University of Iowa
Iowa Research Online

Theses and Dissertations

2010

On low power and circuit parameter independent tests, and a new method of test response compaction

Joseph Michael Howard
University of Iowa

Copyright 2010 Joseph Michael Howard

This dissertation is available at Iowa Research Online: http://ir.uiowa.edu/etd/815

Recommended Citation
Howard, Joseph Michael. "On low power and circuit parameter independent tests, and a new method of test response compaction." dissertation, University of Iowa, 2010.
http://ir.uiowa.edu/etd/815.

Follow this and additional works at: http://ir.uiowa.edu/etd

Part of the Electrical and Computer Engineering Commons

ON LOW POWER AND CIRCUIT PARAMETER INDEPENDENT TESTS, AND A NEW METHOD OF TEST RESPONSE COMPACTION

by

Joseph Michael Howard

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisor: Professor Sudhakar M. Reddy

ABSTRACT

Testing an integrated circuit once it has been manufactured is required in order to identify faulty and fault-free circuits. As the complexity of integrated circuits increases so does the difficulty of creating efficient and high quality tests. Three issues facing manufacturing test are the power consumed during testing, process variation, and test data volume.

In regards to the power consumed during testing, abnormal switching activity, far above that seen by functional operation, may occur due to the testing technique of scan insertion. While scan insertion greatly simplifies test generation for sequential circuits, it may lead to excessive switching activity due to the loading and unloading of scan data and when the scan cells are updated using functional clocks. This can potentially damage the circuit due to excessive heat or inadvertently fail a good circuit due to current supply demands beyond design specifications.

Stuck-at tests detect when lines are shorted to either the power supply or ground. Open faults are broken connections within the circuit. Some open faults may not be detected by tests generated for stuck-at faults. Therefore tests may need to be generated in order to detect these open faults. The voltage on the open node is determined by circuit parameters. Due to the feature size of the circuit it may not be possible to determine these circuit parameters, making it very difficult or impossible to generate tests for open faults.

Automated test equipment is used to apply test stimuli and observing the output response. The output response is compared to the known fault-free response in order to determine if it is faulty or fault-free. Thus, automated test equipment must store the test stimuli and the fault-free responses in memory. With increased integrated circuit complexity, the number of inputs, outputs, and faults increase, increasing the overall data required for testing. Automated test equipment is very expensive, proportional to the

memory required to store the test stimuli and fault-free output response. Simply replacing automated test equipment is not cost effective.

These issues in the manufacturing test of integrated circuits are addressed in this dissertation. First, a method to reduce power consumption in circuits which incorporate data volume reduction techniques is proposed. Second, a test generation technique for open faults which does not require knowledge of circuit parameters is proposed. Third, a technique to further reduce output data volume in circuits which currently incorporate output response compaction techniques is proposed. Experimental results for the three techniques show their effectiveness.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

ON LOW POWER AND CIRCUIT PARAMETER INDEPENDENT TESTS, AND A NEW METHOD OF TEST RESPONSE COMPACTION

by

Joseph Michael Howard

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

December 2010

Thesis Supervisor: Professor Sudhakar M. Reddy

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

PH.D. THESIS

This is to certify that the Ph.D. thesis of

Joseph Michael Howard

has been approved by the Examining Committee
for the thesis requirement for the Doctor of Philosophy
degree in Electrical and Computer Engineering at the December 2010
graduation.

Thesis Committee: _____
Sudhakar M. Reddy, Thesis Supervisor

_____
Jon G. Kuhl

_____
Gary E. Christensen

_____
Er-Wei Bai

_____
Palle E. Jorgensen

To My Parents

We'll ride the spiral to the end and may just go where no one's been. Spiral out. Keep going …

Maynard James Keenan
Lateralus

ACKNOWLEDGMENTS

I would like to thank my advisor, Professor Sudhakar M. Reddy, for all his support, guidance and especially his patience throughout the years of my graduate work.

I would like acknowledge the Semiconductor Research Corporation (SRC) for their financial support (2004-TJ1243, 2007-TJ1642).

I would like to express my appreciation to my committee members for their time and effort.

To fellow graduate students Kent Hutchinson, Manish Vemulapauli, Santiago Remersaro, and Sharada Jha, thank you for your friendship through my years as a graduate student.

To my best friend R. Bruce Kinner, thank you for the years of chess and conversation. They kept me sane.

Finally, I would like to thank my parents for their love and support through all the years. Thank you for being with me for all the good times, but especially for the difficult times.

ABSTRACT

Testing an integrated circuit once it has been manufactured is required in order to identify faulty and fault-free circuits. As the complexity of integrated circuits increases so does the difficulty of creating efficient and high quality tests that can detect a variety of defect types that can occur throughout the manufacturing process. Three issues facing manufacturing test are the power consumed during testing, addressing different types of fault, and test data volume.

In regards to the power consumed during testing, abnormal switching activity, far above that seen by functional operation, may occur due to the testing technique of scan insertion. While scan insertion greatly simplifies test generation for sequential circuits, it may lead to excessive switching activity due to the loading and unloading of scan data and when the scan cells are updated using functional clocks. This can potentially damage the circuit due to excessive heat or inadvertently fail a good circuit due to current supply demands beyond design specifications.

Stuck-at tests detect when lines are shorted to either the power supply or ground. Open faults are broken connections within the circuit. Some open faults may not be detected by tests generated for stuck-at faults. Therefore tests may need to be generated in order to detect these open faults. The voltage on the open node is determined by circuit parameters. Due to the feature size of the circuit it may not be possible to determine these circuit parameters, making it very difficult or impossible to generate tests for open faults.

Automated test equipment is used to apply test stimuli and observing the output response. The output response is compared to the known fault-free response in order to determine if it is faulty or fault-free. Thus, automated test equipment must store the test stimuli and the fault-free responses in memory. With increased integrated circuit complexity, the number of inputs, outputs, and faults increase, increasing the overall data

required for testing. Automated test equipment is very expensive, proportional to the memory required to store the test stimuli and fault-free output response. Simply replacing automated test equipment is not cost effective.

These issues in the manufacturing test of integrated circuits are addressed in this dissertation. First, a method to reduce power consumption in circuits which incorporate data volume reduction techniques is proposed. Second, a test generation technique for open faults which does not require knowledge of circuit parameters is proposed. Third, a technique to further reduce output data volume in circuits which currently incorporate output response compaction techniques is proposed. Experimental results for the three techniques show their effectiveness.

TABLE OF CONTENTS

LIST OF TABLES ix

LIST OF FIGURES xi

CHAPTER

I. INTRODUCTION 1

1.1 Motivation 1
1.2 Testing Overview 3
1.2.1 Defects, Fault Modeling and Test Generation 3
1.2.2 Scan Design 5
1.2.3 Test Data Volume 6
1.2.3.1 Test Compression 7
1.2.3.2 Response Compaction 9
1.2.4 Fault Diagnosis 16
1.3 Organization of the Thesis 17

II. REDUCED SWITCHING ACTIVITY TESTS FOR BROADCAST SCAN BASED DESIGNS 18

2.1 Introduction 18
2.2 Review of Prior Work 19
2.2.1 Preferred Fill 20
2.2.2 Repeat Fill 20
2.2.3 Reconfigurable Switch Based Broadcast Scan 21
2.3 Proposed Method 22
2.3.1 Relaxing and Filling 23
2.3.2 Purpose of Mux 26
2.4 Experimental Results 27
2.5 Conclusions 30

III. IMPROVED DEFECT DETECTION IN THE PRESENSE OF PROCESS VARIATION 33

3.1 Introduction 33
3.2 Review of Related Work 34
3.3 Proposed Method 36
3.4 Experimental Results 45
3.5 Conclusions 46

IV. ATE ASSISTED TEST RESPONSE COMPACTION 47

4.1 Introduction 47
4.2 Preliminaries 50
4.3 ATE Assisted Test Response Compaction 50
4.3.1 ATE Assisted Compaction Procedure 52
4.3.2 Experimental Results 53
4.4 Fault Diagnosis Aware ATE Test Response Compaction 56
4.4.1 Fault Diagnosis 56
4.4.2 Procedure for Fault Aware ATE Assisted Compaction 60

4.4.3 Fault Detection and Diagnosis.....63
4.4.4 Experimental Results.....64
4.4.5 Cost of Fault Aware ATE Assisted Compaction.....70
4.5 Processing CUT Responses on ATE 77
4.6 Conclustions 79

V. CONCLUSIONS 81

5.1 Completed Research 81
5.2 Future Research 83
5.2.1 Enhancing Multi-Site Testing for Aging ATEs.....83
5.2.2 Increased ATE Assisted Compaction.....84

REFERENCES 86

LIST OF TABLES

Table

1. Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s13207.....28

2. Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s15850.....28

3. Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s35932.....28

4. Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s38417.....29

5. Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s38584.....29

6. Necessary Neighbor Assignments for Fault Activation 43

7. Necessary Neighbor State Assignments Under Condition 1 44

8. Necessary Neighbor State Assignments for Unspecified Values 44

9. Possible Neighbor State Assignments for Unspecified Neighbors N_5 and N_6 45

10. CPI-R Experimental Results.....46

11. ATE Assisted Compaction (CR = 10) 55

12. ATE Assisted Compaction (CR = 20) 55

13. ATE Assisted Compaction (CR = 30) 55

14. Fault Equivalence Class Example.....58

15. Fault Aware ATE Assisted Compaction (CR = 10) 65

16. Fault Aware ATE Assisted Compaction (CR = 20) 65

17. Fault Aware ATE Assisted Compaction (CR = 30) 66

18. Single Stuck-at Equivalence Class (CR = 10) 67

19. Single Stuck-at Equivalence Class (CR = 20) 67

20. Single Stuck-at Equivalence Class (CR = 30) 68

21. Double Stuck-at Fault Diagnosis (CR = 10).....69

22. Double Stuck-at Fault Diagnosis (CR = 20).....69

23. Double Stuck-at Fault Diagnosis (CR = 30).....70

24. Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 10) 71

25. Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 20) 71

26. Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 30) 72

27. Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 10)....73

28. Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 20)....73

29. Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 30)....74

30. Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 10) 74

31. Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 20) 75

32. Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 30) 75

33. Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 10)....76

34. Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 20)....76

35. Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 30)....77

LIST OF FIGURES

Figure

1. Conventional External Testing with On-Chip Decompressor and Compactor.....8

2. Illustration of (a) Fault Aliasing and (b) Fault Masking.....11

3. Time Compactor 13

4. Space Compactor 14

5. A Circuit with 4 Internal Scan Chains 21

6. Two Possible Configurations.....22

7. Scan Input Data.....23

8. MUX Implementation.....26

9. Test Data Volume Reduction Factor 31

10. Percent Switching Activity Reduction.....32

11. A Parity-Tree 51

12. On-Chip Test Response Compaction.....51

13. Illustration of Diagnosis Parameters.....60

14. ATE Processing Steps (a) First response from the CUT corresponding to a pair R2, (b) First response of the pair stored in the ATE, (b) Second response of the pair from the CUT, (c) The linear sum of the first and the second response is compared with the expected response stored in the ATE 79

CHAPTER I: INTRODUCTION

The semiconductor industry has grown tremendously over the past few decades due in part to very large scale integration (VLSI) techniques. The complexity of digital integrated circuits (ICs) has reached densities of over one billion transistors on a single chip. Increased IC complexity has resulted in longer test times and increased data volume in order to guarantee proper operation and reliability. Improved testing techniques, that are both efficient and high quality, are required to address the issues of longer test times and increased data volume.

1.1 Motivation

With the tremendous growth of the semiconductor industry, a naive view of testing integrated circuits cannot be taken. Increasing transistor counts and shrinking transistor feature sizes introduce new issues pertinent to high quality manufacturing of integrated circuits. In this work, techniques to address three such issues will be addressed.

As the transistor density increases, the power consumed during testing becomes of concern. The design-for-test (DFT) technique of scan insertion greatly simplifies test generation for sequential circuits. However, this simplification comes at the cost of increased switching activity during testing. Loading tests, capturing responses, and unloading of responses all contribute to the switching activity of the circuit during scan. Abnormal switching activity can occur when the circuit under test (CUT) is forced to operate outside of the normal functional operations during scan testing. Two issues related to the abnormal switching activity seen during scan testing are excessive supply current demands and power dissipation. Excessive supply current demands can cause voltage supply droops leading to larger gate delays, causing good circuits to fail tests. Excessive power dissipation may cause hot spots that could damage the CUT. For this

issue, a technique is proposed to reduce switching activity for broadcast scan based designs.

The second issue addresses the problem of detecting open faults in the presence of process variation. Open faults are disconnections of lines within the circuit. Open defects occur in contacts and vias. The disconnected node is said to be floating. The voltage on the floating node is determined by circuit parameters such as coupling capacitances to neighboring signal nodes, the coupling capacitances to power and ground lines and to the substrate, initial trapped charge, and the internal capacitances of the gates driven by the floating node. Unfortunately precise knowledge of these parameters may be difficult or impossible to determine as feature size decreases. Thus, tests must be generated to detect open faults without requiring knowledge these circuit parameters. To address this issue, a technique is proposed to detect open faults utilizing only the signal values on neighboring lines within the circuit.

The third issue addressed is the growth in test data volume. A major contributor to test data volume is the amount of data generated by the output response. For each test applied, the resulting output response must be compared to a known fault-free output response. The known fault-free responses must be stored in automated test equipment (ATE). ATEs are very expensive and their cost is proportional to the memory required to store the fault-free responses. Simply replacing them is not cost effective and so they must be kept in operation as long as possible. In order to reduce the amount of output response data required to be stored on the ATE, output response compaction techniques have been introduced. While output response compaction techniques do reduce the amount of output response data there is typically a discrepancy between the desired and actual reduction in output response data volume. This is due to requiring additional tests to maintain appropriate fault coverage. The difference in actual and desired output response data makes it difficult to match the memory required to store the fault-free responses and the memory available in the ATE. To address this issue, a technique is

proposed to help bridge the gap between the data volume required to be stored on the ATE and the memory available on the ATE.

1.2 Testing Overview

The general purpose of testing any system is to determine if it behaves correctly. To determine if the system behaves correctly, stimuli must be applied to the system and the corresponding response must be analyzed. The stimuli applied must target potential faults which can occur within the system. Targeting potential faults during the generation of stimuli requires that potential faults must be modeled. If the response to applied stimuli is determined to be incorrect then diagnosing the root cause of the incorrect behavior can help isolate problems within the manufacturing of integrated circuits.

The remaining of this section is as follows. Defects, fault modeling and test generation are introduced in 1.2.1, the design-for-test concept of scan is introduced in section 1.2.2, test data volume is discussed in section 1.2.3 and fault diagnosis will be discussed in section 1.2.4.

1.2.1 Defects, Fault Modeling and Test Generation

In order to detect physical faults within a circuit, logical fault models have been developed. The simplest and most common is the *single line stuck-at* fault model. Given a line i in the circuit, if a short occurs between line i and the power supply or ground, the voltage level at line i will remain fixed. The line stuck-at fault model represents the effect of this physical fault as the line being fixed, or stuck, at the logical value 1 or 0, denoted as s-a-v, where $v \in \{0,1\}$. Detection of a fault within the circuit occurs when a test is applied and the observed output response is different from the expected output response. For a multiple input and multiple output combinational circuit, let the logic function of the circuit be represented by $Z(x)$, where x represents an arbitrary input vector. Given a specific input vector t, the expected output response is $\underline{Z}(\underline{t})$, which is also

a vector. Due to the presence of a fault f in the circuit, the logic function of the circuit is now $Z_f(x)$. Applying a test t detects the fault f if $Z(t) \neq Z_f(t)$. It is possible to simulate a circuit with test t, both with and without the fault f. Results that are different for the two cases are represented by v/v_f where v represents the correct value and v_f represents the faulty value. Hence, if $v/v_f \in \{1/0, 0/1\}$ is observed at one or more outputs of the circuit, the fault f is detected. Other well established fault models such as open, bridging and delay exist. These fault models have been developed to target specific physical faults. The open fault model targets broken connection, the bridging fault model targets shorts between internal lines and the delay fault model targets timing related faults. Although these fault models target different physical faults within a circuit, simplifying assumptions are made that relate each of them to the stuck-at fault model and fault detection occurs when v/v_f is observed at an output.

Generating tests for combinational circuits involves two fundamental steps; *fault activation* and *fault propagation*. Input values must be set to activate the fault effect and propagate the fault effect to an output. Generation of tests for combinational circuits is much easier than for sequential circuits. The outputs of combinational circuits depend only on its current input values. In contrast, the outputs of sequential circuits depend on their current and past input values. Thus, sequential circuits contain memory elements. In order to generate a test for a specific fault, these memory elements need to be initialized prior to fault activation and propagation, making sequential test generation difficult.

Tests are generated using *automatic test pattern generators* (ATPG) that target potential fault locations; this is known as *deterministic test generation*. When a test is generated using an ATPG, only a subset of inputs have specified values in order to detect the target fault. The specified inputs are referred to as care bits, unspecified inputs are referred to as *don't-care* bits. Don't-care bits can be arbitrarily set to either 0 or 1 without affecting the detection of the target fault. An ATPG may have the option of

filling the don't-care bits randomly, generating a fully specified test vector or test pattern. If the ATPG does not specify the don't-care bits, this is known as a *test cube*. The percentage of care bits in the test cube is known as the *fill rate*.

1.2.2 Scan Design

In order to alleviate the difficulty of sequential test generation, a *design-for-testability* (DFT) technique called *scan* has been introduced. Scan designs introduce registers, or flip-flops, to increase the ability to observe and control internal nodes of the circuit. These flip-flops are connected together which generate a *scan chain*. Data can be shifted into the scan chain to specify individual flip-flop values, increasing the ability to control internal node values. Alternatively, data can be shifted out of the scan chains to observe individual flip-flop values, increasing the ability to observe internal node values. This ability to shift data into and out of the scan chain changes the test generation of sequential circuits to that of a combinational circuit, simplifying the test generation procedure.

Since the values of the flip-flops can be controlled and observed in scan designs just as if they were inputs and outputs of the circuit, the terms inputs and outputs becomes slightly ambiguous. Therefore to distinguish between an input to the circuit and an input due to the introduction of a flip-flop, the term *primary input* (PI) is used to describe the inputs of the circuit and *pseudo-primary inputs* (PPI) is used to describe an input of the circuit due to introducing a flip-flop to the circuit. In a similar manor, outputs of the circuit are referred to as *primary outputs* (PO) and outputs of internal nodes captured by flip-flops are referred to as *pseudo-primary outputs* (PPO).

The use of scan in a circuit design increases the ability to control the values of internal nodes. However, there are situations where not all line values are known within the circuit. The origins of these unknown values stem from bus contentions, uninitialized floating busses, uninitialized storage elements, and inaccurate simulation models. In

order for simulation tools to handle these unknown values a separate logic value, denoted as $X \in \{0,1\}$, is used to indicate an unknown value.

All the scan elements of the circuit are connected together in one long *single scan chain* design, requiring only one additional input pin to shift in the input data to the PPI and one additional output pin to shift out the values in the PPOs. Alternatively, instead of connecting all of the flip-flops together to generate one long scan chain, a *multiple scan chain* design may be used. In a multiple scan chain design it is typical to balance the scan chains, having near equal number of flip-flops in each scan chain. Multiple scan chain designs require additional input and output pins corresponding to the number of scan chains.

1.2.3 Test Data Volume

Testing an integrated circuit involves applying test stimuli on the inputs of the *circuit under test* (CUT) and capturing the response on the outputs. The captured response is then compared against the known correct output response to determine if the response to the test produced a *faulty* or *fault-free* response. *Automated test equipment* (ATE) supplies the test stimuli to the CUT, observes the output response, and compares the output response to the known correct response via an output response analyzer (ORA). Thus, both the test vector and known fault-free output response must be stored on the ATE. This is referred to as *conventional external testing*. In general, as the complexity of the IC increases, the number of inputs, outputs, and faults increase, this in turn increases the size of the test vector, output response and the number of test vectors, respectfully. These factors all contribute to the increased data volume required to be stored on the ATE. This increased data volume has become a problem area in IC manufacture testing due to the limited memory of the ATE.

Data volume reduction can come from reducing the number of test vectors, the amount of data required to apply test stimuli or the amount of output response data

observed. Techniques are typically implemented to minimize the number of test vectors necessary to achieve high fault coverage. In order to maintain high fault coverage, the number of test vectors cannot be reduced without sacrificing fault coverage. This limits techniques of data volume reduction in external testing to reducing the data required to apply test stimuli or the amount of output response data observed. Techniques to reduce the data required to apply test stimuli is known as *test data compression*, or *test compression* for short. Techniques to reduce the amount of output response data observed are known as *output response compaction*, or *response compaction* for short.

