TESTING OF BOARD INTERCONNECTS USING BOUNDARY SCAN ARCHITECTURE

Abu S. M. Hassan

VLSI Design Laboratory
Department of Electrical Engineering
McGill University
Montreal, CANADA

A thesis submitted to the Faculty of Graduate Studies
and Research in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Electrical Engineering)

November 1989

Abstract

The testing of printed circuit board (PCB) interconnects is a complex task that requires enormous amount of resources. With the increasing use of new technologies like surface mounting technology (SMT), testing PCB interconnects using the available techniques, like in-circuit testing and functional testing, is becoming very difficult. To make testing manageable, it must be considered earlier in the design process. This is known as 'design for testability' (DFT). A hierarchical DFT approach known as *boundary scan architecture* has recently become an increasingly attractive solution for PCB interconnect testing problems. This framework provides a scan path for electronic access to the interconnect test points, thus removing the need for accessibility through electro-mechanical contacts known as 'bed of nails'.

In the recent past, several researchers have proposed different schemes for PCB interconnect testing based on the boundary scan architecture.

In this dissertation, a new approach, based on the concept of built-in self-test (BIST), is developed using the boundary scan architecture for PCB interconnect testing. BIST, at the component level, generally consists of incorporating additional circuitry on the chip to generate test patterns and to compact the response of the circuit under test into a reference signature. For the PCB level BIST, the board is considered as the unit under test. A family of BIST schemes are developed for board interconnect testing utilizing the properties of the boundary scan architecture. The BIST approach has removed the dependence on automatic test equipment (ATE) for generation of test vector sets and analysis of output data sets. Techniques are developed for the generation of test vector sets which require very simple test generation hardware. Test vector sets are shown to be independent of the order of the input/output (I/O) scan cells in the boundary scan chain and of the structural complexity of the interconnects under test. Response compaction techniques proposed in the schemes are such that fault detection and diagnosis can be done independent of the topological information about the interconnects. These response compaction techniques can be implemented within each boundary scan cell or outside the boundary scan chain, providing a trade-off in terms of test time and hardware complexity. The various uses of the boundary scan architecture make the proposed schemes more attractive and advantageous than the existing approaches for board interconnect testing.

Moreover, a family of interconnect testing schemes is proposed for a *partial* boundary scan environment. Partial boundary scan environment refers to a board with a mix of boundary scan and non-boundary scan components. Such an environment is more complex compared to a complete boundary scan environment. The proposed schemes are BIST-able despite the inherently complex test environment. However, fault coverage is limited because of the reduced accessibility of the partial boundary scan environment.

Résumé

Le test d'interconnexions de circuits imprimmés (CI) est une tâche complexe qui requiert d'énormes ressources. L'usage de plus en plus fréquent de technologies nouvelles, telles que la technique de montage de surface (TMS), rends difficile le test d'interconnexions de CI par les techniques courantes, comme le contrôle intérieur direct ou le test fonctionnel. Pour rendre le test réalisable, des concepts de test doivent être pris en considération plus tôt dans le processus de conception (Conception Adaptée à la Vérification.CAV). Une approche CAV hiérarchique, à chaîne de balayage périphérique, est récemment devenue de plus en plus intéressante pour solutionner le problème de test d'interconnexions de CI. Ce cadre fournit une chaîne de balayage donnant accès aux points de test des interconnexions, évitant l'usage de connecteurs à fiches multiples.

Récemment, plusieurs chercheurs ont proposé différents procédés de test d'interconnexions dans le cadre d'une chaîne de balayage périphérique.

Dans cette dissertation, une nouvelle approche, axée sur le concept de circuits à vérification intégrée (CVI), est développée en utilisant l'architecture à chaîne de balayage périphérique pour le test d'interconnexions de CI. Le CVI, au niveau composante, consiste en général en l'incorporation de montages additionnels sur la puce pour générer des vecteurs de test et pour comprimmer la réponse du circuit sous test en une signature de référence. Pour le CVI au niveau CI, le circuit imprimmé en entier est considéré comme étant l'unité sous test. Une famille de procédés CVI à été développée pour le test d'interconnexions de circuits imprimmés utilisant une chaîne de balayage périphérique. L'approche CVI a remplacé la dépendance sur les systèmes d'équipement automatique de test (EAT) pour la génération de vecteurs de test et l'analyse de l'ensemble de données de sortie. Des techniques ont été développées pour la génération de vecteurs de test, techniques requérant un matériel de génération de test très simple. Des ensembles de vecteurs de test sont démontrés comme étant indépendants de l'ordre des cellules d'entrée/sortie (E/S) de la chaîne de balayage et de la complexité structurelle des interconnexions sous test. Les techniques de compression de réponses proposées sont telles que la détection et le diagnostic de défauts peuvent être faits indépendamment de l'information

topologique sur les interconnexions. Ces techniques de compression de réponses peuvent être mises en application à l'intérieur de chaque cellule de balayage périphérique ou à l'exterieur de la chaîne, dans un compromis entre le temps de test et la complexité du matériel. Les divers usages de l'architecture à chaîne de balayage périphérique, dans les procédés proposés, les ont rendus attrayants et plus avantageux que les approches existantes de test d'interconnexions de circuits imprimmés.

De plus, une famille de procédés de test d'interconnexions est proposée pour un environnement à chaîne de balayage périphérique partielle. Dans un tel environnement, un mélange de composantes avec et sans chaîne de balayage est utilisé, résultant en une complexité plus élevée que dans un environnement a chaîne de balayage complète, mais demeurant réalisable dans un contexte CVI. L'accessibilité réduite de la chaîne partielle limite toutefois la couverture des défauts.

Acknowledgements

I wish to acknowledge my co-advisor, Professor Vinod K. Agarwal, for his invaluable and instructive guidance throughout the course of my research. I am grateful to him for his insight, knowledge, motivation and enthusiasm.

I also gratefully acknowledge my other co-advisor Professor Janusz Rajski for the guidance, time, and help he offered during the course of this work.

I also acknowledge the Design For Testability group at Bell-Northern Reserach in Ottawa for the sharing of ideas and for helping me with various resources during this work.

I would like to extend my thanks to Mr. Fadi Mamaari for his help in writing the french abstract, and to Mr. Ajai Jain and Mr. Chris Little for their useful comments on the presentation of this material.

Very special thanks go out to my wife and parents, without whose support and encouragement the task would have been much more difficult.

I am grateful to Trang Nguen and Lynn-Marie Holland for their sincere co-operation and help that they offered me in the VLSI Design Laboratory.

Finally, I am grateful also to the Canadian Commonwealth Scholarship Plan of Canada and the Natural Sciences and Engineering Research Council of Canada for providing financial support during the period of this research.

Table of Contents

Page

Abstract ii

Résumé iv

Acknowledgements vi

List of Tables x

List of Figures xii

Claim of Originality xiv

Chapter 1 Introduction 1

1.1 Perspective and Dissertation Outline 1

1.2 Testing of ICs 5

1.2.1 Classical Approaches 5

1.2.2 Design for Testability 6

1.2.2.1 Scan Path Testing 6

1.2.2.1 Built-In Self-Test 6

1.3 Printed Circuit Board Interconnect Testing 7

1.3.1 Classical Approaches 8

1.3.2 Design for Testability 9

1.3.2.1 Boundary Scan Architecture 9

1.3.2.2 Approaches based on Boundary Scan Architecture 21

1.3.2.3 Proposed Approach 21

Chapter 2 Notations and Preliminaries 22

2.1 Structure of the Interconnects 22

2.2 Fault Model 26

2.3 The Boundary Scan Test Environment 29

2.3.1 Loading and Application of Test Vectors 29

2.3.2 PCB Level BIST 32

2.3.3 Extended Control Cell with By-pass Capability 34

2.3.4 Compatibility with the JTAG Design 36

Chapter 3 Testing of Shorts and Stuck-At Faults 37

3.1 Existing Schemes for Shorts and Stuck-Ats Detection 37

3.1.1 Minimal-Size Test Set Scheme for Shorts Detection 37

3.1.2 Minimal-Size Test Set for Shorts and SAs Detection 38

3.2 Existing Schemes for Shorts and SAs Diagnosis 39

3.2.1 One Step Diagnosis Schemes 39

3.2.1.1 True/Complement Test Set Scheme 39

3.2.1.2 Min-Weight Test Set Scheme 39

3.2.1.3 Max-Independence Test Set Scheme 40

3.2.2 Adaptive Diagnosis Scheme 41

3.2.2.1 One-Test Adaptive Scheme 41

3.2.2.2 W-Test Adaptive Scheme 42

3.2.2.3 Optimal C-Test Adaptive Scheme 43

3.3 Limitations of the External Schemes 43

3.4 Summary of Existing Schemes 45

3.5 Proposed Schemes 49

3.5.1 Detection of Shorts and SAs 49

3.5.1.1 Order Independent Vector Set Scheme 49

3.5.1.2 Walking Sequence 52

3.5.2 Diagnosis of Shorts and SAs 67

3.5.2.1 Order Independent Vector Set Scheme 67

3.5.2.2 Walking Sequence Scheme 72

3.6 Summary of Proposed Detection and Diagnosis Schemes 75

3.7 Comparison of External Schemes and BIST Schemes 77

Chapter 4 Testing of Open Faults 79

4.1 Detection of Inet Open Faults 79

4.1.1 LI/CLI Vector Scheme 80

4.1.2 Order Independent Vector Set Scheme 82

4.1.3 Walking Sequence Scheme 82

4.2 Diagnosis of Inet Faults 83

4.2.1 LI/CLI Vector Scheme 83

4.2.1.1 Order Independent Vector Set Scheme 84

4.2.1.3 Walking Sequence Scheme 84

4.3 Diagnosis of Inet Conductor Open Fault 84

4.4 Summary of Proposed Schemes 88

Chapter 5. Testing of Glue Logic Interconnects 90

5.1 Preliminaries 90

5.2 Fault Model 95

5.3 GLI Test Problems 96

5.3.1 Controllability and Observability of Interconnects 96

5.3.2 Increasing Observability of GLI_{IN} 97

5.3.3 Application of Test Vectors on $GLI_{OUT}s$ 98

5.4 Fault Detection and Diagnosis 99

5.4.1 Walking Sequence based Scheme 99

5.4.2 GLFI Scheme 108

5.5 Application of Test Schemes 111

5.5.1 Test Vector Sets for Example IC 111

5.5.2 Test Vector Set for Example PCB 116

5.6 Summary of Proposed Schemes 117

6. Conclusion 119

References 124

List of Tables

1.1 Use of controller states for interconnect testing. 15

1.2 Reserved instructions. 16

1.3 Input and output mode controls for boundary scan cells. 17

2.1 Input vectors for Inet testing. 30

3.1 Minimal-size vector set for shorts detection. 38

3.2 True/Complement vector set for shorts and SAs diagnosis. 40

3.3 Min-weight test set. 41

3.4 Max-independence test set. 42

3.5 Vector set for one-step adaptive diagnosis. 43

3.6 Comparison of test times. 50

3.7 Walking one sequence. 52

3.8 Walking zero sequence. 53

3.9 Walking one sequence. 62

3.10 Compacted responses due to walking sequences. 64

3.11 $2log_2N$ vectors for shorts and SAs diagnosis. 68

3.12 Possible combinations of input bit-pairs. 68

4.1 Input vectors. 87

4 2 Output responses for diagnosis. 88

5.1 Partial walking one sequence. 105

5.2 Expected outputs due to partial walking one sequence. 105

5.3 Compacted responses due to partial walking one sequence. 105

5.4 Expected outputs due to partial walking zero sequence. 106

5.5 Compacted responses due to partial walking one and zero sequences.. 107

5.6 2.2^i Input vectors for GLFI scheme. 110

5.7 Faulty output bit-pairs. 110

5.8 Diagnostics table. 110

5.9 Walking one sequence for demultiplexer. 112

5.10 Pseudo walking one sequence for demultiplexer. 113

5.11 Pseudo walking one sequence input set for demultiplexer. 113

5.12 Input vector for detection. 114

5.13 Expected output for detection. 114

5.14 $2.logN$ vectors for demultiplexer. 115

5.15 2.2^i vectors for demultiplexer. 115

List of Figures

1.1 PCB with boundary scan. 3

1.2 Simple input pin scan cell. 10

1.3 Test access port. 12

1.4 Controller main state diagram. 13

1.5 Scan state diagram. 14

1.6 Input pin scan cell with output holding register. 18

1.7 Output pin scan cell. 19

1.8 Scan cells associated with tri-state output pin. 19

1.9 Scan cells associated with bidirectional pin. 20

2.1 Inet under test. 23

2.2 PCB level interconnects 24

2.3 Different Inet structures. 25

2.4 Input vector loading through B-Scan chain. 30

2.5 Application of a test vector. 32

2.6 Short between two Inets. 33

2.7 Architecture for PCB level BIST. 33

2.8 Block diagram of control cell with by-pass capability. 35

2.9 Control cell design with by-pass capability. 35

3.1 Hardware to generate $\lceil log_2(N+2) \rceil$ vectors. 51

3.2 Hardware to generate walking sequence. 53

3.3 Walking sequence loading. 55

3.4 Inets under test. 61

3.5 Extended output scan cell for walking sequence response compaction. 62

3.6 Block diagram of extended output scan ce!l for walking sequence response compaction. 63

3.7 XOR tree. 63

3.8 Extended output scan cell for local response compaction. 69

3.9 Block diagram of extended output scan cell for local response compaction. 69

3.10 Arrangements for external diagnosis. 71

4.1 Inet conductor open fault diagnosis. 87

5.1 Interconnects connecting IC components. 91

5.2 !nterconnects under test. 92

5.3 GLI structures. 93

5.4 Special GLI Structures. 94

5.5 Special GLI structures redefined. 95

5.6 Input scan cell with response compaction capability. 98

5.7 Extended input scan cell for partial walking sequence. 103

5.8 Demultiplexer under test. 112

Claim of Originality

The author claims originality for the following contributions of the dissertation :

- In Chapters 3 and 4, BIST schemes are developed using the boundary scan framework for testing board interconnect faults. Unlike the boundary scan path based existing schemes, the proposed schemes have simplified the process of test vector generation and loading, response compaction, and fault detection and diagnosis.
- BIST schemes use techniques to generate test vector sets independent of the complexity of the interconnect structures and the order of the I/O scan cells in the boundary scan chain. The processes of detection and diagnosis are also independent of the topology of wire routing. These novel techniques make the proposed BIST schemes 'circuit' independent.
- Various uses of the boundary scan architecture, for interconnect testing, are shown in the proposed BIST schemes. In walking sequence based schemes, the boundary scan chain is used for internal generation of the test vector sets. Boundary scan cells are used for local response compaction techniques.
- BIST schemes are developed with trade-off between the two following constraints : the test time and the hardware extension of the boundary scan cells. Schemes are proposed with no extension of the existing boundary scan architecture which require more test time. Schemes with low test time are developed with various extensions of the boundary scan cells.
- In Chapter 5, various schemes proposed in Chapters 3 and 4 are extended and modified for a partial boundary scan environment. These BIST schemes test faults in 'glue logic' interconnects as well as faults in interconnects of boundary scan components in an environment of increased structural complexity and reduced accessibility compared to the complete boundary scan environment.

