so called "second order" effects referred to earlier, as accurately as possible during interconnect BIST. A library of P-TPG components is constructed. The suitable components from this

library are interconnected in specific ways to recreate the switching activity profile of the interconnect being tested.

## 6.1 Proposed Approach

Given the switching activities and the correlations of the IC line drivers, the goal is to superimpose a BIST architecture on the boundary scan cells of MCMs so that the same activities are recreated during BIST. This approach involves the following steps.

- 1) Design of B-bit P-TPGs and the construction of a library of various P-TPG components in such a manner that each component in the library has known switching activities and switching correlations of its outputs.
- 2) If an IC has N line drivers (I/O interconnect width), then  $\lceil N/B \rceil$  P-TPG<sub>i</sub> components are used as shown in Figure 16. The problem is how to determine which P-TPG component



**FIGURE 16. Interconnect BIST Scheme**

from the library is to be used for P-TPG<sub>i</sub> and how these B-bit P-TPGs should be interconnected so that the prescribed switching profiles and correlations (which are obtained from a simulation model of the MCM) of N bit line drivers get recreated during BIST.

## 6.2 Precharacterized TPG Architecture

Typical BIST strategies use TPGs with linear feedback network such as the linear feedback shift register (LFSR) or cellular automata register (CAR). Two basic types of LFSRs are, generally, referred to as external- and internal- XOR LFSRs, respectively. In the following derivations, we consider only the internal-XOR LFSR. The schematic diagram of an internal-XOR LFSR is shown in Figure 17. An n-stage LFSR is characterized by its feedback polynomial  $P(x) = c_0 + c_1x + c_2x^2 + \dots + c_{n-1}x^{n-1} + c_nx^n$ ;  $c_0$  and  $c_n$  are always equal to 1. If the feedback polynomial is primitive, then an n-stage LFSR (initialized to any non-zero state) generates a sequence of length  $2^n - 1$ . Such LFSR is called a maximum length LFSR(ML-LFSR).

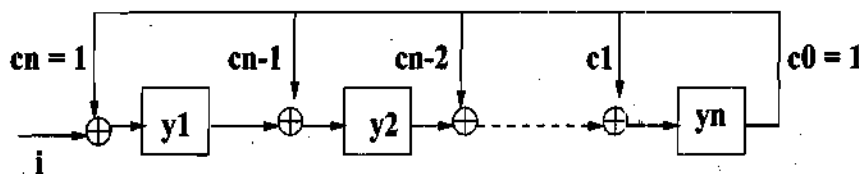


FIGURE 17. Schematic of Internal-XOR LFSR

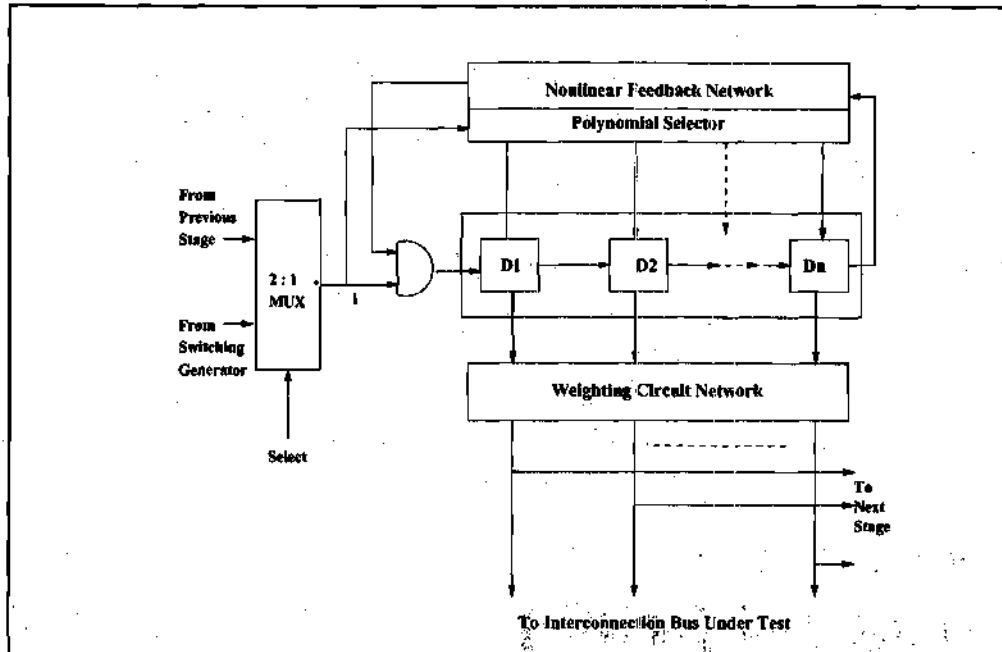
Each sequence generated at each clock cycle is the state  $i$  of LFSR state machine. The future state of  $i$  after  $n$  clock cycles is denoted by  $i_n$ . Thus the immediate next state of  $i$  can be denoted by  $i_1$ . The state  $i_1$  is given by the relationship:

$$\begin{bmatrix} y_{j1} \\ y_{j2} \\ y_{j3} \\ \dots \\ y_{jn} \end{bmatrix} = \begin{bmatrix} 0 & 0 & \dots & 0 & c_n \\ 1 & 0 & \dots & 0 & c_{n-1} \\ 0 & 1 & \dots & 0 & c_{n-2} \\ 0 & 0 & \dots & 0 & \dots \\ 0 & 0 & 0 & 1 & c_1 \end{bmatrix} \begin{bmatrix} y_{j1} \\ y_{j2} \\ y_{j3} \\ \dots \\ y_{jn} \end{bmatrix} \oplus \begin{bmatrix} i \\ 0 \\ 0 \\ \dots \\ 0 \end{bmatrix} \text{ where } i \text{ is the input bit, coefficients } c_k$$

are 1, if and only if, feedback exists. Thus the coefficients  $c_k$  represents the feedback link assignments. The  $k^{\text{th}}$  feedback link is connected (disconnected) when  $c_k = 1(0)$ . The  $n \times n$  matrix (T) is termed as transition matrix of a LFSR. The next state of the LFSR is statistically dependent only on the present state; therefore, the behavior of the LFSR can be represented as a Markov chain [71][72].

The internal-XOR type LFSR described above, is reconfigured into a specialized component called *Precharacterized TPG* by using nonlinear feedback, modification of feedback taps with AND gates (gated feedback) and weighted interconnection network as shown in Figure 18 . Weighting Circuit network [73] is used to control the switching probability of individual output bits. Various configurations of weighting circuits can be implemented to generate desired activity profile for P-TPG under construction. By controlling the value of input bit  $i$  (0 or 1) we can control the AND gates on the feedback taps to either enable or disable the feedback. This feature can control the feedback polynomial

of the LFSR since the feedback taps represent the coefficients  $c_0, c_1, \dots, c_n$  of the feedback polynomial. Consequently, we have two different LFSR structure realizations for  $i = 0$  and  $i = 1$  as shown in Figure 19.

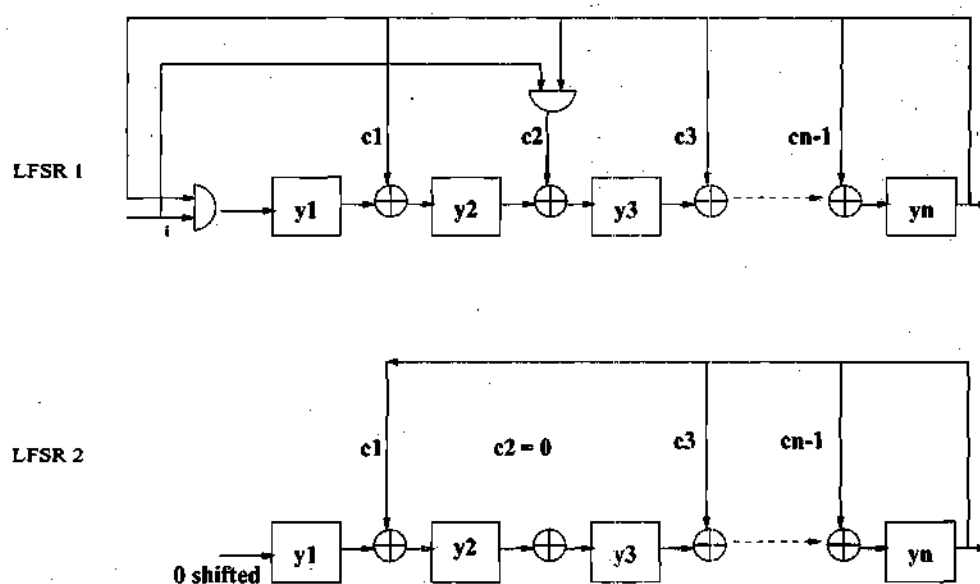


**FIGURE 18. Precharacterized TPG Component**

Because of these two different finite machine realizations, state transitions of the LFSR also differ. Hence, the switching probabilities on each net driven by the output bit of LFSR change with the configuration, and thereby change with the switching probability of input bit  $i$ . This programmable feature of using a multiplexer to select input bit  $i$  is a fundamental feature of the P-TPG architecture. We use this feature to build a cascaded structure of P-TPGs by concatenating two P-TPGs such that the certain output bit of the P-TPG component acts as input bit  $i$  for the next P-TPG component. The complete cascaded P-TPG is then used to generate test sequences so that a switching profile can be recreated



on the  $N$  bit IUT driven by the P-TPG output bits. It is observed that we can recreate reliably, the activity profile of an  $N$  bit IUT by cascaded P-TPG structure consisting of  $N/B$  P-TPG components where  $B$  is the size of P-TPG component. Each component is selected to match the corresponding  $B$  bit segment of the IUT.



**FIGURE 19. Two LFSR Configurations**

### 6.3 P-TPG Design Techniques

To achieve a reliably matched recreation of an activity profile, the absolute switching probabilities generated by P-TPG sequences should match the respective values of the pre-computed switching profile of the IUT. The entries in the correlation matrix obtained by the designed P-TPG must match the precomputed correlation matrix for the given IUT. Next, we discuss the techniques that enable a designer to generate different activity pro-

files on the outputs of the P-TPGs. The first technique consists of a proper choice of feedback polynomial.

- **Design Technique 1:** Switching probability of output bits of an n-stage P-TPG can be changed by changing an associated feedback polynomial.

By the **transitions** property [74] of maximum length LFSR sequences, the number of transitions between 1 and 0 that the sequence makes in one period is  $(m+1)/2$  where  $m = 2^n - 1$ . We illustrate this property in Table 8 by an example of 4 bit LFSR sequences with primitive feedback polynomial  $1+x+x^4$ . For each bit we have total 8 transitions

**TABLE 8. Maximum Length Sequence for 4 bit external-XOR type LFSR**

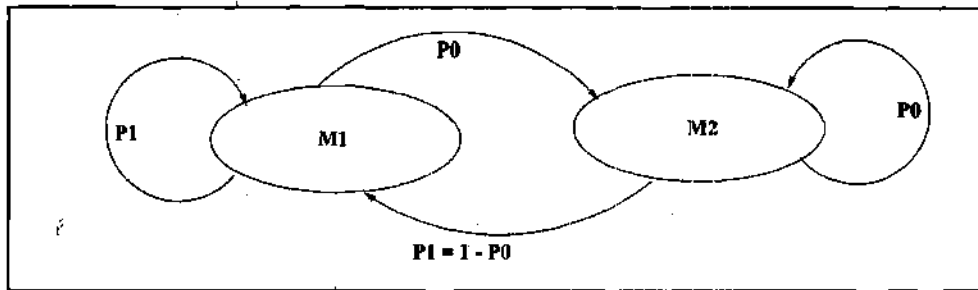
Sequence Bit 3210	Transitions Bit 3	Transitions Bit 2	Transitions Bit 1	Transitions Bit 0
0001			8	8
1000	1			1
1100		1		
1110			1	
1111				2
0111	2			
1011	3	2		
0101	4	3	2	
1010	5	4	3	3
1101		5	4	4
0110	6		5	5
0011		6		6
1001	7		6	
0100	8	7		7
0010		8	7	

between values 1 and 0 or 0 and 1. We define an **absolute switching probability** for a particular output bit as the total number of transitions between 1 and 0 divided by the total number of sequence pairs over which the transitions are considered. In the above example

absolute switching probabilities of all the output bits converge to  $(8/16) = 0.5$ . The concept of 'run' as introduced in [74] is the maximal contiguous grouping of symbols (0 or 1). To count maximal contiguous grouping the period must start at the transition from run of 1's to run of 0's or vice versa. The run property states that in every period of the maximum length sequence, one-half the runs have length 1, one-fourth runs have length 2, one-eighth runs have length 3, and so on. The runs of 1's and 0's terminate with runs of length  $n$  and  $n-1$ , respectively. For example, we have two runs of length 1, one run of length 2 for both symbols 0 and 1 for bit 1. Run of 1's and 0's terminate with runs of length 4 and 3 respectively. It is easily seen that by splitting run of length 4 for any output bit of this LFSR, (for example 1111  $\rightarrow$  1011), we can increase the number of transitions on bit thereby increasing the corresponding switching probability of bit to  $(10/16) = 0.62$ . Similarly by merging two consecutive runs of length 1 (for example 10  $\rightarrow$  11), we can reduce the transitions thereby reducing the corresponding switching probability of bit to  $(7/16) = 0.44$ . These 'run lengths' sequences are the characteristics of a primitive feedback polynomial. Thus, by changing the feedback polynomial from a primitive polynomial to a non-primitive polynomial before the period is complete, the *run lengths* of 1's and 0's can be changed. The non-primitive polynomial implements a different finite state machine where run property does not hold and this causes perturbation of *run lengths*. This eventually changes the transitions property since transition is associated with the beginning of a run. As the total number of transitions between 1 and 0 changes, the switching probability of an output bit also changes.

- **Design Technique 2:** Switching probability of an input bit of a P-TPG controls the switching probabilities of its output bits.

To illustrate this, we develop a Markov Model of the P-TPG component shown in Figure 19 . As we change input bit  $i$  from 1 to 0, we switch from a Markov chain model of a LFSR1 to that of a LFSR2 as shown in Figure 20 . A transition on  $i$  from 0 to 1 switches the machine back to the Markov chain model of LFSR1. Thus, we obtain switching between two Markov chains. These transitions between the two models is governed by input probability of  $i$  being 1; i.e., switching probability of input  $i$ .



**FIGURE 20. Markov Model of P-TPG Component**

Let  $M1$  and  $M2$  denote the Markov chain models of LFSR1 and LFSR2 obtained by assignments  $i = 1$  and  $i = 0$  respectively. We assume that initially P-TPG component runs as a LFSR1 (i.e. Markov chain model  $M1$ ). Let  $\{X_n\}$  denote the state of a P-TPG. Let  $P_{m1}(N)$  and  $P_{m2}(N)$  be the probabilities of a P-TPG component being in the state  $m1$  (Markov chain model  $M1$ ) and the state  $m2$  respectively after  $N$  clock periods starting from the state  $M1$ . Thus the initial conditions are

$$P_{m1}(0) = 1 \text{ and } P_{m2}(0) = 1 - P_{m2}(0) = 0 \quad (1)$$

The P-TPG remains in a state  $m1$  as long as input  $i = 1$ . Let  $P_0$  and  $P_1$  denote signal probability of an input bit  $i$  to be 0 and 1 respectively. By simple probability theory, we have  $P_0 = 1 - P_1$ . So the probability of the P-TPG being in a state  $m1$  or  $m2$  at time  $N$  can be computed given the state of P-TPG at time  $N-1$ . Thus the above relationship can be expressed in terms of conditional probabilities as  $Prob[X_n = m2 | X_{n-1} = m1]$  and  $Prob[X_n = m1 | X_{n-1} = m2]$ .