1.2.3.1 Test Compression

A deterministic test generated by an ATPG which produces a test cube contains care bits and don't-care bits. Analysis of ATPG algorithms has shown that fill rates are very low for industrial designs. Test data compression techniques take advantage of the low percentage of care bits and the ability to arbitrarily specify don't-care bit values to reduce the data volume of test stimuli stored on the ATE. Test data compression requires additional on-chip logic that decompresses test stimuli received from the ATE. This on-chip logic, known as a *decompressor*, is placed in front of the scan chains and is shown in Figure 1. Since deterministic tests are generated by an ATPG, compression of test stimuli must be lossless. After the compressed test stimuli from the ATE is expanded through the decompressor, all of the care bits in the test cube must be reproduced.

Test data compression techniques can be generally placed within three categories; code based, linear decompression based, and broadcast scan based schemes [1]. Code based schemes use the encoding/decoding to compress/decompress the test data. Common code based schemes are run-length, dictionary, and statistical based codes. Linear decompressors are either combinational or sequential and are composed of wires, flip-flops and exclusive-or (XOR) gates. Combinational linear decompressors use a network of fanout and XOR gates to decompress the data. Sequential linear

decompressors are based on linear finite state machines such as linear feedback shift registers (LFSR), cellular automata (CA), and ring generators. In broadcast scan based schemes, a single tester input is broadcast to multiple scan chains. In order to reduce dependencies, broadcast scan based schemes may be designed to be reconfigurable.

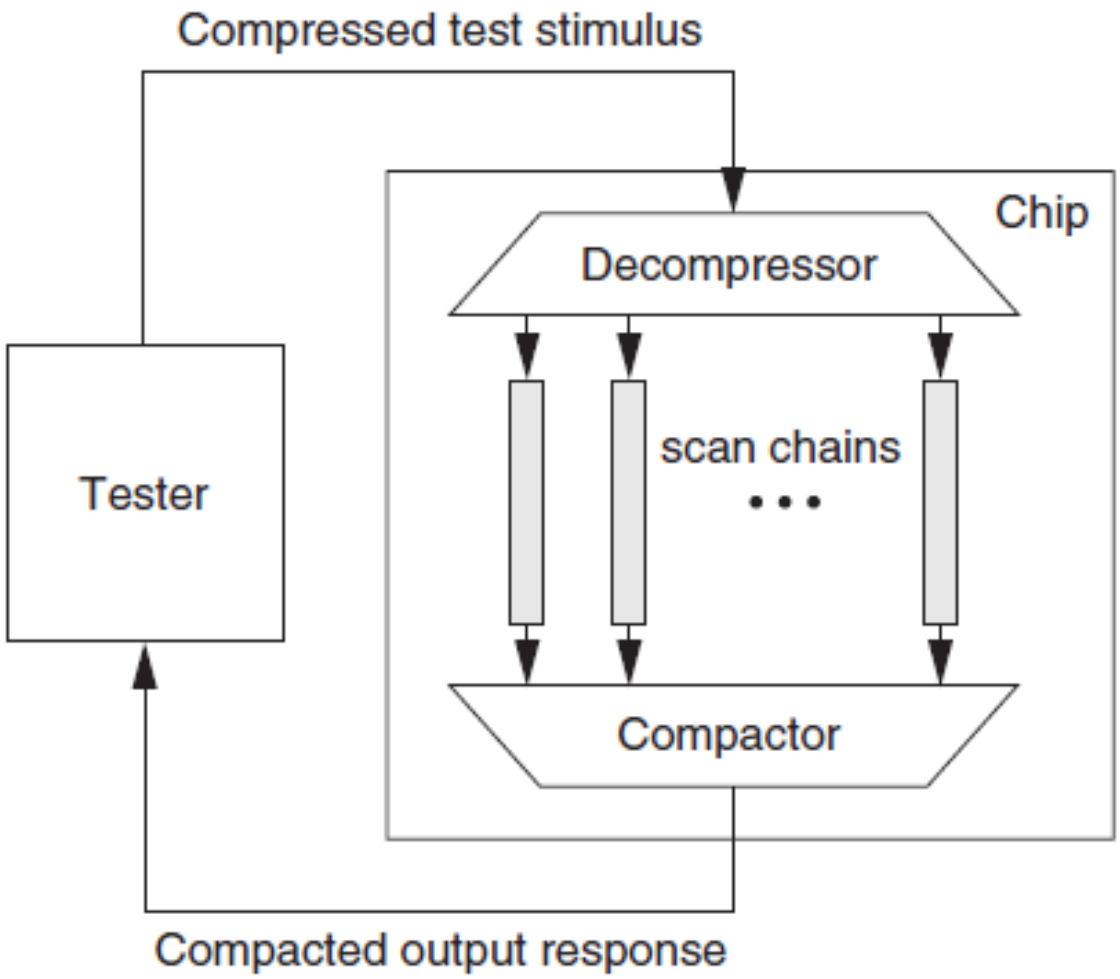

Figure 1: Conventional External Testing with On-Chip Decompressor and Compactor

1.2.3.2 Response Compaction

The other major contributor to data volume is the amount of data generated by the output response. For each test applied, the resulting output response must be compared to a known fault-free output response. In order to reduce the amount of output response data, response data compaction techniques have been introduced. While test compression techniques are required to be lossless in order to maintain the care bits produced by an ATPG, response compaction techniques are not required to be lossless. While compaction techniques reduce the amount of output response data, this reduction of output response data comes at a cost. The cost comes in the form of reducing the ability of tests to determine if the circuit being tested is good or bad, requiring additional on-chip logic and reducing diagnosis capabilities. Therefore the following five merits are used to evaluate the quality of a given compaction technique.

Compaction Ratio: The reduction of output response data is given in terms of a ratio of response data without compaction techniques to response data with compaction techniques. It is desired to achieve the highest compaction ratio as possible.

Fault Coverage: The fault coverage can be measured in two ways; actual loss of fault coverage due to fault simulation or estimated by the probability of aliasing.

Area Overhead: The amount of additional on-chip logic required to implement the compaction technique defines the amount of area overhead.

Design Complexity: Attributes such as CPU run-time for simulation, test generation, amount and routing of additional logic, and required control logic cause an increase in design complexity.

Diagnosis Resolution: The ability to quickly and uniquely identify the location of the fault is defined as the diagnosis resolution.

The ideal compactor will maximize compaction ratio and diagnosis resolution while minimizing the loss of fault coverage and area overhead. Of course, an ideal compactor is difficult, if not impossible, to obtain in most cases. In actual industrial designs, a compaction technique is employed in order to reduce the data volume to a desired amount, or in other words to achieve a desired compaction ratio.

Conventional external testing takes the approach of a bit-by-bit comparison of the output response with the known good output response in order to determine that the CUT is fault-free for all tests applied. Due to the size and complexity of current industrial ICs, the amount of memory required to store the data of the entire known fault-free output response for each test is quite large. In order to reduce the amount of data required to be stored, the output response of the CUT is compressed into a *signature*. To determine if the CUT is fault free, the observed signature for each test is compared to the known fault-free signature.

Response compaction requires additional on-chip logic, or area overhead, that compacts the output response before it is observed by the ATE. This on-chip logic, known as a *compactor*, is placed after the scan chains, shown in Figure 1. If the complexity of the compactor is high, the area overhead required may be large and routing the additional logic may be cumbersome. In addition, increased compactor complexity will result in increased CPU run-time for simulations and test generation.

When data is compacted there is an opportunity for loss of information. This loss of information can result in reduced fault coverage and fault diagnosis resolution. Two cases where reduced fault coverage occurs are *aliasing* and *fault masking*. Aliasing occurs when an output response prior to the compactor is faulty but is mapped to a fault free response when it is compacted. For example, if both inputs to the XOR gate shown in Figure 2 have faulty values, the resulting faulty response is the same as the fault free response. Fault masking occurs when an unknown (*X*) value appears in the output response prior to the compactor and prevents the error from propagating to the output of the compactor. Take for example the two input XOR gate shown in Figure 2. It has a faulty value at one input and an X on the other input. The X on the input can be either 0 or 1, therefore the output cannot be uniquely determined and therefore masks the fault effect.

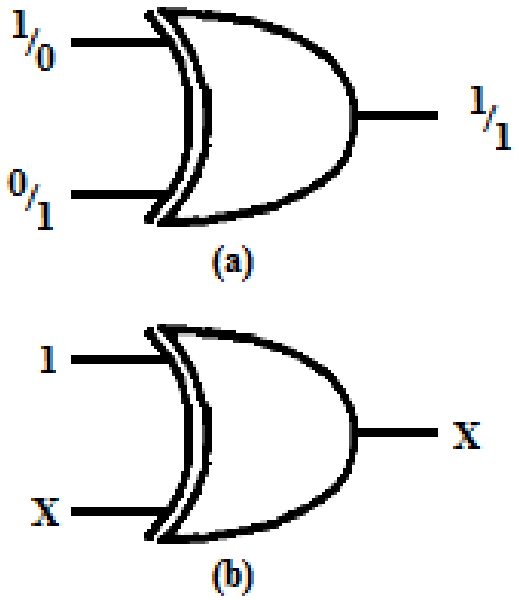

Figure 2: Illustration of (a) Fault Aliasing and (b) Fault Masking

In fault diagnosis, it is desired to easily and uniquely determine the location of a detected error within the circuit. *Diagnosis resolution* indicates the ability to easily and uniquely determine an error location within a circuit. If a particular circuit has high diagnosis resolution, it is relatively easy to uniquely determine the error location within the circuit. Due to compaction, or more specifically the loss of information, the ability to uniquely determine the location of a detected error diminishes.

Output response compaction techniques can be generally placed within three categories; time, space and finite memory. Time compactors reduce the overall length of the output response. Space compactors reduce the number of outputs which need to be observed. Finite memory compactors incorporate elements of both time and space compaction.

Initial response compaction techniques made the underlying assumption of the absence of unknown (*X*) values. This assumption was considered valid for the compaction techniques used in the early stages of integrated circuits. However, with the increase in integrated circuit complexity, the introduction of unknown values (*X*) has invalidated this assumption. Thus new compaction techniques must account for this behavior by either developing compaction techniques that are tolerant of unknown (*X*) values or introduce additional logic to block unknown (*X*) values from entering the compactor. In general it is impractical to eliminate all X sources due to timing constraints, overhead, simulation inefficiencies and inaccuracies in modeling behaviors of certain memory, custom logic, and analog circuit blocks. In addition, X values are discovered very late in the design stage or during chip debug after manufacturing has begun. In this case it is difficult or impossible to insert additional DFT logic. The presence of X values creates the challenge of designing compactor that can tolerate X values while maintaining high fault coverage [23].

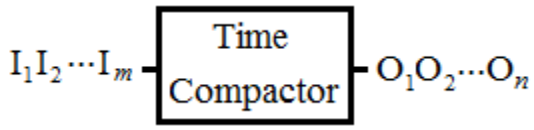

Figure 3: Time Compactor

Time compactors use sequential logic to reduce the length of the output response. Typically the output response which is an *m*-bit data stream is reduced to an *n*-bit data stream where ($n < m$). In Figure 3 a general time compaction scheme is shown. Most time compactors generate a single signature for the entire set of tests. Techniques such as parity checking [3], ones counting [5], transition counting [9] and signature analysis [11,12], also known as a multiple input signature register (MISR), are some of the most popular time compaction techniques in the absence of X values. Parity checking uses the parity of the output sequence as the signature. Ones counting uses the total number of ones in the output sequence as the signature. Transition counting creates a signature based on counting the number of transitions ($0 \rightarrow 1$, $1 \rightarrow 0$) occurring in the output sequence. Signature analysis is based on the error detection technique of *cyclic redundancy checking* (CRC) using polynomial division [13]. Signature analysis is the most popular compaction technique due to low aliasing probability, high compaction ratio and its guarantee to detect all single bit errors.

Due to the sequential nature of time compactors, any introduction of X values will result in *signature corruption*, creating multiple fault-free signatures. If an X is introduced early enough into the test sequence, the entire signature may result in X values. Therefore the signature obtained from the CUT cannot be compared with one unique fault-free signature. Techniques such as those proposed in [25, 26 28, 40] use

control logic to block Xs from entering the time compactor. The technique proposed in [27] uses stochastic coding to minimize the appearance of X values entering the compactor. The technique in [29] uses modular compactors of relatively prime length which guarantees the detection of an error in the presence of a single X value entering the compactor. The technique in [30] uses a MISR and generates a signature invariant of the Xs through symbolic simulation.

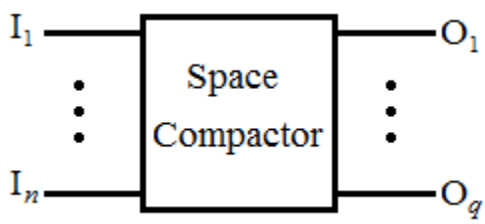

Figure 4: Space Compactor

Space compactors use combinational logic to reduce the number of outputs that need to be monitored. The n ($n > 1$) outputs of a circuit are supplied to a space compactor which produces q ($q < n$) outputs, resulting in a compaction ratio of n:q. In Figure 4 an n-to-q space compactor is shown. Typically space compactors are designed such that $q = 1$ in order to achieve the highest compaction ratio. Unlike most time compactors, space compactors generate a signature for each test pattern. The technique of parity tree space compactors [15] is the most popular space compaction technique. It consists of XOR gates, having minimal overhead and design complexity. Due to the lack of a controlling value, he XOR gate has excellent error propagation. While this technique will only detect faults appearing on an odd number of circuit outputs, it has been

experimentally shown [16] that most single line stuck-at faults are detected using a party tree compactor. Other techniques such as those proposed in [2, 17-20, 22] use combinational logic to construct a space compactor. These techniques attempt to improve error detection over the parity tree technique but are typically more complex, have higher area overhead and are test set dependent.

While the introduction of a single X value corrupts the signature of a time compactor, space compactors are more tolerant of X values. The effects of an X appearing at the output of the circuit which propagates to the output of the compactor is only present for that cycle. Therefore, fault-free signatures may contain X values in some bit positions. In this case, the bit positions of the X values are ignored when comparing the signature of the CUT to the fault-free signature. The techniques in [31-33] use control logic to mask certain scan chains to block the propagation of X values. A technique called X-compact [34] uses XOR gates to construct a compactor based on X-codes [73]. Similarly a technique called X-filter [36] is proposed where the compacted output is processed so dependences on Xs are eliminated. The approach in [35] uses control logic to modify the output response such that either a constant value or a shifted version is delivered to the compactor.

Finite memory compactors are a relatively new class of compactors, combining space and time dimensions. Finite memory compactors use combinational logic in conjunction with memory elements but do not have feedback. Without feedback, finite memory compactors have a finite impulse response and therefore can tolerate X values. Any X that enters finite memory compactor is flushed out in a relatively small number of cycles. Techniques such as convolutional compactors [37] and block compactors [38] are some of the recently proposed finite memory compactors. Convolutional compactors are based on polynomials corresponding to certain code words of k-out-of-M codes. Block compactors are based on parity check matrices of block codes.

1.2.4 Fault Diagnosis

Fault diagnosis is an important aspect of testing ICs. Fault diagnosis algorithms can be placed in two categories [77]; cause-effect and effect-cause. Cause-effect methods build a fault behavior database for modeled faults. Due to the database size, this approach is not practical for large designs. The effect-cause approach analyzes actual responses and determines the most likely fault(s) which could have caused the failure and is therefore much more practical.

Due to the loss of information, response compaction reduces the ability of fault diagnosis techniques. One alternative to improving diagnosis is to incorporates additional on-chip logic that allows observing all scan chains directly, bypassing the on-chip compactor. Doing so provides direct access to the scan cells, enabling the application of the well-established standard ATPG based fault diagnosis techniques. Unfortunately, the approach of bypassing the on-chip logic has two drawbacks. First, additional overhead is required to facilitate the bypass mode. While this overhead may be acceptable, the second drawback is that two separate test sets are required; a compressed test set for volume production testing and an uncompressed test set for fault diagnosis purposes. In addition, such an approach does not facilitate on-line diagnosis.

Another approach (called *indirect diagnosis* in [76]) performs diagnosis for compactor-based designs through two phases. Phase 1 consists of identifying scan cells which should observe failures. In phase 2, the identified scan cells are used with ATPG based diagnostic algorithms designed for circuits without compactors. This approach assumes only a single error exists at any scan cycle and multiple errors cannot be uniquely identified. In reality, a single defect may produce multiple errors, therefore fault diagnosis is limited with this approach.

An alternative, called *compactor independent direct diagnosis* proposed in [76], provides a generalized effect-cause approach to fault diagnosis for designs with

compactors. The technique introduces a *transformation function* that describes the compaction function. The function, Φ_i, is given by the expression $P_i = \Phi_i (C_i)$, where P_i is the *i*'th compactor output and C_i is the set of values prior to compaction. Each scan cell corresponds to a fan-in logic cone within the circuit. For a faulty signature, suspect faults within the logic cones are simulated, including the compactor logic, and are compared to the faulty signature. If the suspect fault does not match the faulty signature the fault is removed from the suspect list. Suspect faults are then ranked based on criteria that are used in the existing diagnosis algorithm. The advantages of this approach are that it can be generally applied to a variety of compactors, existing diagnosis algorithms can be used, it is an on-line methodology and it does not make the single error assumption.

1.3 Organization of the Thesis

The thesis is organized as follows. Chapter II presents the proposed technique for generating low power tests for broadcast scan based designs. Chapter III presents the proposed technique for generating tests to detect open faults. Chapter IV presents the proposed technique for reducing output response data. Chapter V reviews and concludes this thesis.

CHAPTER II: REDUCED SWITCHING ACTIVITY TESTS FOR BROADCAST SCAN BASED DESIGNS

Switching activity of a circuit may be large during scan testing, resulting in abnormal power dissipation. We investigate the use of a recently proposed procedure called preferred fill to reduce switching activity in the presence of on-chip test data reduction circuits. Specifically we consider broadcast scan based designs. Experimental results presented for the larger ISCAS-89 benchmark circuits show that a high test data compression ratio and reduction in test power can be simultaneously obtained.

2.1 Introduction

The current standard for manufacture test is the scan based approach. Two issues in manufacturing test of VLSI circuits are power consumption and the reduction of test data volume and test application time. Recently, methods have been proposed to address each of these issues independently.

Regarding the issue of power consumption, it has been observed that during scan testing switching activity may far exceed the activity during normal operation of the circuit [41-43]. Excessive switching activity is caused by scan tests requiring the circuit under test (CUT) to operate outside of the normal functional operation. Excessive switching activity during the application of scan tests are caused both during scan chain shifts to load tests and unload test responses as well as when the scan cell contents are updated using functional clocks in what are referred to as capture cycles. Abnormal switching activity causes abnormal peak as well as average power dissipation and supply currents. Excessive power dissipation may cause hot spots that could damage the CUT.

In [44] a method referred to as Preferred Fill was used to reduce switching activity during capture cycles. Unspecified entries in a test cube are filled according to a simplified signal probability calculation. The Preferred Fill method is applied to generate

launch off capture tests for transition delay faults. However, this method can also be applied to generate tests for other fault models such as stuck-at and path delay faults. In [45] a method referred to as Repeat Fill (also called Adjacent Fill) was used to reduce switching activity during shift cycles. Unspecified entries in a test cube are filled with the previously specified bit.

Methods to reduce test data volume use a large number of internal chains which are driven by a small number of external scan inputs through an encoder [31,46-50]. These methods are based on either broadcast scan [46] or linear feedback shift register encoding [47]. In this work we consider broadcast scan based methods and specifically use the design proposed in [50]. This method uses multiple internal scan chains in conjunction with a reconfigurable switch to distribute test data from a limited number of external inputs.

2.2 Review of Prior Work

In this section we first review Preferred Fill for reduction of capture cycle switching activity for stuck-at faults in standard scan designs. Next we review Repeat Fill for reduction of shift cycle switching activity for stuck-at faults in standard scan designs. Last we briefly discuss the reconfigurable switch and its implementation with multiple scan chain designs. Since the focus of this paper is stuck-at faults (SAFs), all discussions assume SAFs.

In this work we use Weighted Switching Activity (WSA) in order to estimate the power consumed during testing. The WSA of a node is the number of state changes at the node multiplied by (1 + node fan-out). The WSA of the entire circuit is obtained by summing the WSA of all nodes in the circuit.