Chapter 1 Introduction

1.1 Perspective and Dissertation Outline

In recent years, structured *design for testability* (DFT) at the *printed circuit board* (PCB) level has become an activity of major interest. This is a natural evolution following a wide acceptance of the structured DFT (i.e. scan) at the integrated circuit (IC) level and the realization that the costs associated with implementing scan cannot be justified unless it can be used to simplify the testing efforts at the PCB and higher levels as well. This, combined with the emergence of the very high density packaging technology at the PCB level, in particular that of *surface mount technology* (SMT), made it essential to develop the concept of DFT at the PCB level and further hierarchical levels of system integration [JTAG 88].

Since the inception of ICs, over two decades ago, the IC fabrication technologies have not ceased improving. Among other aspects, these technology improvements have been synonymous with continuously increasing scales of integration. Hence, in only two decades, the levels of integration have been qualified from small (SSI) to medium (MSI), large (LSI), very large (VLSI), and ultra large (ULSI).

The technology of PCB design and manufacturing has been improving equally in pace [Bateson 85], [Bennetts 82], [Pardee 88] with the IC development and manufacturing technology. More than 30 years after they appeared, these PCBs are still the most important means of connecting components into electronic systems. They have evolved from simple systems of etched copper strips connecting 30-50-component terminals on one side of a piece of phenolic board, to multiple layers of copper on polymer substrates connecting upward of 10,000 terminals in a three-dimensional pattern. On early boards, the copper routes were laid out by hand and conductor paths were 100 mils (1 mil = 0.001 inch or 0.025 millimeter) wide; now the entire process is computer-controlled and the conductors are as narrow as 3 mils.

Until a few years ago, almost all the components were placed on a PCB with their leads passing through holes in the PCB and soldered to copper 'lands' - small terminal spots on the other side of the PCB. The leads needed holes 40 to 50 mils in diameter that penetrated every layer of the multilayer PCB, while lands could add a further 1 5 mils for the diameter. With ICs in dual in-line packages (DIPs) mounted this way, little PCB area was left for printed wiring. SMT has changed all that. Soldering component terminals to pads on the PCB's surface reduces the need for through-holes and leaves more space on every layer for interconnects. Moreover, terminals can lie on all four edges of a square package, instead of two available on a DIP. The much smaller IC package that results allow room for many more components, hence many more functions can be crammed on to the PCB. However, with SMT, the density of component terminals per square inch has grown from approximately 45 chips in 16 pin DIP packages to more than 250 chips in chip-carriers with terminals only 25 mils apart [Pardee 88]. This results in a tremendous increase in the density and the number of wiring interconnects on the PCB.

Of course, today's possibility of fabricating extremely complex PCBs is very attractive for several reasons. Unfortunately, as in any other engineering endeavor, trade-offs arise in the overall enterprise. Besides the ever increasing difficulty for designers to correctly design (layout) the intended PCB, the problem of testing the manufactured PCB against manufacturing defects is increasing with the growing complexity of the PCB fabricated [JTAG 88]. Hence, although many problems associated with testing appeared early in the history of PCBs, the recent soaring complexity of PCBs is forcing the experts to re-examine the testing problem from a new perspective: namely, recognizing that the testing problem plays a key role in the economical success of a product.

The recognition of testing's crucial role at the component level has spawned a number of research efforts whose fruits have been various. Perhaps, the most important is the proposition of new design techniques collectively known as design for testability (DFT) [Williams 82]. Recently, this concept of DFT has been extended to the PCB and further hierarchical levels of system design to solve the testing problem in a structured manner. In particular, the boundary scan architecture, as detailed in Boundary Scan Architecture Standard Proposal, Version 2.0 [JTAG 88] produced by Joint Test Action Group (JTAG) and later submitted as the IEEE Standard Proposal P1149.1, provides the basis for a hierarchical DFT approach in order to solve the PCB level test problems.

The boundary scan concept allows one to access and control all the primary input and output pins of each component on the PCB from outside. This is done by connecting all the primary inputs and outputs of each IC into a shift register that has a boundary scan input and a boundary scan output. The shift registers on all the ICs of a PCB can be connected together to form a longer shift register with a single scan in edge and a single scan out edge, as shown in Figure 1.1. Thus, in effect, the boundary scan concept provides a sort of electronic in-circuit testing facility.

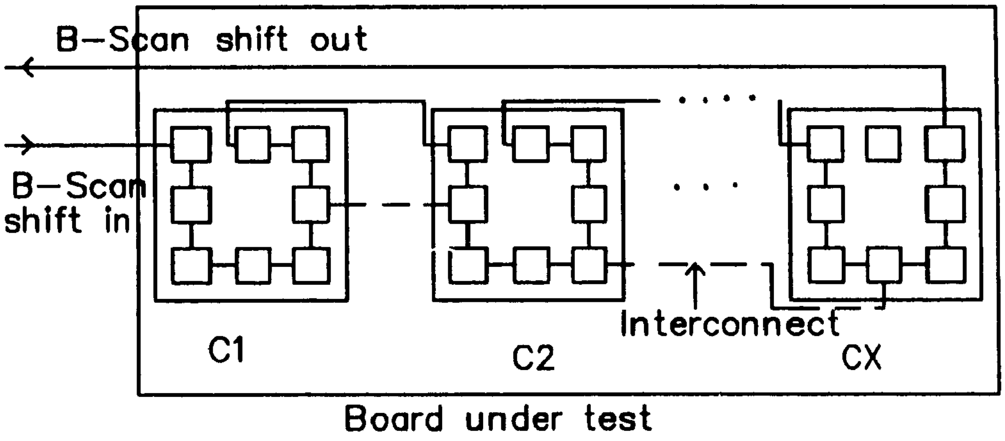


Figure 1.1 PCB with boundary scan.

Using this concept at the PCB level, it should be possible to confirm that each IC performs its required function, that the ICs are interconnected in the correct manner, and that the ICs interact correctly so that the complete PCB performs its intended function. The broad motivation for this dissertation is that of using this concept of boundary scan to verify that the ICs are interconnected in the correct manner.

Several researchers have addressed the PCB level interconnect testing problem using the boundary scan architecture as the framework. This dissertation addresses the problem with focus on a technique known as *built-in self-test* (BIST) [Design & Test 85]. Component level BIST is becoming increasingly popular since it is believed to have the potential of greatly simplifying the test procedure, and therefore decrease the overall cost of testing a system. The work in this dissertation develops BIST techniques for PCB level interconnect testing based on the boundary scan architecture. BIST schemes developed in this dissertation use easy to generate test vector sets and include simple procedures for fault detection and diagnosis. The boundary scan architecture has been used not only as a means for electronic access to the test points, but also for various steps like test vector generation and output response compaction in the test procedure. These various uses of the boundary scan architecture have made the proposed BIST schemes very time-efficient. At the same time, the proposed uses of the boundary scan architecture have made the framework more attractive and justifiable to the designer. The major contribution of this dissertation is the family of structured BIST schemes based on the boundary scan architecture for PCB level interconnect testing.

Briefly, the dissertation is organized as follows. The remainder of this chapter further elaborates on the problems of component and PCB level testing. It is shown that the issues related to testing are similar at these two levels and, consequently, similar approaches are taken to solve the problems. This chapter also describes the essential features of the boundary scan architecture. In Chapter 2, various basic notions related to interconnects and interconnect faults are described. The test environment for the proposed BIST schemes is also introduced in this chapter. Chapter 3 begins with an overview of the existing interconnect test schemes and proposes a number of detection and diagnosis schemes for the testing of shorts and stuck-at faults in the interconnects. Test schemes for open faults are described in Chapter 4. PCB level BIST implementation is emphasized in the schemes proposed in Chapters 3 and 4. Schemes are proposed with various extents of BIST (for example, response compaction outside the boundary scan chain, response compaction within the boundary scan chain, etc.) to show the implementation trade-offs. Chapter 5 describes the problems of interconnect testing in a partial boundary scan environment and proposes a number of test schemes for this more complex test environment. Concluding remarks and research directions are given in Chapter 6.

1.2 Testing of ICs

1.2.1 Classical Approaches

Testing of digital systems generally consists of three major steps : the first being the generation of a set of input patterns, the second being the application of these patterns to the circuit under consideration, and the third being the analysis of the collected output data with an expected response to finally declare if the circuit is good or faulty. The above test procedure may be applied at various fabrication levels of a product, i.e., chip, board, system levels [Mann 80] [Myers 83]. Generally, it is believed that the cost of testing increases by an order of magnitude from one level to the next, and that the overall cost is minimal when the defects are detected as early as possible [Williams 82]. These observations have resulted in a major portion of the recent research in testing to focus on the chip or board levels.

Given a circuit under test (CUT), a fault in the CUT is said to be detected when a particular input pattern applied to the CUT produces an incorrect logic response on one or more of its outputs. Such incorrect responses are called errors, and such input patterns constitute a test for that fault. A set of input patterns that detects a set of faults is called a test set, and a set of test patterns that detects all the faults is said to constitute a complete test set. If a set is not complete, then the most commonly used measure of the quality of the test set is the fault coverage measure which is simply the ratio (usually expressed as a percentage) of the detected faults to the total number of modeled faults in the CUT.

Several test pattern generation algorithms have been developed in the past to generate test sets, e.g., [Roth 67], [Goel 81], [Fujiwara 83]. Although they are generally highly desirable, complete test sets are also generally prohibitively expensive to generate, even with the best available test pattern generation algorithms. One process often used in combination with test pattern generation algorithms is called fault simulation [Chang 70], [Armstrong 72], [Levendel 81], [Bose 82], [Maamari 83] which consists of determining all the faults that a particular set of patterns may detect. Fault simulation is combined with test pattern generation since its computational complexity is less than that of test pattern generation. Nevertheless, in spite of constant advances in the algorithms for test pattern generation and fault simulation, in the case of large circuits these processes may simply be too expensive to be feasible. When the situation is such, alternative methods for classical test pattern generation and fault simulation must be used. Several of these have been proposed recently.

1.2.2 Design for Testability

1.2.2.1 Scan Path Testing

The limitations of the classical approaches have forced the IC manufacturing and design communities to propose new approaches for testing complex VLSI circuits. Such approaches are commonly known as design for testability (DFT) techniques [Williams 82]. Perhaps the most well known example of these techniques is one that arose because of the difficulty in testing sequential circuits. The idea is the notion of scan design [Eichelberger 78], [Stewart 77], [Funatsu 75], [Ando 80] which essentially reduces the problem of testing a sequential circuit to one of testing a combinational circuit. In scan design, the circuit's latches are extended to have two modes of operation. In normal mode, the latches perform as ordinary latches. In test mode, the normal feedback paths of the sequential circuit are broken and the latches form a shift register. With the shift register configuration, a test vector can be scanned in the register from one primary input of the circuit. Then, this vector may be applied to the circuit by returning the circuit to its normal mode for one clock cycle. Once the response of the test vector is latched, the circuit may be returned to test mode to scan out this response through a primary output of the circuit. The response then may be compared to that expected.

The most obvious advantage of scan design is the reduced complexity of generating test sets for sequential circuits. This is accomplished by treating the latter as a combinational circuit when it is being tested. The principal disadvantages are the hardware overhead, arising from extensions in the latches and the extra control circuitry, extra primary inputs and outputs, and the time required for scanning tests in and out of a circuit.

1.2.2.2 Built-In Self-Test

Another DFT approach which can be well coupled with scan design is the idea of built-in self-test (BIST) [Design & Test 85], [Bardell 87]. BIST is usually understood as a scheme where all circuitry associated with testing a chip is part of the chip itself. Thus, an input pin is required to select between normal and self-test modes. In self-test mode, test patterns are generated on the chip, and the responses are observed and analyzed on the chip. Finally, a simple form of 'go'/'no-go' is delivered at one of the chip's output pins. For sequential circuits, scan design is usually assumed. Therefore, in self-test mode, the CUT is combinational, and the latches form part of the test circuitry.

BIST requires an on-chip source of test patterns. The most popular scheme for generating test patterns is to configure certain latches into a linear feedback shift register (LFSR) [Bardell 87]. BIST usually implies a relatively large number of test patterns, and hence considerable response data, especially if the circuit has several outputs. Thus, in BIST, some form of data compaction is used to reduce this output data into a signature of only a few bits. At the end of the test sequence, the signature is compared to a stored reference signature.

Several compaction schemes have been proposed for BIST, e.g [Carter 82], [Frohwerk 77], [Hassan 84], [Hayes 76], [Hurst 85], [Miller 84], [Muzio 83], [Savir 80], [Saxena 86], [Susskind 81], [Zorian 86]. The reader is referred to these references for details about the schemes. One of the better known schemes is signature analysis [Frohwerk 77]. In signature analysis, an LFSR can be used to compact the output responses of a single-output CUT. The mathematical interpretation of compaction by an LFSR is rather straightforward. The process can be interpreted as polynomial division over a Galois field [Goloumb 82], [Peterson 72]. The LFSR compactor can also be extended to the multiple-output circuit, using a multiple-input shift register (MISR). Compaction by an LFSR is attractive for its simplicity and low overhead, compared to other BIST schemes. In some cases, the compaction circuit can be combined with the test pattern source [Koenemann 79].

There are several advantages of BIST. Since pseudorandom patterns are applied, no costly test generation is required. Also, a BIST can be applied at close to the normal operating speed of the CUT. The extra test circuitry required is small. Finally, BIST introduces the possibility of easy and inexpensive field testing.

The most obvious disadvantage of BIST is the area overhead, which is highly design dependent. Also, if exhaustive testing is performed, circuits with a large number of inputs may require an exceedingly large test set and associated test time. However, the most serious difficulty with BIST is the loss of information incurred by the compaction stage. This phenomenon is described as masking or aliasing [McCluskey 85].