It is obvious that these conditional probabilities are not affected by state of P-TPG at time  $N-2$  thus satisfying Markov property. Then we can model P-TPG as a two-state Markov process with the following state transition probabilities.

$$\begin{aligned} P_{m1,m1} &= 1 - P_0 & P_{m1,m2} &= P_0 \\ P_{m2,m1} &= P_0(1 - P_0) & P_{m2,m2} &= 1 - P_0(1 - P_0) \end{aligned} \quad (2)$$

where  $P_{ij}$  = probability of going from state  $i$  to  $j$  in one clock cycle. Let  $m1$  = State of being in Markov chain M1 and  $m2$  = State of being in Markov chain M2. The state diagram of the above Markov process is given in Figure 20. The state transition matrix of this two state Markov process, denoted by  $P$ , is given by

$$T = \begin{bmatrix} 1 - P_0 & P_0 \\ P_0(1 - P_0) & 1 - P_0(1 - P_0) \end{bmatrix} \quad (3)$$

Since the initial state of P-TPG is assumed to be  $m1$ , we have

$$\begin{bmatrix} P_{m1}(N) & P_{m2}(N) \end{bmatrix} = \begin{bmatrix} 1 & 0 \end{bmatrix} T^N \quad (4)$$

Therefore  $P_{m1}(N)$  (probability of P-TPG being in state m1 after N cycles) is given by

$$P_{m1}(N) = \begin{bmatrix} 1 & 0 \end{bmatrix} T^{N-1} \begin{bmatrix} 1-P_0 \\ P_0(1-P_0) \end{bmatrix} \quad (5)$$

To complete our proof, we use mathematical definition of eigen value and eigen vector and state a well-known theorem. Its proof can be found in [72].

**Theorem:** Let  $\lambda_1, \lambda_2, \dots, \lambda_n$  be distinct eigen values of an  $n \times n$  matrix A, and let  $x_1, x_2, \dots, x_n$  be linearly independent eigen vectors associated with these eigen values respectively.

Define matrices L and  $\Lambda$  such that  $L = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix}$  and  $\Lambda = \begin{bmatrix} \lambda_1 & 0 & \dots & \dots & 0 \\ 0 & \lambda_2 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 0 & \dots & \dots & 0 & \lambda_n \end{bmatrix}$

$$\text{then } A^k = L^{-1} \Lambda^k L \quad (6)$$

The characteristic polynomial of a transition matrix T is given by the expression  $|T - \lambda I|$ , where I is identity matrix. The roots of the characteristic polynomial give the eigen values of T. So eigen values are

$$\lambda_1 = 1 \text{ and } \lambda_2 = (1-P_0)^2 \quad (7)$$

Using the definition, the eigen vectors can be found to be  $x_1 = (1, 1/(1-P_0))$  and  $x_2 = (-1, 1)$ . Since these vectors are linearly independent, inverse of matrix  $L$  exists as follows:

$$L = \begin{bmatrix} 1 & 1/(1-P_0) \\ -1 & 1 \end{bmatrix} \text{ and } L^{-1} = (1/(2-P_0)) \begin{bmatrix} 1-P_0 & -1 \\ 1-P_0 & 1-P_0 \end{bmatrix} \quad (8)$$

From equations (6), (7) and (8), we have

$$T^{N-1} = L^{-1} \begin{bmatrix} 1 & 0 \\ 0 & (1-P_0)^2 \end{bmatrix}^{N-1} L \quad (9)$$

Substituting (8) and (9) in equation (5) we get

$$P_{m1}(N) = (1-P_0)/(2-P_0) [1 + (1-P_0)^{2N-1}] \quad (10)$$

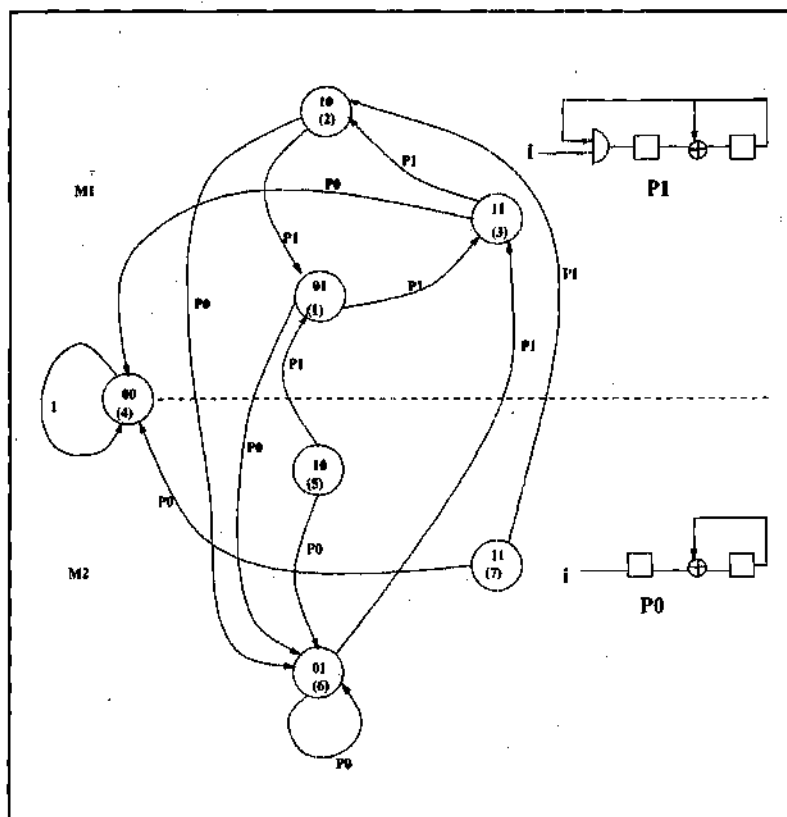
This is the expression for steady state probability of P-TPG being in state m1. It is trivial to verify that as  $P_0 \rightarrow 0$ ,  $P_{m1}(N) \rightarrow 1$  and as  $P_0 \rightarrow 1$ ,  $P_{m1}(N) \rightarrow 0$ . This proves that for large  $N$ , (large number of test sequences applied), the state of P-TPG is governed by switching probability of input bit namely  $P_0$ . So with change in the value of  $P_0$ , P-TPG switches between states m1 and m2. This switching forces two different Markov chains to be active at one time. Let M1 represent the Markov model of a maximum length LFSR (which is a typical choice as TPG) and M2 represent the Markov model of a LFSR with a nonprimitive polynomial, then switching probability  $P_0$  causes dynamic switching

between feedback polynomials of M1 and M2. This changes switching probability of output bits of P-TPG as discussed in design technique 1.

- **Design technique 3:** Switching probability of an input bit of a P-TPG controls the correlations of its output bits.

We prove this with an example of the finite Markov chain model of 2 bit P-TPG

(Figure 21 ) with the primitive polynomial  $(1 + x + x^2)$ .



**FIGURE 21. Markov Chain Model of 2-bit P-TPG**



The state transitions indicate switching between states of a machine M1 and M2 depending on  $P_0$  and  $P_1$  i.e. the input signal probability. To find the correlations of the output bits, we need to find the probabilities of transitions between the states where both bits switch simultaneously. For example, transition from a state 2 to a state 1. For this transition, the correlations between a bit 0 & a bit 1 is the probability that there is transition from the state 2 to the state 1 given that the P-TPG is already in the state 2. Since the Markov chain is non-irreducible (ergodic) with the state 00 (4) being an absorbing state, no matter where the process starts, the probability after  $n$  steps that the P-TPG is in an absorbing state(4) tends to 1 as  $n$  approaches infinity. Hence we have to find the probabilities that the process is in transient states (2-7). This can be computed by the use of fundamental matrix  $N$  and theorem noted in [75].

$$T = \begin{bmatrix} 0 & 0 & P_1 & 0 & 0 & P_0 & 0 \\ P_1 & 0 & 0 & 0 & 0 & P_0 & 0 \\ 0 & P_1 & 0 & P_0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ P_1 & 0 & 0 & 0 & 0 & P_0 & 0 \\ 0 & 0 & P_1 & 0 & 0 & P_0 & 0 \\ 0 & P_1 & 0 & P_0 & 0 & 0 & 0 \end{bmatrix}$$

**FIGURE 22. Transition Matrix of 2-bit P-TPG**

To make use of this theorem we write the transition probability matrix of the Markov chain shown in Figure 22 in its canonical form by uniting all absorbing and transient states

as submatrices  $\begin{bmatrix} S & O \\ R & Q \end{bmatrix}$  where S deals with the absorbing states, O consists of all zeros, R concerns the transitions from transient to absorbing states and Q correspond to transient states. Then diagonal entries of  $N = (I - Q)^{-1}$ , ( $I$  = identity matrix) give the means of the total number of times the process is in respective transient states. With trivial computation we get

$$N = \begin{bmatrix} 1 & 0 & -P_1 & 0 & -P_0 & 0 \\ -P_1 & 1 & 0 & 0 & 0 & 0 \\ 0 & -P_1 & 1 & 0 & -P_0 & 0 \\ 0 & 0 & -P_1 & 1 & -P_0 & 0 \\ 0 & -P_1 & 0 & 0 & 1 - P_0 & 0 \\ -P_1 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}^{-1}$$

The diagonal elements of this matrix N can be computed in terms of  $P_0$  and  $P_1$ . For 2 different values of input signal probabilities, the average of total number of times P-TPG is in the given transient state is shown in Table 9. Thus the probability of being in these

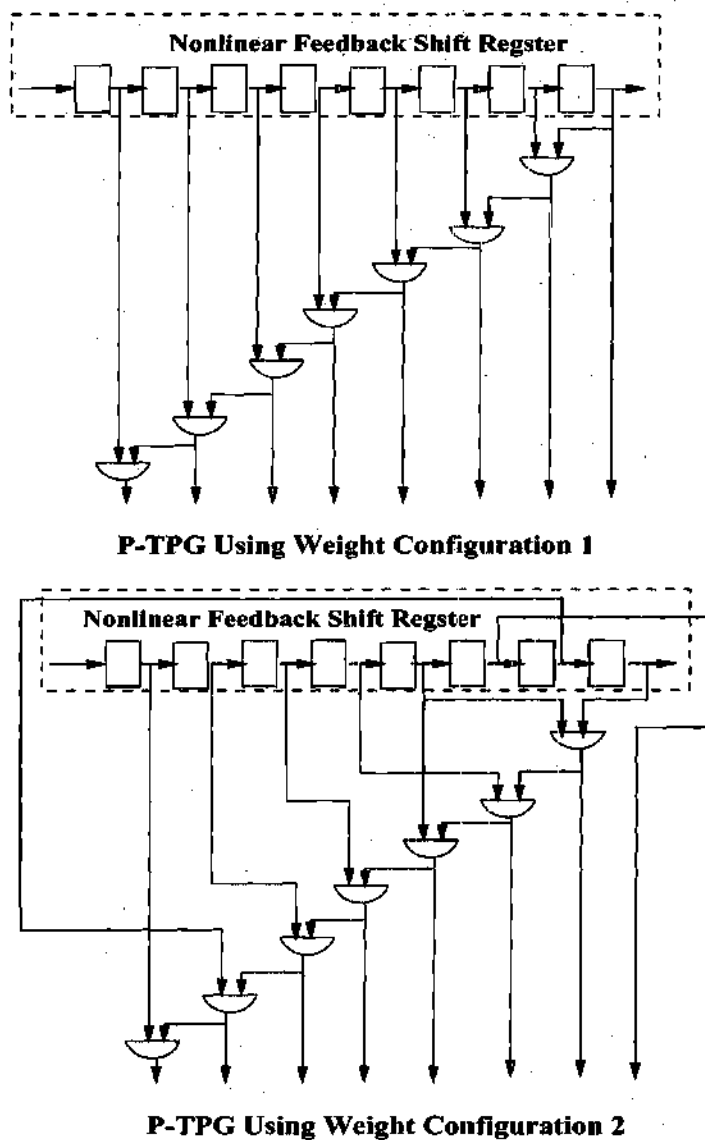
**TABLE 9. Average of Total Number of Times P-TPG is in Given Transient State**

States	$P_0 = 0.2, P_1 = 0.8$	$P_0 = 0.5, P_1 = 0.5$
state 2	5	2
state 3	5	2
state 1	4.2	1.5
state 5	1	1
state 6	3	3.5
state 7	1	1

states is controlled by the values of  $P_0$  and  $P_1$ . Consequently the correlations of output bits which is a conditional probability of transition between the states is controlled by the input switching probability.

- **Design technique 4:** The weighting circuit configurations of a P-TPG controls the correlations of its output bits.

To illustrate this, we design a P-TPG component generating same type of switching profile (increasing) using two different weighting circuit configuration as shown in Figure 23.



**FIGURE 23. Two Weight Circuit Configurations for Increasing Switching Profile**

It is trivial to see that each state of a Markov model of a P-TPG before weight circuit configurations maps to another state after the weighting circuit configuration and this another state is governed by values of bits used to realize the respective weight circuit configuration. As we change the weighting circuit configurations, this mapping changes causing different mapped states (after the weighting circuit configurations) for the states of the Markov chain model before weighting circuit network. Therefore, the correlations among the output bits also change.

- **Design technique 5:** The feedback polynomial of a P-TPG controls the correlations of its output bits.

This can be deduced easily from the design techniques 1, 3 and 4 since the states of a Markov model of a P-TPG change with the non-primitive polynomial used for implementing Markov chain M2. Therefore the correlations of the output bits of a P-TPG change with associated feedback polynomial, by the similar argument used to prove the design technique 4.

To summarize, the following properties make P-TPGs easily synthesizable structures to recreate IUT activity profiles.

- **Modularity:** Fixed length P-TPGs can be cascaded to realize a test pattern generator for a given IUT.
- **Flexibility:** Realizing accurate IUT profile becomes simple since numerous P-TPGs can be predesigned.

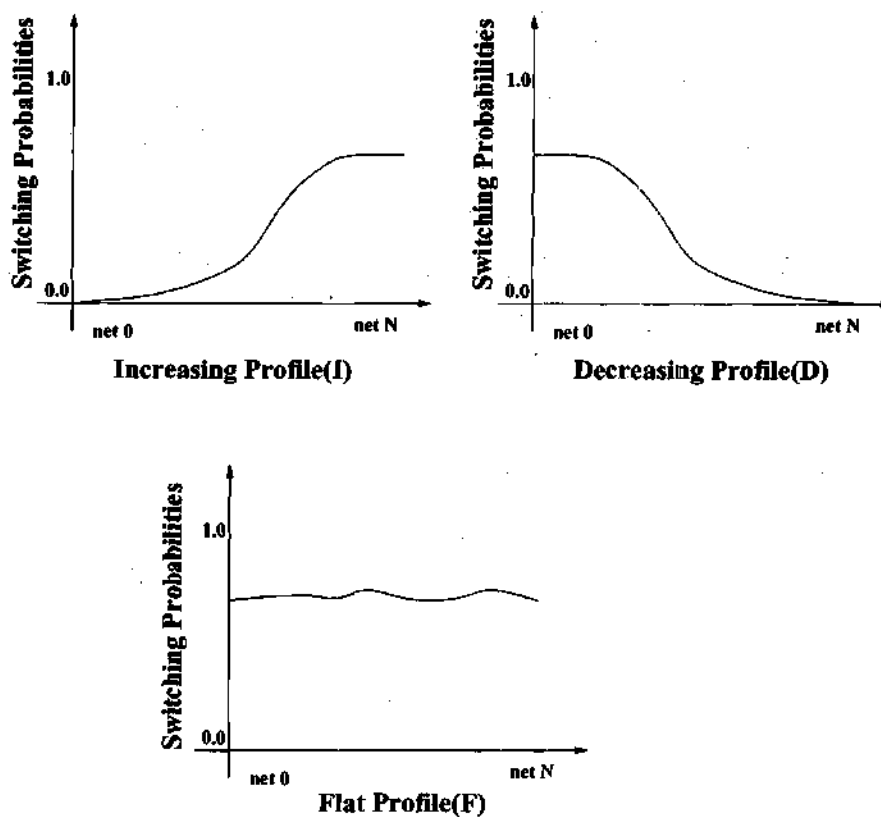
- **Granularity:** Better control over the test sequence generation due to complete precharacterization of individual P-TPGs.

## 6.4 P-TPG Design Specifications

The proposed scheme is based on complete characterization of the basic component i.e. B bit P-TPG. For each P-TPG configuration, there are four design specifications: a feedback polynomial, a profile type, a configuration type and an input threshold. Below, each specification is described in detail.