2.2.1 Preferred Fill

Consider the test vector <v> and test response <f(v)> for SAF testing. The test vector can be written as v = (PI, PPI), where PI and PPI correspond to the primary inputs and pseudo-primary inputs in the test vector and f(v) = (PO, PPO), where PO and PPO correspond to the primary outputs and pseudo-primary outputs of the test response. The goal of Preferred Fill in SAF testing is to minimize the Hamming distance between the applied test vector v and the test vector response f(v). More specifically, to minimize the Hamming distance between <v: PPI> and <f(v): PPO>. When the Hamming distance is minimized, the number of changes to the input of the circuit during the capture cycle is reduced. Thus, the switching activity during the capture cycle is reduced.

Minimizing the Hamming distance between <v: PPI> and <f(v): PPO> is done as follows. Preferred values for each scan cell or equivalently each PPI_j, $1 \leq j \leq n$ are determined in a preprocessing step. The preferred value for a PPI_j is 0 (1) if the probability that the corresponding scan cell holds its value at 0 (1) when the circuit is clocked is higher than 0.5. Thus, one can determine the preferred values by computing the signal probabilities of PPOs [44].

2.2.2 Repeat Fill

As mentioned in [45], the logic value of the k-th PPI does not change if the (k+1)th PPI is specified to the same logic value during the shift cycle. Therefore, filling unspecified bits with the previous PPI logic value will not increase switching activity during shift cycles.

For example, consider the following case where we have 0X0. Filling the unspecified value to 1 will produce 010, causing two transitions. However, filling the unspecified value with 0 produces 000, causing no transitions. Filling unspecified values to the same value as the previously filled value is called repeat fill.

2.2.3 Reconfigurable Switch Based Broadcast Scan

In the broadcast scan method proposed in [50] a reconfigurable switch is used to distribute test data to internal scan chains from a limited number of external scan inputs. A chip with four internal scan chains and two external scan inputs is shown in Figure 5. A two-bit vector is supplied to the two external scan inputs and the reconfigurable switch produces the appropriate four-bit vector to be scanned into the four internal scan chains. An example of two different configurations is shown in Figure 6. In the first configuration, shown in Figure 6a, the external scan input 1 is connected to internal scan chains 1 and 2 while external scan input 2 is connected to internal scan chains 3 and 4. In the second configuration, shown in Figure 6b the external scan input 1 is connected to internal scan chains 1 and 3 while external scan input 2 is connected to internal scan chains 2 and 4. Tests to detect some faults may exist in one configuration and not the other. In general, a switch with several configurations may be necessary in order to generate tests for all the detectable faults. To set the switch into the desired configuration, a controller circuit with a register or a counter and an appropriate decoder is needed.

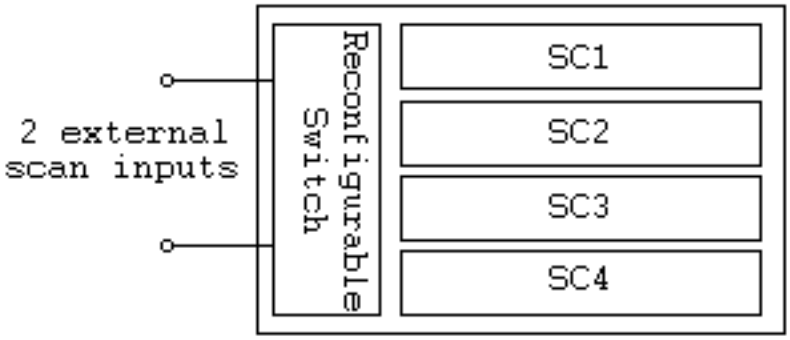

Figure 5: A Circuit with 4 Internal Scan Chains

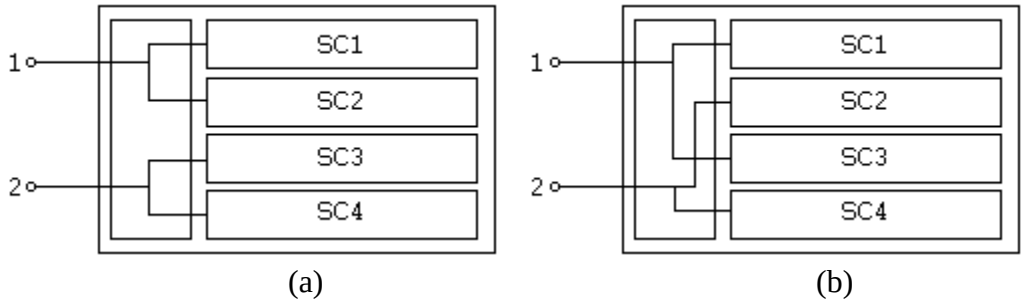

Figure 6: Two Possible Configurations

2.3 Proposed Method

In this section the method called Compact Preferred Fill (CPF) to fill unspecified values in test cubes for multiple scan chain designs is described. The design of broadcast scan shown in Figure 6 is augmented to include a MUX at the input of each internal scan chain. The test generation procedure for this design first obtains a fully specified test set from the test generator used in [50] and a test relaxation procedure similar to [51] is used to un-specify several inputs in the tests. The unspecified entries are next filled using preferred fill. The process of relaxing and filling will be described in Section 3.1 and the use of the MUXs at the inputs to the internal scan chains is described in Section 3.2.

The test set provided by the method in [50] includes configuration information for the reconfigurable switch for each test vector. This information is necessary for proper filling of unspecified values. The internal scan chains can be represented by a matrix whose rows are individual scan chains. Depending on the configuration, the columns of all scan chains driven by a particular external scan input must contain the same test data. For example, consider the design shown in Figure 7. This design has four internal scan chains and two external scan inputs. The configuration shown in Figure 7 has external input 1 connected to internal scan chains 1 and 2 while external input 2 is connected to

internal scan chains 3 and 4. When the input data is shifted into the scan chains, the logic values in the columns corresponding to internal scan chains 1 and 2 (3 and 4) are consistent.

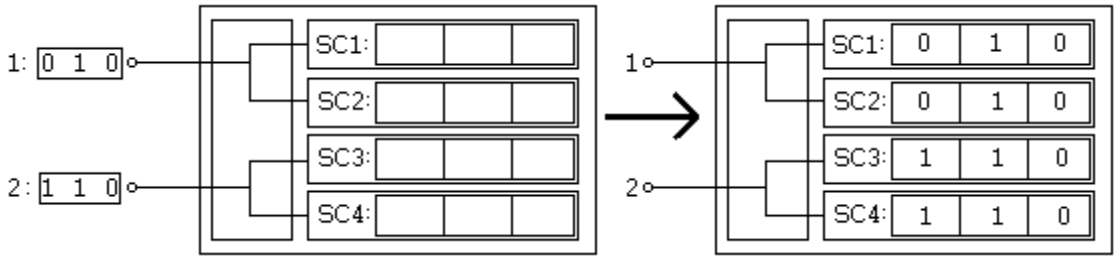

Figure 7: Scan Input Data

2.3.1 Relaxing and Filling

In this section we provide details of the proposed post-processing procedure to modify a given set of SAF tests to obtain tests with lower switching activity during both capture and shift cycles. The basic step of the modification is to relax a test by un-specifying some of the specified entries in the test and then filling the unspecified values with preferred values. We use a procedure similar to that of [51].

Fault coverage of the relaxed tests must not be lower than that of the original test set. Achieving this is done by providing a set of faults, called target faults, which must be detected by a relaxed test. In [51] the target faults for a test are determined by simulating a fully specified test when it is picked for relaxation. In our procedure, we initially determine the set of target faults for each test simulating an ordered set of tests in

reverse order. The set of target faults for a test are updated after fault simulating a newly relaxed test in which the unspecified values are filled according to the following criteria:

(1) If a row in the scan chain matrix is completely unspecified, fill it with the majority preferred fill value. Mark that row as filled.

(2) Excluding rows marked as filled, if the column for the internal scan chains corresponding to a particular external scan input are completely unspecified, fill the column with the majority preferred fill value.

(3) Excluding rows marked as filled, if the column for the internal scan chains corresponding to a particular external scan input has a specified value, fill the column with the corresponding specified value

The complete procedure is given below.

Procedure:

1. Obtain a SAF test set T. Each t in T will include configuration information.

2. For each test t in T compute the WSA of the capture cycle and order the tests in T in decreasing order of WSA. Let the ordered test set be T_0.

3. Fault simulate the tests in T_0 in reverse order using fault dropping and delete any test that does not detect any yet undetected faults. For each test t_i in T_0 let F_i be the set of faults detected by t_i.

4. Set T' = Ø.

5. For i=1 to $|T_0|$ do

6. If $F_i \neq \emptyset$ then

7. Relax t_i with target set of faults F_i.

8. If a row in the scan chain matrix is completely unspecified, fill it with the majority preferred fill value and mark the row as filled

9. Excluding rows marked as filled, if the column for the internal scan chains corresponding to a particular external scan input are completely unspecified, fill the column with the majority preferred fill value.

10. Excluding rows marked as filled, if the column for the internal scan chains corresponding to a particular external scan input has a specified value, fill the column with the corresponding specified value.

11. Let the resulting test be t_i'

12. Fault simulate t_i' and delete the detected faults from F_j, for all j>i.

13. $T' = T' \cap \{t_i'\}$

14. End If

15. Else drop t_i

16. End for

17. Return T'

End Procedure

2.3.2 Purpose of MUX

As discussed above, in [50], all internal scan chains corresponding to a particular external scan input will obtain the same test data. In our method, we wish to control each individual internal scan chain by placing a two-input MUX between the reconfigurable

switch and the internal scan chains. We show a configuration in Figure 8. The purpose of the MUX is to select between the data to be scanned in from the external scan input or from the constant value determined from the majority preferred fill value for an individual scan chain. In Figure 8 this constant for each scan chain are shown as F1, F2, F3 and F4 which correspond to SC1, SC2, SC3 and SC4, respectively. This allows control over the process of filling both the unspecified rows and columns in the scan chain matrix with the majority preferred fill value, thus reducing switching activity during capture cycles. In case the MUXs are not used one can fill columns of unspecified values in scan chains driven by the same external scan input using preferred fill. However we found that this was not effective in reducing the capture or scan shift WSA.

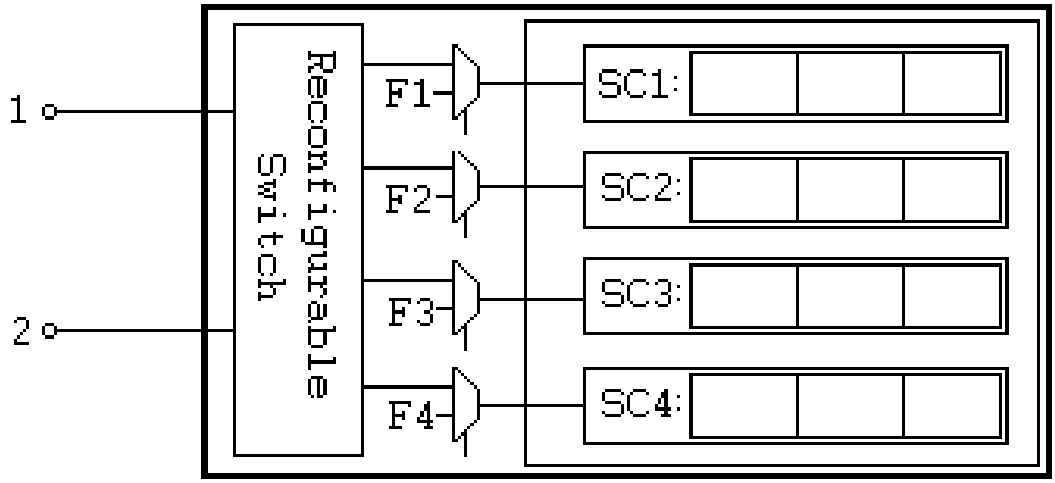

Figure 8: MUX Implementation

2.4 Experimental Results

The proposed method was implemented in C language. Experiments were conducted on the larger ISCAS-89 benchmark circuits using a Pentium 4 2.8 GHz PC with 1 GB of RAM operating under Linux. Test vectors were obtained using an academic ATPG. Results reported in this paper are for tests to detect all the detectable single line stuck-at faults. For each benchmark circuit, several different numbers of internal scan chains were used in the experiments. For each benchmark circuit, except for s15850, the number of internal scan chains used in the experiments was 16, 32 and 64. For s15850 the number of internal scan chains used in the experiments was 16 and 32. The number of external scan inputs used for all experiments was 4.

The format of Table 1–5 are the same. In the first column we list the benchmark circuit and under (#SC, #IP) we give the number of internal scan chains (#SC) and the number of external scan inputs (#IP). In the second column we give the test data volume reduction factor compared to a single scan chain architecture using random fill. The first reduction factor does not include the additional MUX control bits, while the second reduction factor includes the additional bits necessary to control the MUXs. Compact Preferred Fill with MUX we give the capture WSA (C-WSA) and shift WSA (S-WSA). The percent reduction of WSA compared to random fill is given below the data for each scan chain configuration. In the next four columns we give similar data for the case when preferred fill is used without using MUXs.

From Tables 1–5 it can be seen that the proposed method using MUXs allows simultaneous reduction of test data volume and reduced power during test for all designs. If MUXs are not used as suggested and preferred fill is used sometimes it may not reduce the switching activity caused by tests for the benchmark circuits compared to random fill.

Table 1: Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s13207

		CPF with MUX				CPF without MUX			
s13207	TDV	C-WSA		S-WSA		C-WSA		S-WSA	
(#SC,#IP)	Red	PEAK	AVE	PEAK	AVE	PEAK	AVE	PEAK	AVE
(64,4)	13.25	3,453	2,315	4,673	882	3,940	3,002	5,544	3,188
	5.40	(17.9%)	(25.5%)	(15.8%)	(73.5%)	(6.3%)	(3.3%)	(0.1%)	(4.3%)
(32,4)	6.24	3,759	2,893	4,953	1,631	4,073	2,973	4,910	3,680
	4.58	(18.0%)	(11.8%)	(4.5%)	(54.5%)	(11.1%)	(9.4%)	(5.3%)	-(2.6%)
(16,4)	3.84	4,540	2,570	4,424	1,352	4,578	2,528	4,552	3,020
	3.52	(8.6%)	(23.2%)	(13.2%)	(62.7%)	(7.8%)	(24.5%)	(10.7%)	(16.6%)

Table 2: Reduction of Capture & Shift power of ISCAS-89 benchmark circuit s15850

		CPF with MUX				CPF without MUX			
s15850	TDV	C-WSA		S-WSA		C-WSA		S-WSA	
(#SC,#IP)	Red	PEAK	AVE	PEAK	AVE	PEAK	AVE	PEAK	AVE
(32,4)	4.65	2,659	1,739	4,699	1,474	3,418	2,124	5,473	2,799
	3.35	(34.3%)	(37.5%)	(22.5%)	(60.2%)	(15.5%)	(23.7%)	(9.7%)	(24.5%)
(16,4)	3.21	3,278	1,742	4,940	1,739	3,067	1,772	5,333	2,334
	2.91	(17.4%)	(38.9%)	(15.0%)	(54.0%)	(22.7%)	(37.9%)	(8.2%)	(38.3%)

Table 3: Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s35932

		CPF with MUX				CPF without MUX			
s35932	TDV	C-WSA		S-WSA		C-WSA		S-WSA	
(#SC,#IP)	Red	PEAK	AVE	PEAK	AVE	PEAK	AVE	PEAK	AVE
(64,4)	7.87	14,610	8,140	14,100	5,356	14,856	8,882	14,100	6,461
	5.04	(18.4%)	(22.9%)	(12.9%)	(32.8%)	(17.0%)	(15.8%)	(12.9%)	(19.0%)
(32,4)	4.82	12,243	8,526	13,235	6,160	12,243	8,650	13,405	6,493
	4.22	(23.7%)	(17.4%)	(15.4%)	(21.2%)	(23.7%)	(16.2%)	(14.3%)	(16.9%)
(16,4)	2.60	11,031	8,341	13,973	5,741	11,041	8,335	13,961	5,848
	2.51	(25.5%)	(20.4%)	(12.1%)	(21.0%)	(25.4%)	(20.5%)	(12.2%)	(19.5%)

Table 4: Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s38417

		CPF with MUX				CPF without MUX			
s38417	TDV	C-WSA		S-WSA		C-WSA		S-WSA	
(#SC,#IP)	Red	PEAK	AVE	PEAK	AVE	PEAK	AVE	PEAK	AVE
(64,4)	3.82	12,514	9,591	14,502	5,551	13,480	9,567	14,773	7,745
	2.37	(7.2%)	(4.3%)	(7.3%)	(43.3%)	(0.0%)	(4.5%)	(5.6%)	(20.9%)
(32,4)	2.90	11,208	9,003	13,703	4,151	11,517	9,244	13,670	6,538
	2.52	(10.0%)	(7.7%)	(1.7%)	(54.3%)	(7.5%)	(5.2%)	(2.0%)	(28.1%)
(16,4)	2.00	11,175	9,191	13,086	4,274	11,733	9,092	12,158	5,210
	1.93	(11.1%)	(9.6%)	(7.0%)	(51.0%)	(6.7%)	(10.6%)	(13.6%)	(40.2%)

Table 5: Reduction of Capture & Shift Power of ISCAS-89 Benchmark Circuit s38584

		CPF with MUX				CPF without MUX			
s38584	TDV	C-WSA		S-WSA		C-WSA		S-WSA	
(#SC,#IP)	Red	PEAK	AVE	PEAK	AVE	PEAK	AVE	PEAK	AVE
(64,4)	10.21	9,094	4,178	10,726	3,296	9,769	5,365	12,106	5,877
	6.02	(24.2%)	(38.6%)	(28.6%)	(58.5%)	(18.6%)	(21.1%)	(19.4%)	(26.0%)
(32,4)	5.75	7,518	4,458	10,881	3,576	6,444	4,829	11,526	4,302
	4.90	(42.1%)	(37.1%)	(27.8%)	(55.8%)	(50.4%)	(31.8%)	(23.6%)	(46.9%)
(16,4)	3.31	7,178	4,458	10,726	3,685	7,487	4,720	11,092	4,267
	3.17	(39.9%)	(37.5%)	(28.1%)	(53.8%)	(37.3%)	(33.9%)	(25.7%)	(46.5%)

Figure 9 shows a comparison of the test data volume reduction factor achieved for the proposed method without the MUX and with the MUX with 4 external scan inputs and 64 and 32 internal scan chain for all circuits except for s15850 which use 32 and 16 internal scan chains. It can be seen that the test data volume reduction factor without the MUX is better than with the MUX due to the additional control data supplied to the MUX. However, it can be seen that if the proposed method is implemented without the MUX the switching activity reduction percentage for both capture and switching power is

lower. Thus a tradeoff must be made between reducing the switching activity and reducing the test data volume.

Figure 10 shows the percent switching activity reduction for the peak capture WSA (C-WSA PEAK), average capture WSA (C-WSA AVE), peak shift WSA (S-WSA PEAK) and average shift WSA (S-WSA AVE) for the proposed method with the MUX. It can be seen that the most significant reduction is in the average shift WSA reduction. This result is due to the fact that when a MUX is set to supply a constant value to the scan chain for a given test, the circuit will see minimal switching activity.

2.5 Conclusions

In this work we have investigated a new method to reduce power consumption during both capture and shift cycles in scan testing in addition to reducing test data volume and test application time for multiple scan chain designs. The proposed method has been shown to reduce switching activity in both capture and shift cycles. However, this comes at a cost of additional test data volume and test application time due to the high amount of configuration data needed to control the large number of internal scan chains. We wish to further study the aspects of the MUX configuration data in order to minimize the test data volume and test application time needed for the proposed method.