1.3 Printed Circuit Board Interconnect Testing

Various approaches in IC testing are reviewed in the previous section. It is shown that scan path testing and BIST techniques have solved the problems of expensive test generation, fault simulation and response analysis in IC testing. The problems in PCB level interconnect testing are discussed in the following chapters. Structured techniques like scan path testing and BIST, similar to the IC level DFT techniques, are described for PCB level interconnect testing. The basic concepts of test set, fault coverage, response compaction etc. are the same for these two levels of testing. Therefore, those concepts, as defined above, are equally applicable in the following discussion.

1.3.1 Classical Approaches

Presently, there are two techniques used for PCB level interconnect testing : in-circuit testing and functional testing [Bateson 85], [Bennetts 82].

In the in-circuit test technique, the IC components of a PCB are accessed by a 'bed of nails' - probes on the automatic test equipment (ATE) that directly make contact with the device I/O pins from pads on the surface of the circuit board. This makes it possible to test each IC and the interconnects between ICs. In-circuit test detects faults caused by the manufacturing process and detects the majority of component faults. In this test technique, tests are applied directly to individual components by over-driving their connections from other devices in the product. The process requires extensive access to the circuit, since every connection must be driven and monitored directly in order to apply the tests to the individual components. This access is provided through the 'bed of nails' mentioned above. Various difficulties with this approach include degraded reliability due to over-driving connections between ICs, physical limitation of through-the-hole accessibility, difficulty of reproducing tests and expenses involved with the 'bed of nails' testing fixtures. Moreover, PCB level interconnect testing cost has increased tremendously due to the increasing use of surface mount technology and high speed and high pin-count testers.

Functional tests are applied through a PCB's normal terminations - edge connectors, for example. The objective is to test the PCB as a single entity. During testing, the functional board tester duplicates final product environment by stimulating the board inputs and measuring the outputs, thereby testing board performance. Thus, the interconnect faults get tested during the course of the process. However, test generation, fault isolation and test application costs are excessive to achieve acceptable fault coverage. For some products it may be impossible to either verify or achieve the desired fault coverage [JTAG 88].

1.3.2 Design for Testability

The main problem in PCB level interconnect testing is one of accessibility. The end points (also known as test points) of every interconnect need to be accessible for testing purposes. An interconnect should be both controllable and observable. This means that controlled input(s) should be applied from the input end(s) and the response(s) should be observed from the output end(s) of the interconnect. Physical access to the control and observation points are obtained through the 'bed of nails' in in-circuit testing, as discussed above. Gaining access to the interconnect test points is difficult because of the increase in the number of interconnects, the decrease in the spacing of interconnects, the multi-layer wiring of interconnects and the increasing use of SMT boards. Therefore, a structured design for testability approach is necessary at the PCB level to solve this problem The boundary scan architecture, as described in the following, is such a framework. Latter sections discuss how this DFT framework can be efficiently used for the testing of PCB level interconnects.

1.3.2.1 Boundary Scan Architecture

The boundary scan architecture is proposed by Joint Test Action Group (JTAG). This section describes the basic elements of this architecture relevant to board level interconnect testing. The description is extracted from Boundary Scan Architecture Standard Proposal, Version 2.0 [JTAG 88].

The boundary scan technique involves the inclusion of a shift register latch (contained in a boundary scan cell) adjacent to each component pin so that signals at component boundaries can be controlled and observed using scan testing principles. Figure 1.2 illustrates one possible implementation for a boundary scan cell which could be used for an input connection to an IC. Dependent on the control signal applied to the multiplexer, data can either be loaded into the scan register from the input port, or driven from the register through the output port of the cell into the core of the IC. An analogous design could be used for boundary scan cells associated with output or bi-directional connections to the component.

The boundary scan cells for the pins of a component are interconnected to form a shift register chain around the border of the design, and this path is provided with serial input and output connections and appropriate clock and control signals. Within a product assembled from several ICs the boundary scan registers for the individual IC's can be connected in series to form a single path through the complete design. Alternately, a board design could contain

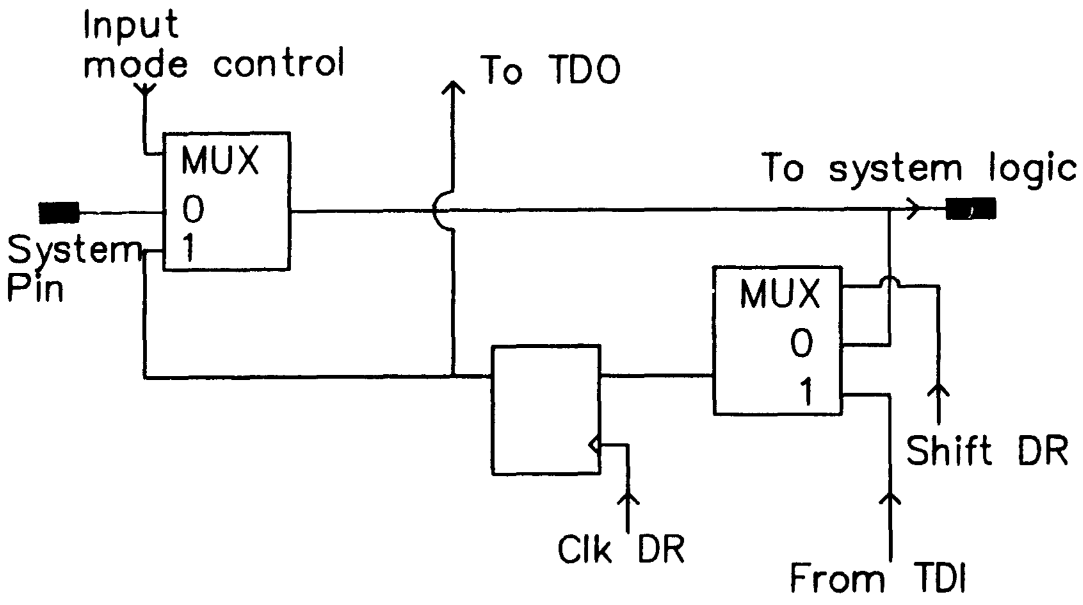


Figure 1.2 Simple input pin scan cell

several independent boundary scan paths.

The boundary scan path through the complete design can be used in two ways :

a) To allow the interconnections between the various components to be tested. Test data can be shifted into all the boundary scan cells associated with component output pins and loaded in parallel through the component interconnections into those cells associated with input pins. This is the *External* boundary scan test.

b) To allow the components on the board to be tested. This is the *Internal* boundary scan test.

Note also that by parallel loading the cells at both the inputs and outputs of a component and shifting out the results, the boundary scan register provides a means of 'sampling' the data flowing through a component without interfering with its behavior. This *Sample* boundary scan test is valuable for design debugging and fault diagnosis.

The following elements are the required components of the design : a test access port (TAP) comprising of four pins, a TAP controller, and a scannable instruction register. Each circuit must include at least two scannable test data registers : a by-pass register and a boundary scan register Optionally, other data registers can be included.

The instruction and test data registers are parallel shift register based paths, having a common serial data input and a common serial data output. both of which are connected to the TAP. The selection between these parallel shift register based paths is made under control of the TAP controller. when necessary the contents of the instruction register is taken into account.

Key features of the design are :

(i) Test Access Port (TAP)

The TAP consists of the three input connections and one output connection for the test logic shown in Figure 1.3. The TAP is a general purpose port which can provide access to many test support functions built into the circuit design. The signals in the TAP form a bus which interconnects a slave device with other slave devices and with a bus master. The bus master may be either an ATE system or a component which interfaces the TAP bus to a higher level test bus.

The signals in the TAP for a slave component are defined as follows :

TCK : *Test Clock.* This is nominally a free running clock signal with a 50% duty cycle. The changes on TAP input signals (TMS and TDI) are clocked into the TAP controller, instruction register or selected data register on the rising edge of TCK.

TDI : *Test Data Input.* TDI is clocked into the selected register on a rising edge of TCK.

TDO : *Test Data Output.* The contents of the selected register are shifted out of TDO on the falling edge of TCK.

TMS : *Test Mode Select.* This serial control input is clocked into the TAP controller on the rising edge of TCK. The signal applied at the TMS pin is decoded to distinguish the principle operations of the test support circuitry.

(ii) TAP controller

The TAP controller is a synchronous finite state machine whereby through the various operations of the JTAG circuitry occurs under control of the TMS signal. The states in the main controller state diagram are as shown in Figure 1.4 In this figure, the signal values shown adjacent to the state transition arcs represent the value of TMS at the rising edge of TCK.

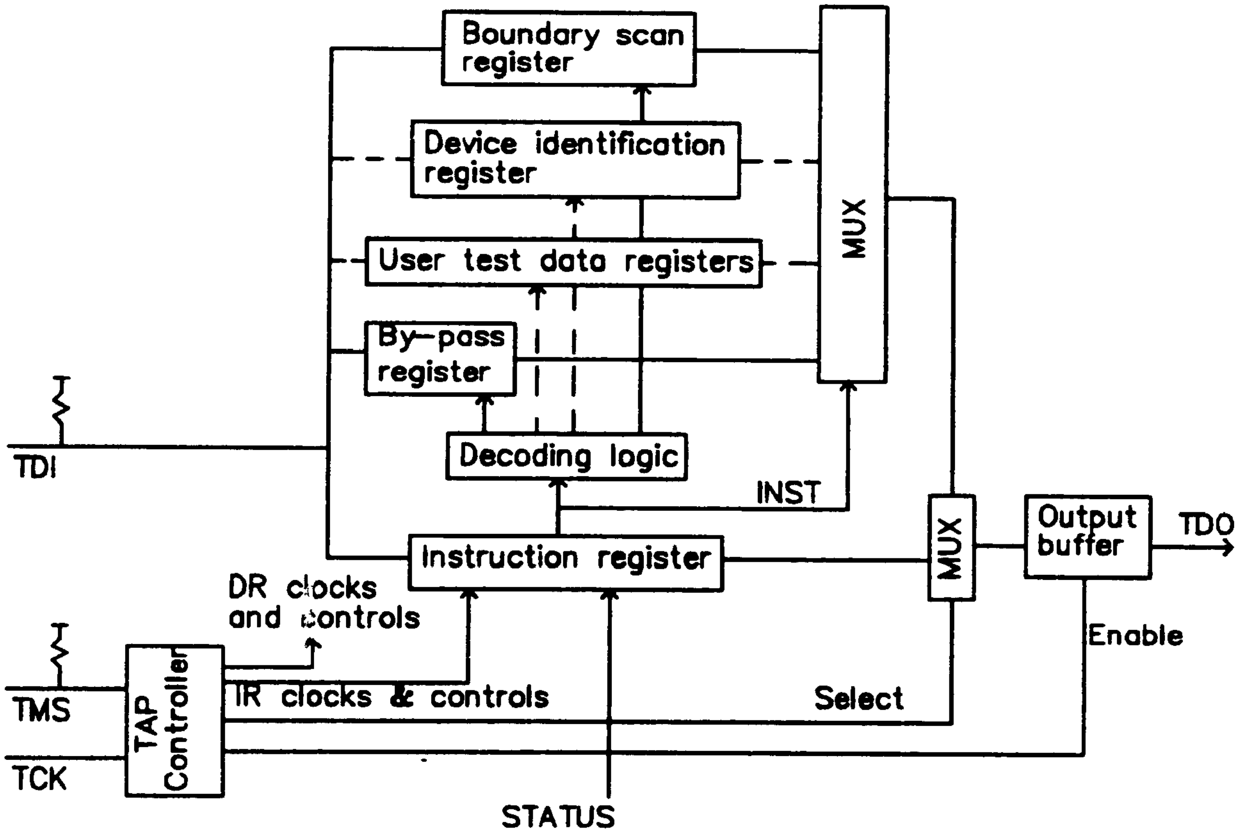


Figure 1.3 Test access port.

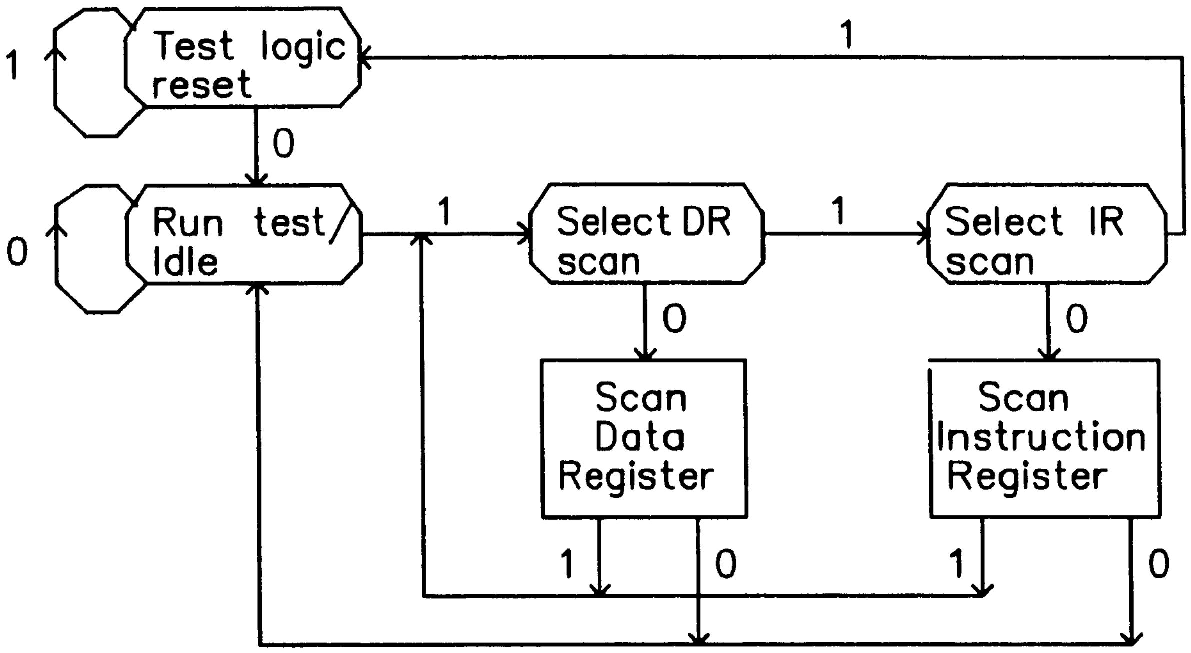


Figure 1.4 Controller main state diagram.