**Feedback Polynomial:** This specification denotes the primitive polynomial of an internal-XOR type LFSR as well as the location of feedback taps to be modified by inserting AND gates for non-linear feedback. This is an implementation of non-primitive feedback polynomial.

**Profile Type:** Typically we classify IUT switching profiles into three broad classes. These are increasing profile, decreasing profile and a flat profile denoted by I, D and F, respectively, and are shown graphically in Figure 24.

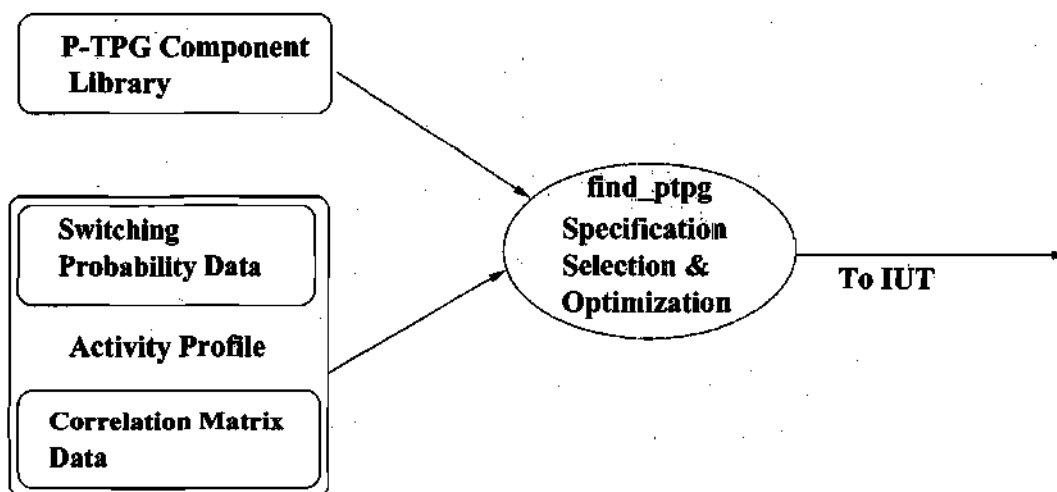


**FIGURE 24. Types of Switching Profiles**

**Configuration Type:** For each switching profile we select one of the two weighting circuit configurations as shown in Figure 23. These are denoted by C1 and C2.

**Input Threshold:** This specification indicate switching probability of an input bit  $i$  of a P-TPG component. This input switching probability value is chosen from the range of 0.1 to 0.9.

## 6.5 P-TPG Design Optimization

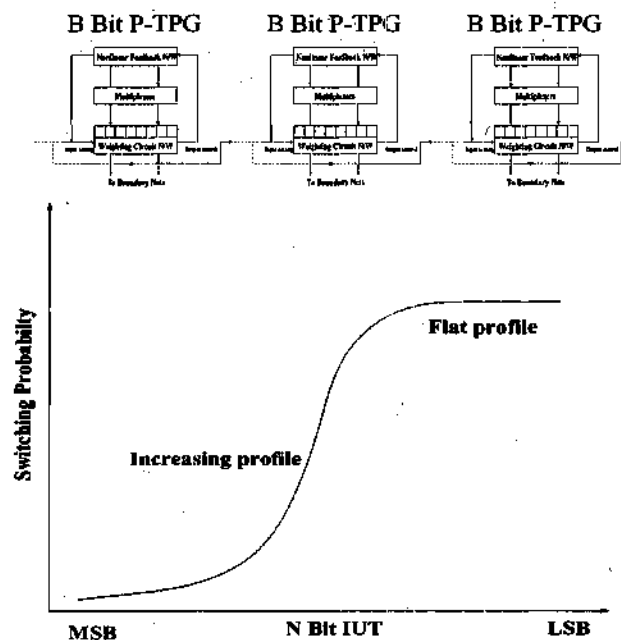


**FIGURE 25. DFT Architecture for Interconnect BIST**

The BIST architecture shown in Figure 25 mainly consists of precharacterized TPG components library. It allows the designer of P-TPG cascaded structure to choose from various precharacterized components. Note that the numerous P-TPG components with specific configurations can be synthesized by simultaneously changing one or more specifications stated above. To find the specifications of a P-TPG component which recreates activity profile of a given IUT in the best possible way is obviously a computationally hard problem and belongs to the category of problems based on combinatorial search and optimization. This *P-TPG Design Optimization Problem* uses a cost function based on an accurate matching of activity profile. The task is to search for a P-TPG component in a

library, to design a P-TPG component with particular specifications, to select N/B P-TPG components and to interconnect them in such a manner that overall activity profile of cascaded structure matches the activity profile of IUT. The newly designed components can be stored in P-TPG component library for future reuse. The algorithm *MatchProfile* which uses procedure *find\_ptpg* has been implemented to achieve accurate matching.

## 6.6 Activity Profile Generation Using P-TPG



**FIGURE 26. Cascaded P-TPG Structure and IUT Switching Profile**

For each P-TPG specification, an activity profile is computed by solving Markov equations discussed in Section 6.3. We create an extensive library of such activity profiles. We then use this profile library to perform optimized matching of activity and correlation of



the B bit section of the IUT. Advantage of this scheme is that there is no simulation overhead since we build analytically precharacterized B bit P-TPG components. Figure 26 shows the cascaded structure of P-TPG components of and associated switching profile. Below, an algorithm MatchProfile is described in detail. The synthesis tool based on this algorithm and the experimental results are described in chapter 8.

## 6.7 Algorithm Description

```

Algorithm MatchProfile (IUT profile, P-TPG library) {
  select B as P-TPG bit size; divide N bit IUT into
  N/B segments;
  select ith segment of IUT;
  P(i) = find_ptpg(i/p activity);
  return i/p activity; increment i;
  repeat
    for each ith segment of IUT {
      P(i) = find_ptpg(output activities of P(0)
      upto P(i-1));
      cascade [P(i-1), P(i)] with output bit of
      P(i-1) corresponding to best o/p activity;
      select cascaded structure with minimum cost;
    }
  until (IUT activity profile is matched);
  return cascaded P-TPG components;
}

Procedure find_ptpg(switching activity) {
  cost_function = sum of difference between values
  of switching probabilities & correlations of P-TPG
  component and IUT segment;
  return P-TPG component with minimum cost;
}

```

# **CHAPTER VII**

## **DISTRIBUTED DIAGNOSIS**

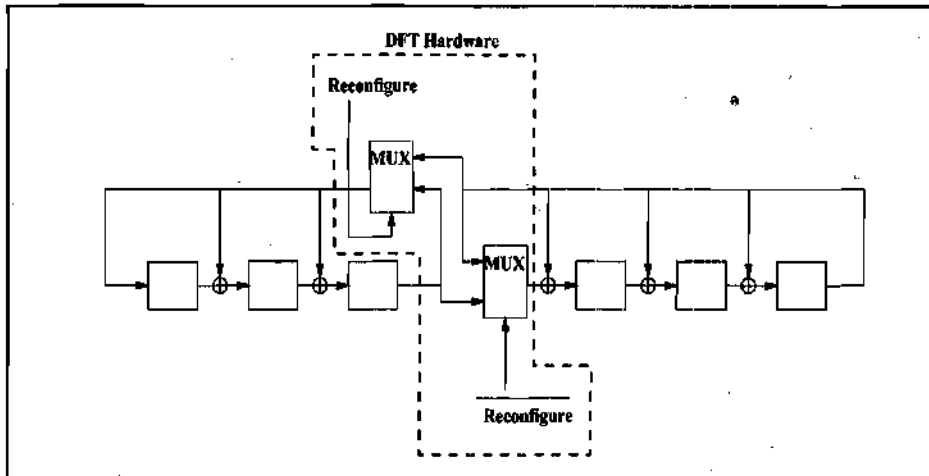
### **USING MISR RECONFIGURATION**

A general description of a distributed BIST diagnosis of MCMs is given. It allows the fault diagnosis of MCM interconnects for dynamic effects. A theory of partitioning of linear registers is applied to devise a two-phase distributed diagnosis strategy. The design of a novel MISR reconfiguration scheme that enables high diagnosis resolution is presented. Simulation results obtained confirm the effectiveness of this BIST technique.

#### **7.1 Proposed Approach**

In the distributed BIST technique, MISRs superimposed on the input boundary scan cells of an IC are used for test response compression. The aliasing probability of a MISR is defined as the probability that two different input data streams to the MISR lead to the same signature. The concatenation properties of LFSRs are studied in [76]. The segmented LFSR structure is presented in [74]. These approaches are suitable for distributed

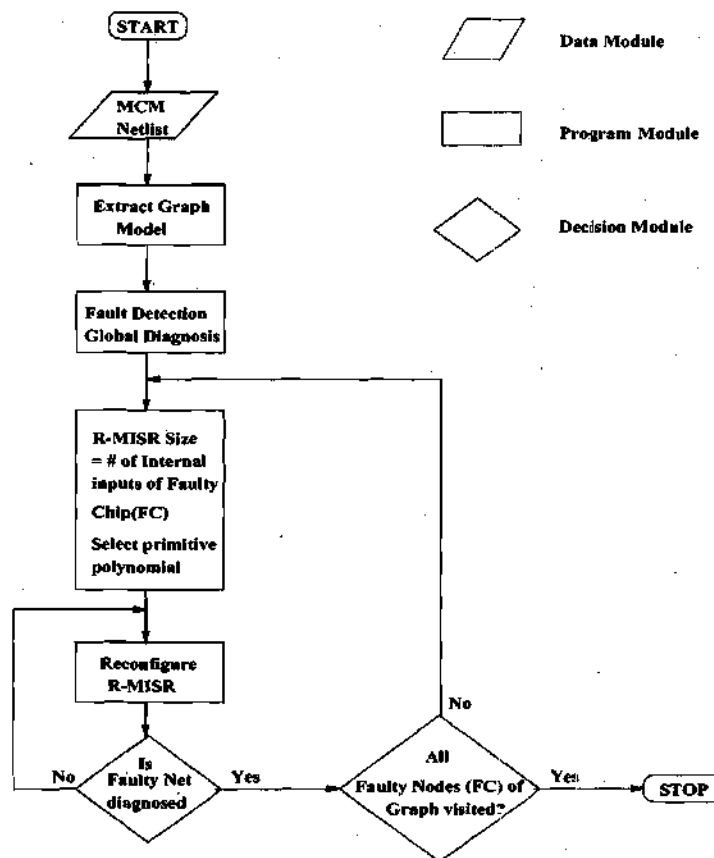
test pattern generation. The partitioning of LFSRs and linear cellular automata registers (LCARs) is studied in [77] and this forms the basis of our distributed BIST diagnosis strategy. The partitioning of a six stage LFSR into two LFSRs of length three each, under multiplexer control, is shown in Figure 27. Such a partitioning strategy when applied to a MISR leads to a new DFT structure called Reconfigurable MISR (*R-MISR*). The dynamic reconfiguration of MISR structures using multiplexers is a fundamental feature of the R-MISR architecture. In the process of reconfiguration, care must be taken to ensure that the signature aliasing probabilities of the reconfigured MISRs are not degraded.



**FIGURE 27. Dynamic Reconfiguration of LFSR**

The proposed distributed BIST-based diagnosis technique consists of two phases: (a) that of determining whether there is any fault in the MCM IC interconnections and (b) that of isolating the fault as accurately as possible. Phase (b), above, consists of two steps: (1) identifying the pairs of ICs with faulty interconnections between them and (2) isolating the fault down to as few sets of relevant IC interconnections as possible. In the following,

the two phases of the proposed distributed BIST-based fault diagnosis algorithm is discussed.



**FIGURE 28. MCM Interconnection Diagnosis Flowchart**

**Fault Detection Phase:** There are two ways in which the LFSRs and MISRs superimposed on the boundary scan output and input cells of the MCM ICs, respectively, can be configured for test pattern generation and output response compression. First, assuming there are on the average  $n_o$  output boundary scan cells per IC and a total of  $N$  ICs mounted

on an MCM substrate, all  $Nn_0$  such boundary scan cells can be connected as one LFSR. Such a LFSR is referred to as a *Global LFSR*. Similarly, a *Global MISR* is defined as one that consists of  $Nn_i$  input boundary scan cells of ICs, where  $n_i$  is the average number of input boundary scan cells per IC. The  $n_0$  output boundary scan cells of an IC can also be configured as an LFSR and this is referred to as a *Local LFSR* with respect to the IC concerned. It is used to generate the test patterns over the output interconnections of the respective IC. Similarly the  $n_i$  input boundary scan cells of an IC can be configured as a MISR and this is referred to as a *Local MISR*. These two ways of configuring a LFSR and a MISR lead to four possible test modes as described below.

#### Test Mode 1: Local LFSR/ Local MISR.

In this mode, all ICs generate test patterns on their output interconnections using the Local LFSR and compress the test responses using the Local MISR. This test mode enables the isolation of the faulty interconnections down to those connected to a single IC by scanning out the signatures of its Local MISR and comparing it with the simulated golden signature. This test mode also enables testing of interconnections between pairs of ICs selectively. Let  $C_1$  and  $C_2$  be a pair of ICs with the Local LFSR of  $C_1$  and the Local MISR of  $C_2$  activated. It is essential that all internal inputs of  $C_2$  that are connected to the ICs other than  $C_1$  be set to known states (0 or 1). Our methodology identifies the receiver IC whose signature is faulty and its communicating driver ICs. Any discrepancy in the signatures with respect to the known good simulated signature isolates the IC fault down to the IC with the incorrect signature. This selective activation of Local LFSR and Local

MISR is called *Incremental Performance Testing*. The advantage of this mode is high diagnostic resolution down to the IC level. The disadvantage is that if an MCM consists of ICs with very few connections to other ICs, then the smaller Local MISR size might result in a higher aliasing probability. This disadvantage is overcome in test mode 2 at the cost of diagnostic resolution.

#### Test Mode 2: Global LFSR/ Global MISR

In this test mode, the Global LFSR sends test patterns over all internal interconnections between the ICs of the MCM-under-test and realizes a precharacterized test pattern generator as described in [57]. This particular configuration enables generation of patterns on internal IC outputs such that they replicate the real time switching activities on the MCM interconnections. The Global MISR compresses the test responses simultaneously and the signature can be scanned out for comparison with the golden signature. The advantage of this configuration is the considerable savings in XOR gates [78] to constructing a maximum length sequence MISR.

#### Test Mode 3: Global LFSR/Local MISR

In this mode, the local MISR for each IC is run with Global LFSR activated. The diagnostic resolution obtained with this mode is between that obtained with the test mode 1 and test mode 2.

#### Test Mode 4: Local LFSR/ Global MISR.

In this mode, incremental performance testing is achieved by a BIST controller. It selectively activates the Local LFSR in one IC of the MCM. The test responses are compressed by a Global MISR. This enables performance testing of interconnects between the selected IC and the rest of the ICs on the MCM. This can be repeated for all of the ICs. It is possible to activate the Local LFSRs of two or more ICs. The BIST controller is responsible for optimized BIST scheduling as well as to selectively activate delay test or cross-talk tests. One such BIST controller design which can be effectively extended for MCMs is described in [79].

**Fault Diagnosis Phase:** The first phase, above, identifies one or more ICs with faulty interconnections to their inputs. Since tests are run at-speed these faults include, for example, interconnections with delay problems. In the following, it is assumed that only a single interconnection of an IC has a delay fault. However, more than one IC can have a faulty input interconnection. To perform diagnosis, we run the Local MISR of each IC in various configurations to exactly identify the faulty interconnect. The diagnosis algorithm described below, takes the MCM netlist as an input and computes how the R-MISRs should be reconfigured to diagnose the fault in a minimum number of reconfigurations.