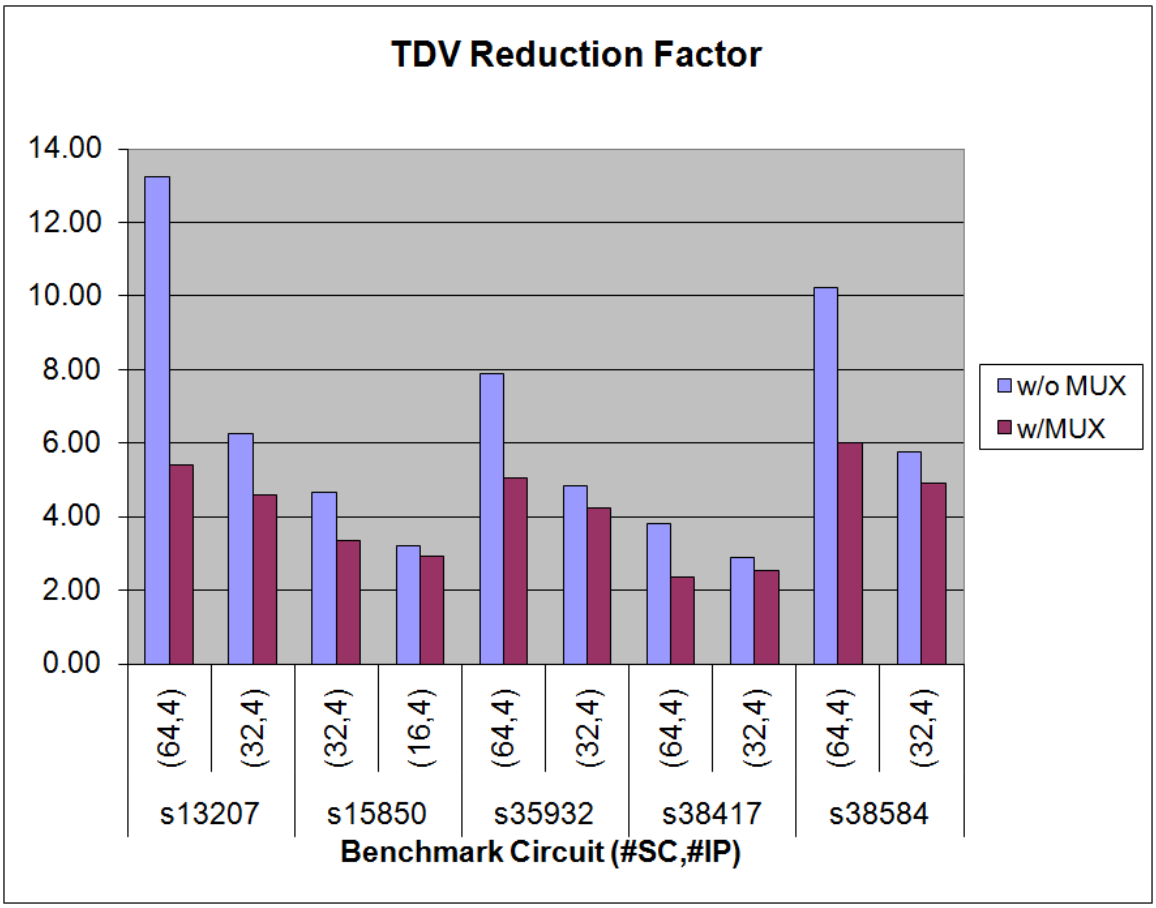

Figure 9: Test Data Volume Reduction Factor

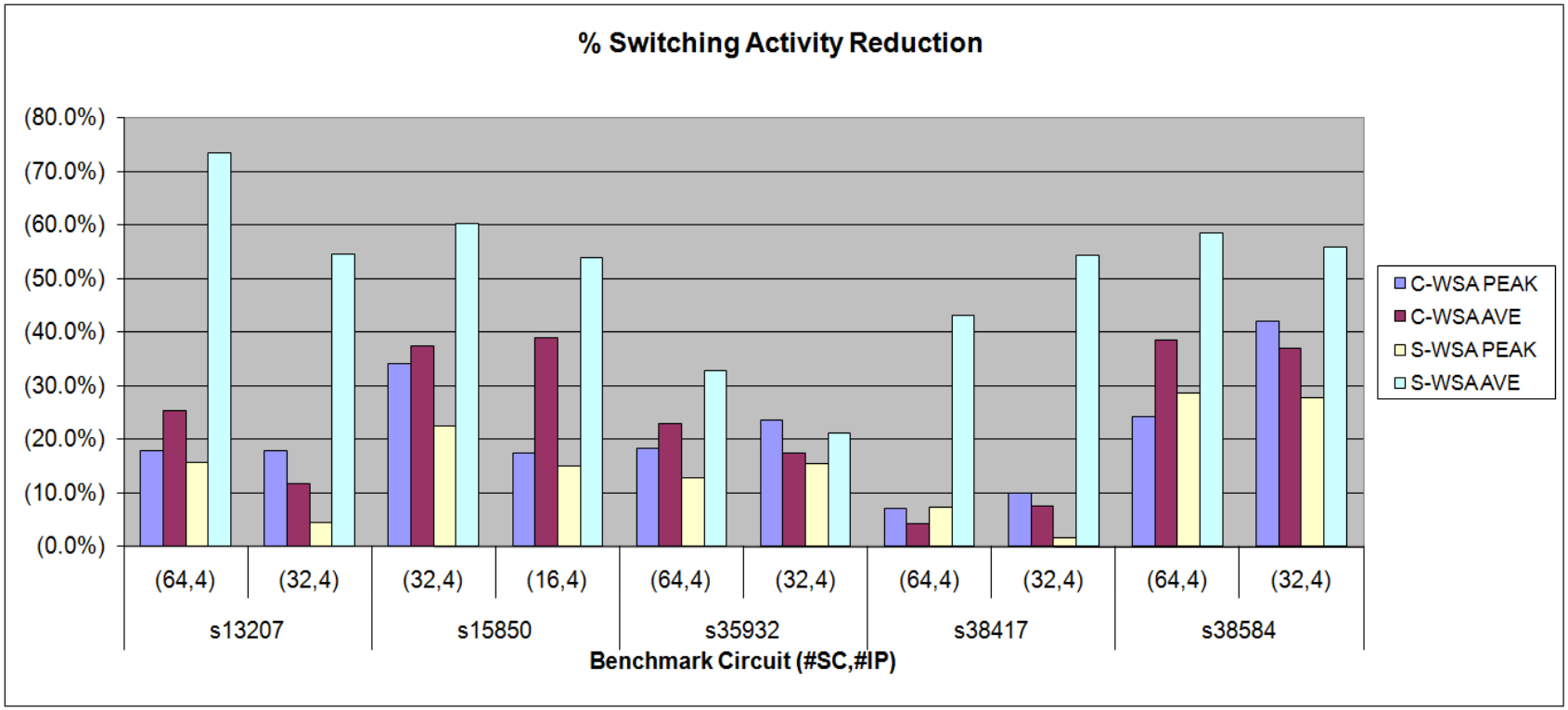

Figure 10: Percent Switching Activity Reduction

CHAPTER III: IMPROVED DEFECT DETECTION IN THE PRESENSE OF PROCESS VARIATION

VLSI designs of future CMOS technologies are expected to suffer unpredictable process variations. Manufacturing tests will need to be effective in the presence of variations. In this work we investigate a class of tests to detect interconnect opens which meet this goal. The proposed voltage based (logic) tests are constituted as a pair of constrained stuck-at fault tests for the circuit node affected by the open defect. In addition, since the test consists of a pair of stuck-at-0 and stuck-at-1 the necessary conditions for activating bridging faults between a node and its neighbors are also met, thus detecting many bridging faults.

3.1 Introduction

Open defects in digital CMOS circuits occur most frequently in contacts and vias. Opens can be classified into two categories; partially open and completely open. A partially open node has finite resistance while a completely open node has infinite resistance. In this work we will only consider completely open vias.

We consider the detection of open defects in digital CMOS circuits by way of voltage based (logic) tests. When a circuit node is disconnected from the gate driving it, the node is said to be floating. The voltage on the floating node is determined by several circuit parameters such as coupling capacitances to neighboring signal nodes, the coupling capacitances to power and ground lines and to the substrate, initial trapped charge, and the internal capacitances of the gates driven by the floating node [53-60]. In addition, the threshold voltage of the gate(s) driven by the disconnected line segment may be interpreted differently by different gates driven by the open node [55, 61]. Even though the effect of trapped charge can be expected to be negligible in technologies with high leakage currents which will drain this charge away the dependence on circuit

parameters such as coupling capacitances and gate threshold voltages can be expected to remain. As the feature size of VLSI circuits decreases, precise knowledge of the circuit parameter values may be difficult, if not impossible, to obtain. Therefore it is important to develop methods to generate tests to detect opens that do not require accurate knowledge of circuit parameters.

Methods to detect via opens in the presence of unknown circuit parameters have been previously investigated. In [62] the method of maximal favorable neighborhood states based tests is proposed and in [63] the method of circuit parameter independent (CPI) tests is proposed. These methods are reviewed in Section 2 and the need for methods that generate more flexible tests to detect via opens is discussed. We then propose a new method of detecting via opens referred to as *Relaxed Circuit Parameter (*CPI-R*) Tests*. CPI-R tests target open via defects independent of circuit parameters, thus generation of tests does not require any knowledge of the circuit parameters and trapped charge. Additionally CPI-R tests are not confined by the stringent requirements of CPI tests, therefore CPI-R tests are able to detect opens that are considered undetectable by CPI tests.

3.2 Review of Related Work

In this section we review the related works on tests to detect via opens and their limitations. In Section 3 we will prove that it is possible to relax the conditions imposed by CPI tests to detect via opens and improve the number of via opens detected. The model for interconnect opens is the same as in [63], which considers only completely open vias. In [64, 65] the voltage, V_F, on a floating node F was shown to be

(1) $V_F = \frac{C_1}{C} V_{dd} + \frac{Q_{trap}}{C}$

(2) $C = C_0 + C_1$

(3) $C_0 = C_{gnd} + C_{a0} + C_{int0}$

(4) $C_1 = C_{vdd} + C_{a1} + C_{int1}$

In the above equations Q_{trap} is the internal trapped charge on the floating node, C_{vdd} and C_{gnd} are the coupling capacitance of node F to the power supply line and the ground/substrate, respectively. C_{a0} and C_{a1} are the sums of the coupling capacitances between the floating node F and its neighbor nodes that have the values of logic 0 and logic 1, respectfully. C_{int0} and C_{int1} are the coupling capacitances internal to the driven gate whose actual values depend on voltage V_F [55].

It can be seen from the above equations that the voltage V_F depends on the voltages of the neighbors of F. The voltage of the neighbors of F depend on the input pattern applied to the circuit. The voltage V_F is interpreted as logic 0 or 1 depending on its magnitude and the threshold voltage of the gate input driven by node F. If the floating node drives more than one gate whose threshold voltages are different, then gates may interpret the voltage on node F differently [55]. In this work we assume that all the gates driven by a specific floating node have the same threshold voltages while gates driven by different open nodes may have different threshold voltages. In addition it is also possible that an oscillation and/or sequential behavior could occur due to the feedback through the coupling capacitance and Miller capacitances [57]. In this work we do not consider the case of oscillations and sequential behavior.

Now we will review the related method of maximal favorable neighborhood states based tests of [62]. A neighbor N of a floating node F is said to be at the favorable value [62] for a stuck-at-0 (1) fault on F if its state is 0 (1). The method in [62] requires generating a stuck-at-0 and a stuck-at-1 tests for each floating node with a maximal number of neighbors at favorable values. Each test is assigned a weight which is the ratio of the number of neighbors set to the favorable value and the total number of neighbors of the target node. The coverage obtained by a set of tests is the sum of the weights of

the tests detecting the target opens. This method of tests generation does not account for the fact that the values of different coupling capacitances to the neighbors of a node could vary considerable. For example, the physical layouts and extracted capacitances values for the ISCAS-85 benchmark circuits is given in [66] where it can be seen that the values of different capacitances differ most frequently by factors of two to four and can be up to an order of magnitude. In addition, it is not clear how to relate the coverage obtained by the proposed test weighting to defect coverage.

Next we review the related method of circuit parameter independent (CPI) tests of [63]. CPI tests are constituted as a pair of constrained stuck-at fault tests for the circuit node affected by the open defect. Detection of an open node, say F, for CPI tests require that the states of all neighboring signal nodes are identical for both the stuck-at 0 and stuck-at 1 tests. Due to this condition the voltage on F (V_F) remains the same for both the stuck-at 0 and stuck-at 1 tests. This voltage is interpreted as either 0 or 1 by the gates driven by F when either of the stuck-at tests are applied to the circuit. If an open exists at node F, either the stuck-at 0 or stuck-at 1 test will detect the defect.

As stated above, the limitations of the maximal favorable neighborhood states based test is that the coverage obtained is unclear and that it does not take into account the variability of coupling capacitances of neighboring nodes. CPI tests are limited in the detection of via opens due to the condition that all neighborhood states must be identical for both the stuck-at-0 and stuck-at-1 tests. If a test cannot be generated with all neighborhood states having identical values, a via open is deemed as undetectable for the CPI test method. In addition, both methods require that all neighborhood states are specified, limiting test compaction.

3.3 Proposed Method

The proposed method, Relaxed Circuit Parameter Independent (CPI-R) tests, relaxes the requirement that the states of the neighboring signal nodes are identical in the

stuck-at-0 and stuck-at-1 test pair. CPI-R tests are also constituted as a pair of constrained stuck-at fault tests, however different conditions must be observed. For a target open node, say F, let t_0 and t_1 be stuck-at-0 and stuck-at-1 tests, respectfully.

CPI-R Conditions:

(1) If a neighbor, say N, has the value 1 (0) for test t_0 (t_1), this implies that N must be 1 (0) for t_1 (t_0).

(2) When a neighbor state is set to 0 (1) for t_0 (t_1) the corresponding neighbor state for t_1 (t_0) is a do not care (X) in a CPI-R test.

Condition (1) ensures that the voltage V_F on the open node F is such that $V_F(t_1) \geq V_F(t_0)$. For t_0 (t_1), 0 (1) is the preferred, or favorable, neighbor state, in order to set the voltage on the open node to 0 (1). From Condition (2) it can be seen that if a neighbor is assigned a do not care (X) this may reduce the number of specified inputs required for the stuck-at fault test. The reduced number of specified inputs allows for increased test compaction. Even more advantageous than reducing the number of specified inputs is the fact that the relaxed conditions for CPI-R tests allow for tests to be generated to detect via opens that may not be possible to generate under the stringent requirements of CPI tests.

Lemma 1: Let t_0 and t_1 be stuck-at-0 and stuck-at-1 tests, respectfully. Let N be a neighbor and let $N(e_0, e_1)$ denote the neighbor state combination of N for tests t_0 and t_1. The only valid neighbor state combinations are N(0,0), N(0,1), N(1,1).

Proof: From Condition (2) when N has the value 0 (1) for t_0 (t_1) it is in the favorable state and N is a do not care (X) for t_1 (t_0). For these two

cases we then have the following neighbor state combinations N(0,X) and N(X,1). Expanding each of these combinations for the do not care (X) value we obtain three neighbor state combinations; N(0,0), N(0,1), and N(1,1). The only remaining possible neighbor state combination is N(1,0). However this neighbor state combination violates Condition 1. Therefore we have only three valid neighbor state combinations, specifically N(0,0), N(0,1), and N(1,1).

Q.E.D.

Theorem 1: A CPI-R test (t_0,t_1) for an open at node F detects the open defect independent of the circuit parameters when the open is modeled as in [63].

Proof: From Equations (1) – (4) it can be seen that the only input pattern dependent components that affect the voltage on F are C_{a0} and C_{a1}, which are the sums of the capacitances to F from the neighbor states at logic 0 and logic 1, respectfully. As shown in Lemma 1, there only exists three valid neighbor state combinations; N(0,0), N(0,1), and N(1,1). For the case when all the neighbor states have the combination of either N(0,0) or N(1,1), is was proven in [63] that the open defect would be detected because the voltage, V_F, would remain the same for both tests. Therefore the only remaining case is when the neighbor state combination is N(0,1). Expanding equation (1) for C_1 we obtain

(5) $V_F = \frac{C_{vdd} + C_{a1} + C_{int1}}{C} V_{dd} + \frac{Q_{trap}}{C}$. If we subtract $V_F(t_0)$ from $V_F(t_1)$ we obtain (6)

(6) $V_F(t_1) - V_F(t_0) = \frac{C_{a1}}{C} V_{dd} \geq 0$. From (6) we obtain (7)

(7) $V_F(t_1) \geq V_F(t_0)$

Q.E.D.

One special case of consideration is when all neighbor states can be set to favorable values for a stuck-at-0 or stuck-at-1 test. This is referred to as the maximum favorable neighborhood state CPI-R test, or MAX CPI-R test. Generating a MAX CPI-R test may be necessary in order to detect a via open in cases where it is not possible to generate a standard CPI-R test. For instance, if a stuck-at-0 (1) fault is redundant for node F, then in order to detect the open at node F a MAX CPI-R stuck-at-1 (0) test must be generated. Another case where it is necessary to generate a MAX CPI-R test is when generating a CPI-R test it is not possible to satisfy Condition 1. In this case it is necessary to generate a MAX CPI-R test in order to detect the open.

Detection of bridging faults is an added feature of both CPI-R and MAX CPI-R tests. Since each CPI-R test consists of both a stuck-at 0 and stuck-at 1 test, the necessary conditions for activating bridging faults between a node and its neighbors are also met by the CPI-R tests. For the case when a MAX CPI-R test is necessary, the conditions for bridging fault detection are also activated. For example if a MAX CPI-R stuck-at-0 (1) test is required, the target node is set to 1 (0) while the neighbor states are set to 0 (1). Thus many bridging faults may be detected with CPI-R and MAX CPI-R tests.

Next we outline the proposed method to generate CPI-R tests and minimize the number of tests. It was shown in [63] that a large percentage of opens could be detected by CPI tests contained in the standard 1-detection stuck-at tests. Therefore, in Phase 1, we first determine the opens detected by CPI-R tests contained in the 1-detection stuck-at

test set. Then in Phase 2 the ATPG is implemented to generate CPI-R tests for the remaining open faults. As a pre-processing step to Phase 2, the necessary assignments for fault activation are derived for each detectable stuck-at fault. Necessary assignments allow neighbor state combination conflicts, such as N(1,0), to be observed before test generation begins. When a conflict is observed this indicates that the open fault is CPI-R Redundant. Finally in Phase 3 we implement steps in order to reduce the test pattern count. The procedure is as follows:

Procedure:

Phase 1: Simulate the 1-detection stuck-at test set as CPI-R and MAX CPI-R tests, mark detected faults *CPI-R_Detected* and add tests to *test_set*

Phase 2: For all faults not detected in Phase 1

1. If a neighbor state combination conflict exists, mark fault *CPI-R_Redundant*. Else,
2. Generate a MAX CPI-R stuck-at-0 test for the current fault. If SUCCESSFUL mark fault as *CPI-R_Detected* and compact test cube into *test_set*. Else,
3. Generate a MAX CPI-R stuck-at-1 test for the current fault. If SUCCESSFUL mark fault as *CPI-R_Detected* and compact test cube into *test_se*t. Else,
4. Determine if all specified neighbors from necessary assignments for both the stuck-at fault tests that fall under Condition (1) can be satisfied. If NO mark fault as *CPI-R_Redundant*. Else,

5. For unspecified neighbor states use SCOAP-Controllability values as a metric for selecting initial neighbor values.
6. Generate a CPI-R test for the current combination of unspecified neighbor states.
   a. If SUCCESSFUL mark fault *CPI-R_Detected* and compact test cubes into *test_set*.
   b. Else if an additional combination exists, apply the new combination to the unspecified neighbor states and return to Step VI.
   c. Else if no additional combinations exist, mark the fault *CPI-R_Redundant*

Phase 3:

1. Random fill unspecified values in test cubes in *test_set* and mark all *CPI-R_Detected* faults as *CPI-R_Undetected*
2. Simulate *test_set* to detect via opens for MAX CPI-R tests. Mark used test patterns, mark detected faults as *CPI-R_Detected* and add used test patterns to *final_test_set*. Simulate used test patterns from *test_set* as CPI-R tests, mark detected faults *CPI-R_Detected*
3. Greedy Set Cover Algorithm for CPI-R pairs
   a. Create CPI-R pairs (T_{ij}) for each non-used test pattern
      ($T_{ij} = \{t_i, t_j\}$; $t_i, t_j \in$ test_set)
   b. **Obtain_Essential_CPI-R pairs (Tij)**
      – Obtain all essential CPI-R pairs
      – Mark tests t_i and t_j as essential

- Add t_i and t_j to *cpir_pair_set*
- Mark detected faults as *CPI-R_Detected*
- Drop CPI-R pairs which do not detect any new open faults

c. While CPI-R pairs remain

- Select the CPI-R pair T_{ij} that detects the largest number of remaining undetected open faults
- Mark t_i and t_j as essential
- Add t_i and t_j to *cpir_pair_set*
- Mark detected faults *CPI-R_Detected*
- Drop CPI-R pairs which do not detect any new open faults
- Implement *Obtain_Essential_CPI-R*

4. Add tests from *cpir_pair_set* to *final_test_set*. Keep essential marks, clear used marks, reverse order of tests in *final_test_set* and mark all detected faults as *CPI-R_Undetected*

5. Simulate test patterns marked essential from *final_test_set* as CPI-R tests and MAX CPI-R tests. Mark detected faults as *CPI-R_Detected* and essential test patterns as used

6. Simulate all test patterns from *final_test_set* as CPI-R tests and MAX CPI-R tests. Mark detected faults CPI-R_Detected and mark used tests

7. Drop any tests which are not marked used

End Procedure

The simulations of Phase 3 in Steps 2-4 first help identify the essential test patterns required for fault detection. The simulations of Steps 5-6 help determine which test patterns are unnecessary and can be removed from the *final_test_set*.