- **Main State Diagram :**

Following are the various states in the main controller state diagram :

a) *Test-Logic-Reset.* This state disables the test logic so that normal operation of the system logic in the component can continue unhindered. The controller enters Test-Logic-Reset whenever the TMS input is held high for at least five rising edges of TCK. The controller remains in this state while TMS is held high.

b) *Run-Test/Idle.* A controller state between scan operations where an internal test previously selected by setting the instruction register may be executed. Once entered, the controller will remain in this state, as long as TMS is held low.

c) *Select-DR-Scan.* If TMS is held low when the controller is in this state, then a scan sequence for the selected test data register is initiated.

d) *Select-IR-Scan.* Similar to Select-DR-Scan.

e) *Scan-Data-Register.* The scanning process defined in the controller scan state diagram (Figure 1.5) is applied to the test data register identified by the current instruction.

f) *Scan-Instruction-Register.* Similar to Scan-Data-Register.

- **Scan State Diagram :**

The scan state diagram for the TAP controller is shown in Figure 1.5. The scanning process contains six states which are defined as follows :

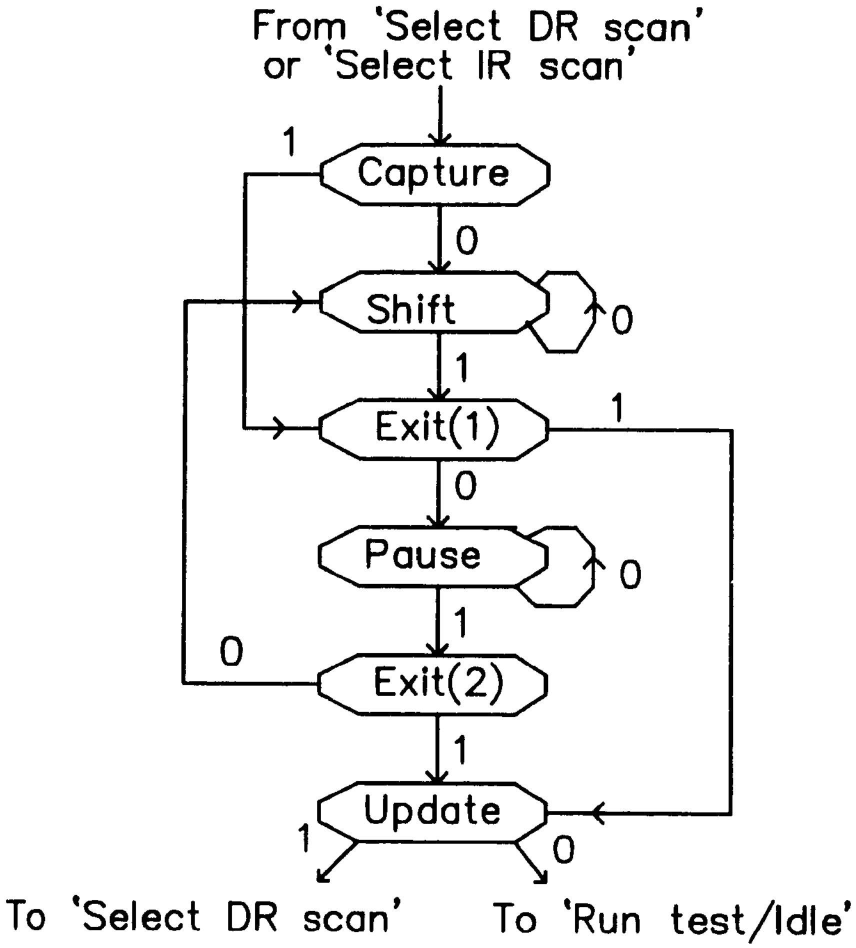


Figure 1.5 Scan state diagram.

a) *Capture.* Data is parallel loaded into the selected test data (instruction) register.

b) *Shift.* The selected test data (instruction) register shifts data one stage towards its serial output on each rising edge of TCK.

c) *Exit(1).* A high signal on the TMS line while in this state causes termination of the scanning process; a low signal causes entry into the *Pause* state.

d) *Pause.* Shifting of the selected test data (instruction) register is temporarily halted.

e) *Exit(2).* A high signal on the TMS line while in this state causes termination of the scanning process; a low causes a return to the *Shift* state.

f) *Update.* Data is transferred from each shift register stage into the corresponding parallel output latch (if the selected register includes a parallel output latch).

The TAP controller must generate signals to control the operation of the test data and instruction registers and the associated multiplexers. The TAP controller states include three basic actions required for testing : stimulus application (Update-DR), execution (Run-Test/Idle), and response capture (Capture-DR). However, all these actions are not necessary for every type of test. The actions required for boundary scan interconnect testing (or, external testing) are shown in Table 1.1.

Update-DR	Run-Test/Idle	Capture-DR
Yes	No	Yes

Table 1.1 Use of controller states for interconnect testing.

(iii) Instruction Register

The instruction register allows an instruction to be shifted into the design. The instruction is used to select the test to be performed and/or the test data register to be accessed. The instruction register must include at least three shift register based cells. Each must contain a shift register stage and a parallel output register/latch. This register/latch is included to ensure that the instruction only changes at the end of the instruction register scanning sequence. Every instruction must select either one of the test data registers or the by-pass register.

At a minimum, instruction must be provided to achieve the following :

- Selection of the by-pass register,
- Selection of the boundary scan register for external (interconnect) test,
- Selection of boundary scan register for sample test,

- Selection of boundary scan register for internal test,
- Selection of device identification register (if included).

Three instructions are reserved as defined in Table 1.2. The reserved instructions are defined to allow easy selection of three key operations without requiring complex instruction data patterns to be shifted into the design. Instructions which are not reserved can be assigned for other operations.

Instruction MSB...LSB	Register selected	Setting of boundary scan mode controls
111...1	By-pass	Sample
000...0	Boundary scan	External
10X...X	Device identification/by-pass	Sample

Table 1.2 Reserved instructions.

(iv) By-pass Register

The by-pass register provides a short circuit route for test data in the data register scanning cycle which can be used when access to other test data registers in the IC is not required.

(v) Device Identification Register

This register allows the manufacturer, part number, and version of a component to be determined through the JTAG TAP. Detail description can be found in [JTAG 88].

(vi) Boundary Scan Register

The boundary scan register is a simple single shift register based path containing cells connected to all system inputs and outputs of the circuit and the system logic, including clock inputs to the system logic of the design, and to output enable or direction control signals associated with system pins.

Boundary scan cells must allow execution of at least three types of tests as determined by the state of mode controls supplied to each cell. These mode controls are generated by decoding the instruction register parallel output.

The three required test types are :

a) *External (Interconnect) Test.* The boundary scan register used to test circuitry external to the IC package - typically the board interconnects. Cells at the output pins are used to apply test stimulus, while those at the input pins capture test results.

b) *Sample Test.* The boundary scan mode controls are set to allow normal operation of the system logic. Signals entering and leaving the circuit can be sampled without affecting circuit operation.

c) *Internal Test.* The boundary scan mode controls allow the test of internal functional circuitry.

The boundary scan mode controls are used to set the various cells for the three test types. The states of these mode controls are coded as shown in Table 1.3.

Boundary scan test	Input mode control	Output mode control
External	0	1
Sample	0	0
Internal	1	0
undefined	1	1

Table 1.3 Input and output mode controls for boundary scan cells.

Different types of boundary scan cells are described in the following :

- **Input Pin Scan Cell :**

Each system input pin must have a boundary scan cell. Figures 1.2 and 1.6 show two versions of the input pin scan cell design. The design in Figure 1.6 includes an output register/latch (not included in the design of Figure 1.2) which is updated from the shift register stage in the Update-DR controller state. This register is included to prevent changes at the output of the shift register stage, during shifting, being applied to the system logic of the IC (which could cause unwanted operation). If the mode controls are set for external test, the signal applied to each system input pin must be loaded into the corresponding shift register during the Capture-DR controller state.

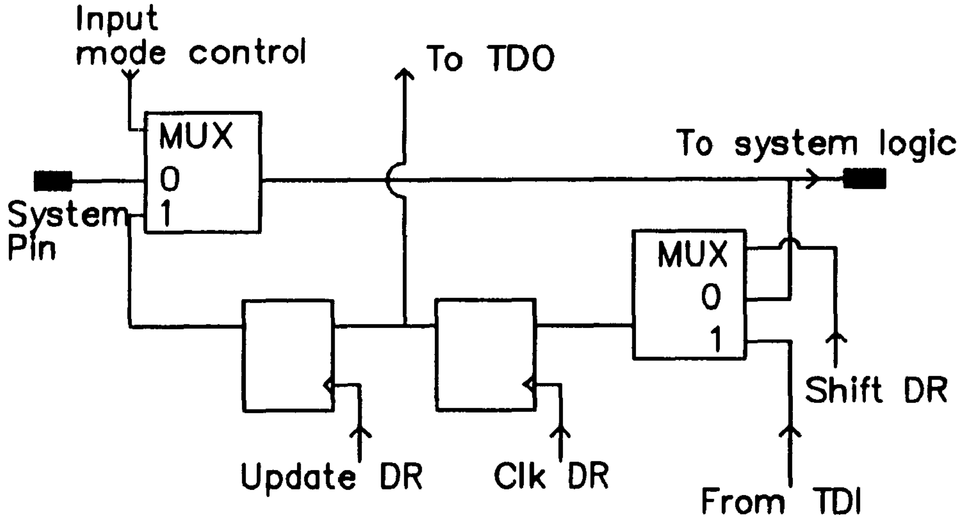


Figure 1.6 Input pin scan cell with output holding register.

- **Output Pin Scan Cell :**

Each uni-directional 2-state or open-collector system output pin must have a boundary scan cell. If the mode controls are set for the external test, the signal fed to the system pin must change only during the Update-DR controller state. During this test, the feedback around the registers via the multiplexer ensures that their state do not change. An output pin scan cell is shown in Figure 1.7.

- **Tri-State Pin Scan Cell :**

Each tri-state system pin must have a boundary scan cell prior to the tri-state buffer fed from the system data output. Such a cell is similar to the output pin scan cell described above. Additionally, the output enable signal must feed its associated output buffers through a boundary scan cell. Such a cell associated with the enable signal is termed as a *control scan cell*. A tri-state output pin scan cell with a control cell is shown in Figure 1.8. If the mode controls are set for external test, the signal fed to the connected buffer(s) from the control cell must change only during the Update-DR controller state.

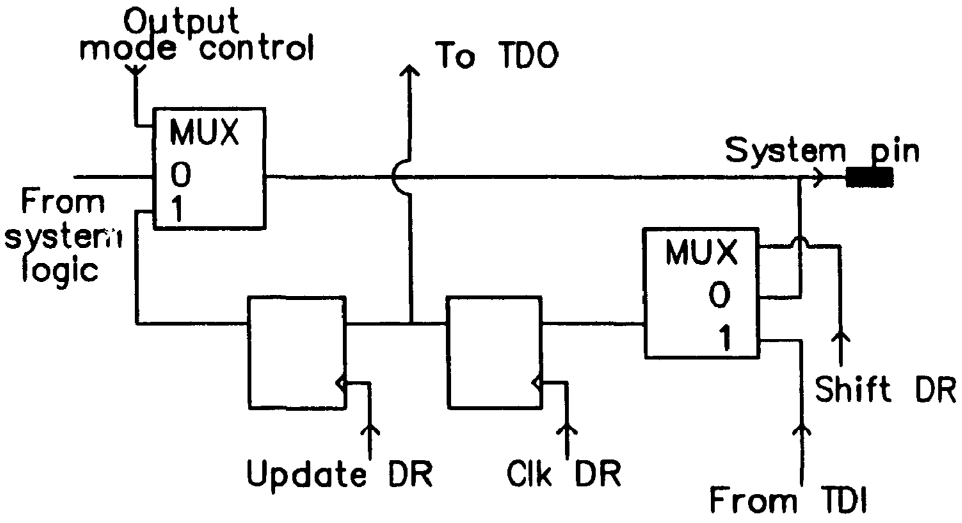


Figure 1.7 Output pin scan cell.

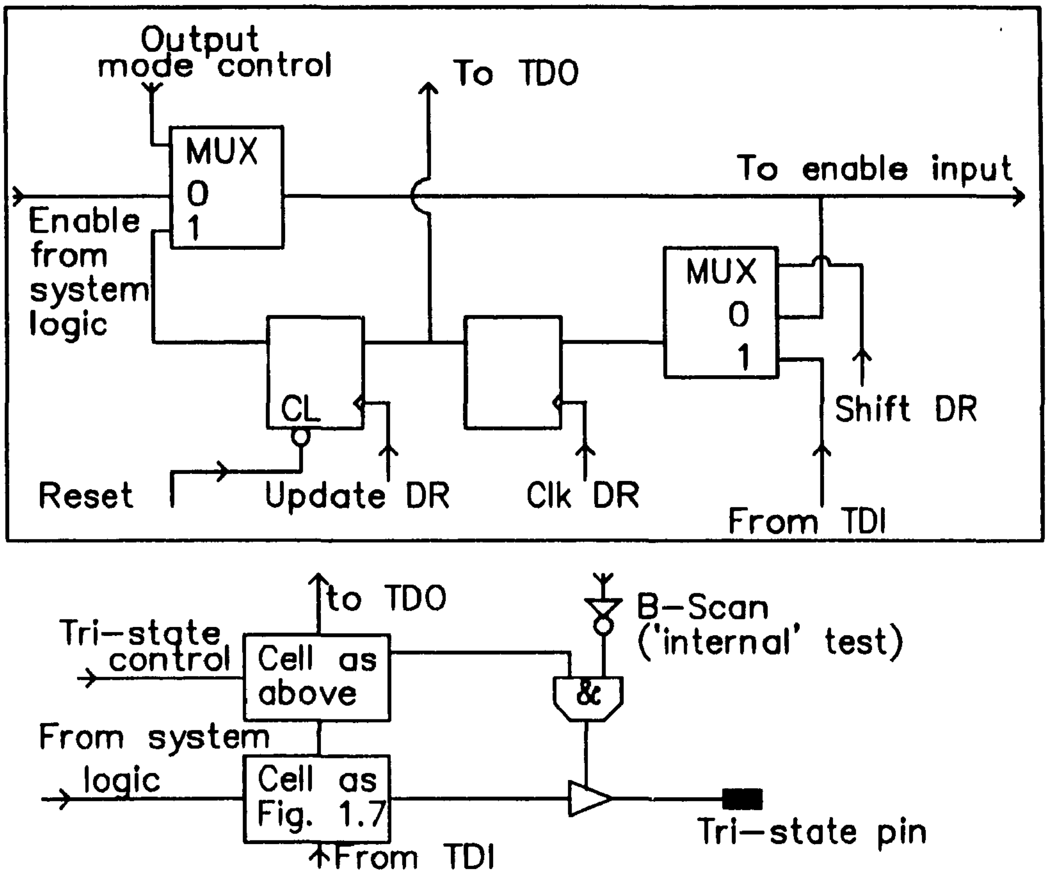


Figure 1.8 Scan cells associated with tri-state output pin.