## Distributed BIST Diagnosis Algorithm

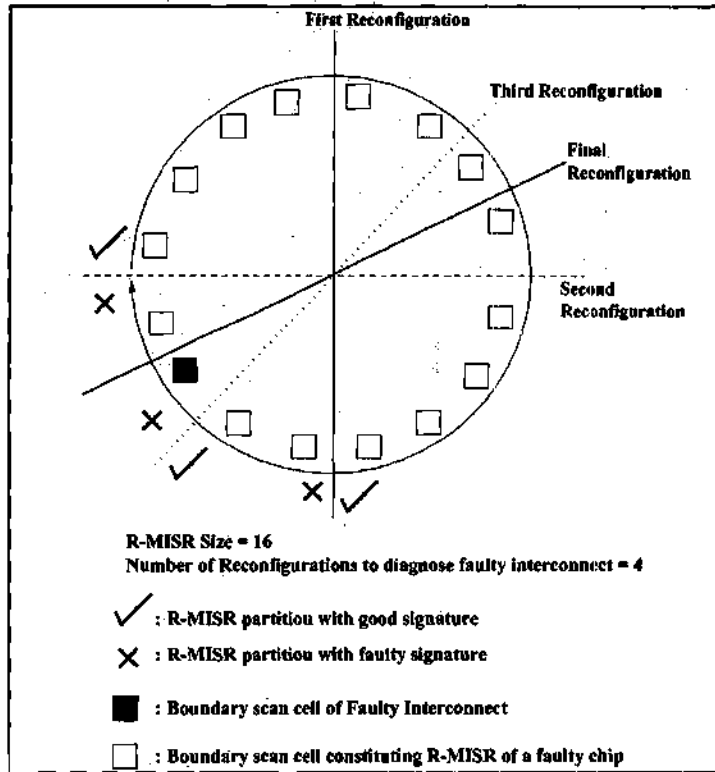
```
Algorithm Distributed BIST Diagnosis () {  
  /* Detection Phase */  
  run Global LFSR/Global MISR mode;  
  if (correct signature)  
    done;  
  else /* Diagnosis Phase */  
  {  
    for the  $i^{\text{th}}$  IC of the MCM under test {  
      run Global LFSR/Local MISR mode  
      or Local LFSR/Local MISR mode;  
      if (correct signature)  
        increment  $i$ ;  
      else  
      {  
        FC = faulty IC;  
        Add FC to the faulty_IC set;  
      }  
    }  
    for each FC in faulty_IC set {  
      run_local_diagnosis (FC);  
      return faulty interconnect;  
    }  
  }  
  
  Procedure run_local_diagnosis(faulty IC) {  
    R-MISR = extract_misr(faulty IC);  
    while (faulty interconnect is NOT diagnosed)  
    {  
      fault list = reconfigure_misr(R-MISR);  
      increment num_reconfigurations;  
    }  
    return faulty interconnect;  
  }  
}
```

## 7.2 MISR Reconfiguration Technique

The principle of MISR reconfiguration for fault diagnosis can be explained with the help of Figure 29. Consider an IC with  $n$  input pins and consider conceptually that the boundary scan cells corresponding to these  $n$  input pins are arranged in a circle as shown



in Figure 29. In practice, all these  $n$  cells are connected as a single MISR with aliasing probability  $2^{-n}$ . We reconfigure R-MISR into two partitions such that aliasing probability of each partition is minimized simultaneously.



**FIGURE 29. Diagnosis Using R-MISR Reconfiguration**

**Theorem 1:** The minimum aliasing probability of a R-MISR of  $n$  bits partitioned into two individual MISRs is maximized when one of the partitions consists of  $\left\lfloor \frac{n}{2} \right\rfloor$  bits and the other consists of  $n - \left\lfloor \frac{n}{2} \right\rfloor$  bits.

**Proof:** Consider  $n$  even. Let one partition of the R-MISR consist of  $x$  cells and the other  $(n - x)$  cells. Further, let  $x \leq \frac{n}{2}$ . Then the minimum of the aliasing probabilities  $2^{-x}$  and  $2^{-(n-x)}$  of the two partitions of a R-MISR is maximized when  $x = \frac{n}{2}$ . The same result is achieved for  $x \geq \frac{n}{2}$ . Hence, the statement of the theorem holds for  $n$  even. It can be proved similarly for  $n$  odd. Q.E.D.

From theorem 1, above, our goal is to recursively partition the original R-MISR of  $n$  bits into two MISRs of  $\left\lfloor \frac{n}{2} \right\rfloor$  and  $n - \left\lfloor \frac{n}{2} \right\rfloor$  bits so that precise fault diagnosis is achieved [59].

### 7.3 MISR Reconfiguration Under Single Fault

Consider that the circular R-MISR of Figure 29 is arbitrarily partitioned into two halves ( $n$  even). The respective cells of each partition are configured into two MISRs. This is shown by "First Reconfiguration" in Figure 29. Since the interconnection leading to the black cell is faulty, the MISR configuration of left partition gives the incorrect signature. Hence, the fault is diagnosed to be on one of the  $n/2$  left half cells. Then the half-circle corresponding to the possible faulty cells is split into half by the "Second Reconfiguration" of Figure 29. In the above, all the  $n/2$  cells above the horizontal line are connected as one MISR. Similarly for the cells below. Note that at each step, the number of cells in each MISR is always  $n/2$ . After the "Second Reconfiguration", the fault is diagnosed to be in the cells corresponding to the lower left quadrant of the circle of Figure 29.

The reconfiguration process can be continued in this manner until faulty cell (interconnection leading into the cell) is located exactly. This leads to the theorem below.

**Theorem 2:** Given a R-MISR superimposed on  $n$  boundary scan cells, the number of reconfigurations necessary and sufficient to completely diagnose a performance fault on an internal input under the single fault assumption is equal to  $\lceil \log_2 n \rceil$ .

**Proof:** Consider Figure 29 in which the input boundary scan cells of an IC are shown conceptually arranged in a circle. Initially the circle is partitioned vertically into two halves, the set of right half cells RH and the set of left half cells LH. Subsequently, the circle is partitioned horizontally into two halves, the set of top half cells TH and the set of bottom half cells BH. Under the single fault model, one of the MISRs in both step 1 and 2, above will yield an incorrect signature. Let's say these correspond to LH and BH, respectively. Then the set of possible faulty cells (or incident input) is given by  $LH \cap BH$ . Note that when  $n$  is even, the cardinality of LH and BH is  $n/2$ , while that of  $LH \cap BH$  is  $n/4$ . After the two steps above, the set of possible faulty cells will be constrained to lie in one of the four quadrants of the circle of Figure 29. The next reconfiguration step consists of constructing two MISRs of length  $n/2$ , such that this quadrant is split into two and process is repeated. From this, the faulty cell or interconnect is located exactly in  $\lceil \log_2 n \rceil$  steps for  $n$  even. Consider the case of  $n$  odd. In this case, one partition of circle of Figure 29 will consist of an even number of cells and the other will consist of an odd number of cells. In the recursive search process, if the partition containing the faulty cells

always one that has an odd number cells, then one additional reconfiguration step will be needed to locate that cell. This is given by  $\lfloor \log_2 n \rfloor + 1$  or  $\lceil \log_2 n \rceil$ . Q.E.D.

**Corollary:** If an  $n$ -bit R-MISR is superimposed on the internal boundary scan input nodes of an IC connected to  $N$  communicating ICs of an MCM under test, such that each contributes  $K$  connections, ( $n = NK$ ), then the minimum number of optimal R-MISR reconfigurations necessary to diagnose a faulty communicating IC under the assumption of multiple faults on interconnections of a single IC is  $\lceil \log_2 N \rceil$ .

### MISR Reconfiguration Algorithm

```

Algorithm reconfigure_misr (R- MISR) {
  let B = R-MISR size(no. of bits);
  create circular MISR structure such that last
  cell of MISR points to the first one;
  randomly select pivot MISR cell;
  pivot_windowsize = B/2;
  use this pivot cell to partition R- MISR into 2
  MISRs of size B/2 namely MISR_A and
  MISR_B;
  run signature analysis on both MISR partitions;
  new_fault_list = net list of faulty partition;
  if (MISR_A signature is faulty)
    pivot = pivot + pivot_windowsize;
  else
    pivot = pivot - pivot_windowsize;
  if (first_reconfiguration == TRUE)
    fault_list = new_fault_list;
  else /* subsequent reconfigurations */
    fault_list = create_common_faultlist
                (fault_list, new_fault_list);
  return (fault_list);
}

```

## 7.4 MISR reconfiguration Under Multiple Faults

Performance related faults such as cross talk and/or ground bounce affect more than one interconnect at a time. Thus it is important to analyze the effectiveness of the MISR reconfiguration technique in the diagnosis of a set of faulty interconnects. Below, some important terms which are used in the following analysis are defined.

**Logical MISR:** This is a Local MISR of a faulty IC identified in fault detection phase. It consists of logical grouping of interconnects (input B-S cells) belonging to each IC communicating to the faulty IC. An example of a logical MISR of a faulty IC C3 with four neighboring ICs (C1, C2, C4 and C5) is shown in the Figure 30. The logical MISR increases wiring complexity, but fault diagnosis can be achieved with optimal MISR reconfigurations.

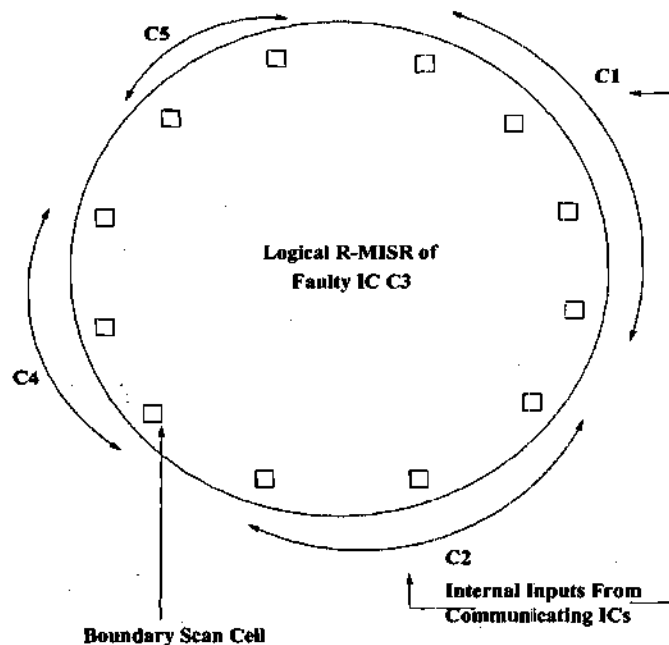
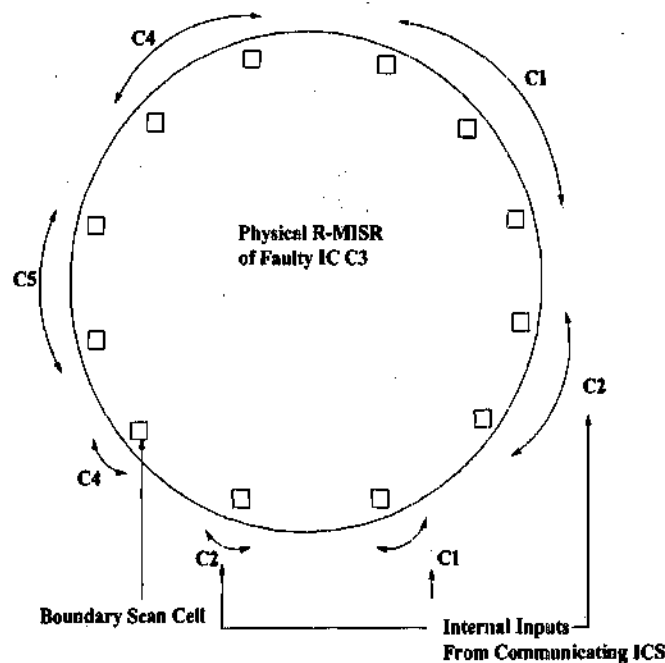


FIGURE 30. Logical MISR

**Physical MISR:** This is a Local MISR of a faulty IC constructed by connecting all input boundary scan cells without the constraint of logical grouping of cells at an IC level. An example of a physical MISR of the faulty IC C3 of Figure 30 can be constructed from layout information and is shown in Figure 31. The physical MISR is easy to implement due to simple wiring but adds to reconfiguration complexity.



**FIGURE 31. Physical MISR**

**Fault Group:** The *fault group* is a set of consecutive internal input boundary scan cells of a Local MISR (Logical or Physical). The size of a fault group is the cardinality of the set which constitutes the fault group.

*Net Level Cut:* A net level cut( $p1, p2$ ) of a  $n$  bit R-MISR is a partition of  $n$  cells into  $p1$  and  $p2 = n - p1$ .

*Edge* is a set of any two R-MISR cells. We say an edge *crosses* the cut( $p1, p2$ ) if one of its cells is in  $p1$  and the other is in  $p2$ . An *Adjacent Edge* is a set of two adjacent or consecutive R-MISR cells.

*Optimal Cut:* A Net Level Cut is optimal if it satisfies theorem 1.

*IC Level Cut:* This is a Net Level Cut such that an adjacent edge that crosses this cut has its two cells connected to two different communicating ICs (or doesn't have its two cells connected to any one IC).

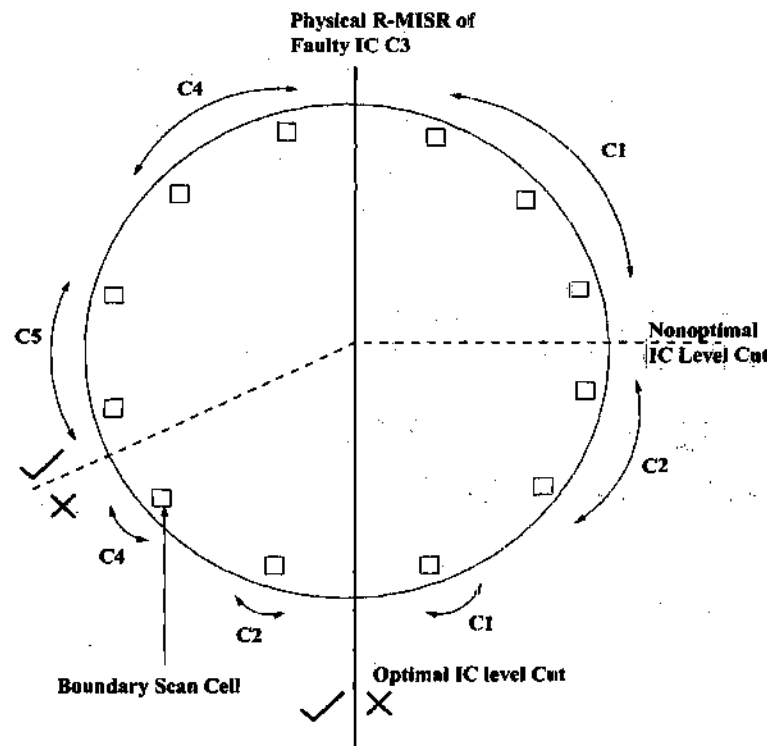
*Fault Group Level Cut:* This is a Net Level Cut such that an adjacent edge that crosses this cut doesn't have its two cells belonging to a single fault group.

*MISR reconfiguration at IC Level:* It is a reconfiguration of a R-MISR using a IC level cut.

*MISR reconfiguration at fault group Level:* It is a reconfiguration of a R-MISR using a fault group level cut.

It is assumed that the size of R-MISR is much larger than the size of the fault group. First consider a simple fault model in which the faulty interconnects in the fault group belong to a single IC. This is typical in the case of performance faults introduced due to a ground bounce. Then by the corollary proved in the previous section, the faulty IC can be diagnosed in  $\lceil \log_2 N \rceil$  reconfigurations of a logical MISR at IC level, where  $N$  is number of

communicating ICs. Here it is assumed that the faulty IC has equal number of inputs ( $K$ ) from each of its communicating ICs. In practice, this might not necessarily be the case. Then the logical MISR reconfiguration at IC level using optimal cut will not be possible as shown in Figure 32. This leads to the violation of theorem 1 and aliasing resulting due to this violation may not be acceptable. Therefore we introduce the notion of a fault group.