Next an example is provided to illustrate the procedure of the ATPG (Phase 2).

Example 1: Let F be the target node having 6 neighbors, N_1, N_2, N_3, N_4 N_5 and N_6. Assume the necessary assignments for fault activation of the stuck-at faults for node F indicate the following neighbor states in Table 6.

Table 6: Necessary Neighbor Assignments for Fault Activation

	N1	N2	N3	N4	N5	N6
SA0	0	X	1	X	X	X
SA1	X	1	X	0	X	X

Neighbors N_1, N_2, N_3, and N_4, have specified values where N_5 and N_6 remain unspecified for the CPI-R test.

Step 1: No neighbor state combination conflict exists for node F, continue to Step II.

Step 2: A MAX CPI-R stuck-at-0 test is not possible due to the necessary assignment on N_3 which must be a 1 in order to activate the target node, continue to Step III.

Step 3: A MAX CPI-R stuck-at-1 test is not possible due to the necessary assignment on N_4 which must be a 0 in order to activate the target node, continue to Step IV.

Step 4: Neighbors N_3 and N_4 receive values from the necessary assignments that fall under Condition 1. Therefore we must have $N_3(1,1)$ and $N_4(0,0)$. Assume that these neighbor states are achievable. Therefore we now have the following neighbor states as shown in Table 7.

Table 7: Necessary Neighbor State Assignments Under Condition 1

	N1	N2	N3	N4	N5	N6
SA0	0	X	1	0	X	X
SA1	X	1	1	0	X	X

Step 5: SCOAP-Controllability - $C_{N5}(4,7)$, C_{N6} (5,2). Therefore the initial values are shown in Table 8.

Table 8: Necessary Neighbor State Assignments for Unspecified Values

	N1	N2	N3	N4	N5	N6
SA0	0	X	1	0	0	X
SA1	X	1	1	0	X	1

Step 6: Using the above neighbor states test generation is done for both stuck-at tests. If test generation fails, the unspecified neighbors are then changed to a new combination, shown in Table 9, and Step VI is repeated until no further combinations exist.

Table 9: Possible Neighbor State Assignments for Unspecified Neighbors N_5 and N_6

N_5 (0,X), N_6 (X,1)	N_5 (X,1), N_6 (X,1)
N_5 (0,X), N_6 (0,X)	N_5 (X,1), N_6 (0,X)

3.4 Experimental Results

We investigated detection of open vias in several ISCAS-85 circuits and the results are given in Table 10 below. The first column indicates the circuit name and the second column represents the total number of vias in the circuit. Column 3, 4, and 5, which fall under the general column heading of *1D Test Set*, represent the number of via opens detected, percentage of via opens detected and number of test patterns, respectfully, for the 1-detection stuck-at fault test set. The test patterns used in the 1-detection stuck-at fault test sets came from an academic ATPG. Columns 6, 7, and 8, which fall under the general column heading of *Plus ATPG*, represent the total number of via opens detected, percentage of via opens detected and number of test patterns, respectfully, for the proposed method. Column 9 represents the number of opens that cannot be detected by CPI-R tests and column 10 represents the number of aborted opens.

Table 10: CPI-R Experimental Results

		1D Test Set			Plus ATPG				
ckt	#VIA	#DET	%DET	#TP	#DET	%DET	#TP	#RED	#ABT
c432	1284	1123	87.46%	60	1196	93.15%	67	88	0
c880	2211	1952	88.29%	57	2080	94.08%	73	131	0
c1355	3303	3083	93.34%	127	3178	96.22%	109	25	100
c1908	5248	4619	88.01%	154	4762	90.74%	128	483	3
c2670	8584	7017	81.75%	145	7347	85.59%	161	1157	80
c3540	10990	8954	81.47%	158	9625	87.58%	237	1351	14
c5315	16099	13745	85.38%	147	14402	89.46%	248	1595	102
c6288	16353	15123	92.48%	33	15954	97.56%	208	314	85
Average			87.27%			91.80%			

3.5 Conclusions

In this work we proposed a relaxed version of a previously proposed class of tests in order to improve the detection of opens in digital CMOS circuits. Experimental results on several ISCAS-85 benchmark circuits show that the proposed method is effective in detecting opens in the presence of unknown circuit parameters. In addition, the stuck-at fault coverage remained on average greater than 99% for the final test set. Another feature, due to the properties of the CPI-R tests, is that many bridging fault defects may also be detected.

CHAPTER IV: ATE ASSISTED TEST RESPONSE COMPACTION

A new method for achieving test response compaction is proposed. The method involves testers to achieve additional compaction, without compromising fault coverage, beyond what may be already achieved using on-chip response compactors. The method does not add additional logic or modify the circuit under test or require additional tests and thus can be used with any design including legacy designs. Simple heuristic procedures are used to achieve additional compaction. Experimental results on larger ISCAS-89 circuits show the effectiveness of the method.

4.1 Introduction

As the size and complexity of integrated circuits increases so does the cost of testing due to increased test data volume, test application time and cost of automated test equipment (ATE). In order to combat the increase in test data volume, test vector compression [1] and test response compaction methods are used [4,15,33,34,68,69]. The focus of this paper is on test response compaction.

In order to determine if a circuit is free of defects, the expected responses to test stimuli must be stored in ATE. The use of test compaction reduces the amount of data to be stored in the ATE [4,15,33,34,68]. Three types of response compactors have been proposed [68]. These are time or infinite memory compactors [4], finite memory compactors [68] and space compactors [15].

Time compactors are sequential circuits with global feedback that combine the current test response with the previous responses to generate a signature used for fault detection. One of the most popular time compactors is the MISR [4]. The compaction level achieved by MISRs is extremely high. However, if unspecified values in responses, also called Xs, exist in test responses MISRs cannot be used unless the Xs are masked by

additional DFT logic and control inputs or the designs are modified to insure suppression of Xs.

Finite memory compactors use feed forward or definite sequential circuits and can accommodate Xs in test responses [68].

Space compactors, which also accommodate Xs, are combinational circuits typically constructed as trees of exclusive OR gates, also called parity trees, to combine the outputs of the circuit under test (CUT) in order to reduce response data volume and reduce the number of pins, often to a single pin, which are monitored by the ATE. The compaction ratio CR of a space compactor or finite memory compactor is defined as N/R, where N is the number of inputs to the compactor and R is the number of compactor outputs. Actual compaction ratio (ACR) achieved is the ratio of the test response data volume without compaction and the response data volume with compaction. Often ACR achieved is less than CR of the space compactor since the number of tests to achieve the same fault coverage with the compactor may be higher than without the compactor.

The choice of compactor used is based on criteria such as, desired ACR, loss of fault coverage and/or reduction of diagnosis resolution and hardware overhead. Loss of fault coverage may be due to aliasing or fault masking. Aliasing occurs when a faulty response is mapped to a fault-free response. Fault masking occurs when an X appears in the response preventing the error from propagating to the output of the compactor. The origin of X's stem from bus contentions, uninitialized floating buses and uninitialized storage elements, inaccurate simulation models etc. [15]. Losses in fault coverage due to aliasing and masking by Xs can be compensated by increasing the number of tests.

A basic characteristic of all space and finite memory compactors is that they compact test response for each test separately and store the compacted responses in the ATE. Specifically, test responses of more than one test are not combined in the compaction process as in the time compactors. We propose that instead of storing in the ATE the compacted response for each test, store the linear sum (exclusive OR) of the

compacted responses of up to K consecutive tests. By doing this the compacted response data stored in the ATE can be reduced by a factor of up to K. We call this *ATE assisted test response compaction* or for short *ATE assisted compaction*. We implemented this proposal for the larger ISCAS-89 benchmark circuits with on-chip parity tree space compactors and storing in the ATE the linear sum of compacted test responses from the on-chip compactor for two consecutive tests. The results show that on top of the compaction obtained from the on-chip compactor the proposed method provides approximately an additional 2X compaction of test response data.

ATE assisted compaction can be used with any of the space compactors and finite memory compactors. It can also be used with MISR based on-chip compactor if the signatures are computed separately for each test to aid defect diagnosis [70].

Since ATE assisted compaction does not require on-chip logic it can be used with any design. Another important issue it might help with is the following. It has been observed that when test data compression techniques are used, often the actual level of compression achieved is quite a bit less than what is expected from the ratio of the number of internal scan chains to the number of compactor outputs [71]. This causes difficulties in matching the tester memory to the test data volume for a given design. The additional compaction due to ATE assisted compaction will help in bridging the gap between the reduction in the expected test data volume compaction and the compaction that was actually obtained. Another important area where ATE assisted compaction can be helpful is in the context of multisite test in which multiple number of chips are simultaneously tested to reduce test costs. In such applications the same test stimuli are applied to all chips on the load board of the tester but the test responses must be observed separately from each tested chip. The number of simultaneously tested chips is limited either by the tester memory or the number of test channels on the tester. In the event that the tester memory is the limiting factor, using ATE assisted compaction which increases

the response compaction level, one can simultaneously test additional chips thus reducing the test costs.

4.2 Preliminaries

In this section we review space and finite memory compactors and how test response data is stored in ATEs.

Typically a VLSI design consists of several scan chains. Let us assume that the number of scan chains is N. The response data in the N scan chains for each test is compacted by the on-chip space/finite memory compactors in to K, K<N, compacted response vectors to be stored in the ATE. For the sake of discussion and in our experiments we assumed a single output parity tree shown in Figure 11 as the on-chip space compactor. In the case when the compactor of Figure 11 is used on-chip the test response is a single vector of length L as illustrated in Figure 12, where L is the length of the longest scan chain in the design. The compacted response vector contains 0s, 1s and Xs and is stored in the ATE using two vectors of length L. One of these vectors is the data vector and the other is the mask vector. The mask vector is used to mask the response from the CUT on the tester in the position where the response vector had an X. For example if the response vector for a test obtained by simulation of the fault free circuit is 1 0 X 1 0 X then the data vector 1 0 0/1 1 0 0/1 and the mask vector 1 1 0 1 1 0 is stored. In the data vector a 0/1 indicates that one can store an arbitrary value in the corresponding position. The response of a CUT from the tester is compared to the stored response only in the positions where the mask vector bit is 1.

4.3 ATE Assisted Test Response Compaction

In this section the proposed ATE assisted response compaction technique is described. Section 3.1 outlines the procedure used to generate the responses stored in the ATE. Section 3.2 outlines how the responses stored in the ATE are used during

manufacturing test and the requirements on the ATE. In Section 3.3 we briefly discuss the effect on fault detection and fault diagnosis when ATE assisted response compaction is used.

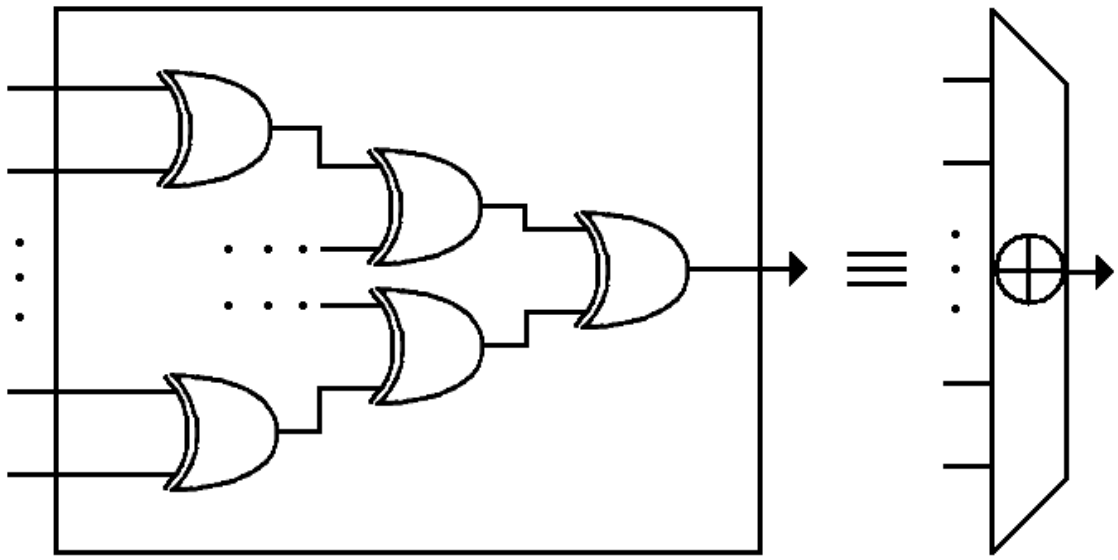

Figure 11: A Parity-Tree

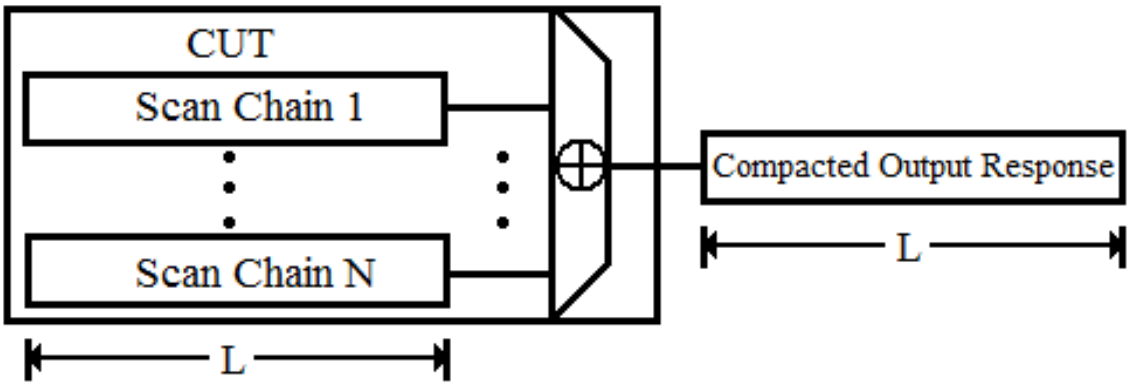

Figure 12: On-Chip Test Response Compaction

The expected responses determined by the procedure given in Section 3.1 are the linear sum (exclusive OR) of two test responses compacted by the on-chip compactor. For example if two responses from the CUT are 1 0 X 0 1 1 and 1 1 0 1 0 X the stored response is 0 1 X 1 1 X.

4.3.1 ATE Assisted Compaction Procedure

The inputs to the compaction procedure are the test set $T = \{t_1, t_2, \ldots, t_n\}$, the set of corresponding fault-free responses from the CUT $R = \{r_1, r_2, \ldots, r_n\}$ and the set of faults F detected by T. There are six steps, outlined below, in the procedure.

Step 1: Determine using fault simulation the set of faults $F_1 = \{f_{11}, f_{12}, \ldots, f_{1q}\}$ detected only once and the set of faults $F_2 = \{f_{21}, f_{22}, \ldots, f_{2r}\}$ detected only twice.

Step 2: Determine using fault simulation the pairs of responses (r_i, r_j), $r_i, r_j \in R$, that should not be merged in to a single response to be stored as an expected response in the ATE. By merging two responses we mean computing the exclusive OR of them. Let R_p be the set of such pairs of responses. We use a heuristic procedure to determine R_p. A pair of responses is included in R_p if merging the pair of responses masks the detection of a fault in F_1 or in F_2 computed in Step 1.

Step 3: In this step we find the pairs of responses that are merged. Let this set of pairs be R_m. Pairs in R_m are determined by a greedy procedure that avoids inclusion in R_m any pair in R_p found in

Step 2. It should be noted that in this step some responses are not merged with any other response to avoid including a pair in R_p in R_m.

Step 4: In this step we fault simulate all faults in F when the compacted responses that use the linear sums of pairs of responses in R_m and the responses that were not merged with any other response are used for detection. It is possible that some faults in F are now not detected due to the merging of some responses. Let the set of faults not detected be F_u.

Step 5: In this step we unmerge some of the pairs of responses in R_m such that using the unmerged responses all faults in F_u are detected. Let the set of responses that are left merged be R_2 and the set of responses that are not merged be R_1.

Step 6: We order the tests in T such that the tests t_i and t_j corresponding to the two responses of a (r_i , r_j) pair in R_2 appear such that t_j appears immediately after t_i or vice versa. The ordered test set is used in manufacturing test.

4.3.2 Experimental Results

The proposed method was implemented in C language. Experiments were conducted on the larger ISCAS-89 benchmark circuits. Test vectors were obtained using an academic ATPG. Results reported in this paper are for tests to detect all detectable single stuck-line faults.

Test generation is done in a two phase process. In the first phase, tests were generated for all detectable faults, not considering the on-chip compaction logic. The generated tests were fault simulated including the on-chip compaction logic. In the second phase tests were generated including the on-chip compaction logic for the faults not detected in the first phase. In order to emulate the existence of Xs in tests for industrial designs we randomly injected unspecified values in the tests during the random fill phase of the test generation. These Xs created Xs in the test responses.

Different levels of on-chip compaction and percentages of unspecified bits in the test patterns were examined. For each circuit we considered expected on-chip compaction CR of 10, 20 and 30. This implies that the numbers of scan chains were 10, 20 and 30 with the single output parity tree on-chip compactor. In conjunction with different on-chip compaction levels we examined the effects of 1% and 2% Xs in the test patterns.

Results are presented in Table 11–13. The format of each table is the same. The main heading for each table lists the level of the expected compaction ratio (CR) and the percentage of unspecified values (X) in the test patterns. The percentages of unspecified values in each test pattern is given by X = 1.0% and X = 2.0%. Column 2 and 5 show the actual compaction ratio achieved by on-chip compaction (ACR1). Columns 3 and 6 show the actual compaction ratio achieved (ACR2) using ATE assisted compaction. Columns 4 and 7 show ΔCR = ACR2/ACR1, the ratio of the actual compactions without and with the ATE assisted compaction.

It can be seen that ACR1 is always lower than the expected compaction ratio. In addition, as the expected compaction ratio CR and the percentage of Xs increase, the actual compaction ACR1 diverges further down from CR. With ATE assisted compaction the increase in compaction, ΔCR, approaches the maximum possible 2X for a majority of the circuits. Due to relatively few test patterns required to test s35932 it does not follow the trend of approaching the maximum possible 2X for ΔCR.

Table 11: ATE Assisted Compaction (CR = 10)

	CR=10, X = 1.0%			CR=10, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	8.72	16.99	1.95	7.56	14.39	1.90
s9234	7.73	15.05	1.95	6.43	12.39	1.93
s13207	9.56	18.96	1.98	9.22	17.95	1.95
s15850	8.78	16.78	1.91	7.74	14.55	1.88
s35932	5.60	7.18	1.28	4.42	5.47	1.24
s38417	8.61	16.32	1.90	7.64	13.30	1.74
s38584	8.39	15.76	1.88	7.32	10.56	1.44

Table 12: ATE Assisted Compaction (CR = 20)

	CR=20, X = 1.0%			CR=20, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	13.78	26.13	1.90	10.51	19.29	1.84
s9234	11.59	22.30	1.92	9.12	16.98	1.86
s13207	16.59	32.14	1.94	14.72	26.90	1.83
s15850	14.74	26.87	1.82	11.47	20.29	1.77
s35932	7.62	10.64	1.40	6.76	7.88	1.17
s38417	14.59	25.43	1.74	10.42	16.69	1.60
s38584	12.71	20.68	1.63	8.80	12.64	1.44

Table 13: ATE Assisted Compaction (CR = 30)

	CR=30, X = 1.0%			CR=30, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	17.75	33.29	1.88	12.10	21.99	1.82
s9234	12.64	23.31	1.84	10.88	19.71	1.81
s13207	23.65	44.55	1.88	17.29	30.02	1.74
s15850	17.37	31.92	1.84	11.70	20.28	1.73
s35932	8.75	11.57	1.32	6.04	7.42	1.23
s38417	16.08	27.37	1.70	9.01	14.40	1.60
s38584	12.54	20.06	1.60	7.53	10.83	1.44

4.4 Fault Diagnosis Aware ATE Test Response Compaction

In order to increase yield and quality, the root cause of failures must be identified. In this capacity fault diagnosis has played a vital role in the success of modern IC manufacturing. The goals of testing and diagnosis methods are to minimize the cost of testing while simultaneously maintaining the ability to uniquely identify the location of a fault within the failing chip. Unfortunately these goals are not easily achieved simultaneously. Fault diagnosis techniques use the output responses of failing chips to determine the failing sites. The use of test response compaction methods may adversely affect the ability of fault diagnosis techniques to accurately diagnose the failing sites since less information is available to the fault diagnosis procedures if the test responses are compacted. Next we investigate an augmentation to the ATE assisted compaction technique which has minimal effect on diagnostic resolution. This technique is called fault aware ATE assisted compaction

4.4.1 Fault Diagnosis

Fault diagnosis techniques can be placed into two categories; cause-effect and effect-cause [77]. Cause-effect methods build a fault behavior database for modeled faults. Due to the database size, this approach is not practical for large designs. The effect-cause approach analyzes actual output responses of failing chips and determines the most likely fault(s) which could have caused the failure. This approach determines possible defect candidates by comparing the observed responses from the failing chip and the simulated responses of the circuit with injected faults, which are typically single or multiple stuck-at faults.