- **Bi-directional Pin Scan Cell :**

Each bi-directional system pin must have one or more boundary scan cells placed between itself and the system logic, so that data flowing in either direction can be controlled or observed from the boundary scan cell(s). Any such scan cell should be either like the input pin scan cell or like the output pin scan cell. Additionally, each system direction control must feed its associated output buffer through a boundary scan cell. The scan cells associated with a bi-directional pin are shown in Figure 1.9.

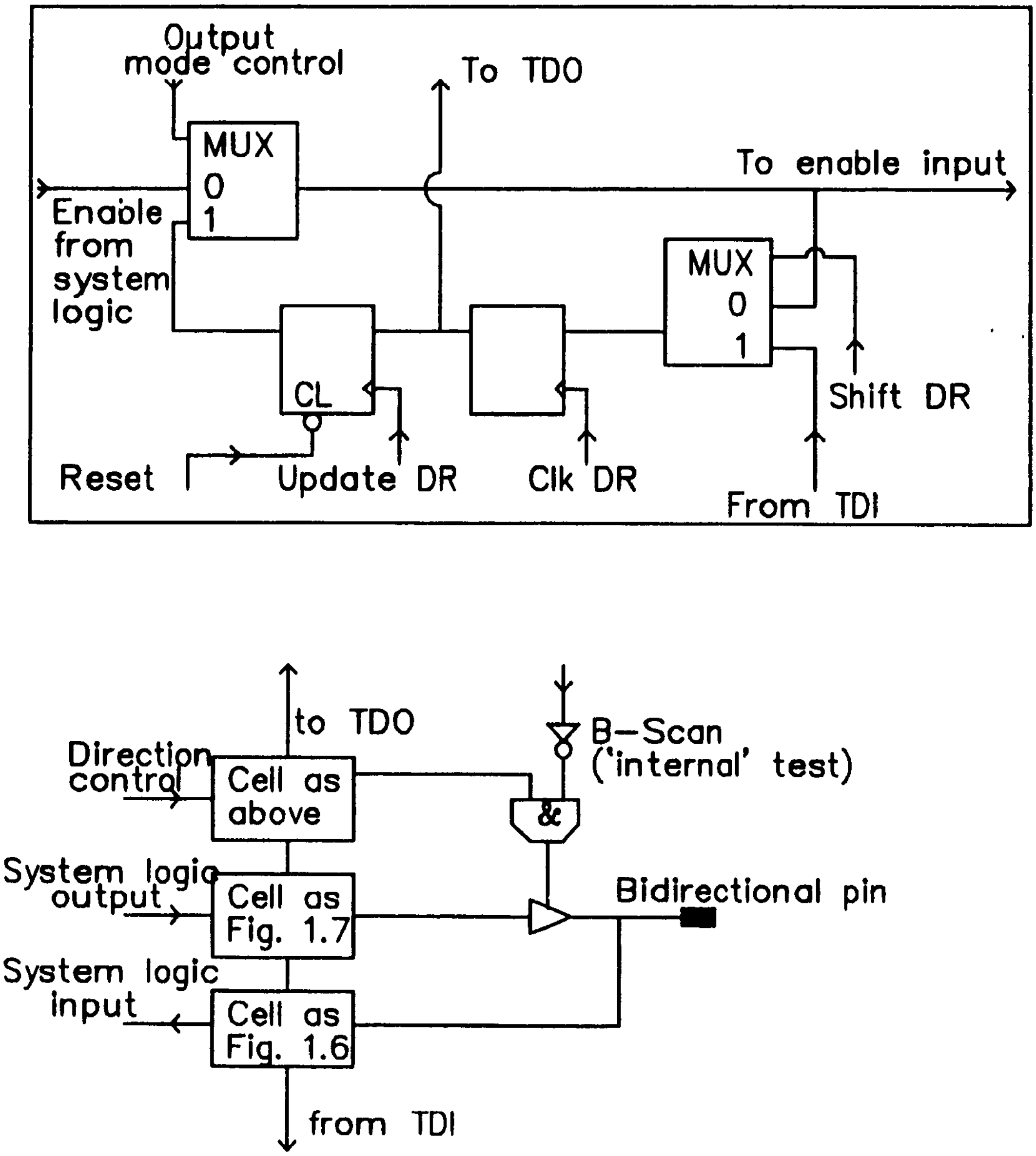


Figure 1 9 Scan cells associated with bidirectional pin.

1.3.2.2 Approaches based on Boundary Scan Architecture

The boundary scan architecture has extended the concept of DFT from the component level to the different hierarchical levels of system integration. In particular, the problem of accessibility in PCB level interconnect testing is solved by introducing the framework of scan path based testing. Test inputs can be shifted to the test site and applied to the interconnects, and the responses can be shifted out for analysis using the boundary scan register. Various algorithms have been proposed by different researchers for boundary scan based interconnect testing. These algorithms are discussed in detail in Chapter 3. These methods are called *external schemes*, because ATE external to the PCB is used for test vector generation and output response analysis. There are several disadvantages with this external approach : it requires considerable resource and time for test vector generation, it takes a long time to load test vectors and shift output responses through the boundary scan chain, it requires simulation to generate the expected outputs and memory space to store expected outputs.

1.3.2.3 Proposed Approach

This dissertation has taken a BIST approach based on the boundary scan architecture in order to solve the problems in external approach. Besides using it as a scan path to access test points, the boundary scan chain is used to generate time-efficient test vectors, to compact output responses, and to simplify the process of detection and diagnosis. The test vector sets used in the proposed BIST schemes are independent of both the topology of the interconnects under test, and the order of the I/O scan cells in the boundary scan chain. This 'circuit independent' test vector generation and loading requires less time and effort compared to the external approach. Techniques have been developed in these schemes to make detection and diagnosis independent of the structural complexity of the interconnects. As a result, no simulation is required to determine the expected outputs or signature. These advantages of the proposed BIST schemes are described in detail in the following chapters.

Chapter 2 Notations and Preliminaries

It was shown in Chapter 1 that the boundary scan architecture is a very convenient framework to overcome the interconnect test problems, like reduced controllability and obse.vability. This chapter introduces the various terms and concepts necessary to develop boundary scan based BiST schemes for interconnects. The interconnect structures, fault model of interest and the test environment are described in the following.

2.1 Structure of the Interconnects

The interconnects on a PCB can have various possible structures. In order to include these different structures as units under test, and to conveniently describe the various testing schemes, the term Inet is defined in the following.

Definition : *An Inet is any group of two or more I/O boundary scan cells and the electrical conductors (i.e., wires) which connect all these cells into one grid.*

The concept of 'grid' implies that if any one conductor in an Inet was set to a logic value, say, '0'('1'), then all the other conductors in that Inet carry the same logic value '0'('1').

Each Inet is named after the boundary scan cells connected to it and each conductor in an Inet is named after the two end-points of the conductor. Figure 2.1 shows an Inet $PQRS$ where P, Q, R and S are the four boundary scan cells and PT, TR, QU, US and TU are the conductors forming this Inet. An arrow in an Inet conductor indicates the direction of signal flow in that conductor.

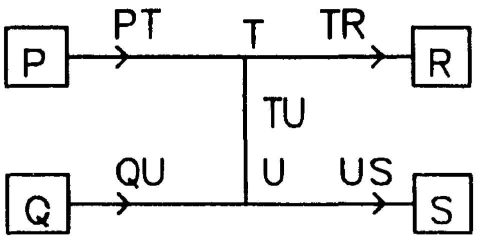


Figure 2.1 Inet under test.

Figure 2.2 shows an example PCB layout with four IC components $IC1$. $IC2$, $IC3$ and $IC4$. Each of these components has sixteen pins around the boundary. Four of these sixteen pins are for the boundary scan test access port (namely. TDI. TDO. TMS and TCK). Four other pins are used for power and system clock. The remaining eight pins in each component are the functional I/O pins. The boundary scan cells associated with these pins are connected in the serial shift register boundary scan path. The boundary scan paths of the four components are connected in a series to form the boundary scan chain of the example PCB. Interconnects connecting these four components are also shown in Figure 2 2. These interconnects with boundary scan cells are the Inets as defined above. The Inet structures of Figure 2.2 are shown in Figure 2.3.

An Inet can be driven by one or more drivers (input scan cells) and can be fanned out to one or more receivers (output scan cells). Therefore. based on the number of drivers and receivers. Inets can have different structures. There are two types of single driver Inets. The simple type is the Inet with a single driver and a single receiver. $1B2B$ and $3B4B$ are such simple Inets as shown in Figure 2.3. The second type of single driver Inet fans out to multiple receivers. $1R2R3R4E$. in Figure 2.3. is such an Inet. where the single driver $1R$ fans out to three receivers $2R$. $3R$ and $4E$.

Multi-driver Inets are also typified two ways : either a wire OR(AND) or a 3-state type. They are referred to as either a wire-Inet or a tri-state Inet. respectively. The logical behavior of wire-Inets is different from the logical behavior of other types of Inets. Wire-Inets function like OR(AND) logic gates. Because of this different nature. the testing of wire-Inets is not treated in this dissertation. Therefore. in the rest of the dissertation. multi-driver Inets will refer only to tri-state Inets. These multi-driver Inets are. again. of two types. In the first type. multiple drivers feed a single receiver. $1F2F3F4F$. in Figure 2.3. is such an Inet. Finally. multiple-drivers can be connected to multiple receivers giving a cluster structure. $1M2M3M4R$.

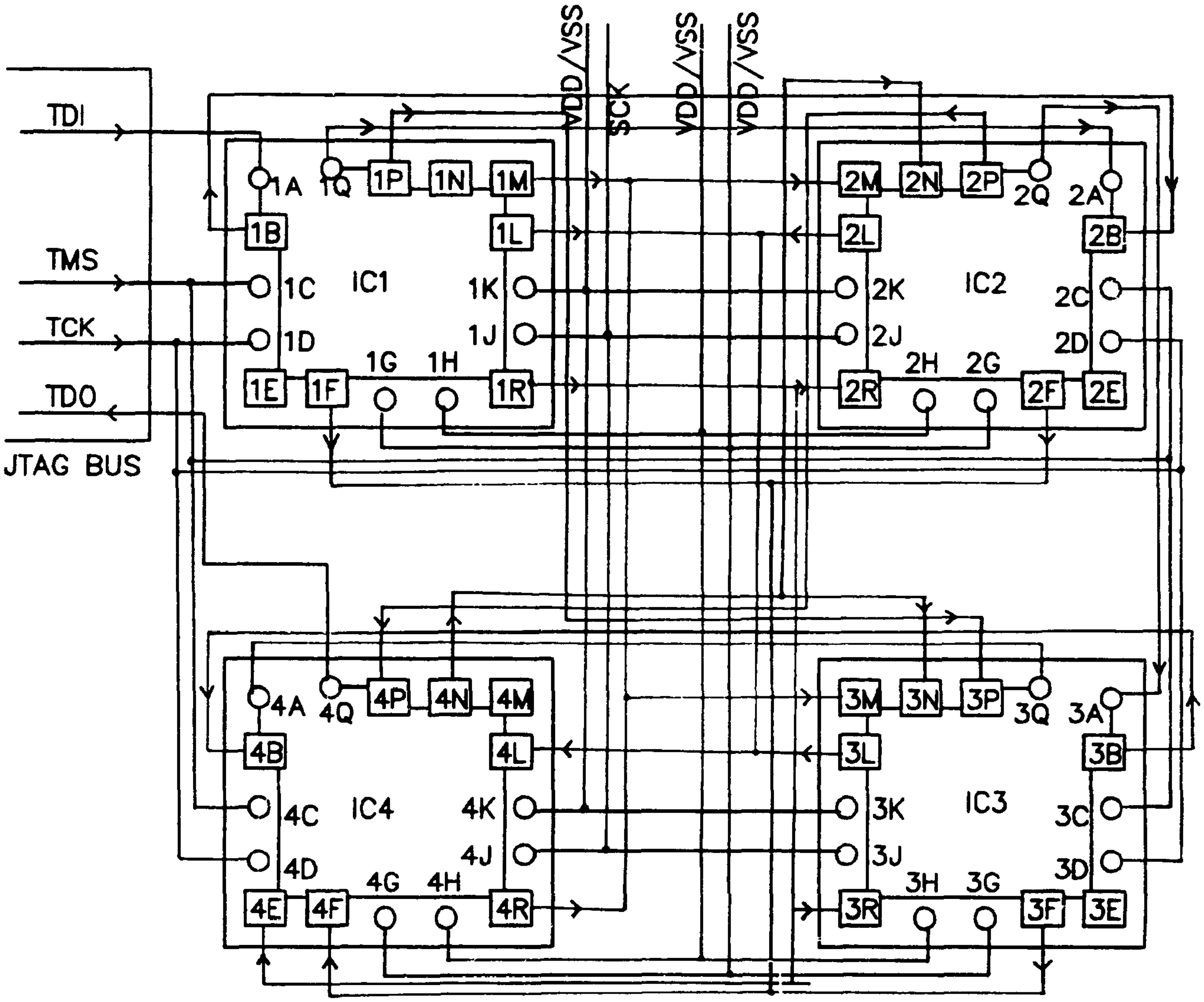


Figure 2.2 PCB level interconnects.

in Figure 2.3, is a cluster Inet.

Each output driver in a multi-driver Inet has tri-state capability. Tri-state is enabled/disabled by the control bit loaded in the associated boundary scan control cell [JTAG 88]. Tri-state drivers are enabled/disabled in such a way that, at any given time, only one driver of every multi-driver Inet is enabled. Such control setting avoids the bus contention in multi-driver Inets.