**FIGURE 32. Non-optimal MISR Reconfiguration at IC Level**

The fault group may contain nets belonging to different ICs. For a R-MISR size of  $n$  bits and  $k$  is the size of a fault group, each fault group will have  $k$  cells if  $n \bmod k$  is 0. In this case, since size of a fault group is fixed for a given R-MISR, the MISR reconfiguration at fault group level can be achieved using optimal cut. Therefore we can achieve diagnosis of a faulty set of interconnects with minimum number of reconfigurations of a logical as



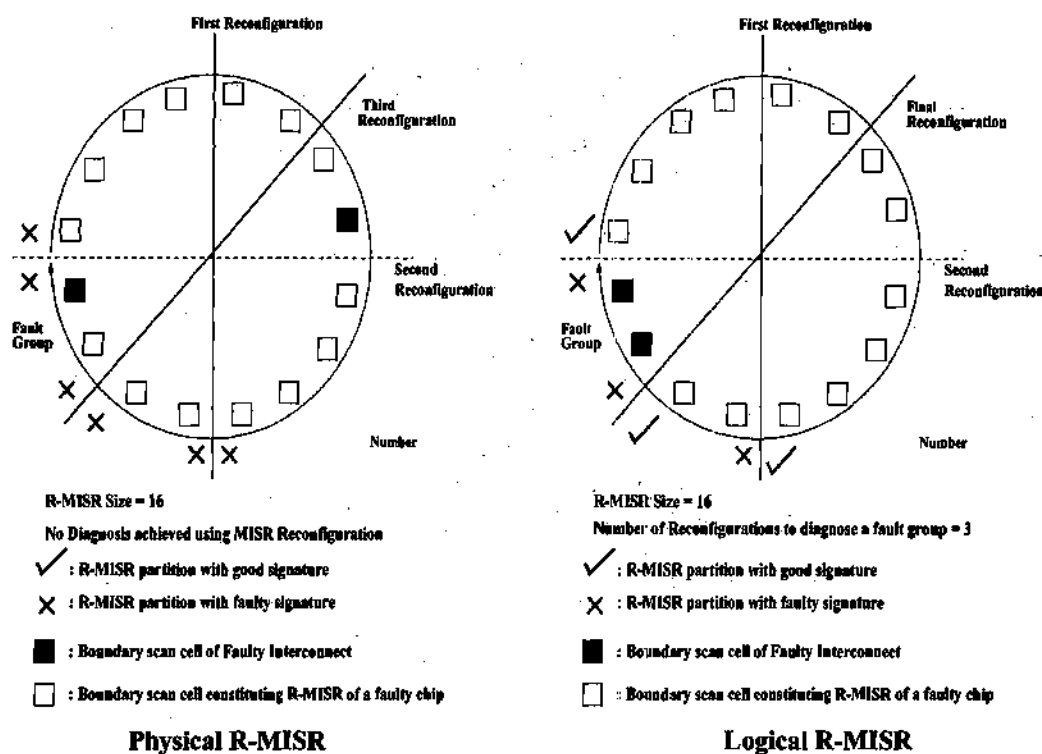
well as a physical MISR as per theorem 2. Under the multiple fault model, MISR reconfiguration at the fault group level consists of a CUT at boundary of predefined fault group. At the end of reconfiguration steps, we identify the fault group which consists of the cells connected to interconnects with performance related fault. A MISR reconfiguration at IC level is a special case of a MISR reconfiguration at a fault group level. In this case, each communicating IC is a fault group. Thus for a Logical MISR, each fault group contains nets from one and only one IC. In case of a physical MISR, each fault group contains nets from one IC but different fault group may contain different nets from the same IC. The fault group size can be a parameter for a MISR reconfiguration algorithm. This parameter can be determined by the statistical analysis of the manufacturing and process data obtained over the period as well as the simulation models. It can also be determined by the layout or routing of the MCM under test and its functional model. This data identifies which adjacent nets are activated in a functional mode or which pair of active and passive nets are adjacent.

## **7.5 Discussion**

The reconfigurable MISR can be easily implemented using the 1149.1 boundary scan standard. The boundary scan architecture allows the following provisions for the boundary scan register (BSR): (1) the BSR cells for the various input/output signals, output enable signals and direction control signals of the system logic may be assembled in the register in any order. (2) Additional cells that are not capable of controlling and/or observing the state of a system pin can be added to the BSR [46]. These provisions can be used

at the expense of minimal hardware overhead to implement R-MISR using split boundary scan register technique for testing board interconnect [80]. The use of split boundary scan register enables implementation of Input Boundary Scan Register(IBSR), Output Boundary Scan Register(OBSR) and Control Boundary Scan Register(CBSR) using multiplexers. They can be concatenated to form a normal Boundary Scan Register. We use an IBSR with the additional hardware to realize R-MISR. We use an OBSR to implement cascadable LFSRs of our distributed BIST technique. Because of the separation of OBSR and CBSR, this distributed BIST technique can be also applied to 3 state bidirectional interconnects. Using the second provision of the boundary scan standard, dummy cells can be inserted in the R-MISR to reduce shift correlation.

There is a trade-off between realizing a Logical R-MISR and a Physical R-MISR. The Logical R-MISR implementation using an IBSR will require special routing and layout design. This adds to the complexity at the physical design level, but reduces the algorithmic complexity to diagnose multiple faults. For example, multiple faults on the interconnects shown in Figure 33 can not be diagnosed accurately with optimal number of reconfigurations and without compromising the diagnostic resolution. But by constructing a Logical R-MISR, the B-S cells of these interconnects can be made adjacent to each other. This enables MISR reconfiguration at fault group level and can be diagnosed using optimal number of MISR reconfigurations. An IBSR for Physical R-MISR can be designed without added complexity of physical design because of the simple routing between adjacent boundary scan cells.



**FIGURE 33. Diagnosis Using Logical R-MISR v/s Physical R-MISR**

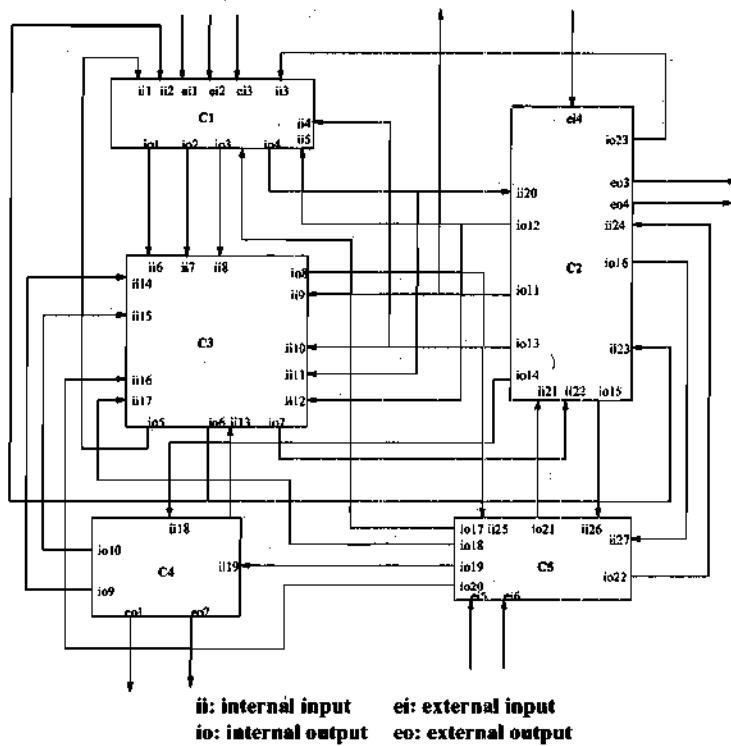
The distributed BIST controller enables distributed global diagnosis through the activation of proper test modes and achieves local diagnosis through MISR reconfiguration. The simplest controller finite state machine has four states for each test mode and can be implemented using only 2 flipflops and a few gates.

General methods to reduce MISR signature aliasing probability are either to increase the size of the MISR or to use repeatable signature technique. With the Reconfigurable MISR, both these techniques can be employed efficiently. Using dummy B-S cells, R-MISR size can be increased and also adjusted so that each R-MISR partition has the same

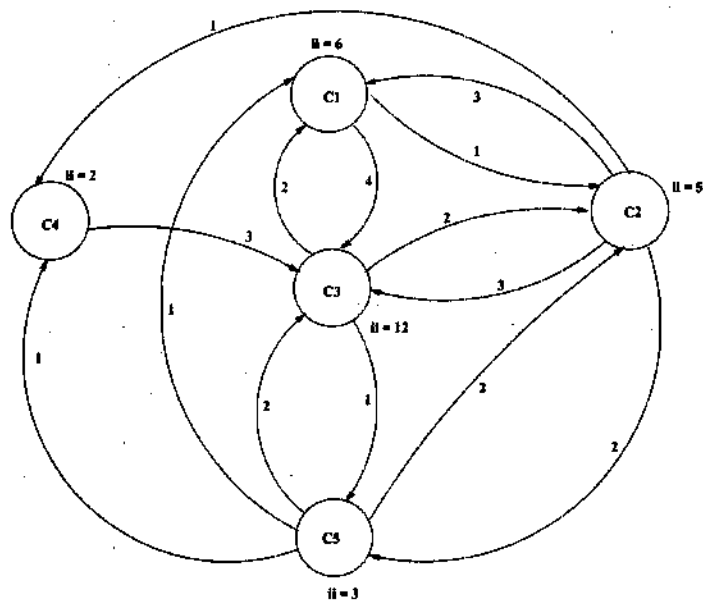
number of cells. The MISR reconfiguration technique checks signature of both partitions after each reconfiguration. Thus repeatable signature checking minimizes the aliasing phenomenon. Moreover, since reconfiguration algorithm achieves diagnosis in optimal number of reconfigurations, the total test time is optimized.

## **7.6 Experimental Results**

A software tool has been developed to implement the distributed BIST diagnosis technique using the proposed MISR reconfiguration algorithm. This tool is parameterized for reading the MCM netlist. The MCM graph model is then extracted from this netlist. The nodes represent constituent ICs of the MCM under test and directed arcs represent internal inputs and outputs of each IC. Global and local diagnosis for injected faults on interchip interconnections is performed. The Reconfigurable MISR of each of the faulty ICs is simulated and the number of optimal reconfigurations required for complete diagnosis is computed. The example MCM under test is shown in Figure 34. The graph model of the example MCM is shown in Figure 35 . The simulation results for various sizes of the R-MISR of the faulty IC with 11 internal inputs are summarized in Table 10 .



**FIGURE 34. Example MCM Under Test**



**FIGURE 35. Graph Model of MCM Under Test**

**TABLE 10. Simulation Results for Distributed BIST Diagnosis Using R-MISR**

Faulty Chip	Number of Reconfigurations for Given R-MISR Size			
Faulty Net Id	12	24	50	100
ii0	4	5	6	7
ii1	4	5	6	7
ii2	4	5	6	7
ii3	4	5	6	7
ii4	4	5	6	7
ii5	4	5	6	7
ii6	4	5	6	7
ii7	4	5	6	7
ii8	4	5	6	7
ii9	4	5	6	7
ii10	4	5	6	7
ii11	4	5	6	7

## 7.7 Hardware Cost

Since ICs are getting more complex and silicon cost is falling, the cost of area required for embedding distributed LFSRs and MISRs in ICs is well justified. The design of the BIST controller makes hierarchical BIST a possibility. The Distributed BIST hardware consists of multiplexers and AND gates required to dynamically reconfigure MISRs and sequential logic for the BIST controller. It is expected that the hardware cost will be smaller for a larger size and hence for a practical length MISR. An implementation prototype of a BIST test pattern generator and a signature analyzer based on nongroup and 90-150 CA has shown encouraging results [81]. The instruction length decoder IC was designed with dynamic CMOS gates with 120K transistors, having an area of 3090 X

3344 microns. Area of embedded LFSM and BIST Controller was found to be just 3% of the total chip area.

# **CHAPTER VIII**

## **DFT TOOL FOR MCM**

### **INTERCONNECT TEST**

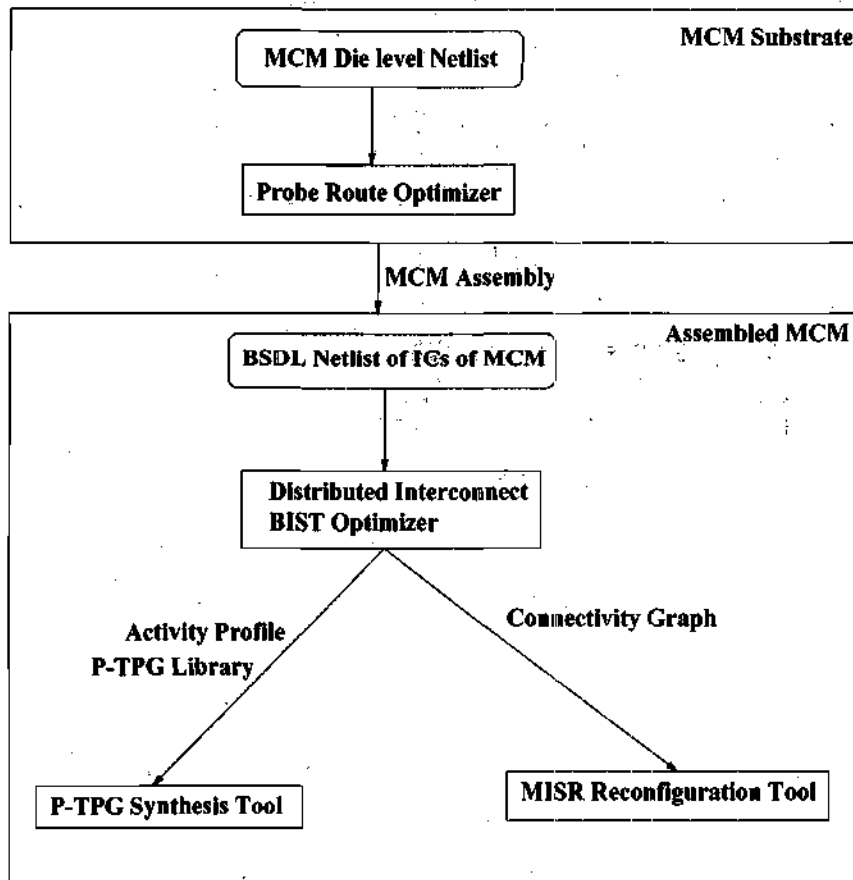
In this chapter, a DFT tool developed in this research is described. This tool provides a seamless integrated CAD environment which enables a designer to automate the design process using algorithms presented in earlier chapters.

#### **8.1 Architectural Framework**

The top level software architectural framework of the DFT tool developed based on the algorithms developed in this research is shown in Figure 36. Three software components (probe route optimizer, P-TPG synthesis tool and MISR reconfiguration tool) are written in 'C' and can be easily integrated together. The structure of P-TPG synthesis tool is shown in Figure 37 . It consists of following components and their intercommunications. A *Design Engine* takes switching probability data of IUT, the size of P-TPG and associated primitive polynomial coefficients. The preprocessor analysis is done to identify



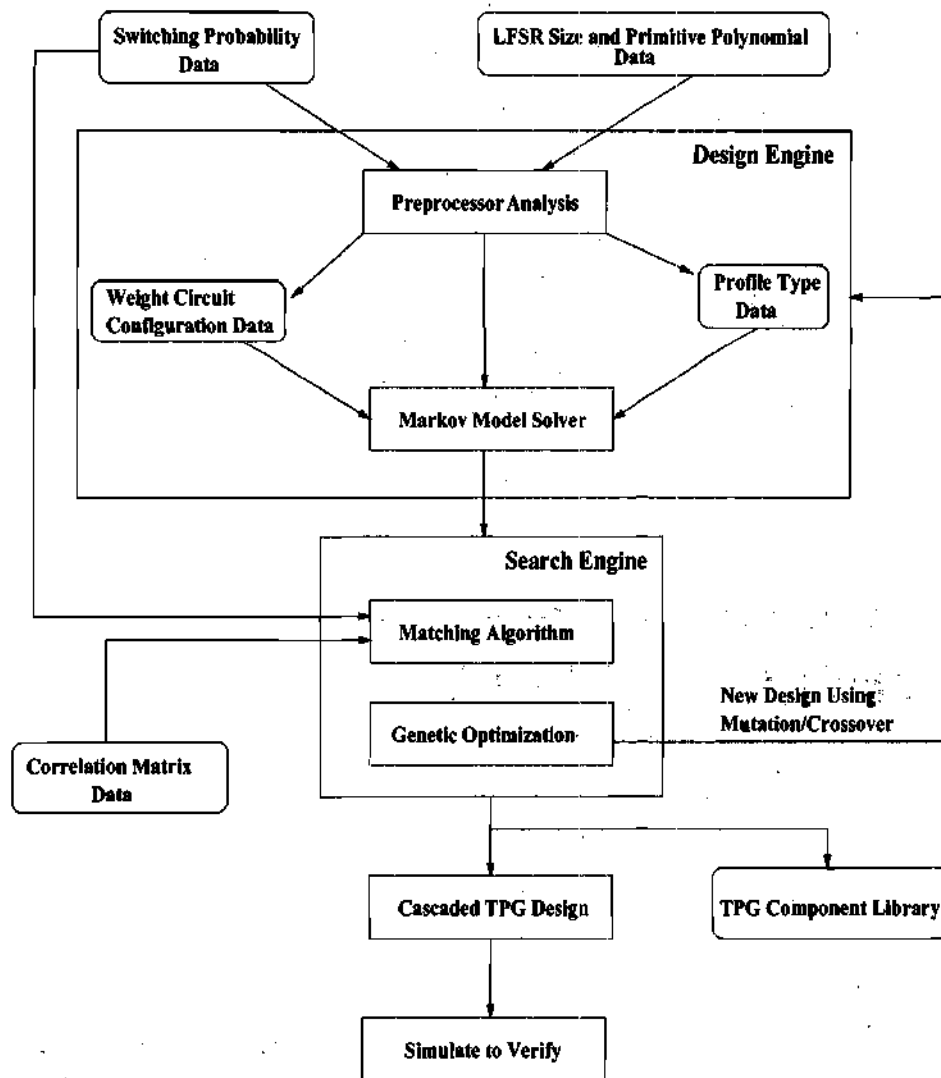
the type of switching profile of the given IUT, for example, the switching profile can be either increasing or flat, etc. The advantage of preprocessor is that it allows use of selective weighting circuit configurations in the P-TPG design process and improves the search efficiency. The design of P-PTG is given by its specifications as described in Section 6.4.