Due to the loss of information, response compaction reduces the ability of fault diagnosis. One alternative to improving fault diagnosis is to incorporate additional on-

chip logic that allows observing all scan chains directly, bypassing the on-chip compactor. Doing so provides direct access to the scan cells enabling the application of well-established fault diagnosis techniques. Unfortunately, the approach of bypassing the on-chip logic has two drawbacks. First, additional overhead is required to facilitate the bypass mode. While this overhead may be acceptable, the second drawback is that two separate test sessions are required; a test session without using bypass mode for fault detection and a test session using bypass mode for fault diagnosis. In addition, such an approach does not facilitate on-line volume diagnosis.

Diagnosis techniques which use the compacted output response can be placed into two categories; indirect and direct [76]. *Indirect diagnosis* performs diagnosis by first identifying scan cells which have captured erroneous values and then the identified scan cells are used with ATPG based diagnostic algorithms designed for circuits without compactors. This approach assumes only a single error exists at any scan cycle and multiple errors cannot be uniquely identified. In reality, a single defect may produce multiple errors, therefore fault diagnostic resolution may be limited. *Direct diagnosis* [76] provides a generalized effect-cause approach for compactor-based designs by using the compacted responses to make diagnosis decisions. The advantages of this approach are that it can be generally applied to a variety of compactors, existing diagnosis algorithms can be used, it is an on-line methodology and it does not make the single error assumption. The diagnosis technique used in this work uses direct diagnosis.

Using ATE assisted compaction all the faults detected without using it are also detected using it. However, when a merged expected response stored in the ATE detects a failing test it could be due to failing the first or the second test of the corresponding pair of tests. This may also have some effect on fault diagnosis.

When analyzing the ability to diagnose single stuck-at faults, an effective measure is to partition the set of single stuck-at faults into equivalence classes based on their output responses for tests. Take for example the responses shown in Table 14. The CUT

for this example has two outputs and all seven single stuck-at faults are detected by three tests. The fault-free response and the responses due to each of the eight single stuck-at faults are shown. The detected faulty outputs are shown in bold. It can be seen that the responses due to faults f_1, f_7 and f_8 are unique. The responses due to f_2 and f_4 are the same and the response due to f_3, f_5, and f_6 are the same. Thus, we can partition the faults into the following equivalence classes: $\{f_1\}$, $\{f_7\}$, $\{f_8\}$, $\{f_2, f_4\}$, $\{f_3, f_5, f_6\}$. If the response seen from applying the three tests is 11 11 11 the fault location cannot be uniquely identified since it could be f_3, f_5, or f_6.

Table 14: Fault Equivalence Class Example

	FF	f1	f2	f3	f4	f5	f6	f7	f8
R1	00	0**1**	00	**11**	00	**11**	**11**	**1**0	**11**
R2	01	01	**1**1	**1**1	**1**1	**1**1	**1**1	**10**	0**0**
R3	10	1**1**	10	1**1**	10	1**1**	1**1**	1**1**	**0**0

Extensive research has been done to develop efficient diagnosis procedures [75,78,79]. A diagnosis procedure adopted from [78] is used to report diagnosis results in this work [81] for double stuck-at faults. This method [80] gave quite a bit better direct diagnosis results than the recent procedure presented in [75]. The approach uses four parameters to estimate how well a candidate fault matches the observed faulty behavior. The initial set of candidate faults are all the structurally collapsed single stuck-at faults using fault equivalences at a gate. Let T be the set of test patterns applied to the device under diagnosis (DUD). An observed output is said to have failed if the observed response is different from the response of the fault-free circuit, otherwise it is said to have

passed. A test in T that failed (passed) the DUD on the tester is referred to as a *failing pattern* (*passing pattern*). For each test pattern t $\in$ T, the output response of fault simulating a candidate fault can be divided into four sets: sftf, sftp, sptf and sptp. Their definitions are as follows:

sftf: The number of observed outputs which fail in the candidate fault simulation and fail in the DUD

sftp: The number of observed outputs which fail in the candidate fault simulation and pass in the DUD

sptf: The number of observed outputs which pass in the candidate fault simulation and fail in the DUD

sptp: The number of observed outputs which pass in the candidate fault simulation and pass in the DUD

These parameters are illustrated in Figure 13 where FM is the circuit with an injected candidate fault. The outputs of DUD and FM which have darker shading represent failing values.

Let FT (PT) be the subset of tests of the test set T that failed (passed) the DUD. Let DS_a be the diagnostic score for candidate fault f_a given by $DS_a = SFTF_a + SPTP_a$, where $SFTF_a = \sum_{t\,\epsilon FT} sftf$ and $SPTP_a = \sum_{t\,\epsilon FT} sptp(t) + 0.5\sum_{t\,\epsilon PT} sptp(t)$. Candidate faults are first ordered in decreasing value according to DS. Faults with the same value of DS are reordered in increasing value of SFTP, where$SFTP = \sum_{t\,\epsilon T} sftp$ [80].

The two diagnostic measures we use are *not diagnosed* and *first hit*. To correctly diagnose a double stuck-at fault, we assume that at least one of the injected faults needs to appear in the top three ranks of the list of faults ordered using the scores described above. Note that in each rank there could be more than one fault if their parameter values are identical. The measure *not diagnosed* reports the number of times the procedure does

not place any one of the injected faults in the top three ranks. The measure *first hit* reports the number of times in which the procedure placed at least one of the injected faults at the top rank.

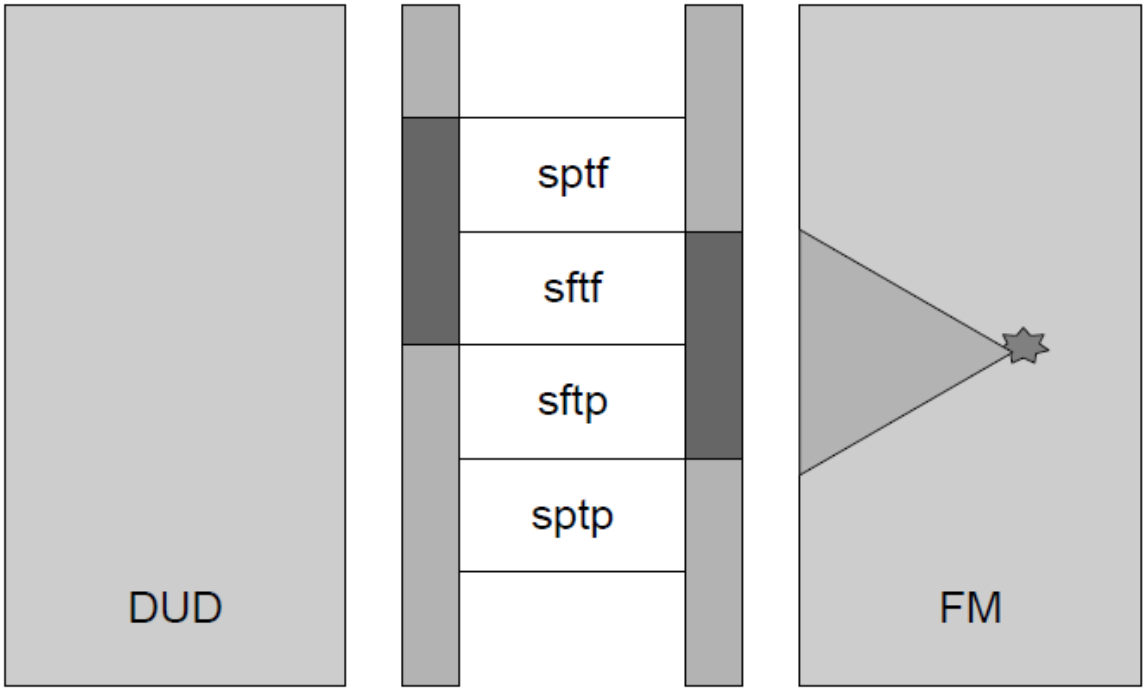

Figure 13: Illustration of Diagnosis Parameters

4.4.2 Procedure for Fault Aware ATE Assisted Compaction

As in the ATE assisted compaction technique [72], the inputs to the compaction procedure are the test set $T = \{t_1, t_2, \ldots, t_n\}$, the set of corresponding fault-free responses from the CUT $R = \{r_1, r_2, \ldots, r_n\}$ and the set of faults F detected by T. There are six steps, outlined below, in the procedure. The procedure determines the pairs of test whose

linear sums are stored (called merging of test responses in this work) in the ATE instead of the two individual responses of the merged pairs. It should be noted that the procedure given below differs from the earlier work [72] in Step 3 which now uses a heuristic to determine pairs of test responses to be merged without affecting diagnostic resolution. Also Step 5 was not included in the earlier procedure and is used to achieve close to maximum possible 2X additional compaction using ATE as the sum of two responses is stored in the ATE.

Step 1: Determine using fault simulation the set of faults $F_1 = \{f_{11}, f_{12}, \ldots, f_{1q}\}$ detected only once and the set of faults $F_2 = \{f_{21}, f_{22}, \ldots, f_{2r}\}$ detected only twice. These sets of faults guide the selection of test pairs whose responses are replaced by their linear sum in ATE assisted compaction. This is a heuristic used to moderate the run time of the procedure.

Step 2: Determine using fault simulation the pairs of responses (r_i, r_j), $r_i, r_j \in R$, that should not be merged in to a single response to be stored as an expected response in the ATE. By merging two responses we mean computing the exclusive OR of them. Let R_p be the set of such pairs of responses. We use a heuristic procedure to determine R_p. A pair of responses is included in R_p if merging the pair of responses masks the detection of a fault in F_1 or in F_2 computed in Step 1.

Step 3: In this step we find the pairs of responses in R that are merged. Let this set of pairs be R_m. Pairs in R_m are determined by the following procedure. It uses the number of fault effects, called

NUM_BEF (r_i, r_j), that alias or are masked when the pair of responses r_i and r_j are merged. It should be noted that NUM_BEF (r_i, r_j) is the count of the fault effects which are propagated to compacted responses and not the faults themselves that are aliased or masked. Let R_c be the set of pairs in R but not in R_p.

```
while Rc ≠ ∅
    for all pairs (ri, rj) ∈ Rc
        obtain NUM_BFE(ri, rj) for the faults in F
    end for
    MIN_BFE_PAIR = pair with minimum NUM_BFE
    Drop the faults in F detected by the merged response of the
        pair MIN_BFE_PAIR
    Remove MIN_BFE_PAIR from Rc and add to Rm
end while
```

It should be pointed out that the procedure includes a pair of responses that minimize the number of blocked/aliased fault effects by each response pair included in R_m. It should also be noted that a response r_j may not be merged with any other response if all possible response pairs that include r_j are in R_p.

Step 4: In this step we fault simulate all faults in F using the compacted responses in R_m and the responses in R that were not merged with any other response. It is possible that some faults in F are not detected due to the merging of some responses since the

procedure only avoids merging pairs in R_p which insures avoiding loss of detection of faults in F_1 and F_2. Let the set of faults in F that are not detected by response merging be F_u.

Step 5: If $F_u \neq \emptyset$, responses in R_m are unmerged until $F_u = \emptyset$. Let the set of responses which remain merged be R_2 and the set of responses that are not merged be R_1. We next consider pairing each response in R_1 with another response in R_1 and include in R_2 those pairs that do not affect fault detection when merged (i.e. whilst maintaining $F_u = \emptyset$). In this step a response in R_1 is merged with at most one other response in R_1.

Step 6: We order the tests in T such that the tests t_i and t_j corresponding to the two responses of a (r_i, r_j) pair in R_2 appear one immediately after the other. The ordered test set is used in manufacturing test.

4.4.3 Fault Detection and Diagnosis

Using ATE assisted compaction all the faults detected without using it are also detected using it. However, merging responses can result in loss of information due to aliasing and fault masking. This loss of information may result in reduced diagnostic resolution. In the previous work on ATE assisted compaction [72], no attempt was made to minimize the effect of reduced diagnostic resolution due to merging of responses. Responses were selected in a greedy manner based on their order. Merging was avoided only for pairs that were placed in R_p. In this work in order to minimize impact on diagnostic resolution, a simple heuristic method, discussed in Step 3 of the procedure, is used to select pairs which minimize the number of fault effects which are blocked by

either aliasing or fault masking. To increase the numbers of merged responses we also use an iterative procedure in Step 5. This allowed obtaining close to the maximum of 2X compaction using ATE assisted compaction whilst minimally compromising diagnostic resolution.

4.4.4 Experimental Results

The proposed method was implemented in C language. Experiments were conducted on the larger ISCAS-89 benchmark circuits for collapsed stuck-at faults. Test vectors were obtained using an academic ATPG. Results reported in this paper are for tests to detect all detectable single stuck-at faults.

Test generation is done in two phases. In the first phase, tests were generated for all detectable faults, not considering the on-chip compaction logic and fault simulated using the on-chip compactor. In the second phase tests were generated including the on-chip compaction logic for the faults not detected in the first phase. In order to emulate the existence of Xs in tests for industrial designs we randomly injected unspecified values in the tests during the random fill phase of the test generation. These Xs created Xs in the test responses. For each circuit we considered expected compaction CR of 10, 20 and 30. This implies that the numbers of scan chains were 10, 20 and 30 with the single output parity tree on-chip compactor. In addition we examined the effects of 1% and 2% Xs in the test patterns.

Compaction data is presented in Tables 15-17, single stuck-at equivalence class data is presented in Tables 18-20 and double stuck-at fault diagnosis data is presented in Tables 21-23. All of the tables follow a similar format. Each table has two sections with headings for compaction ration (CR) and unknown percentage (X).

For Tables 15-17, Column 2 and 5 show the actual compaction ratio achieved by on-chip compaction (ACR1). Columns 3 and 6 show the actual compaction ratio achieved (ACR2) using fault aware ATE assisted compaction. Columns 4 and 7 show

ΔCR = ACR2/ACR1, the ratio of the actual compaction without and with the ATE assisted compaction.

Table 15: Fault Aware ATE Assisted Compaction (CR = 10)

	CR=10, X = 1.0%			CR=10, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	8.72	17.22	1.97	7.56	15.04	1.99
s9234	7.73	15.32	1.98	6.43	12.77	1.98
s13207	9.56	18.96	1.98	9.01	17.82	1.98
s15850	8.78	17.45	1.99	7.78	15.38	1.98
s35932	5.60	8.84	1.58	4.42	6.05	1.37
s38417	8.61	17.12	1.99	7.60	14.41	1.90
s38584	8.39	16.60	1.98	7.25	10.95	1.51

Table 16: Fault Aware ATE Assisted Compaction (CR = 20)

	CR=20, X = 1.0%			CR=20, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	13.78	27.26	1.98	10.56	20.85	1.98
s9234	11.72	23.27	1.99	9.20	18.35	1.99
s13207	16.90	33.67	1.99	15.15	30.00	1.98
s15850	14.74	28.98	1.97	11.47	22.55	1.97
s35932	7.50	12.36	1.65	6.73	8.63	1.28
s38417	14.46	27.36	1.89	10.42	19.94	1.91
s38584	12.61	24.43	1.94	8.67	15.54	1.79

Table 17: Fault Aware ATE Assisted Compaction (CR = 30)

	CR=30, X = 1.0%			CR=30, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	17.83	35.49	1.99	12.29	24.51	1.99
s9234	20.00	39.65	1.98	11.08	22.06	1.99
s13207	22.89	45.62	1.99	18.01	35.74	1.98
s15850	17.37	34.43	1.98	11.84	23.48	1.98
s35932	8.75	14.22	1.63	6.15	9.35	1.52
s38417	15.77	31.30	1.98	9.26	18.08	1.95
s38584	12.54	24.75	1.97	7.35	14.48	1.97

It can be seen that ACR1 is always lower than the expected compaction ratio. In addition, as the expected compaction ratio CR and the percentage of Xs increase, the actual compaction ACR1 diverges further down from CR. With fault aware ATE assisted compaction the increase in compaction, ΔCR, is approximately the maximum possible 2X for most circuit experiments. The one exception is circuit s35932. This circuit requires relatively few test patterns to detect all single stuck-at faults. Therefore a substantial number of responses remain unmerged by the proposed method in order to maintain single stuck-at fault coverage and diagnostic resolution.

In Tables 18-20, results on the sizes of equivalence classes of single stuck-at faults are reported for the on-chip compactor (Comp), the fault aware ATE assisted compaction (AAC) technique and their differences (Δ). For both Comp and AAC the maximum equivalence class size (Mx) and average equivalence class size (Av) are reported. Columns 2 and 8 report Mx for Comp while columns 4 and 10 report Mx for AAC. Columns 3 and 9 report Av for Comp while columns 5 and 11 report Av for AAC.

Columns 6 and 12 report Mx for Δ. Mx for Δ is the difference of Mx for AAC and Comp. Columns 7 and 13 report Av for Δ. Av for Δ is the percentage increase in Av from Comp to AAC.

For most cases the maximum value of Δ, Mx, is less than 3. While there are a few cases in which there is a significant increase in Mx for AAC, this has little impact given that the maximum value of the average value Av of Δ is only 4.09% for all cases.

Table 18: Single Stuck-at Equivalence Class (CR = 10)

	CR=10, X = 1.0%						CR=10, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	7	1.11	7	1.12	0	0.92%	7	1.11	7	1.13	0	2.33%
s9234	7	1.25	10	1.26	3	1.36%	9	1.25	10	1.27	1	1.45%
s13207	9	1.22	9	1.23	0	0.88%	18	1.22	18	1.23	0	1.16%
s15850	28	1.22	28	1.23	0	0.88%	28	1.22	28	1.23	0	1.10%
s35932	7	1.45	9	1.46	2	0.99%	6	1.46	10	1.48	4	1.48%
s38417	12	1.12	21	1.13	9	0.87%	13	1.12	16	1.13	3	0.69%
s38584	5	1.09	6	1.09	1	0.46%	6	1.09	7	1.09	1	0.41%

Table 19: Single Stuck-at Equivalence Class (CR = 20)

	CR=20, X = 1.0%						CR=20, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	4	1.12	5	1.14	1	1.54%	5	1.12	6	1.13	1	1.29%
s9234	6	1.24	9	1.26	3	1.46%	7	1.24	8	1.27	1	1.98%
s13207	9	1.23	9	1.26	0	2.47%	18	1.23	18	1.24	0	0.93%
s15850	28	1.22	28	1.23	0	0.53%	28	1.22	28	1.23	0	0.75%
s35932	7	1.43	10	1.46	3	1.73%	8	1.45	17	1.49	9	2.63%
s38417	13	1.11	13	1.12	0	0.48%	15	1.11	15	1.12	0	0.69%
s38584	6	1.09	6	1.09	0	0.66%	5	1.09	7	1.09	2	0.83%

Table 20: Single Stuck-at Equivalence Class (CR = 30)

	CR=30, X = 1.0%						CR=30, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	4	1.12	6	1.15	2	2.80%	4	1.11	5	1.14	1	3.03%
s9234	5	1.25	8	1.27	3	1.43%	7	1.25	10	1.28	3	2.42%
s13207	9	1.23	9	1.28	0	3.85%	13	1.23	19	1.26	6	2.47%
s15850	28	1.22	28	1.23	0	0.79%	28	1.22	28	1.24	0	1.11%
s35932	8	1.43	13	1.47	5	2.81%	11	1.44	25	1.50	14	4.09%
s38417	12	1.11	12	1.12	0	0.90%	17	1.11	17	1.11	0	0.80%
s38584	6	1.09	7	1.09	1	0.76%	5	1.08	7	1.10	2	1.18%

In Tables 21-23, double stuck-at fault diagnosis results are reported for the on-chip compactor (Comp) and the fault aware ATE assisted compaction (AAC) technique. For these experiments, 10,000 randomly selected double stuck-at faults were injected. Results are reported for the number of instances where the diagnosis procedure did not place either of the injected faults in the top three ranks (Not_Diag) and the number of instances when the diagnosis procedure placed at least one of the injected double faults in the top rank (First_Hit). Columns 2 and 6 report Not_Diag for Comp while columns 3 and 7 report Not_Diag for AAC. Columns 4 and 8 report First_Hit for Comp while columns 5 and 9 report First_Hit for AAC.