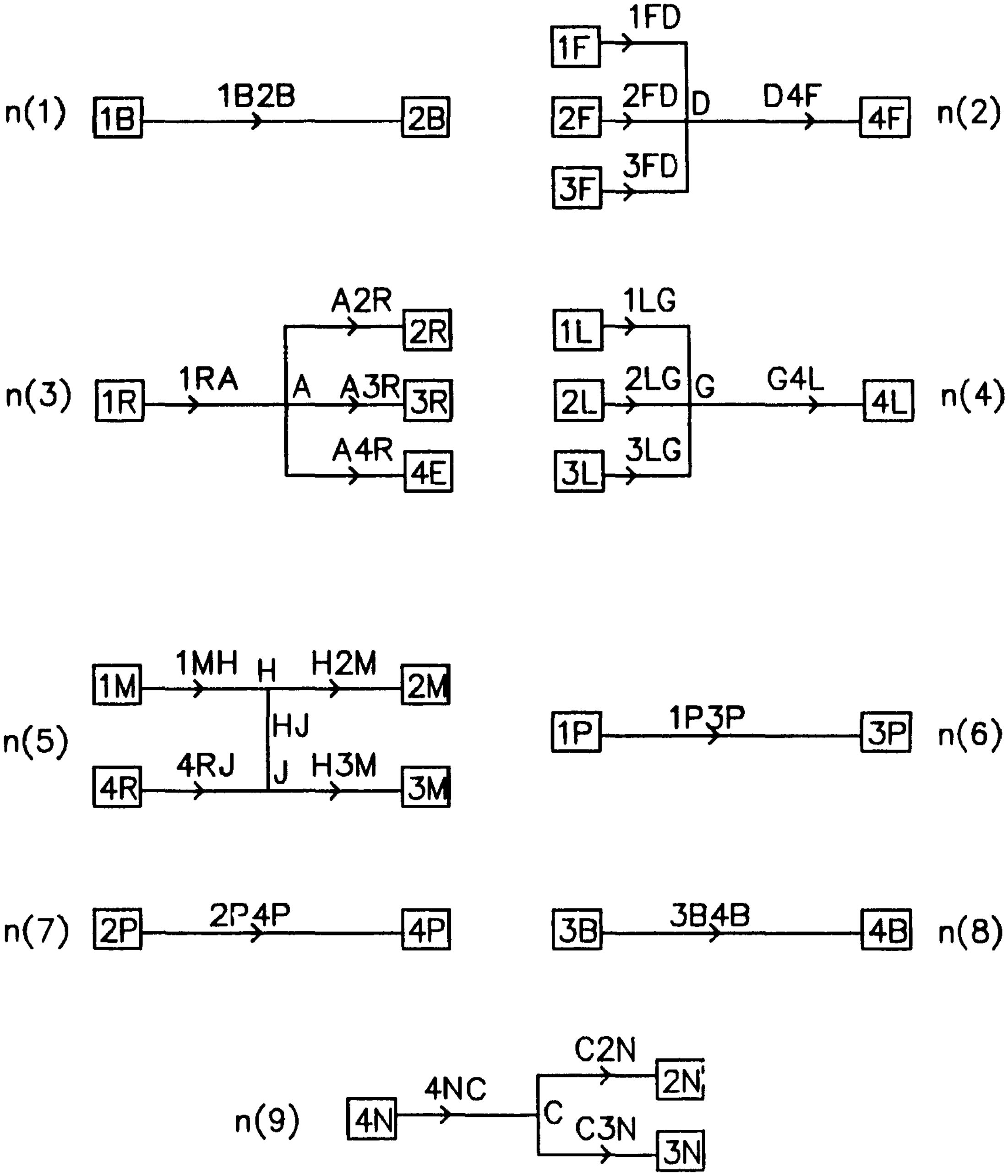


Figure 2.3 Different Inet structures

Two useful terms related to Inet structures are introduced in the following.

Path : *A path, in an Inet, is the collection of the conductors connecting one input scan cell to one output scan cell in that Inet.*

There exists a path, and only one path, between every input scan cell and every output scan cell in an Inet. Multiple paths between a pair of input-output scan cells indicate that there is redundancy in PCB wiring. It is assumed that there is no such redundancy in the design. Thus, the number of paths in an Inet, with x input scan cells and y output scan cells, is exactly $x + y$. In Figure 2.1, P_{QR} is a path connecting the input scan cell Q to the output scan cell R of Inet $PQRS$. The conductors QU, TU and TR are the members of this path P_{QR}.

A conductoı in an Inet can be a member of a number of paths. The term cpath is defined involving this concept.

Cpath : *A cpath corresponding to a conductor in an Inet is the collection of all the paths which include this conductor as a member.*

In Figure 2.1, the cpath corresponding to the conductor TU consists of the paths P_{PS} and P_{QR}. Since there is no wiring redundancy, the cpath of every conductor is different from the cpaths of all other conductors in that Inet.

2.2 Fault Model

The fault model of interest for Inet testing should be based on the failures likely to be observed in the Inets on PCBs. As mentioned in the literature [Bateson 85], [Bennetts 82], interconnect faults are induced by the manufacturing processes of PCB assembly and soldering. Assembly problems result in bent, broken or missing component leads. The soldering problems cause unwanted opens and shorts. For instance, opens can result from lack of solder, improper movement of parts, or bare printed circuit etches of feedthrough holes. Shorts are mainly caused by solder mask and solder flow. The classic assembly fault distribution is said to be between 25 and 55 percent. During soldering, the classic fault distribution ranges from 35 to 65 percent.

It is also mentioned in [Bateson 85] that the manufacturing process defect depends upon the type of PCB. For analog PCBs, these defects are low compared to digital or hybrid PCBs. This is because analog PCBs contain fewer components and analog components have fewer leads that require soldering. Consequently, the components, as well as the traces, are widely spaced. Contrarily, digital PCBs very often have a high density of components and tightly spaced traces (interconnects), plus multiple leads to be soldered for each component. As a result, digital PCBs experience a large number of manufacturing process defects. The hybrid PCB defect percentages are essentially between the analog PCB and the digital PCB, almost proportional to the mix of analog and digital components.

Based on the data available in the literature [Bateson 85], [Bennetts 82], [Goel 82], [Wagner 87], the following types of Inet faults are addressed.

(i) Short or Bridging Fault

This fault creates a (unwanted) physical connection between two or more Inets. The behavior of the Inets depend upon the driver characteristics of the individual Inets which are shorted together. The behavior can be deterministic or non-deterministic. Deterministic behavior can be characterized as follows :

a) **AND Short** : If the drivers of the shorted Inets are such that a logic '0' dominates, then the resultant logic value is an AND of the logic values of the individual Inets.

b) **OR Short** : If the drivers of the shorted Inets are such that a logic '1' dominates, then the resultant logic value is an OR of the logic values of the individual Inets.

Of these two types of shorts, depending upon the technology used in the individual component, either the AND type or the OR type will occur, but not both. However, components with different technologies can be used on the same PCB. Therefore, to make the testing schemes technology independent, this dissertation treats the simultaneous presence of both the AND and the OR type of shorts on a single PCB. These two short types are treated extensively in the dissertation.

A short between Inets can create an undefined or non-deterministic logic state at the receivers of the faulty Inets. The shorted outputs can be weak '0's or '1's. Analog testing is necessary to detect this type of short. Detecting the undefined faulty value is beyond the scope of the digital logic testing schemes developed in this dissertation. Therefore, this type of short is not considered in the fault model.

(ii) Stuck-at (one/zero) Fault

The faulty Inet gets stuck to a logic value of '1' (stuck-at-one)/ '0' (stuck-at-zero) due to this failure.

(iii) Stuck-open or Open Fault

This fault creates a break or open in the faulty lnet. Due to the break in the lnet, the logic value applied from the input end does not reach the output end. However, it is possible that the output end of an lnet is 'charged' to a logic '1'('0') due to the pad protection circuitry, parasitic capacitance etc. [Gruetzner 89] and a logic '1'('0') is applied from the input end for testing. In this case, the open fault will not be detected. Therefore, to test for an open fault, it is necessary to apply a '1'-'0' ('0'-'1') transition from the input end and, to observe the same transition at the output end of the lnet.

There is a difference between the way a short (SA) and an open fault affects an lnet. Short (SA) affects the entire lnet whereas open fault may affect only a part of the lnet. Due to this nature of short (SA), it is sufficient to apply test vectors from only one driver (input end scan cell) and to monitor the outputs from only one receiver (output end scan cell) of each lnet during short (SA) testing. In other words, the entire lnet can be sensitized from any input end scan cell. Also, if any conductor is shorted (SA), the fault effect can be observed from any output end scan cell of the faulty lnet irrespective of the location of the conductor in the lnet. A short is detected and diagnosed considering the lnet as the unit under test. No attempt is made to locate the faulty part or faulty conductor in the lnet.

Contrarily, open fault in every conductor of an lnet may not be sensitized from a single input end scan cell and may not be observed from a single output end scan cell of that lnet. For example, there are two input end scan cells, P and Q, in lnet PQRS as shown in Figure 2.1. Open fault in conductor PT cannot be sensitized by inputs applied from input end scan cell Q. Similarly, open fault in conductor TR cannot be observed from the output end scan cell R. Thus, in order to sensitize open fault in every conductor in an lnet, input vectors should be applied from every input end scan cell, one at a time. Also, responses should be observed at every output end scan cell of each lnet. Diagnosis of open fault is treated in two categories. In the first category, lnets are considered as units under test as in short and SA testing. The second category involves diagnosing within the faulty lnet. In this approach, lnet conductors are treated as the units under test and the faulty conductor in each lnet is diagnosed. It is assumed that a single conductor in each lnet can be faulty.

The fault model assumes that both single and multiple faults can occur in the Inets under test.

2.3 The Boundary Scan Test Environment

2.3.1 Loading and Application of Test Vectors

The following notations and definitions are useful for the test environment used in this dissertation.

N : Length of the boundary scan chain. This is the total number of input end scan cells, output end scan cells and control cells in the boundary scan chain. In Figure 2.2, $N = 32$.

N_i : i-th scan cell in the boundary scan chain (where $i = 1$ corresponds to the boundary scan cell closest to the scan-in edge of the PCB).

n : Number of Inets under test, $n < N$. In Figure 2.3, $n = 9$.

n_i : i-th Inet under test (where $i = 1$ corresponds to the Inet connected to the input end scan cell closest to the boundary scan-in edge of the PCB).

p_i : Degree of i-th Inet. This is the total number of input scan cells (or drivers) connected to the i-th Inet. In Figure 2.3, n_2 has three input drivers. Therefore, p_2=3.

P : max(p_i). In Figure 2.3, $P = 3$.

- **Input Vector** V_i : This is a set of input bits applied in parallel to all the Inets under test. Table 2.1 shows a set input vectors V_1, V_2, V_3 and V_4. Figure 2.4 shows the loading of vector V_4 into the boundary scan chain of Figure 2.2. Note that the vector is padded with 'X's and '0's('1's) so that the 'actual' data bits (shown in Table 2.1) are loaded in the appropriate input end scan cells and the control cells are loaded with required control bits.

- **Response Vector** R_i : This is a set of output bits received in parallel from all the Inets under test.

Inets	V_1	V_2	V_3	V_4
n_1	0	0	0	1
n_2	0	0	1	0
n_3	0	0	1	1
n_4	0	1	0	0
n_5	0	1	0	1
n_6	0	1	1	0
n_7	0	1	1	1
n_8	1	0	0	0
n_9	1	0	0	1

Table 2.1 Input vectors for Inet testing

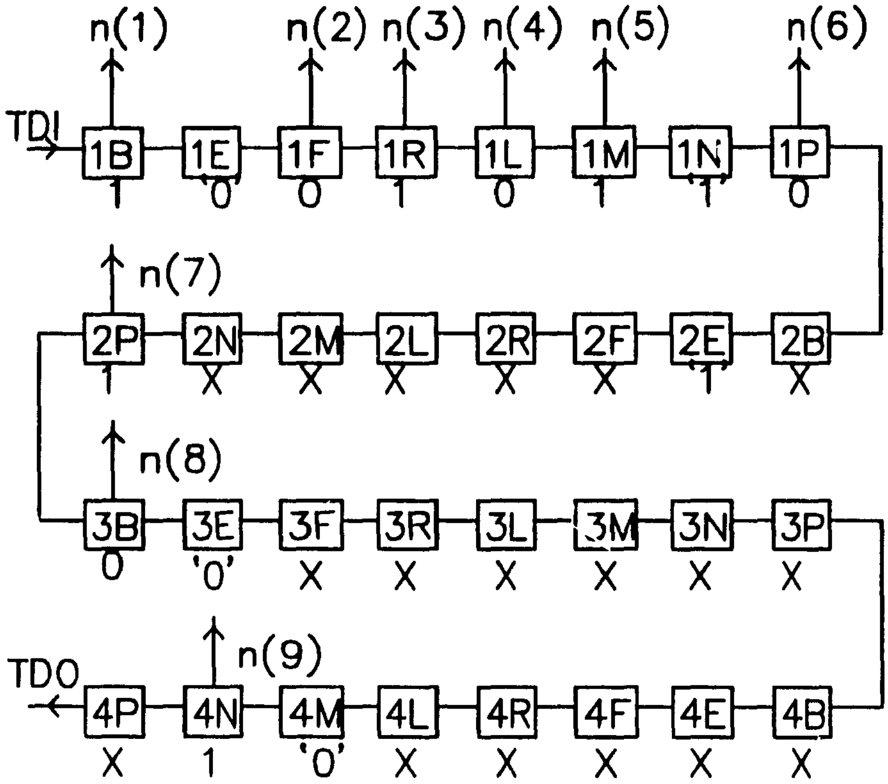


Figure 2.4 Input vector loading through B-Scan chain

- **Input Bit Set (IBS)** : This is the set of input bits applied to an Inet, over a period of time, by a number of input vectors. Different rows of Table 2.1 show the IBSs applied to different Inets n_1, n_2, n_3 etc.

- **Response Bit Set (RBS)** : This is the response of an Inet to an IBS. IBS and RBS are identical for a fault-free Inet. However, RBS should be different from the corresponding IBS for a detectable fault.

The basic elements of the boundary scan architecture are described in Chapter 1. It was shown that in a typical interconnect testing scenario, all the boundary scan cells associated with output connections of all the ICs would be first loaded with test data using the boundary scan register. In the second step, this test data would be applied and collected at the corresponding boundary scan cells associated with the input connections. In the final step, the collected responses are shifted out and verified. Therefore, interconnect tests are applied from output pin cells and received by input pin cells. As far as interconnect testing is concerned, output pins are input ends of the interconnects and sources of inputs, whereas input pin cells are the output ends of the interconnects and receivers of output responses. Therefore, in the remainder of the dissertation, the output pin scan cells will be called *input end scan cells* or, simply, *input scan cells* and the input pin scan cells will be called *output end scan cells* or *output scan cells*.