**FIGURE 36. Software Architecture of DFT Tool**

There are numerous combinations of such designs which can recreate the IUT activity profile and to select the best design out of these combinations is a classical combinatorial optimization problem. In the literature, genetic algorithms have been shown to be very effective for this kind of search and optimization problems. Therefore *Search Engine* of

our BIST synthesis tool is based on genetic algorithm. Each P-TPG design is evaluated by computing analytically the switching probabilities and correlations using *Markov Model Solver* based on the theory of gaussjordon elimination and computing associated cost function.



**FIGURE 37. Structure of P-TPG Synthesis Tool**

Genetic algorithm to perform efficient search over various configurations while minimizing the cost function for optimized matching. At each iteration of the genetic algorithm, a fixed length chromosome is operated on using a mutation operator and a crossover operator. The chromosome is constructed by concatenating strings of genes. Each gene represents a binary digit 0 or 1. First string of genes has a length equal to a given length of P-TPG and represents AND gate locations for a given primitive polynomial. Second string selects the profile type to be implemented to match that of a given IUT. Third string of genes indicate weighting circuit configuration to be designed at the output of the P-TPG. The last string of genes chooses one of different input switching probability values within a range of 0.1 to 0.9. For example, the chromosome consisting of 13 genes is represented by a string **1100001010101**. First eight bits 11000010 represent location of AND gates at the position represented by 1 in the string. Next gene indicates type of profile to be generated. A value of 1 indicates flat profile and a value of 0 indicates increasing profile. Next gene selects one of the two weighting circuit configurations (C1 or C2) on the output of P-TPG depending on whether its value is 0 or 1. Last three genes denote one of eight values of input switching probabilities (0.2 to 0.9). Thus this chromosome represents a design of a 8 bit P-TPG component with non-primitive polynomial 11000010 implementing increasing switching profile using weighting circuit configuration C1 and input switching probability of 0.7. Any new chromosome obtained from this one by the operation of mutation or crossover represents a design of a new P-TPG component with new non-primitive polynomial, input switching, and weighting circuit configuration depending upon the value of the respective genes. The fitness value of this new chromosome is evaluated by computing

the cost function using Markov model solver, as described before. The *Matching Algorithm* described in section 6.7 is used to match switching activity profile of the P-TPG design to that of the IUT by using this cost function. The genetic algorithm iterates over large range of generations applying mutation and crossover operations and selects the chromosome with the best fitness. This chromosome gives the design of optimal P-TPG for a given IUT. A *TPG Component Library* consists of P-TPG components with best fitness, making them reusable for future designs. The synthesis tool implementing this CAD methodology is parameterized for selecting various options such as LFSR options (feedback polynomial, seed, bit size, internal/external type), configuration options (profile, weighting circuit configuration, multiplexer configuration) and cascade options (i/p switching probability and output bit selection to control i/p of next stage). Experiments were done with 8-bit internal-type LFSRs with primitive polynomial  $x^8 + x^6 + x^5 + x + 1$ .

**TABLE 11. Switching Profile Matching by Cascaded P-TPG Structure**

o/p bit p- tpg	Cascaded P1-P2			P-TPG		
	$P_{in} = 0.22$ $P_{next} = 0.52$			$P_{in} = 0.22$ $P_{next} = 0.52$		
	Increasing Profile(P2) with Configuration C1			Decreasing Profile(P2) with Configuration C2		
	P1	P2	P2'	P1	P2	P2'
0	0.02	0.00	0.00	0.02	0.61	0.51
1	0.02	0.00	0.01	0.02	0.28	0.38
2	0.03	0.01	0.02	0.03	0.09	0.16
3	0.09	0.02	0.03	0.09	0.09	0.16
4	0.17	0.07	0.08	0.17	0.06	0.13
5	0.24	0.13	0.22	0.24	0.00	0.01
6	<b>0.50</b>	0.45	0.46	<b>0.50</b>	0.00	0.01
7	0.50	0.45	0.46	0.50	0.00	0.00

Table 11 demonstrates that the matching profile of a IUT segment using P-TPG component P2' with input switching probability 0.52 can be recreated by cascaded structure P1-P2 when input to P2 is controlled by output Bit 6 of P1 with switching probability 0.5.

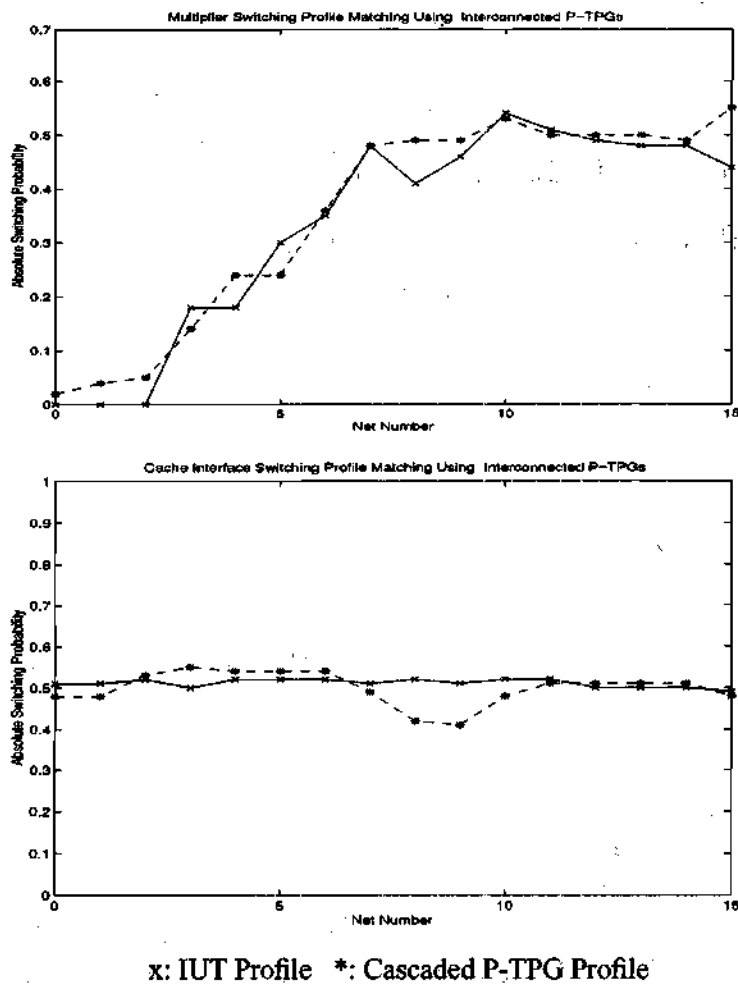
P-TPG components with different type of profiles (increasing-decreasing) can also be interconnected. The results of the switching profile matching with cascaded P-TPGs for two IUTs namely floating point multiplier data path and cache-processor interface are shown in Table 12 and Table 13 respectively. Figure 38 shows respective plots comparing actual profile and matched profile. The correlation matrix entries were also matched.

**TABLE 12. IUT Profile Matching for Floating Point Multiplier**

Out-put Bit of P-TPG	Increasing-Flat profile Switching Profile of 16 bit IUT				$P_{in} = 0.22$	
	Segment 1		Segment 2		Switching Profiles of 8 bit P-TPG Components	
					P1	P2
0	0	0.00	7	0.41	0.02	0.49
1	1	0.00	8	0.46	0.04	0.49
2	2	0.00	9	0.54	0.05	0.53
3	3	0.18	10	0.51	0.14	0.50
4	4	0.18	11	0.49	0.24	0.50
5	5	0.30	12	0.48	0.24	0.50
6	6	0.35	13	0.48	0.36	0.49
7	7	0.48	15	0.44	0.48	0.55
Correlations of Segment 2						
$\begin{bmatrix} 1.00 & 0.19 & 0.20 & 0.20 & 0.24 & 0.21 & 0.16 & 0.19 \\ 0.19 & 1.00 & 0.27 & 0.28 & 0.20 & 0.24 & 0.19 & 0.20 \\ 0.20 & 0.27 & 1.00 & 0.23 & 0.23 & 0.25 & 0.26 & 0.22 \\ 0.20 & 0.28 & 0.23 & 1.00 & 0.26 & 0.22 & 0.23 & 0.24 \\ 0.24 & 0.20 & 0.23 & 0.26 & 1.00 & 0.25 & 0.23 & 0.23 \\ 0.21 & 0.24 & 0.25 & 0.22 & 0.25 & 1.00 & 0.26 & 0.21 \\ 0.16 & 0.19 & 0.26 & 0.23 & 0.23 & 0.26 & 1.00 & 0.20 \\ 0.19 & 0.20 & 0.22 & 0.24 & 0.23 & 0.21 & 0.20 & 1.00 \end{bmatrix}$						
Correlations of P2						
$\begin{bmatrix} 1.00 & 0.28 & 0.27 & 0.26 & 0.26 & 0.25 & 0.23 & 0.30 \\ 0.28 & 1.00 & 0.26 & 0.24 & 0.26 & 0.26 & 0.25 & 0.26 \\ 0.27 & 0.26 & 1.00 & 0.26 & 0.27 & 0.25 & 0.27 & 0.29 \\ 0.26 & 0.24 & 0.26 & 1.00 & 0.23 & 0.23 & 0.24 & 0.28 \\ 0.26 & 0.26 & 0.27 & 0.23 & 1.00 & 0.23 & 0.23 & 0.25 \\ 0.25 & 0.26 & 0.25 & 0.23 & 0.23 & 1.00 & 0.23 & 0.32 \\ 0.23 & 0.25 & 0.27 & 0.24 & 0.23 & 0.23 & 1.00 & 0.25 \\ 0.30 & 0.26 & 0.29 & 0.28 & 0.25 & 0.31 & 0.25 & 1.00 \end{bmatrix}$						

**TABLE 13. IUT Profile Matching for Cache-Processor Interface**

Output Bit of P-TPG	Flat profile Switching Profile of 16 bit IUT				$P_{in} = 0.9$	
	Segment 1		Segment 2		Switching Profiles of 8 bit P-TPG Components	
					P1	P2
0	0	0.51	7	0.52	0.48	0.36
1	1	0.51	8	0.51	0.48	0.36
2	2	0.52	9	0.52	0.53	0.51
3	3	0.50	10	0.52	0.55	0.54
4	4	0.52	11	0.50	0.54	0.53
5	5	0.52	12	0.50	0.54	0.53
6	6	0.52	13	0.50	0.54	0.53
7	7	0.51	15	0.49	0.49	0.48
Cost Function					0.2	0.44



**FIGURE 38. Plots of Matched Switching Profiles**

Switching probabilities of bit 7 and bit 5 (bold faced values) act as  $P_{next}$  to P2. It is observed that switching probability greater than 0.5 can be obtained by choosing 2 least correlated bits and XORing them. Such XOR operation for bit 1 and bit 6 to get higher input switching for P2 is performed to get improved matching for cache interface. An improvement in the cost function is shown in Table 14.

**TABLE 14. Improved Profile Matching for Cache-Processor Interface Using XOR**

Output Bit of P-TPG	Increasing profile Switching Profile of 16 bit IUT				$P_{in} = 0.9$	
	Segment 1		Segment 2		Switching Profiles of 8 bit P-TPG Components	
					P1	P2
0	0	0.51	7	0.52	0.48	0.42
1	1	0.51	8	0.51	0.48	0.41
2	2	0.52	9	0.52	0.53	0.48
3	3	0.50	10	0.52	0.55	0.51
4	4	0.52	11	0.50	0.54	0.51
5	5	0.52	12	0.50	0.54	0.51
6	6	0.52	13	0.50	0.54	0.51
7	7	0.51	15	0.49	0.49	0.48
Cost Function					0.2	0.29

The activity profile for a multiplier IUT was matched using the synthesis tool. The number of generations allowed by the genetic algorithm were varied as shown in the first column of Table 15. The best P-TPG designs obtained and the corresponding matching costs are shown in columns 2 and 3 for the first segment of the IUT. Similarly columns 4 and 5 show the best components for matching a second segment of the IUT. It is obvious that by increasing the number of generations, an optimal cost can be obtained at the cost of CPU time. This cost was computed as a sum of difference between the values of given switching probabilities and the correlations of the IUT and the same values computed analyti-

cally using the Markov model solver for the best design obtained in columns 2 and 4. These analytically computed designs were simulated by implementing all the specifications of the P-TPG both in hardware and software.

A hardware prototype consisted of the entity-architecture pairs for a P-TPG behavioral

**TABLE 15. Analytical Computation of Cost Function Using Genetic Algorithm and Markov Solver**

Multiplier IUT	Analytical Design				
	Component 1		Component 2		CPU Time (seconds)
Number of Generations	Best Fit Chromosome	Cost Function	Best Fit Chromosome	Cost Function	
2	1100001000011	1.404	0000101011101	2.198	1843.45
4	0001010101110	1.382	00011111110110	1.909	2043.98
10	0010101000010	1.348	1110011111110	1.712	2906.23
20	1010110101110	1.3004	1001001010100	1.5	4198.03

model and a tester behavioral model which consisted a clock generator to test a given P-TPG. A testbench entity-architecture pair was designed to interface the tester to a P-TPG design to be simulated and the VHDL procedures were used to compute the activity profiles using a file I/O. The comparison of the analytical designs and the simulated designs is divided into three tables. Table 16 compares the matching cost for a given IUT. An analytical design section of the table elaborates various specifications obtained from the genetic algorithm and used in the hardware simulations. It can be seen that hardware prototype cost shown in the last column matches well with the one computed analytically. Table 17 and Table 18 denote the matching of 16 bit multiplier IUT switching profile and its correlations respectively using the interconnection of two best 8 bit P-TPG components obtained from the design engine and the search engine of the synthesis tool and verified by



a hardware prototype. The results show two best designs obtained by 10 and 20 generations of a genetic algorithms. The correlations of each component match very well with each segment of the given IUT, in both the cases as shown in columns of Table 18 . The simulated design verify an analytical design satisfactorily, confirming the thesis.