From the results reported in Tables 21-23, it can be seen that no significant difference exists between Comp and AAC in the values for both Not_Diag and First_Hit for all of the different compaction ratios (CR) and percentage of unknowns (X). On average there are no more than seven faults which are detected with the on-chip compactor which are not detected with the merged responses of the fault aware ATE assisted compaction technique. On average there are no more than 14 faults that are

ranked as First_Hit in the on-chip compactor which are not ranked at First_Hit for the merged responses of the fault aware ATE assisted compaction technique.

Table 21: Double Stuck-at Fault Diagnosis (CR = 10)

	CR=10, X = 1.0%				CR=10, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	20	31	9937	9931	18	32	9935	9923
s9234	47	48	9929	9922	49	53	9924	9921
s13207	9	14	9971	9962	6	11	9978	9971
s15850	17	24	9959	9952	14	16	9977	9974
s35932	115	120	9875	9870	101	100	9887	9886
s38417	1	5	9988	9983	0	2	9992	9994
s38584	16	17	9982	9981	27	31	9969	9965
Ave	32	37	9949	9943	31	35	9952	9948

Table 22: Double Stuck-at Fault Diagnosis (CR = 20)

	CR=20, X = 1.0%				CR=20, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	21	35	9920	9904	21	21	9917	9926
s9234	61	65	9901	9892	56	59	9912	9899
s13207	18	28	9953	9946	26	31	9940	9936
s15850	21	27	9960	9950	10	29	9969	9944
s35932	111	120	9881	9870	115	119	9873	9872
s38417	2	2	9991	9982	2	4	9982	9980
s38584	22	23	9970	9970	18	21	9976	9974
Ave	37	43	9939	9931	35	41	9938	9933

Table 23: Double Stuck-at Fault Diagnosis (CR = 30)

	CR=30, X = 1.0%				CR=30, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	35	46	9885	9855	45	40	9880	9882
s9234	53	57	9915	9905	73	68	9897	9878
s13207	25	37	9940	9919	23	32	9957	9943
s15850	20	26	9948	9936	26	31	9948	9930
s35932	92	104	9889	9877	107	104	9865	9879
s38417	3	10	9985	9968	3	6	9991	9980
s38584	23	20	9971	9970	17	20	9971	9971
Ave	36	43	9933	9919	42	43	9930	9923

4.4.5 Cost of Fault Aware ATE Assisted Compaction

There is a cost associated with achieving the additional output response compaction while maintaining fault diagnosis resolution. This cost is in terms of the additional CPU run-time required to search for responses to merge that minimally affect fault diagnosis resolution. In ATE assisted compaction, a simple greedy algorithm is used to selected responses to be merged. The greedy algorithm selects the first response in the set and determines if the next response in the list does not violate the condition that a pair in R_p is not included in R_m. If so, the responses are merged, otherwise the next response in the set is considered. In fault aware ATE assisted compaction all valid combinations of pairs are considered when selecting which responses to merge. The valid responses are held in R_c, where no pair in R_p is included in R_c. Pairs in R_c are evaluated to determine how many fault effects are blocked from being detected due to aliasing or masking. This results in many additional comparisons in order to determine the pair of merged responses that minimally impacts fault diagnosis resolution.

The results of comparing the cost of ATE assisted compaction and fault aware ATE assisted compaction can be seen in Tables 24-26. Column 2 and 4 lists the time, in seconds (s), it takes to process the selection of merged responses for ATE assisted compaction. Column 3 and 5 lists the time, in seconds (s), it takes to process the selection of merged responses for fault aware ATE assisted compaction.

Table 24: Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 10)

	CR=10, X = 1.0%		CR=10, X = 2.0%	
Ckt	AAC (s)	FA-AAC (s)	AAC (s)	FA-AAC (s)
s5378	1.45	80.32	1.92	95.56
s9234	3.20	396.34	3.88	580.25
s13207	16.57	1602.44	18.02	1447.20
s15850	10.35	510.73	12.18	586.24
s35932	27.97	44.22	37.49	96.40
s38417	81.34	4734.72	89.34	3929.08
s38584	99.81	3138.72	119.30	12523.03

Table 25: Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 20)

	CR=20, X = 1.0%		CR=20, X = 2.0%	
Ckt	AAC (s)	FA-AAC (s)	AAC (s)	FA-AAC (s)
s5378	0.99	69.59	1.33	116.46
s9234	2.26	414.34	2.93	728.15
s13207	9.70	927.30	10.86	1111.13
s15850	6.46	353.24	8.37	587.89
s35932	21.64	116.58	25.92	56.45
s38417	47.63	2520.37	65.32	4009.99
s38584	69.43	3060.36	98.13	10320.48

Table 26: Run-time Comparison of ATE Assisted Compaction and Fault Aware ATE Assisted Compaction (CR = 30)

	CR=30, X = 1.0%		CR=30, X = 2.0%	
Ckt	AAC (s)	FA-AAC (s)	AAC (s)	FA-AAC (s)
s5378	0.80	72.53	1.18	185.04
s9234	1.35	241.38	2.54	887.22
s13207	7.34	794.56	9.20	1261.38
s15850	5.64	401.92	8.38	1189.48
s35932	18.20	98.74	26.47	128.46
s38417	44.40	2493.73	72.65	7865.89
s38584	70.00	5494.52	112.53	23233.59

Independent experiments were also conducted for the ISCAS-89 benchmark circuits without fault collapsing. Compaction data is presented in Tables 27-29, single stuck-at equivalence class data is presented in Tables 30-32 and double stuck-at fault diagnosis data is presented in Tables 33-35. All of the tables follow a similar format. Each table has two sections with headings for compaction ration (CR) and unknown percentage (X).

The format of Tables 27-29 is the same as Tables 15-17. Column 2 and 5 show the actual compaction ratio achieved by on-chip compaction (ACR1). Columns 3 and 6 show the actual compaction ratio achieved (ACR2) using ATE assisted compaction. Columns 4 and 7 show ΔCR = ACR2/ACR1, the ratio of the actual compaction without and with the ATE assisted compaction.

The results for the experiments with no fault collapsing are similar to those with fault collapsing. ACR1 is lower than the expected compaction ratio and as the expected compaction ratio CR and the percentage of Xs increase, the actual compaction ACR1 diverges further down from CR. In addition, the fault aware ATE assisted compaction

technique provides an increase in compaction, ΔCR, which is approximately the maximum possible 2X for most circuit experiments. As mentioned in the results for the experiments with fault collapsing, the circuit s35932 is an exception due to requiring relatively few test patterns to detect all single stuck-at faults. Therefore a substantial number of responses remain unmerged in order to maintain single stuck-at fault coverage and diagnostic resolution.

Table 27: Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 10)

	CR=10, X = 1.0%			CR=10, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	8.40	16.57	1.97	7.39	14.68	1.99
s9234	8.55	16.93	1.98	6.68	13.31	1.99
s13207	9.27	18.40	1.98	8.89	17.59	1.98
s15850	8.56	16.79	1.96	7.87	15.43	1.96
s35932	5.49	8.45	1.54	5.11	6.38	1.25
s38417	9.05	17.73	1.96	8.39	15.66	1.87
s38584	8.19	15.59	1.90	7.19	11.50	1.60

Table 28: Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 20)

	CR=20, X = 1.0%			CR=20, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	13.31	26.46	1.99	10.24	20.39	1.99
s9234	12.75	25.30	1.98	9.79	19.46	1.99
s13207	17.12	34.00	1.99	15.30	30.32	1.98
s15850	13.79	26.98	1.96	10.45	20.72	1.98
s35932	8.10	12.15	1.50	7.04	8.97	1.27
s38417	14.54	28.28	1.95	10.60	19.49	1.84
s38584	12.06	23.44	1.94	8.42	14.52	1.73

Table 29: Fault Aware ATE Assisted Compaction - No Fault Collapsing (CR = 30)

	CR=30, X = 1.0%			CR=30, X = 2.0%		
Ckt	ACR1	ACR2	ΔCR	ACR1	ACR2	ΔCR
s5378	16.11	31.76	1.97	12.49	24.70	1.98
s9234	16.32	32.53	1.99	11.15	22.14	1.99
s13207	22.75	45.20	1.99	17.90	35.43	1.98
s15850	17.33	34.34	1.98	11.99	23.75	1.98
s35932	9.07	14.51	1.60	6.10	9.23	1.51
s38417	15.35	29.30	1.91	8.19	15.87	1.94
s38584	11.20	22.07	1.97	6.80	13.48	1.98

The format of Tables 30-32 are the same as for Tables 18-20. Columns 2 and 8 report Mx for Comp while columns 4 and 10 report Mx for AAC. Columns 3 and 9 report Av for Comp while columns 5 and 11 report Av for AAC. Columns 6 and 12 report Mx for Δ. Mx for Δ is the difference of Mx for AAC and Comp. Columns 7 and 13 report Av for Δ. Av for Δ is the percentage increase in Av from Comp to AAC.

As expected, without fault collapsing the Av for both Comp and AAC are larger. However, similar to the results with fault collapsing, no significant difference exists for ΔAv for the experiments without fault collapsing.

Table 30: Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 10)

	CR=10, X = 1.0%						CR=10, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	56	3.57	56	3.60	0	0.86%	56	3.57	56	3.64	0	1.88%
s9234	100	5.11	100	5.18	0	1.49%	100	5.10	100	5.11	0	0.18%
s13207	82	5.18	82	5.24	0	1.16%	82	5.18	82	5.22	0	0.84%
s15850	131	5.21	131	5.23	0	0.53%	131	5.21	131	5.24	0	0.59%
s35932	24	3.58	24	3.62	0	1.09%	23	3.64	24	3.69	1	1.46%
s38417	69	4.20	69	4.22	0	0.56%	69	4.19	69	4.23	0	0.90%
s38584	53	3.32	62	3.33	9	0.46%	53	3.31	53	3.33	0	0.52%

Table 31: Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 20)

	CR=20, X = 1.0%						CR=20, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	56	3.61	56	3.69	0	2.34%	56	3.60	58	3.71	2	3.20%
s9234	100	5.08	100	5.15	0	1.26%	100	5.07	100	5.20	0	2.50%
s13207	82	5.21	82	5.32	0	2.07%	82	5.20	82	5.26	0	1.14%
s15850	131	5.22	131	5.24	0	0.43%	131	5.21	131	5.30	0	1.78%
s35932	24	3.60	24	3.68	0	2.19%	24	3.60	34	3.69	10	2.72%
s38417	69	4.15	69	4.17	0	0.62%	69	4.15	69	4.18	0	0.67%
s38584	53	3.31	53	3.33	0	0.64%	53	3.30	53	3.33	0	0.74%

Table 32: Single Stuck-at Equivalence Class - No Fault Collapsing (CR = 30)

	CR=30, X = 1.0%						CR=30, X = 2.0%					
	Comp		AAC		Δ		Comp		AAC		Δ	
Ckt	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av	Mx	Av
s5378	56	3.59	56	3.64	0	1.51%	56	3.59	56	3.66	0	1.69%
s9234	100	5.11	100	5.19	0	1.61%	100	5.10	100	5.21	0	2.08%
s13207	82	5.25	82	5.33	0	1.45%	82	5.23	82	5.32	0	1.71%
s15850	131	5.23	131	5.31	0	1.57%	131	5.23	131	5.31	0	1.42%
s35932	23	3.55	26	3.66	3	3.17%	25	3.57	25	3.73	0	4.34%
s38417	69	4.13	69	4.15	0	0.54%	69	4.13	69	4.16	0	0.84%
s38584	53	3.30	53	3.32	0	0.56%	53	3.29	53	3.33	0	1.07%

The format of Tables 33-35 are the same as those reported for Tables 21-23, double stuck-at fault diagnosis results are reported for the on-chip compactor (Comp) and the ATE assisted compaction (AAC) technique. For these experiments, 10,000 randomly selected double stuck-at faults were injected. Results are reported for the number of instances where the diagnosis procedure did not place either of the injected faults in the top three ranks (Not_Diag) and the number of instances when the diagnosis procedure placed at least one of the injected double faults in the top rank (First_Hit). Columns 2

and 6 report Not_Diag for Comp while columns 3 and 7 report Not_Diag for AAC. Columns 4 and 8 report First_Hit for Comp while columns 5 and 9 report First_Hit for AAC.

Table 33: Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 10)

	CR=10, X = 1.0%				CR=10, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	17	24	9942	9929	19	18	9943	9929
s9234	56	60	9913	9904	46	44	9923	9902
s13207	9	16	9970	9958	17	21	9966	9957
s15850	15	20	9965	9951	8	20	9972	9953
s35932	109	115	9881	9877	109	111	9883	9878
s38417	3	4	9992	9988	3	6	9985	9984
s38584	28	27	9966	9964	33	31	9964	9963
Ave	34	38	9947	9939	34	36	9948	9938

Table 34: Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 20)

	CR=20, X = 1.0%				CR=20, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	37	33	9890	9864	31	39	9908	9874
s9234	57	63	9912	9876	65	57	9897	9882
s13207	44	54	9929	9913	31	40	9936	9911
s15850	36	49	9945	9902	27	40	9941	9918
s35932	116	123	9870	9862	126	126	9861	9858
s38417	7	10	9986	9975	13	19	9981	9970
s38584	31	33	9957	9954	26	28	9963	9964
Ave	47	52	9927	9907	46	50	9927	9911

Table 35: Double Stuck-At Fault Diagnosis - No Fault Collapsing (CR = 30)

	CR=30, X = 1.0%				CR=30, X = 2.0%			
	Not_Diag		First_Hit		Not_Diag		First_Hit	
Ckt	Comp	AAC	Comp	AAC	Comp	AAC	Comp	AAC
s5378	57	41	9858	9853	59	34	9842	9857
s9234	66	61	9888	9854	89	67	9853	9844
s13207	61	61	9907	9870	54	66	9894	9879
s15850	40	63	9926	9876	35	57	9930	9891
s35932	116	124	9864	9856	107	111	9863	9868
s38417	11	25	9978	9966	16	30	9970	9949
s38584	38	41	9949	9948	29	32	9956	9959
Ave	56	59	9910	9889	56	57	9901	9892

Similar to the results reported for experiments with fault collapsing, the results reported in Tables 33-35 show that no significant difference exists between Comp and AAC in the values of both Not_Diag and First_Hit for all of the different compaction ratios (CR) and percentage of unknowns (X). On average there are no more than five faults which are detected with the on-chip compactor which are not detected with the merged responses of the fault aware ATE assisted compaction technique. On average there are no more than 21 faults that are ranked as First_Hit in the on-chip compactor which are not ranked at First_Hit for the merged responses of the fault aware ATE assisted compaction technique.

4.5 Processing CUT Response on ATE

To illustrate how the responses from the CUT on the test floor are processed by ATE using ATE assisted compaction we use the following example.

Example: Assume that for a CUT there are ten tests with the corresponding 10 responses at the output of the on-chip compactor. Assume that the procedure given above determined $R_2 = \{(r_1, r_3), (r_2, r_5), (r_4, r_8), (r_7, r_{10})\}$, $R_1 = \{r_6, r_9\}$. To satisfy the requirement in Step 6 regarding the order of tests to be applied one can use the order of tests t_1, t_3, t_2, t_5, t_4, t_8, t_7, t_{10}, t_6, t_9 or t_3, t_1, t_5, t_2, t_4, t_8, t_{10}, t_7, t_9, t_6 as well as many other orders.

The ATE will need to keep track of whether the response coming in is the first or the second response for the tests corresponding to a pair of responses in R_2 and be able to generate the linear sum of a stored response and the incoming response. If the ATE is receiving the first response of a pair it is temporarily stored. If the ATE is receiving the second response in the sequence, the linear sum of the first response and the second response is generated as the second response is being received. This can be seen in Figure 14. For ease of readability we show stored responses containing 0s, 1s and Xs instead of the pair of vectors normally used to store such vectors as discussed in Section 2. The result of the linear sum is then compared to the expected response. In Figure 14a the first response is ready to be read into the ATE. In Figure 14b the first response is temporarily stored in the ATE. In Figure 14c the second response is ready to be read into the ATE to compute the linear sum of the first and second response. In Figure 14d the linear sum has been computed and is ready for comparison with the known fault free response stored in the ATE.

The additional processing on the ATE can be seen to be nominal. The ATE does need memory to store one response from the CUT corresponding to the first of a pair of responses in R_2.

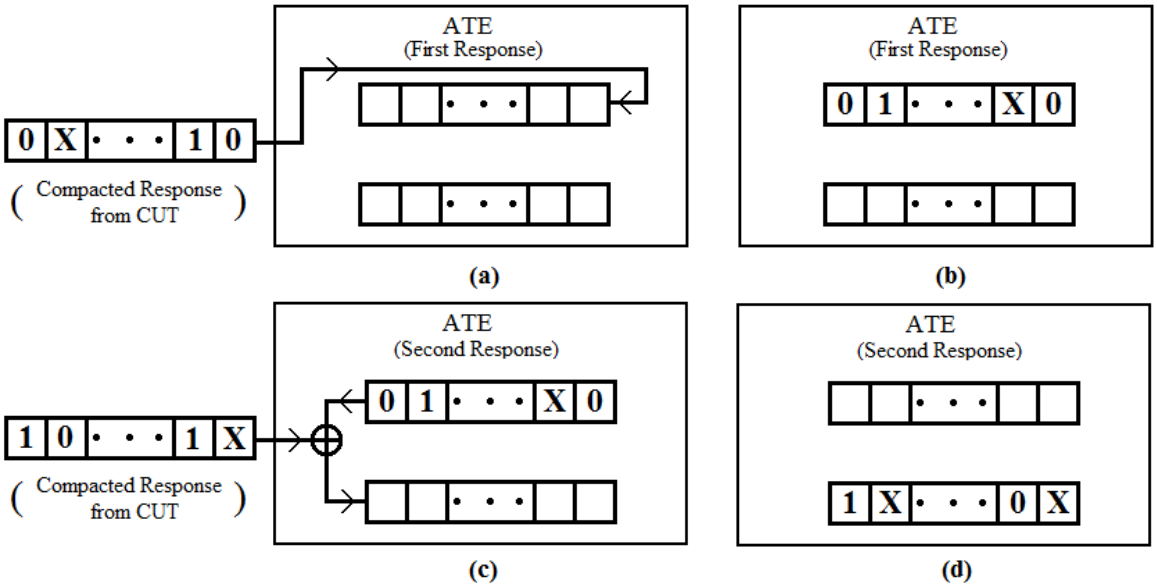

Figure 14: ATE Processing Steps (a) First response from the CUT corresponding to a pair R2, (b) First response of the pair stored in the ATE, (c) Second response of the pair from the CUT, (d) The linear sum of the first and the second response is compared with the expected response stored in the ATE

4.6 Conclusions

In this work we have investigated a new method of test response compaction in order to reduce test response data volume for multiple scan chain designs. The proposed method, called ATE assisted compaction, uses testers to achieve additional compaction, without compromising fault coverage, over what is achieved through on-chip compaction. Instead of storing an expected response for each test, the linear sum of two consecutive tests is stored. Simple heuristics are used to guide the procedure in order to combine compatible pairs of responses to avoid the effects of aliasing. Since the proposed method does not add any on-chip logic, it is applicable to any design. Experimental results show that the proposed method is effective in reducing test response data without loss of fault coverage.

An augmentation of this work was investigated in order to reduce test response data volume for multiple scan chain designs while not adversely affecting diagnostic resolution. Simple heuristics are used to guide the procedure in order to combine compatible pairs of responses to avoid the effects of aliasing and fault masking. Experimental results show that the proposed method is effective in reducing test response data without loss of fault coverage and not adversely affecting diagnosis resolution. In addition, experiments were conducted without fault collapsing. These experiments were consistent with the experiments which incorporated fault collapsing.