The process of application of a test vector to Inets is explained in the following. Figure 2.5 shows an Inet $BCEF$ with two input scan cells B and E and two output scan cells C and F. Since $BCEF$ is a multi-driver Inet, the drivers at B and E are enabled/disabled from the control cells A and D respectively. Suppose, a '1' is to be applied to this Inet. This requires that a '1' be loaded in one of the input scan cells of this Inet. This scan cell must be enabled from the corresponding control cell. The remaining input scan cells must be kept disabled from their corresponding control cells. In Figure 2.5, logic '1' is to be applied from input scan cell B. The driver at B is enabled from A (which is loaded with '1'). The driver at E is disabled from D (which is loaded with '0'). Input scan cell B is loaded with '1' (the bit to be applied to the Inet) and the scan cell E is loaded with 'X' ('0'/'1'), since E is disabled. The output scan cells C and F are initialized to '0'. Due to the application of '1' from input scan cell B to the Inet $BCEF$, output scan cells C and F receive '1's. These output bits are shifted for response analysis. Similarly, a '0' can be applied to an Inet.

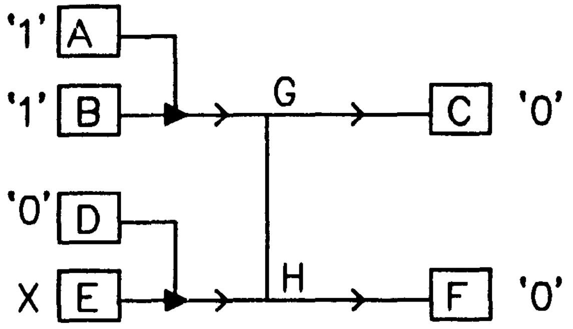


Figure 2.5 Application of a test vector.

In the case of a SA-0(SA-1) fault, the output scan cells C and F receive all '0's(all '1's) irrespective of what is applied from B.

For an open fault in any of the conductors BG, GC, GH and HF, the output scan cells C and F do not receive the input ('1') applied from B. Output scan cells C and F retain their initialized values ('0's). Note that in order to sensitize an open fault in conductor EH, input vector must be applied from input scan cell E (and input scan cell B must be kept disabled). Thus, as explained in Section 2.2, every driver in a multi-driver Inet must be enabled, one at a time, to apply input vectors for open testing. A set of input vectors must be applied P times with P different control settings so that vectors are applied from every driver in a multi-driver Inet, one at a time.

A short involves at least two Inets. An example of short between two Inets $BCEF$ and $LMPQ$ is shown in Figure 2.6. A '1' is applied to Inet $BCEF$ from input scan cell B and a '0' is applied to Inet $LMPQ$ from input scan cell P. For an OR-short between these two Inets, all the output scan cells C, F, M and Q receive '1's.

2.3.2 PCB Level BIST

As mentioned in Chapter 1, this dissertation has taken a BIST approach for board interconnect testing using the boundary scan architecture. This approach removes the need for any ATE. An IC or a group of ICs termed as *Test and Diagnosis Controller* (TDC) on the board under test controls the test procedure. The design of such a TDC is described in [Wang 89] and is shown in Figure 2.7.

Test vectors are generated in the TDC and are loaded through the boundary scan chain. However, in some of the proposed BIST schemes, the boundary scan chain is used to generate

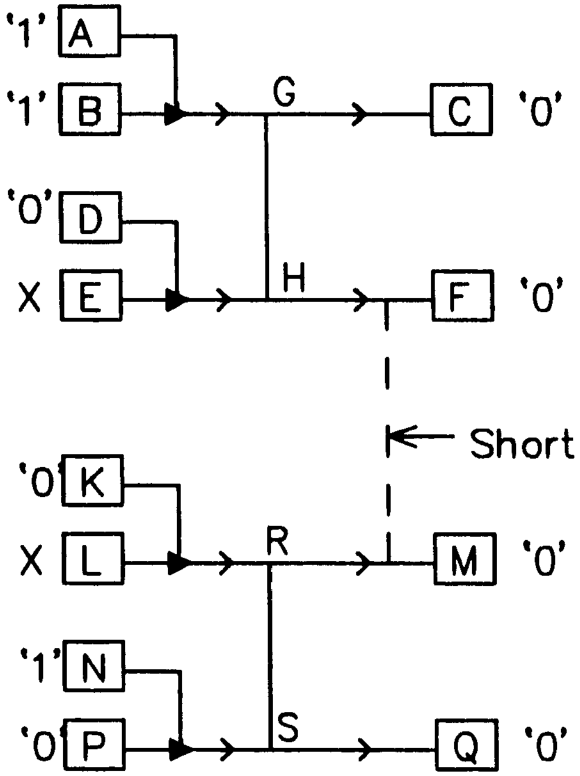


Figure 2.6 Short between two Inets.

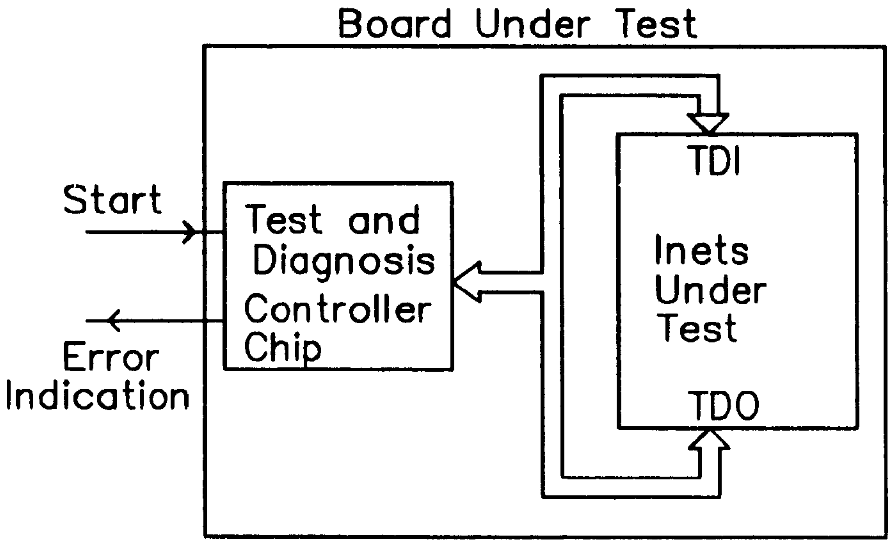


Figure 2.7 Architecture for PCB level BIST

the test vectors by internal shifting of the first vector loaded from the TDC. This use of the boundary scan chain for generation of test vectors leads to very time-efficient test schemes.

Similar to the component level BIST schemes, output responses are compacted in the proposed schemes. Response compaction techniques used in this dissertation are of two types. In one type, JTAG design [JTAG 88] of output scan cells are extended so that output responses can be compacted within each scan cell. The compacted responses are shifted out in the TDC for detection and diagnosis. This technique, called *Local Response Compaction*, reduces the shift out time. In the second approach, called *External Response Compaction*, output responses are shifted out and compacted in the TDC. This approach does not require any extension of the JTAG design of the scan cells. In both of these approaches, response compaction techniques are such that detection and diagnosis are independent of the structural complexity of the Inets under test. This has removed the need for generation and storage of any expected signature.

The boundary scan cells proposed by JTAG are extended to perform various functions in the proposed BIST schemes. The extensions are shown in the following chapters of the dissertation together with the schemes requiring these extensions.

Various schemes proposed in the following chapters are described using a Pascal-like notation.

Finally, a comparison of the proposed BIST schemes with the component level BIST schemes is in order. Most commonly, on-chip BIST schemes use pseudc random test patterns generated by certain configuration of LFSRs [Bardell 87]. On-board BIST schemes proposed in this dissertation use deterministic vector sets. However, these vector sets are independent of the interconnect structures under test, like the pseudorandom vector sets, and are easy to generate. This allows the generation of vector sets on the board under test. Response compaction techniques like bit-pair comparison and parity checking (using single-bit LFSRs) are used in the proposed schemes. In most of the proposed response compaction schemes there is no fault masking. However, in some of the schemes, multiple faults can mask each other.

2.3.3 Extended Control Cell with By-pass Capability

In some of the proposed test schemes, test vectors are generated by internal shifting operation within the boundary scan chain. This requires that the contents of each scan cell is changed after each apply-compact-shift cycle. On the other hand, the contents of each control cell is to be kept fixed during the application of a complete set of test vectors. These two requirements contradict each other. Therefore, internally generated vector sets cannot be implemented using the control cells described in [JTAG 88]. An extension of this control cell design is proposed in the following.

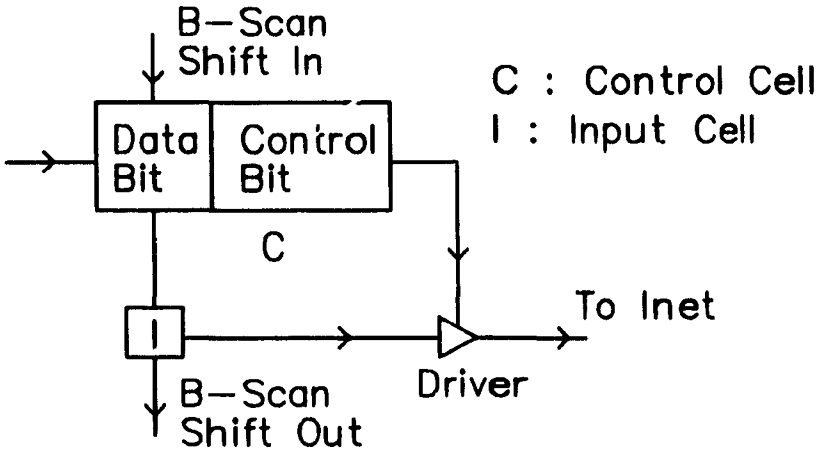
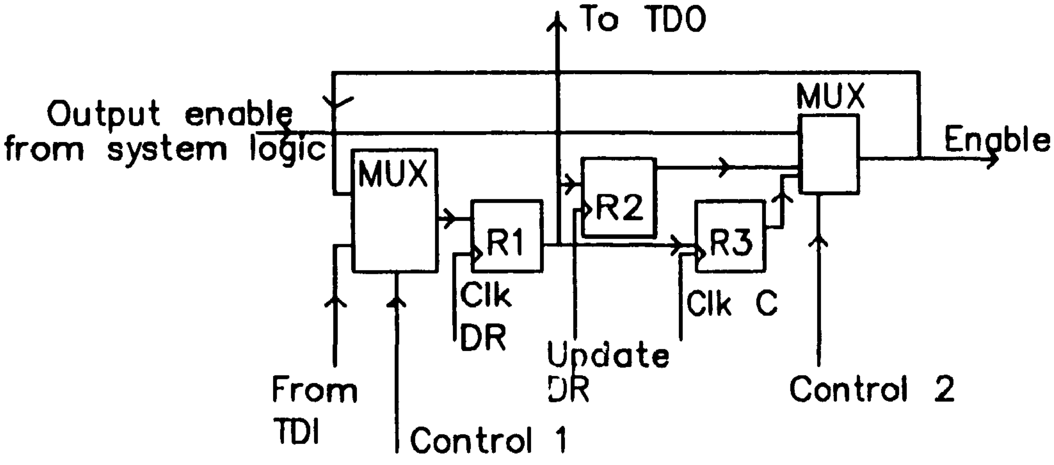


Figure 2.8 Block diagram of control cell with by-pass capability



Clk C = Update DR * Inst

where Inst is the bit from Instruction register
to load Register R3.

Figure 2.9 Control cell design with by-pass capability

In this design, the control cell has by-pass capability. The block diagram and the detailed design of this extended cell are shown in Figures 2.8 and 2.9 respectively. The register $R1$, shown in Figure 2.9, is used for shifting a bit-stream along the scan chain. Register $R3$ is loaded with the control bit. At the beginning, the control bits are shifted in using $R1$ in each control cell and loaded in $R3$. This loading is performed using proper instruction from the instruction register. After loading the control cells, test vectors are shifted in and loaded in the I/O scan cells. During test vector shifting and loading, $R1$ in each control cell is used to by-pass the corresponding control cell. This by-pass capability allows the generation of test vectors by internal shifting of the loaded vector, while the contents of the control cells remain fixed. The register $R2$ is used to load the control data when by-pass capability is not needed.

2.3.4 Compatibility with the JTAG Design

As mentioned above, JTAG design [JTAG 88] of the boundary scan cells are extended in the proposed schemes for various uses like internal shifting, response compaction, by-pass etc. However, all of the proposed extensions are compatible with the JTAG designs. This means that the extensions are flexible enough so that, if necessary, the extended cells can be set to function exactly in the same manner as the corresponding JTAG cells. For example, the control cell shown in Figure 2.9 has by-pass capability. If every control cell does not have this extension, then the extended control cells can be used as regular control cells (as proposed by JTAG) utilizing the register pairs $R1$ and $R2$. The same is true for other extended cell designs. Note that the advantages of the proposed BIST schemes can be exploited only if all the boundary scan cells have the required extensions. If that is not so, the scan cells can be set to perform their regular functions (proposed by JTAG). In such a case, the proposed scheme will lose its efficiency.

Chapter 3 Testing of Shorts and Stuck-At Faults

This chapter describes the existing schemes and proposes a BIST approach for the testing of shorts and SA faults in Inets. It is shown in the following that the existing schemes are incomplete from the implementation point of view. These schemes are actually test generation algorithms and, therefore, they do not address the various issues of implementation. Moreover, the existing schemes are suitable only for external test environment using an external test equipment. Contrarily, this dissertation takes an approach towards developing test schemes for PCB level BIST environment. The proposed schemes are complete from the implementation point of view. Therefore, a broad spectrum of issues, like number of test vectors, ease of test generation and loading, test application time, ease of detection and diagnosis are addressed in each of the proposed schemes. Techniques are used to develop optimal schemes from the BIST point of view. These BIST schemes can also be implemented in an external testing environment like the existing schemes.

3.1 Existing Schemes for Shorts and Stuck-Ats Detection

3.1.1 Minimal-Size Test Set Scheme for Shorts Detection

It has been shown in [Kautz 74] that a set of $\lceil log_2 n \rceil$ vectors is necessary and sufficient to detect every possible short in a network of n unconnected terminals. The terminals are checked by physical contact using multiple probe continuity test. This set of $\lceil log_2 n \rceil$ vectors is used in [Goel 82] and [Wagner 87] for testing shorts in n Inets.

In the following, the scheme is described with an example. Four vectors are required for the nine Inets of Figure 2.3, as shown in Table 3.1. It can be seen from the table that $\lceil log_2 n \rceil$ vectors are applied to n Inets, such that each Inet is assigned a unique binary number. Because of this assignment, each Inet IBS differs from the IBS of every other Inet by at least one-bit position. For example, the IBSs of n_1 and n_2 (see Table 3.1) differ in V_3. Therefore, the corresponding RBSs are also bound to be different in the fault free case. But in the case of a short between this pair of Inets, the response bits corresponding to V_3 are not different any more. Therefore, the short is detected at the output. This is true for every pair of Inets in the set. The same argument holds for multiple Inets shorted together.