**TABLE 16. Comparison of Analytical Design and Simulated Design**

Multiplier IUT	Analytical Design						Simulated Design for Primitive Poly. 10001101
	Best Fit Chromosome	Cost Function	Non-prim. Polynomial	Profile Type	Weight Config.	Input Switching	Cost Function
Component 1	1100001000011	1.40	11000010	I	C1	0.5	1.34
	0001010101110	1.38	*00010101	I	C2	0.8	1.90
	0010101000010	1.35	00101010	I	C1	0.4	1.32
	1010110101110	1.30	*10101101	I	C2	0.8	1.19
Component 2	0000101011101	2.19	00001010	F	C2	0.7	1.98
	0001111110110	1.90	*00011111	F	C1	0.8	1.83
	1110011111110	1.71	11100111	F	C2	0.8	1.69
	1001001010100	1.54	*10010010	F	C1	0.6	1.64

**TABLE 17. Switching Profile Matching of Multiplier IUT Using Analytical and Simulation Design**

Output Bit of P-TPG	Increasing-Flat profile Switching Profile of 16 bit IUT				Switching Profiles of 8 bit P-TPG Components							
					P-TPG Component Design I (10 Generations)				P-TPG Component Design II (20 Generations)			
					Analytical Design		Simulated Design		Analytical Design		Simulated Design	
	Segment 1		Segment 2		P1	P2	P1	P2	P1	P2	P1	P2
0	0	0.00	7	0.41	0.00	0.48	0.01	0.44	0.00	0.33	0.00	0.36
1	1	0.00	8	0.46	0.00	0.50	0.02	0.51	0.00	0.53	0.01	0.47
2	2	0.00	9	0.54	0.00	0.50	0.05	0.52	0.00	0.57	0.01	0.48
3	3	0.18	10	0.51	0.13	0.48	0.11	0.53	0.13	0.53	0.11	0.46
4	4	0.18	11	0.49	0.16	0.51	0.20	0.51	0.16	0.49	0.14	0.50
5	5	0.30	12	0.48	0.16	0.50	0.20	0.50	0.16	0.49	0.14	0.50
6	6	0.35	13	0.48	0.26	0.50	0.35	0.50	0.37	0.49	0.36	0.50
7	7	0.48	15	0.44	0.57	0.50	0.48	0.49	0.49	0.52	0.53	0.50

**TABLE 18. Correlation Matrix Matching of Multiplier IUT Using Analytical and Simulation Design**

16 bit IUT	Analytical Design	Simulated Design
P-TPG Design I Obtained with 10 Generations		
Correlations of Segment 1	Correlations of P1	Correlations of P1
<div> 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 0.00 1.00 0.09 0.09 0.06 0.10  0.00 0.00 0.00 0.09 1.00 0.11 0.06 0.08  0.00 0.00 0.00 0.09 0.11 1.00 0.17 0.16  0.00 0.00 0.00 0.06 0.06 0.17 1.00 0.28  0.00 0.00 0.00 0.10 0.08 0.16 0.28 1.00 </div>	<div> 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 0.00 1.00 0.12 0.12 0.10 0.08  0.00 0.00 0.00 0.12 1.00 0.16 0.15 0.12  0.00 0.00 0.00 0.12 0.16 1.00 0.15 0.12  0.00 0.00 0.00 0.10 0.15 0.15 1.00 0.15  0.00 0.00 0.00 0.08 0.12 0.12 0.15 1.00 </div>	<div> 1.00 0.01 0.01 0.01 0.01 0.01 0.01 0.01  0.01 1.00 0.02 0.02 0.02 0.02 0.02 0.01  0.01 0.02 1.00 0.05 0.04 0.04 0.03 0.02  0.01 0.02 0.05 1.00 0.01 0.10 0.06 0.05  0.01 0.02 0.04 0.10 1.00 0.20 0.16 0.09  0.01 0.02 0.04 0.10 0.20 1.00 0.16 0.09  0.01 0.02 0.03 0.06 0.16 0.16 1.00 0.17  0.01 0.01 0.02 0.05 0.09 0.09 0.17 1.00 </div>
Correlations of Segment 2	Correlations of P2	Correlations of P2
<div> 1.00 0.19 0.20 0.20 0.24 0.21 0.16 0.19  0.19 1.00 0.27 0.28 0.20 0.24 0.19 0.20  0.20 0.27 1.00 0.23 0.23 0.25 0.26 0.22  0.20 0.28 0.23 1.00 0.26 0.22 0.23 0.24  0.24 0.20 0.23 0.26 1.00 0.25 0.23 0.23  0.21 0.24 0.25 0.22 0.25 1.00 0.26 0.21  0.16 0.19 0.26 0.23 0.23 0.26 1.00 0.20  0.19 0.20 0.22 0.24 0.23 0.21 0.20 1.00 </div>	<div> 1.00 0.25 0.24 0.23 0.24 0.25 0.24 0.25  0.25 1.00 0.25 0.25 0.25 0.26 0.25 0.26  0.24 0.25 1.00 0.24 0.25 0.23 0.25 0.26  0.23 0.25 0.24 1.00 0.25 0.24 0.24 0.24  0.24 0.25 0.25 0.25 1.00 0.26 0.26 0.26  0.25 0.26 0.23 0.24 0.26 1.00 0.26 0.25  0.24 0.26 0.25 0.24 0.26 0.26 1.00 0.25  0.25 0.26 0.26 0.24 0.26 0.25 0.25 1.00 </div>	<div> 1.00 0.22 0.22 0.21 0.23 0.21 0.23 0.21  0.22 1.00 0.28 0.26 0.24 0.27 0.26 0.24  0.22 0.28 1.00 0.29 0.25 0.26 0.25 0.24  0.21 0.26 0.29 1.00 0.27 0.27 0.27 0.27  0.23 0.24 0.25 0.27 1.00 0.24 0.25 0.26  0.21 0.27 0.26 0.27 0.24 1.00 0.24 0.25  0.23 0.26 0.25 0.27 0.25 0.24 1.00 0.25  0.21 0.24 0.24 0.27 0.26 0.25 0.25 1.00 </div>
P-TPG Design II Obtained with 20 Generations		
Correlations of Segment 1	Correlations of P1	Correlations of P1
<div> 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 0.00 1.00 0.09 0.09 0.06 0.10  0.00 0.00 0.00 0.09 1.00 0.11 0.06 0.08  0.00 0.00 0.00 0.09 0.11 1.00 0.17 0.16  0.00 0.00 0.00 0.06 0.06 0.17 1.00 0.28  0.00 0.00 0.00 0.10 0.08 0.16 0.28 1.00 </div>	<div> 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 1.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 1.00 0.00 0.00 0.00 0.00 0.00  0.00 0.00 0.00 1.00 0.13 0.13 0.07 0.06  0.00 0.00 0.00 0.13 1.00 0.16 0.10 0.07  0.00 0.00 0.00 0.13 0.16 1.00 0.10 0.07  0.00 0.00 0.00 0.07 0.10 0.10 1.00 0.17  0.00 0.00 0.00 0.06 0.07 0.07 0.17 1.00 </div>	<div> 1.00 0.00 0.00 0.00 0.00 0.00 0.00 0.00  0.00 1.00 0.01 0.01 0.00 0.00 0.00 0.00  0.00 0.01 1.00 0.01 0.00 0.00 0.00 0.00  0.00 0.01 0.01 1.00 0.09 0.09 0.08 0.05  0.00 0.00 0.00 0.09 1.00 0.14 0.13 0.07  0.00 0.00 0.00 0.09 0.14 1.00 0.13 0.07  0.00 0.00 0.00 0.08 0.13 0.13 1.00 0.22  0.00 0.00 0.00 0.05 0.07 0.07 0.22 1.00 </div>
Correlations of Segment 2	Correlations of P2	Correlations of P2
<div> 1.00 0.19 0.20 0.20 0.24 0.21 0.16 0.19  0.19 1.00 0.27 0.28 0.20 0.24 0.19 0.20  0.20 0.27 1.00 0.23 0.23 0.25 0.26 0.22  0.20 0.28 0.23 1.00 0.26 0.22 0.23 0.24  0.24 0.20 0.23 0.26 1.00 0.25 0.23 0.23  0.21 0.24 0.25 0.22 0.25 1.00 0.26 0.21  0.16 0.19 0.26 0.23 0.23 0.26 1.00 0.20  0.19 0.20 0.22 0.24 0.23 0.21 0.20 1.00 </div>	<div> 1.00 0.21 0.22 0.19 0.18 0.20 0.20 0.19  0.21 1.00 0.22 0.22 0.23 0.24 0.25 0.22  0.22 0.22 1.00 0.22 0.24 0.24 0.24 0.26  0.19 0.22 0.22 1.00 0.22 0.24 0.27 0.22  0.18 0.23 0.24 0.22 1.00 0.23 0.23 0.24  0.20 0.24 0.24 0.24 0.23 1.00 0.26 0.25  0.20 0.25 0.25 0.27 0.23 0.26 1.00 0.25  0.19 0.22 0.26 0.22 0.24 0.25 0.25 1.00 </div>	<div> 1.00 0.19 0.19 0.17 0.20 0.17 0.18 0.20  0.19 1.00 0.22 0.20 0.25 0.24 0.23 0.22  0.19 0.22 1.00 0.22 0.24 0.27 0.22 0.23  0.17 0.20 0.22 1.00 0.24 0.24 0.22 0.25  0.20 0.25 0.24 0.24 1.00 0.24 0.25 0.23  0.17 0.24 0.27 0.24 0.24 1.00 0.24 0.26  0.18 0.23 0.22 0.22 0.25 0.24 1.00 0.25  0.20 0.22 0.23 0.25 0.23 0.26 0.25 1.00 </div>

## 8.2 Discussion

This methodology has several advantages. The plots of matched switching profiles compare the IUT profile with the profile obtained by software simulation using the synthesis tool as well as profile obtained by the hardware simulation of the synthesized design using Leapfrog VHDL simulator from Cadence.

### 8.2.1 Interconnect Test Generation for Catastrophic Faults

Random patterns are excellent for detecting typical interconnect defects such as opens and shorted traces and the bridging faults because of high bit toggle rate [55]. In *catastrophic test mode*, P-TPG is to be used in a shift register mode where input bit  $i$  is set to 0, polynomial selector selects a polynomial with all zero coefficients and weighting circuitry is bypassed. It can then be used to implement a boundary walking test to achieve 100% coverage of stuck-at 1 faults, stuck-at 0 faults and faults due to AND & OR type of shorts [82]. The distributed BIST application using the P-TPG components and an interconnect fault diagnosis strategy is explained in detail in [59].

### 8.2.2 Interconnect Test Generation for Performance Faults

In *performance test mode*, P-TPG feature is restored and is used as described in section 5 to generate an activity profile. This mode helps generate tests for the cross-talk faults as well as the ground bounce related faults.

Both the above modes are used carefully so that issues such as activation of multiple drivers, bidirectional pins are taken care of. An ATPG driver selection algorithm [83] as

well as split boundary scan register technique can be integrated with the P-TPG design methodology to address these important issues.

### **8.2.3 Test Compatibility with Boundary Scan Standard**

In *normal test mode*, a P-TPG acts as a normal boundary scan register compatible with the 1149.1 standard and all related functionality can be activated.

### **8.2.4 Interconnect BIST Overhead Analysis**

Typical percentage of boundary nets among total number of nets is 5-10% [84]. Few gates will be required to implement cascading MUX, P-TPG and associated controller. Experiments were performed using Cadence CAD tools and proprietary gate libraries. Functional verification of a P-TPG component was performed using Leapfrog VHDL simulator. Two types of analysis is presented below. First, we assume the intermixed register design, where the XOR gate is already implemented as a part of a register cell. In this case, the area overhead consists of 8 AND gates for non-linear feedback and 7 AND gates required to implement a weighting circuit. In addition, a cascading 2 to 1 multiplexer to select input from previous P-TPG component or input switching generator (used when input switching probability required is greater than 0.5) can be implemented using 4 gates. The input switching generator can be implemented using just an XOR gate with most uncorelated bits of a P-TPG feedback register as its inputs. The polynomial selector is a simple inverter gate which activates non-primitive polynomial when input bit  $i = 0$ . Thus, total area overhead computed for 8 bit P-TPG with intermixed design is approximately 22 gates. If intermixed design is not used, then 7 XOR gates required to implement feedback

shift register part of an 8 bit P-TPG. This increase the overhead to 29 gates. The BIST controller has three test modes: Normal, Catastrophic Test, and Performance Test. A 2-bit BIST controller state machine has can be implemented with additional 30 gates. Thus, total area overhead in the worst case is approximately 59 gates.

### **8.2.5 Complexity Analysis of MatchProfile Algorithm**

MatchProfile algorithm complexity is determined by the parameters of genetic algorithm. The preprocessor component is important as it can reduce the run time of a genetic engine considerably by reducing the search space. This is because the preprocessor analysis helps reduce the string length of the chromosome undergoing mutation and crossover.

## **CHAPTER IX**

# **CONCLUSIONS AND FUTURE WORK**

This chapter summarizes the key contributions of this research and outlines some future directions to extend this work.

In this research, the optimization algorithms and the BIST techniques for functional and performance testing of MCM interconnections before and after mounting of the ICs are investigated. This comprehensive framework combines single test probe traversal optimization with a distributed BIST technique, enables low-cost functional testing of MCM substrates as well as performance testing of assembled MCM interchip interconnections.

### **9.1 Single Probe Traversal Optimization**

#### **9.1.1 Summary Of Contributions**

Efficient single test probe traversal algorithms are given so that total time to test the MCM substrate interconnections is reduced. The functional test cost of MCM

interconnections is therefore reduced. Single probe testing has clear advantages over two probe testing. In this research, tight bound for test times of MCM substrates with single test probe has been derived. The single probe traversal problem has been shown to be NP-hard and heuristics for computing near-optimal probe routes have been developed. The results show that the proposed heuristic algorithm is an effective tool for solving the single probe traversal problem. It gives about 40% improvement in test time over arbitrary insertion for the netlist with 800 nets. It is also 2.5 times faster than a double probe test.

### **9.1.2 Future Research**

There is a potential for developing more sophisticated heuristic algorithms using latest developments in the algorithms for solving traveling salesman problem. The application of theoretical bounds on a probe traversal cost for an MCM substrate presented in section 3.6 can be extended to real designs. Actual test time computations are needed for different kind of MCMs such as an MCM for automotive controller and an MCM for mainframe CPU unit.

## **9.2 Precharacterized TPG Design**

### **9.2.1 Summary Of Contributions**

A comprehensive test strategy for generating real switching activity profiles on the interconnections is invented to aid performance testing of MCM interconnections. A BIST scheme for interconnect performance testing based on cascaded structure of precharacter-

ized test pattern generators has been developed. It is theoretically proven and experimentally validated on practical interconnects. Analytical design technique saves a considerable amount of overhead for circuit simulation. The P-TPG design methodology and synthesis of BIST hardware can be easily automated. This BIST scheme has applications in performance testing of MCM interchip interconnects as well as inter-core interconnects in core-based systems.

### **9.2.2 Future Research**

Testing of embedded-core based system chips is posing a new challenge to the test community. This problem can be solved by using P-TPG design techniques for embedded cores and effectively test inter-core interconnections for performance faults. It can also be used for high density printed circuit board interconnect testing. For MCMs, further research is needed to extend this scheme to test random logic between the ICs at the speed of operation.

## **9.3 MISR Reconfiguration**

### **9.3.1 Summary Of Contributions**

A distributed diagnosis strategy to diagnose performance related faults such as cross-talk and ground bounce is developed. A diagnostic resolution upto a level of a single interconnect or a set of interconnects can be achieved. A method for optimal reconfiguration of a MISR has been proposed and validated theoretically. This reconfiguration scheme reduces the aliasing probability as well as the test time for interconnect diagnosis when



intermediate signatures are scanned out for comparison. A key concept of design trade-off between algorithmic complexity of a Physical MISR and wiring complexity of a physical MISR is presented.

### **9.3.2 Future Research**

The MISR reconfiguration scheme needs to be implemented on real designs and verified under the control of a real BIST controller to estimate its effectiveness. The effect of choosing physical versus logical MISR reconfiguration needs to be analyzed systematically in terms of the reconfiguration cost.