CHAPTER V: CONCLUSIONS

Transistor counts for integrated circuits (IC) continue to increase, resulting in increased transistor densities. This, in turn, increases the complexity of developing efficient and high quality manufacturing tests that guarantee proper functionality. Improved testing techniques attempt to address such issues as power consumption during test, detection of a variety of modeled faults, data volume and fault diagnosis.

Increased IC complexity results in increased gate counts, number of scan elements and number of inputs and outputs. Increased gate counts result in an increase in the number of potential fault locations, leading to an increased number of tests. Increased numbers of scan elements, inputs, and outputs results in an increased amount of data required for each test. Thus, as IC complexity increases, both the number of tests and the data required for each test increases, resulting in increased data volume. With the increase in the number of tests and the data required to test the IC, the power consumed during test becomes a concern. In addition, due to increased transistor densities, all physical faults do not behave according to a single fault model.

The main objective of this thesis was to address issues of power consumed during testing, tests to detect faults across multiple fault models, reduction of data volume and reduction of data volume with minimal impact on fault diagnosis resolution.

5.1 Completed Research

First, in Chapter II, a technique to reduce the power consumed during shift and capture cycles in scan testing for multiple scan chain designs is investigated. The stuck-at fault model was used in this investigation. In this work, additional hardware and control data were used in conjunction with a reconfigurable switch broadcast scan based design to control how each of the scan chains receive test data. Experiments were conducted on ISCAS-89 benchmark circuits. Experimental results show that switching

activity for both shift and capture power can be reduced simultaneously in addition to reducing test data volume. In order to achieve the reduction in shift and capture power, additional control data was required. Thus a tradeoff must be made between reduction of shift and capture power and the reduction of data volume.

Second, in Chapter III, a test generation technique was investigated to detect interconnect open faults in the presence of unknown circuit parameters. The technique uses the stuck-at fault model along with circuit layout information to generate tests. Circuit layout information is used to determine neighboring lines within the circuit in order to specify their values. For each interconnect open fault location, a pair of stuck-at-0 and stuck-at-1 tests are generated. Values on neighboring lines are specified in order to control their effect on a floating node within the circuit. Experiments were conducted on ISCAS-85 benchmark circuits. Experimental results show that on average 91.8% of the interconnect opens can be detected with this technique while simultaneously detecting on average over 99% of the detectable stuck-at faults. In addition, due to using neighboring line information for test generation of interconnect open faults, many bridging fault may be detected. Therefore, the technique is useful in detecting a variety of potential faults that may occur due to process variation.

Third, in Chapter IV, a technique to further reduce the output data volume for circuit designs with output compactors was investigated. In this work, additional output compaction is achieved by taking the linear sum of two output responses and comparing this result with the known fault free response. The stuck-at fault model was used in this investigation. Experiments were conducted on ISCAS-89 benchmark circuits. Initial experimental results show that using this technique an additional factor of 2X of output compaction can be achieved while maintaining the stuck-at fault coverage.

An augmentation of this method was also investigated. In regards to fault diagnosis, loss of information due to output compaction makes uniquely determining the location of a fault within the circuit more difficult. Thus, in this work an approach to

achieve an additional 2X output compaction while not adversely impacting diagnosis resolution was investigated. Experimental results show that an additional 2X compaction can be achieved with only minimal impact on diagnosis resolution. In addition, independent experiments without fault collapsing show similar results to those reported for experiments with fault collapsing.

5.2 Future Research

In our completed research described in Chapter IV, an output compaction technique was investigated for designs which contain an output compactor. In that work, the proposed technique showed that an additional 2X compaction could be achieved without adversely affecting diagnosis resolution. However no attempt was made to improve the additional compaction factor above 2X or to implement the technique on-chip. In the following sections, these ideas will be addressed as potential future work.

5.2.1 Enhancing Multi-Site Testing for Aging ATEs

Increased throughput in the manufacturing of ICs can be achieved by multi-site testing. Multi-site testing consists of testing many devices in parallel by the same ATE. All devices receive the same input stimuli but the output response must be addressed individually to determine if they produce the correct output response. One solution to this problem was suggested in [75] such that the ATE feeds the fault-free output response to each of the CUT which is equipped with an on-chip comparator. However, if the amount of output response data required to be stored for future designs exceeds the ATE capacity, the ATE may become obsolete. Implementing a secondary on-chip compactor to further reduce the output data volume can extend the life of the ATE.

First, as suggested in [75], an on-chip comparator will be required. In addition, a dedicated pin from the ATE to the CUT will be required to transfer the fault-free responses. Second, the compacted responses must be temporarily stored so it can be

compared with the known fault-free response, which will require additional overhead in the form of flip-flops. Last, control logic will be required to facilitate the compaction of output responses and comparison to with the fault free response.

Implementing additional pins to a design is generally a difficult task, requiring design changes which may not be possible. Therefore it will be important to identify how existing pins can be used to transfer the fault-free response from the ATE. Similarly, data supplied to control logic will need to be carefully investigated in to avoid using additional pins to transfer the control data. In regards to overhead, investigation of how to efficiently compact and store output responses with minimal additional logic will be required.

5.2.2 Increased ATE Assisted Compaction

As ICs become more complex, data volume increases beyond the limitations of current ATE. Due to the extremely high cost of ATE, it is not cost effective to simply purchase new ATE. Data volume reduction techniques have extended the life of older ATEs. Unfortunately on-chip compactor designs do not always achieve the reduction in data volume as expected, exceeding current ATE limitations. Redesigning the on-chip compactor may not be an option due to cost or time limitations. To avoid the costly purchase of new ATE, further reduction of data volume is necessary. Utilizing ATE resources, the work in [72] has achieved an additional 2X reduction in the data volume required to be stored on the ATE. Depending on the limitations of current ATEs this additional 2X reduction may not be sufficient. Thus, an improved technique to further reduce the data volume required to be stored on the ATE is necessary.

The approach in [72] used a simple greedy procedure to select two output responses to be merged together. Only limited information regarding fault detection was used within the procedure for selecting candidates for merging. A 3-detect fault

simulation was done such that responses that were only detected twice were not merged together in order to avoid aliasing.

Using an N-detect fault simulation for a much higher value of N would help guide the procedure in selecting up to k ($k \geq 2$) candidates for merging in order to achieve higher compaction. Fault simulation after merging each of the candidates can further guide the procedure to identify pairs of responses that should not be merged to avoid aliasing and if a fault is no longer detected.

REFERENCES

[1] N. A. Touba, "Survey of Test Vector Compression Techniques", *IEEE Design & Test of Computers*, Vol. 23, No. 4, pp. 294-303, April 2006

[2] Y. Zorian, "Programmable Space Compaction for BIST", *International Symposium on Fault-Tolerant Computing*, pp. 340-349, June 1993

[3] W. C. Carter, "The Ubiquitous Parity Bit", *Proceedings 12th International Symposium On Fault-Tolerant Computing*, pp. 289-296, June 1982

[4] P. H. Bardell, William H. McAnney, J. Savir, *Built-in Test for VLSI: Pseudorandom Techniques*, John Wiley & Sons, New York, 1987

[5] J. P. Hayes, "Check Sum Methods for Test Data Compression", *J. Design Automation Fault-Tolerant Computing*, Vol. 1, pp. 3-7, January 1976

[6] M. Abramovici, M. A. Breuer, A. D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, New York, 1990

[7] J. Savir, "Syndrome-Testable Design of Combinational Circuits", *IEEE Transactions On Computing*, Vol. C-29, No. 6, June 1980

[8] A. K. Susskind, "Testing by Verifying Walsh Coefficients", *IEEE Transactions on Computing*, Vol. C-32, No. 2, February 1983

[9] J. P. Hayes, "Transition Count Testing of Combinational Logic Circuits", *IEEE Transactions on Computing*, Vol. C-25, No. 6, pp. 613-620, June 1976

[10] K. I. Diamantraras, N. K. Jha, "A New Transition Count Method for Testing of Logic Circuits", *IEEE Transactions on Computer-Aided Design*, Vol. 10, No. 3, March 1991

[11] N. Benowitz et al., "An Advanced Fault Isolation System for Digital Logic", *IEEE Transactions on Computing*, Vol. C-24, No. 5, 489-497, May 1975

[12] R. A. Frohwerk, "Signature Analysis: A New Digital Field Service Method", *Hewlett-Packard Journal*, pp. 2-8, May 1977

[13] W. W. Peterson, E. J. Weldon, Jr., *Error-Correcting Codes*, the MIT Press, Cambridge, Massachuesetts, 1972

[14] T. W. Williams et al., "Aliasing Errors in Signature Analysis Registers", *IEEE Design & Test of Computing*, Vol. 4, pp. 39-45, April 1987

[15] K. K. Saluja, M. Karpovsky, "Testing Computer Hardware Through Data Compression in Space and Time", *Proceedings International Test Conference*, pp. 83-88, 1983

[16] S. R. Das et al., "Fault Tolerance in Systems Design in VLSI Using Data Compression Under Constraints of Failure Probabilities", *IEEE Transactions on Instruments and Measure*, Vol. 50, No. 6, December 2001

[17] K. Chakrabarty, J. P. Hayes, "Test Response Compaction Using Multiplexed Parity Trees", *IEEE Transactions on Computer-Aided Design of Circuits and Systems*, Vol., 15, No. 11, November 1996

[18] Y. K. Li, J. P. Robinson, "Space Compression Methods with Output Data Modification", *IEEE Transactions on Computer-Aided Design*, Vol. CAD-6, No. 2, March 1987

[19] W.-B. Jone, "DSC – A Space Compression Method", *IEEE International Symposium Circuits & Systems*, pp. 2756-2759, May 1990

[20] M. G. Karpovsky and P. Nagvajara, "Optimal Robust Compression of Test Responses", *IEEE Transactions on Computing*, Vol 39, No. 1, January 1990

[21] K. P. Parker, E. J. McCluskey, "Probabilistic Treatment of General Combinational Networks", *IEEE Transactions on Computers*, Vol. C-24, No. 6, June 1975

[22] M. Seuring, K. Chakrabarty, "Space Compaction of Test Responses for IP Cores Using Orthogonal Transmission Functions", *IEEE VLSI Test Symposium*, pp. 213-219, May 2000

[23] S. Mitra et al., "X-Tolerant Test Response Compaction", *IEEE Design & Test of Computing*, Vol. 22, No. 6, November 2005

[24] S. Z. Hassan, D. J. Lu, E. J. McClusky, "Parallel Signature Analyzers – Detection Capability and Extensions", *26th IEEE Computer Society International Conference, COMCON*, pp. 440-445, February 1983.

[25] I. Pomeranz, S. Kundu, S. M. Reddy, "On Output Response Compression in the Presence of Unknown Output Values", *Proceedings of Design Automation Conference*, pp. 255-258, June 2002

[26] M. Naruse et al., "On-chip Compression of Output Responses with Unknown Values Using LFSR Reseeding", *Proceedings International Test Conference*, Vol. 1, pp. 1060-1068, Sept. 2003

[27] S. Mitra, S. S. Lumetta, M. Mitzenmacher, "X-tolerant Signature Analysis", *Proceedings of International Test Conference*, pp. 432-441, 2004

[28] V. Chickermane, B. Foutz, B. Keller, "Channel Masking Synthesis for Efficient On-chip Test Compression", *Proceedings of International Test Conference*, pp. 452-461, 2004

[29] W. Rajski, J. Rajski, "Modular Compactor of Test Responses", *Proceedings of VLSI Test Symposium*, pp. 242-251, April 2006

[30] N. A. Touba, "X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR", *Proceedings of International Test Conference*, pp. 1-10, October 2007

[31] J. Rajski et al., "Embedded Deterministic Test for Low Cost Manufacturing Test", *Proceedings International Test Conference*, pp. 301-310, October 2002

[32] H. Tang et al., "On Efficient X-handling Using a Selective Compaction Scheme to Achieve High Test Response Compaction Ratios", *International Conference on VLSI Design*, pp. 59-64, 2005

[33] J. Rajski et al., "X-Press: Two-Stage X-tolerant Compactor with Programmable Selector", *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, Vol. 27, No. 1, pp. 147-159, January 2008

[34] S Mitra, K. S. Kim, "X-compact: An Efficient Response Compaction Technique", *IEEE Transactions on Computer-Aided Design of Integrated Circuits & Systems*, Vol. 23, No. 3, pp. 421-432, March 2004

[35] M. C.-T. Chao et al., "Response Shaper: A Novel Technique to Enhance Unknown Tolerance for Output Response Compaction", *IEEE/ACM International Conference on Computer-Aided Design*, pp. 80-87, November 2005

[36] M Sharma, W.-T. Cheng, "X-filter: Filtering Unknowns from Compacted Test Responses", *Proceedings of International Test Conference*, Paper 42.1, November 2005

[37] J. Rajski et al., "Convolutional Compaction of Test Responses", *Proceedings of International Test Conference*, pp. 745-754, 2003

[38] C. Wang et al., "On Compacting Test Response Data Containing Unknown Values", *Proceedings International Conference on Computer-Aided Design*, pp. 885-862, November 2003

[39] V. K. Agarwal, "Increased Effectiveness of Built-In-Testing by Output Data Modification", *International Symposium On Fault-Tolerant Computing*, pp. 227-234, 1983

[40] S. Wang et al., "X-Block: An Efficient LFSR Reseeding-Based Method to Block Unknowns for Temporal Compactors", *IEEE Transactions on Computers*, Vol. 57, No. 7, pp. 978-989, July 2008

[41] J. Rearick, "Too Much Delay Fault Coverage is a Bad Thing", *Proceedings International Test Conference*, pp. 624-633, 2001

[42] J. Saxena et al., "A Case Study of IR-drop in Structured At-Speed Testing", *Proceedings International Test Conference*, pp. 1098-1104, 2003

[43] P. Girard, "Low Power Testing of VLSI Circuits: Problems and Solutions", *Proceedings International Symposium on Quality Electronic Design*, pp. 173-179, 2000

[44] S. Remersaro et al., "Preferred Fill: A Scalable method to Reduce Capture Power for Scan Based Designs", *Proceedings International Test Conference*, pp. 1-10, 2006

[45] S. Kajihara, K. Ishida, K. Miyase, "Test Vector Modification for Power Reduction During Scan Testing", *Proceedings VLSI Test Symposium*, pp.160-165, 2002

[46] K. Lee, J. Chen, C Huang, "Using a Single Input to Support Multiple Scan Chains", *Proceedings International Conf. Computer-Aided Design*, pp. 74-78, November 1998

[47] B. Koneman, "LFSR-Coded Test Patterns for Scan Designs", *Proceedings European Test Workshop*, pp. 237-242, 1993

[48] I. Hamzaoglu, J.H. Patel, "Reducing Test Application Time for Full Scan Embedded Cores", *Proceedings International Symposium Fault-Tolerant Computing*, pp. 260-267, 1999

[49] S. Samaranayake et al., "A Reconfigurable Shared Scan-in Architecture", *Proceedings VLSI Test Symposium*, pp. 9-14, 2003

[50] H. Tang, S. M. Reddy, I. Pomeranz, "On Reducing Test Data Volume and Test Application Time for Multiple Scan Chain Designs", *Proceedings International Test Conference*, pp. 1079-1088, 2003

[51] A. El-Maleh, A. Al-Suwaiyan, "An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits", *Proceedings VLSI Test Symposium*, pp. 53-59, 2002

[52] J. M. Howard et al., "Reduced Switching Activity Tests for Broadcast Scan Based Designs", *Latin American Test Workshop*, March 2007

[53] W. Maly, P.K.Nag, P.Nigh, "Testing Oriented Analysis of CMOS ICs with Opens", *Digest IEEE Int. Conf. on Computer-Aided Design*, pp. 344-347, November 1988

[54] R. Rodriguez-Montanes, J. Figueras, "Electrical and Topological Characterization of Interconnect Open Defects", *IEEE International Workshop on Current and Defective Based Testing*, 2005

[55] H. Konuk, "Voltage- and Current-Based Fault Simulation for Interconnect Open Defects", *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, Vol. 18, No. 12, pp. 1768-1779, December 1999

[56] M. Renovell, G.M. Cambon, "Electrical Analysis and Modeling of Floating-Gate Fault", *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, Vol. 11, No. 11, pp. 1450-1458, November 1992

[57] H. Konuk, F. J. Ferguson, "Oscillation and Sequential Behavior Caused by Opens in the Routing in Digital CMOS Circuits", *IEEE Transactions on Computer-Aided Design of Integrated Circuits*, Vol. 17, No. 11, pp. 1200-1210, November 1998

[58] H. Konuk, F. J. Ferguson, "An Unexpected Factor in Testing for CMOS Opens: The Die Surface", *Proceedings VLSI Test Symposium*, pp. 422-429, 1996

[59] V. H. Champac, A. Zenteno, "Detectability Conditions for Interconnect Open Defects", Prof. VTS, pp. 305-311, 2000

[60] D. Arumi, R. Rodriquez-Montanes and J. Figueras, "Defective Behaviours of Resistive Opens in Interconnect Lines", *Proceedings of European Test Symposium*, pp. 28-33, 2005

[61] S. M. Reddy et al., "On Testing of Interconnect Open Defects in Combinational Logic Circuits with Stems of Large Fanout", *Proceedings International Test Conference*, pp. 83-89, 2002

[62] R. Gomez, A. Giron, V. Champac, "Test of Interconnection Opens Considering Coupling Signals", *Proceedings International Symposium on Defect and Fault Tolerance in VLSI Sys.*, pp. 247-255, 2005

[63] S. M. Reddy, I. Pomeranz, Chen Liu, "On Tests to Detect Via Opens in Digital CMOS Circuits", *Design Automation Conference*, pp. 840-845, 2008

[64] S. Rafiq et al., "Testing for Floating Gates Defects in CMOS Circuits", *Proceedings Asian Test Symposium*, pp. 228-236, 1998

[65] R. Aitken et al., "Defect-Oriented Test", *Tutorial at International Test Conference*, pp. 62-68, 2000

[66] Xiang, 2008, ISCAS-85 Benchmark Circuit Physical layouts and extracted capacitance values, http://ece.tamu.edu/~xiang/iscas.html, (accessed 2008)

[67] J. M. Howard, S. M. Reddy, I. Pomeranz, "Improving Defect Detection in the Presence of Process Variation", *Design for Reliability and Variability Workshop*, October 2008

[68] P. Wohl, J. A. Waicukauski, T. W. Williams, "Design of Compactors for Signature-Analyzers in Built-In Self-Test," in *Proceedings International Test Conference*, pp. 54-63, 2001

[69] J. Rajski et al., "Finite Memory Test Response Compactors for Embedded Test Applications", IEEE TCAD, pp. 622-634, April 2005

[70] W.-T. Cheng et al., "Signature Based Diagnosis for Logic BIST", *Proceedings International Test Conference*, paper 8.3, 2006

[71] P. Krishnamurthy, LSI Corp., Private Communication, 2007

[72] J. M. Howard, S. M. Reddy, I. Pomeranz, "ATE Assisted Test Response Compaction", *Proceedings International Symposium on VLSI Design, Automation and Test*, pp. 112-115 April 2010

[73] S. S. Lumetta, S. Mitra, "X-Codes: Error control with Unknowable Inputs", *Proceedings International Symposium on Information Theory*, pp. 102, 2003

[74] C. Barnhart et al., "OPMISR: The Foundation for Compressed ATPG Vectors", *Proceedings International Test Conference*, pp. 748-757, October 2001

[75] S. Holst, H.-J. Wunderlich, "A Diagnosis Algorithm for Extreme Space Compaction", *Design, Automation & Test in Europe Conference*, pp.1355-1360, April 2009

[76] W.-T. Chen et al., "Compactor Independent Direct Diagnosis", *Proceedings Asian Test Symposium*, pp. 204-209, November 2004

[77] M. Abramovici, M. A. Breuer, "Fault Diagnosis Based on Effect-Cause Analysis: An Introduction", *Conference on Design Automation*, pp. 69-76, June 1980

[78] V. Boppana, R. Mukherjee, J. Jain, M. Fujita and P. Bollineni, "Multiple Error Diagnosis Based on XLists", *Proceedings Design Automation Conference*, pp. 660-665, 1999

[79] S. Huang, "Speeding Up the Byzantine Fault Diagnosis Using Symbolic Simulation", *Proc VLSI Test Symposium*, pp. 193-198, 2002

[80] S. Jha, S. M. Reddy, I. Pomeranz, "Test Pass May be Equally Useful as Test Fail for Defect Diagnosis", Technical Report, ECE Department, University of Iowa, Iowa City, IA

[81] J. M. Howard, S. M. Reddy, I. Pomeranz, "Fault Aware ATE Assisted Test Response Compaction", *Asian and South Pacific Design Automation Conference*, January 2011