## Bibliography

- [1] R. Tummala, "Multichip Packaging-A Tutorial," Proceedings of the IEEE, Vol. 80, No. 12, pp. 1924-1941, December 1992.
- [2] Y. Zorian, Personal Communication, 1998.
- [3] D. C. Keezer, "Electronic Test Methods for MCMs," Electronic Materials and Processing Congress, pp. 131-137, Sep. 1993.
- [4] A. Chatterjee, R. Pendurkar and K. Sasidhar, "Module Level Test of MCMs," MCM Test V, IMAPS Advanced Technology Workshop, September 1998.
- [5] J. Demmin, "MCM test Becomes A Practical Reality," Electronic Packaging and Production, pp. 64-66, February 1995.
- [6] M. Lubaszewski, M. Marzouki and M. Touati, "A Pragmatic Test and Diagnosis Methodology for Partially Testable MCMs," IEEE Multi-Chip Module Conference, pp. 108-113, 1994.
- [7] M. Abadir, Parikh A., Sandborn P., Drake K. and Bal L., "Analyzing Multichip Module Strategies," IEEE Design & Test of Computers, pp. 40-52, 1994.
- [8] Sinnadurai N., "Testability, A Vital Ingredient For MCM Technology," Proc. of ISHM, pp. 378-383, 1995.
- [9] Parag K. Lala, *Digital Circuit Testing and Testability*, Academic Press, 1997.

- [10] Abramovici M., Breuer M. and Friedman A, *Digital Systems Testing and Testable Design*, IEEE Press, 1990.
- [11] H. B. Bakoglu, *Circuits, Interconnections and Packaging for VLSI*, Addison-Wesley Publishing Company, 1990.
- [12] M. Sriram and S.M. Kang, *Physical Design for Multichip Modules*, Kluwer Academic Publishers, 1994.
- [13] N. Sherwani, Y. Qiong and S. Badida, *Introduction to MultiChip Modules*, John Wiley & Sons, 1995.
- [14] Rao R. Tummala and Eugene J. Rymaszewski, *Microelectronics Packaging Handbook*, Van Nostrand Reinhold, NewYork, 1989.
- [15] W. K. Kautz, "Testing of Faults in Wiring Interconnects," IEEE Transactions on Computers, Vol. C-23, No. 4, pp. 358-363, April 1974.
- [16] P.Goel and M. T. McMahon, "Electronic Chip in Place Test" Proc. of the International Test Conference, pp. 83-90, 1982.
- [17] P.T. Wagner, "Interconnect Testing with Boundary Scan," Proc. of the International Test Conference, pp. 52-57, 1987.
- [18] A. Hassan, J. Rajski and V. Agarwal, "Testing and Diagnosis of Interconnects Using Boundary Scan Architecture," Proc. of the International Test Conference, pp. 126-137, 1988.
- [19] N. Jarwala and C. Yau, "A New Framework for Analyzing Test Generation and Diagnosis Algorithms for Wiring Interconnects," Proc. of ITC, pp. 63-70, 1989.

- [20] C. Yau and N. Jarwala, "A Unified Theory for Designing Optimal Test Generation and Diagnosis Algorithm for New Framework for Board Interconnects," Proc. of ITC, pp. 71-77, 1989.
- [21] B. C. Kim, A. Chatterjee, M. Swaminathan and D. Schimmel, "A Novel Low-Cost Approach To MCM Interconnect Test," International Test Conference, pp. 803-812, 1995.
- [22] Hilbert C. and Rathmell C. "Design and Testing of High Density Interconnection Substrates," Proc. of the NEPCON West, Vol. 2, pp 1391-1403, 1990.
- [23] J. Marshall, F.C. Chong, D. Modin and S. Westbrook, "CAD-Based Net Capacitance Testing of Unpopulated MCM Substrates," IEEE Transactions on Components, Packaging and Manufacturing Technology-Part B: Advanced Packaging, Vol. 17, No. 1, pp. 50-55, Feb. 1994.
- [24] D. Golladay, N.A. Wagner, J. R. Rudert and R. N. Schmidt, "Electron-beam technology for open/short testing of multi-chip substrates," IBM J. Res. Develop. Vol. 34. No. 2/3, pp. 250-259, Mar/May 1990.
- [25] M. Swaminathan, B. Kim and A. Chatterjee, "A Survey of Test Techniques for MCM Substrates", Journal of Electronic Testing: Theory and Applications Vol. 10, No. 1&2, pp. 27-38, 1997.
- [26] M. Beiley, et.al., "Array probe card," IEEE Multi-Chip Module Conference, pp. 28-31, 1992.
- [27] S. Hilla, "Boundary Scan Testing For Multichip Module," International Test Conference, pp. 224-231, 1992.

- [28] D. C. Keezer, K. Newman and J. S. Davis, "Improved Sensitivity for Parallel Test of Substrate Interconnections," Proc. of the International Test Conference, pp. 228-233, 1998.
- [29] W. Maly and A. Gattikar, "Feasibility Study Of Smart Substrate Multichip", Modules," Proc. of International Test Conference, pp. 41-49, 1994.
- [30] Agarwal V. D., Kime C.R., and Saluja K.K., "A tutorial on Built-In Self-Test, Part 1: Principles," IEEE Design and Test of Computers, Vol. 10, pp. 73-82, March 1993.
- [31] Agarwal V. D., Kime C.R., and Saluja K.K., "A tutorial on Built-In Self-Test, Part 2: Applications," IEEE Design and Test of Computers, pp. 69-77, June 1993.
- [32] T. W. Williams and K. P. Parker, "Design for Testability-A Survey," The Proceedings of the IEEE, Vol. 71, No. 1, pp. 98-112, January 1983.
- [33] R. Pendurkar, "Hierarchical Built-In Self Test Strategies for Functional and Interconnect Test of MCMs," Technical Report, School of Electrical & Computer Engineering, Georgia Institute of Technology, 1997.
- [34] Electronic Engineering Times, CMP Publications, Issue 994, February 23, 1998.
- [35] The National Technology Roadmap for Semiconductors, Semiconductor Industry Association, 1998.
- [36] S.Z. Yao, N. C. Chou, C. K. Cheng and T. C. Hu, "A Multi-Probe Approach for MCM Substrate Testing," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, pp. 110-121, June 1994.

- [37] N. C. Chou and C. K. Cheng, "Optimal Test Size and Efficient Probe Routing for Substrate Verification Using Two-Probe Testers," International Symposium on Hybrid Microelectronics, pp. 276-281, 1993.
- [38] J. C. Crowell, R. J. Keogh and J. A. Conti, "Moving probe bare board tester offers unlimited testing flexibility," Industrial Electronics Equipment Design New York:McGraw-Hill, Sept. 1984.
- [39] E. L. Lawler, J. K. Lenstra, A.H.G. RinnooyKan and D. B. Shmoys, "*The Traveling Salesman Problem, A Guided tour of Combinatorial Optimization*," John Wiley & Sons, 1987.
- [40] W. L. Pearn, "Augment Insert Algorithm for the Capacitated Arc Routing Problem," Computers Ops.Res. Vol. 18 No. 2, pp. 189-198, 1991.
- [41] S. E. Butt and T. M. Cavalier, "A Heuristic For The Multiple Tour Maximum Collection Problem," Computers Ops.Res. Vol. 21 No. 1, pp. 101-111, 1994.
- [42] R. Pendurkar, C. Tovey and A. Chatterjee, "Single-Probe Traversal Optimization for Testing of MCM Substrate Interconnections," IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Vol. 18 No. 8, pp. 1178-1191, August 1999.
- [43] R. Pendurkar, A. Chatterjee and C. Tovey, "New Heuristic Approach for Efficient Single Probe Routing in Testing MCM Substrate," MCM Test III, Advanced Technology Workshop, September 1996.

- [44] R. Pendurkar, A. Chatterjee and C. Tovey, "A Novel Single Probe Scheduling Algorithm for High Throughput MCM Substrate Interconnect Test," poster presentation at International Symposium On Microelectronics, October 1997.
- [45] R. Pendurkar, A. Chatterjee and C. Tovey, "Optimal Single Probe Traversal Algorithm for Testing of MCM Substrates," International Conference on Computer Design, pp. 396-401, October 1996.
- [46] Standard Test Access Port and Boundary Scan Architecture, IEEE Std. 1149.1, 1990.
- [47] T. Storey, "A Test Methodology for VLSI Chips on Silicon," Proc. of ITC, pp. 359-368, 1993.
- [48] A. Frisch, "Use of Embedded At Speed Test for KGD," Tektronix EAST Technology Internal Report, 1996.
- [49] A. Flint, "Multichip Module Self-Test Provides Means to test at Speed," EE-Evaluation Engineering, pp. 46-55, September 1995.
- [50] J. Hagge and R. Wagner, "High Yield Assembly of Multichip Modules Through Known-Good ICs and Effective Test Strategies," Proc. of the IEEE, pp. 1965-1994, 1992.
- [51] N. Jarwala, "Designing "Dual Personality" IEEE 1149.1 Compliant Multi-Chip Modules," Proc. of ITC, pp. 446-455, 1994.
- [52] N. Jarwala and C. Yau, "Achieving Board-Level BIST using the Boundary-Scan Master," Proc. of ITC, pp. 649-658, 1991.

- [53] Y. Zorian, "A Universal Testability Strategy for Multi-Chip Modules Based on BIST and Boundary Scan," Proc. Int'l Conf. on Computer Design, pp. 59-66, 1992.
- [54] Y. Zorian and H. Bederr, "Designing Self Testable Multi-Chip Modules," Proc. of ETC, pp. 181-185, 1996.
- [55] J. Koeter and S. Sparks, "Interconnect Testing Using BIST Embedded in IEEE 1149.1 Designs," Proc. of International ASIC Conference & Exhibition, pp. P11-2.1-P11-2.4, 1991.
- [56] R. Pendurkar, A. Chatterjee and Y. Zorian, "A Novel DFT Scheme for Performance Testing of MCM Interconnects," presented at Fifth International Test Synthesis Workshop, 1998.
- [57] R. Pendurkar, A. Chatterjee and Y. Zorian, "Synthesis of BIST Hardware for Performance Testing of MCM Interconnects," Proc. of International Conference on Computer-Aided Design, pp. 69-73, November 1998.
- [58] R. Pendurkar, A. Chatterjee and Y. Zorian, "A Reconfiguration Technique for Distributed BIST of MCM Interconnections," MCM Test V, IMAPS Advanced Technology Workshop, September 1998.
- [59] R. Pendurkar, A. Chatterjee and Y. Zorian, "A Distributed BIST Technique for Diagnosis of MCM Interconnections," Proc. of International Test Conference, pp. 214-221, October 1998.
- [60] A. Hassan, V. Agarwal, B. Nadeau-Dostie and J. Rajske, "BIST of PCB Interconnects Using Boundary-Scan Architecture," IEEE Trans. on Computer Aided Design, Vol. 11, No. 10, pp. 1278-1287, 1992.



- [61] C. Su, "Random Testing of Interconnects in a Boundary Scan Environment," Proc. of International Test Conference, pp. 372-381, 1992.
- [62] C. Chang and C. Su, "An Universal BIST Methodology for Interconnects," Proc. of International Test Conference, pp. 1615-1618, 1993.
- [63] Chih-Ang Chen and S. K. Gupta, "BIST/DFT for Performance Testing of Bare Dies and MCMs," Proc. of Electro '94, pp. 803-812, 1994.
- [64] O. Torreiter, U. Goecke and K. Melocco, "Testing the Enterprise IBM S-390<sup>TM</sup> Multi Processor," Proc. of International Test Conference, pp. 115-123, 1997.
- [65] S. Pilarski and A. Pierzynska, "BIST and Delay Fault Detection," Proc. of Custom Integrated Circuits Conference, pp. 13.2.1-13.2.4, 1992.
- [66] H. Chang and J. Abraham, "Delay Test Techniques for Boundary Scan Based Architectures," Proc. of International Test Conference, pp. 263-273, 1986.
- [67] L.T. Wang and E. J. McCluskey, "Hybrid Designs Generating Maximum-Length Sequences," IEEE Trans. on Computer Aided Design, Vol. 7, No.1, pp. 91-99, 1988.
- [68] K. Furuya and E. J. McCluskey, "Two Pattern Test Capabilities of Autonomous TPG Circuits," Proc. of International Test Conference, pp. 704-711, 1991.
- [69] Chih-Ang Chen and S.K. Gupta, "Design of Efficient BIST Test Pattern Generators for Delay Testing," Proc. of IEEE Transactions on CAD of Integrated Circuits and Systems, Vol 15, No. 12, pp. 1568-75, 1996.
- [70] S.K. Gupta and Chih-Ang Chen, "BIST TPGs for Faults in Board Level Interconnect via Boundary Scan," Proc. of VLSI Test Symposium, pp. 376-382, 1997.

- [71] T. W. Williams, W. Daehn, M. Gruetzner and C. W. Stark, "Bounds and Analysis of Aliasing Errors in Linear Feedback Shift Registers," *Proc. of IEEE Trans. Computer Aided Design*, Vol. 7, pp. 75-83, 1988.
- [72] D. Isaacson and R. Madsen, *Markov Chains Theory and Applications*, John Wiley & Sons, 1976.
- [73] J. Waicukauski, E. Lindbloom, E. Eichelberger and O. Forlenza, "A Method for Generating Weighted Random Test Patterns," *IBM Journal of Research and Development*, Vol 33, No. 2, pp. 149-160, 1989.
- [74] P. Bardell, W. Mcanney and J. Savir, *Built-In Test for VLSI: Pseudorandom Techniques*, John Wiley & Sons, 1987.
- [75] J. G. Kemeny and J. L. Snell, *Finite Markov Chains*, D. Van Nostrand Company, Inc., 1960.
- [76] D. K. Bhavsar, "Concatenable polydividers: Bit -sliced LFSR chips for board self-test," *Proc. of International Test Conference*, pp. 88-93, 1985.
- [77] E. Kontopidi and J. Muzio, "The partitioning of linear registers for testing applications," *Microelectronics Journal*, Vol. 24, pp. 533-546, 1993.
- [78] P. H. R. Scholefield, "Shift Registers generating maximum-length sequences," *Electronic Technology*, pp 389-394, October 1960.
- [79] Y. Zorian, "A Distributed BIST Control Scheme fro Complex VLSI Devices," *Proc. of IEEE VLSI Test Symposium*, pp. 4-9, 1993.
- [80] Haider N. and Kanopoulos N., "The split boundary scan register technique for testing board interconnects," *Proc. of VLSI Test Symposium*, pp. 43-48. 1992.

- [81] R. Pendurkar, "CA-BIST For Testing Instruction Length Decoder Chip," Technical Report, Strategic CAD Lab, Intel Corporation, 1997.
- [82] Chan J. C., Boundary Walking Test: An Accelerated Scan Method for Greater System Reliability," IEEE Transactions on Reliability, Vol. 41, No. 4, pp. 496-503, 1992.
- [83] W. Her, L. Jin and Y. El-Ziq, "An ATPG Driver Selection Algorithm for Interconnect Test with Boundary Scan," Proc. of the International Test Conference, pp. 382-388, 1992.
- [84] P. Gillis and F. Woytowich, "Delay Test of Chip I/Os Using LSSD Boundary Scan," Proc. of the International Test Conference, pp. 83-90, 1998.

## VITA

Mr. Rajesh Yashavant Pendurkar was born on December 21st, 1964 in Nipani, India. He received Bachelors degree in Electronics and Telecommunications Engineering from Government College of Engineering, University of Pune, India in 1986. He was a recipient of national merit scholarship and government open merit scholarship. From July 1986 to December 1988, he was employed with Applied Electronics Laboratories, India as a production/test engineer in power electronics division. He earned his Master of Science degree in Computer Systems Engineering from Northeastern University, Boston, Massachusetts in 1993. From March 1993 to February 1994, he worked as a software engineer at Parametric Technology Inc., Waltham, Massachusetts. He was a member of technical staff at GTE Laboratories, Waltham and Strategic CAD Laboratory, Intel Corporation, Hillsboro, Oregon for the summer of 1994 and 1997 respectively. He earned his M.S.E.E. degree from Georgia Institute of Technology in 1998. His research interests include design for testability techniques with emphasis on built-in self-test applications, CAD software design, optimization algorithms and MCM testing. He is a member of IEEE and IMAPS. Currently, he is a member of technical staff in microelectronics division at Sun Microsystems, Inc., Sunnyvale, CA. His other interests include music, athletics, mountaineering, travel and cooking.