Inets	V_1	V_2	V_3	V_4
n_1	0	0	0	0
n_2	0	0	0	1
n_3	0	0	1	0
n_4	0	0	1	1
n_5	0	1	0	0
n_6	0	1	0	1
n_7	0	1	1	0
n_8	0	1	1	1
n_9	1	0	0	0

Table 3.1 Minimal-size vector set for shorts detection.

3.1.2 Minimal-Size Test Set for Shorts and SAs Detection

An Inet stuck-at-one (SA-1) can be detected by applying a '0' as one of the input bits. Similarly, a SA-0 can be detected by applying a '1'. For example, if an IBS '0001' is applied to an Inet which is SA-1, the faulty RBS is '1111'. Therefore, a SA-1 in that Inet can be detected. For this reason, SA faults in most of the Inets can be detected by the set of $\lceil log_2 n \rceil$ vectors used for shorts detection in Section 3.1.1. However, notice that '0000' is assigned to

n_1 in that example. Clearly, '0000' will not detect a SA-0. Similarly, an IBS of all-one will not detect a SA-1. It was shown in [Wagner 87] that instead of $\lceil log_2 n\rceil$ vectors, if $\lceil log_2(n+2)\rceil$ vectors are applied (thus, avoiding all-zero and all-one IBSs) to n lnets, every possible SA and short can be detected. Therefore, $\lceil log_2(n+2)\rceil$ vectors are necessary and sufficient to detect every possible (single and multiple) SA and short in a system of n lnets.

3.2 Existing Schemes for Shorts and SAs Diagnosis

The existing schemes can be divided into two categories. As described in [Jarwala 89], the first category is One Step Test and Diagnosis where a set of test patterns is applied and the response is analyzed for fault diagnosis. The second category is Adaptive Test and Diagnosis where the test is applied, response analyzed and then one or more additional tests may be applied to aid diagnostics.

3.2.1 One Step Diagnosis Schemes

3.2.1.1 True/Complement Test Set Scheme

This diagnosis scheme proposed by Wagner in [Wagner 87] applies $2\lceil log_2(n+2)\rceil$ vectors (although $2\lceil log_2 n\rceil$ vectors are sufficient) to a system of n lnets under test. $\lceil log_2(n+2)\rceil$ vectors of Section 3.1.2 constitute the true vector set. Additional $\lceil log_2(n+2)\rceil$ vectors are obtained by complementing the true set of vectors. This true/complement test set can diagnose every possible single and multiple shorts in a system of n lnets. The true set of $\lceil log_2(n+2)\rceil$ vectors identify at least one of the lnets involved in each short. The complementary set then isolate the other lnets which were not identified by the true set. A true/complement vector set for the lnets in Figure 2.3 is shown in Table 3.2.

3.2.1.2 Min-Weight Test Set Scheme

In this scheme, proposed by Yau and Jarwala [Yau 89], the designer specifies the total number of input vectors, p, to be produced, in advance. During test generation, the Min-Weight Algorithm sequentially assigns a unique (p-bit) IBS of minimum weight to each of the

Inets	True set				Complement set			
	V_1	V_2	V_3	V_4	V'_1	V'_2	V'_3	V'_4
n_1	0	0	0	1	1	1	1	0
n_2	0	0	1	0	1	1	0	1
n_3	0	0	1	1	1	1	0	0
n_4	0	1	0	0	1	0	1	1
n_5	0	1	0	1	1	0	1	0
n_6	0	1	1	0	1	0	0	1
n_7	0	1	1	1	1	0	0	0
n_8	1	0	0	0	0	1	1	1
n_9	1	0	0	1	0	1	1	0

Table 3.2 True/Complement vector set for shorts and SAs diagnosis.

n Inets. Here, weight refers to the total number of '1's in an IBS. A sample test produced by this algorithm is shown in Table 3.3, where $p = 4$ and $n = 9$.

The vector set shown in Table 3.3 does not guarantee the diagnosis of every possible short. For example, if n_1 and n_5 are OR shorted together, the RBS corresponding to n_1 changes from '1000' to '1100'. However, the RBS corresponding to n_5 remains unchanged to '1100'. Therefore, this short can be detected but all the faulty Inets involved in this short cannot be diagnosed.

In general, the larger the p (where $\lceil log_2(n+2)\rceil \leq p \leq n$), the better the diagnostic capability of the vector set. It is interesting to observe the lower and upper bound conditions of this algorithm. If $p = \lceil log_2(n+2)\rceil$, the test is the most compact, but diagnostically least helpful. On the other hand, if $p = n$, the test is diagnostically most helpful, but spatially least compact.

3.2.1.3 Max-Independence Test Set Scheme

In this scheme, also proposed by Yau and Jarwala [Yau 89], test vectors are generated using Inet adjacency or wire routing information, as well as information about the maximum size of expected shorts or solder defects (the size is the number of Inets affected by the defect). Given the maximum extent of shorts E, minimum number p of vectors required

Inets	Vectors				Weights
	V_1	V_2	V_3	V_4	
n_1	1	0	0	0	1
n_2	0	1	0	0	1
n_3	0	0	1	0	1
n_4	0	0	0	1	1
n_5	1	1	0	0	2
n_6	1	0	1	0	2
n_7	1	0	0	1	2
n_8	0	1	1	0	2
n_9	0	1	0	1	2

Table 3.3 Min-weight test set.

for unambiguous diagnosis of all expected shorts is obtained. Complete or partial adjacency ordering of the n lnets are also obtained. Subsets of p-bit IBSs are generated (excluding the all-0 and all-1 IBSs) so that each subset is made of all possible IBSs which have the same potential weights and same Hamming weights. (Here, potential weight refers to the distance between the '1's at the lowest and the highest bit positions in an IBS.) Subsets of IBSs are generated starting from the smallest potential weight (i.e., 1) to higher potential weights until every lnet is assigned a unique IBS. A max-independence test set for $n = 12$, $E = 4$ and $p = 5$ is shown in Table 3.4.

This algorithm guarantees that as long as the extent of a given short never exceeds a certain upper bound, the fault can always be unambiguously diagnosed. The larger the maximum defect extent E, the larger the value of p should be to guarantee unambiguous diagnosis of all expected shorts.

3.2.2 Adaptive Diagnosis Schemes

3.2.2.1 One-Test Adaptive Scheme

This scheme is proposed by Jarwala and Yau [Jarwala 89]. Here, $\lceil log_2(n+2) \rceil$ vectors introduced in Section 3.1.2 are applied in the first step. These vectors identify at least one

Adjacency	Vectors					Potential
Ordered Inets	V_1	V_2	V_3	V_4	V_5	Weights
n_1	1	0	0	0	0	1
n_2	0	1	0	0	0	1
n_3	0	0	1	0	0	1
n_4	0	0	0	1	0	1
n_5	0	0	0	0	1	1
n_6	1	1	0	0	0	2
n_7	0	1	1	0	0	2
n_8	0	0	1	1	0	2
n_9	0	0	0	1	1	2
n_{10}	1	0	1	0	0	3
n_{11}	0	1	0	1	0	3
n_{12}	0	0	1	0	1	3

Table 3.4 Max-independence test set.

of the Inets involved in each short. For example, if n_1 and n_3 in Table 3.5 are OR shorted, RBS of n_1 will be changed from '0001' to '0011'. However, RBS of n_3 will remain the same ('0011'). Therefore, at least one of the Inets (in this case n_1) can be identified as faulty. But n_3 cannot be identified as faulty. This situation is described as aliasing in [Jarwala 89]. In Step 2 of the scheme, a single vector is applied to identify the Inets whose RBSs are aliased to. In this vector, the bits applied to the Inets whose IBSs are aliased to, are set to '1'. The remaining bits are set to '0'. This step can be performed for all aliased cases in parallel and hence the second step requires only one test vector.

3.2.2.2 W-Test Adaptive Scheme

Goel and McMahon have described a diagnosis scheme in [Goel 82] which is also divided into two steps. In the first step, $\lceil log_2(n+2) \rceil$ vectors used for detection in Section 3.1.2 are applied to identify a subset W of the faulty Inets. The shorts must involve these Inets as well as other Inets. In the second step, a unique test is applied to each member w of W. In this test, w is assigned a '0' (or, '1') and the remaining $(n-1)$ Inets are assigned a '1' (or, '0').

Inets	Vectors			
	V_1	V_2	V_3	V_4
n_1	0	0	0	1
n_2	0	0	1	0
n_3	0	0	1	1
n_4	0	1	0	0
n_5	0	1	0	1
n_6	0	1	1	0
n_7	0	1	1	1
n_8	1	0	0	0
n_9	1	0	0	1

Table 3.5 Vector set for one-step adaptive diagnosis.

This indicates which Inets are shorted to w. Thus, W tests in the second step can identify all the shorted Inets.

3.2.2.3 Optimal C-Test Adaptive Scheme

This scheme is proposed in [Jarwala 89]. In step one of this scheme, $\lceil log_2(n+2)\rceil$ vectors are applied to identify some (not all) faulty Inets which are shorted. But, suppose there are two shorts. One short involves c_1 Inets and the second short involves c_2 Inets. These two shorts are such that the faulty RBSs are the same for the two groups of faulty Inets. Therefore, one cannot diagnose if the two groups of Inets are shorted together or separately. This is described as the confounding syndrome [Jarwala 89]. The degree of the confounding syndrome is defined by the total number of shorts which have the same RBS. This scheme applies $(C-1)$ vectors in the second step where C is the maximum degree of the confounding syndrome. One more vector is required to diagnosis the aliasing syndrome. Therefore, C tests in the second step provides complete diagnosis.

3.3 Limitations of the External Schemes

Several detection and diagnosis schemes are described in the previous sections. These schemes have limitations from a practical or implementational point of view. These schemes are test generation algorithms. They describe the number of test vectors required for detection and/or diagnosis of faults and how to generate these vectors. Other issues like loading of test vectors through the boundary scan chain, structural complexity of the Inets and analysis of output response are not addressed. These schemes are more of test generation algorithms than of test schemes. They are developed for use in an external test environment with ATEs. The shortcomings of these external schemes, in the context of boundary scan architecture, are described in the following.

(i) Test Generation and Loading :

In all of the existing schemes, test vectors are generated based on the number of Inets. Each vector consists of n bits, since n is the total number of Inets under test. However, in the boundary scan chain, input scan cells, output scan cells and control cells are connected along the same chain. The length of this scan chain is N, where $N > n$. Therefore, the lengths of the scan chain and the test vector are not equal. Moreover, the scan cells are connected along the scan chain in a certain order. Therefore, a generated vector is to be reformatted or restructured before loading through the scan chain. Each vector is to be padded with extra '0's (or, '1's) in the reformatting operation, so that 'actual' bits of the vector are loaded in the appropriate input scan cells. This requires information about the order of the scan chain. Also, the reformatting operation involves extra time and storage. The assignment of test vectors in Max-Independence Scheme depends upon the ordering (or partial ordering) of the Inets. In this case, test vector generation also depends upon the structural information about the Inets. Ordering of Inets requires some pre-processing time.

(ii) Analysis of Output Response :

In the external schemes, output response is shifted out of the boundary scan chain and compared with the stored expected response for detection and diagnosis. The expected responses are generated based on the actual structure of the Inets. Using the wire-routing information, fault-free simulation is done to obtain the expected responses. Therefore, detection and diagnosis depend upon the structural complexity of the Inets. Moreover, this approach requires pre-processing time and storage for the expected responses.

(iii) Adaptive Schemes :

Adaptive algorithms attempt to reduce the total number of test vectors. However, in each of the three adaptive schemes described above, test generation in the second step depends upon the outcome of the first step. Response analysis in these schemes is more involved and requires more inform:ation about Inet structures compared to the one-step schemes. Also, test generation in the second step is to be done in real time (instead of one time generation and storage). This can increase the total test time.

(iv) ATE :

The boundary scan architecture has provided electronic access to the interconnect test points, thus, removing the need for expensive ATEs and test interfaces. However, the external test schemes depend upon ATEs for the generation of input vectors and expected responses and for the analysis of output responses.

3.4 Summary of Existing Schemes

Various properties and limitations of the existing external test schemes, as described above, are summarized in the following.

(i) Minimal-Size Test Set Scheme for Shorts Detection :

- Number of vectors $\lceil log_2 n \rceil$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $N.\lceil log_2 n \rceil$ bits
- Detects every short
- Detection Inet structure dependent
- ATE required
- Time complexity $O(Nlogn)$

(ii) Minimal-Size Test Set Scheme for Shorts and SAs Detection :

- Number of vectors $\lceil log_2(n+2) \rceil$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $N.\lceil log_2(n+2) \rceil$ bits
- Detects every short and SA
- Detection Inet structure dependent
- ATE required
- Time complexity $O(N logn)$

(iii) One-step schemes for Short and SA Diagnosis

(a) True/Complement Test Set Scheme :

- Number of vectors $2.\lceil log_2(n+2) \rceil$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $2.N.\lceil log_2(n+2) \rceil$ bits
- Diagnoses every short and SA
- Diagnosis Inet structure dependent
- ATE required
- Time complexity $O(N logn)$

(b) Min-Weight Test Set Scheme :

- Number of vectors p, where $\lceil log_2(n+2) \rceil \leq p \leq n$
- Test vectors to be reformatted
- Simulation to obtain expected response

- Expected response storage requirement pN bits
- Diagnostic capability depends upon p
- Diagnosis Inet structure dependent
- ATE required
- Time complexity $O(pN)$

(c) Max-Independence Test Set Scheme :

- Number of vectors p, where p depends upon E and n
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement pN bits
- Diagnostic capability depends upon E
- Diagnosis Inet structure dependent
- ATE required
- Time complexity $O(pN)$

(iv) Adaptive Schemes for Short and SA Diagnosis

(a) One-Test Adaptive Scheme :

- Number of vectors $\lceil log_2(n+2)\rceil + 1$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $N.(\lceil log_2(n+2)\rceil + 1)$ bits
- Diagnoses every short and SA
- Diagnosis Inet structure dependent

- ATE required
- Time complexity $O(N logn)$

(b) W-Test Adaptive Scheme :

- Number of vectors $\lceil log_2(n+2)\rceil + W$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $N.(\lceil log_2(n+2)\rceil + W)$ bits
- Diagnoses every short and SA
- Diagnosis Inet structure dependent
- ATE required
- Time complexity $O(N[logn + n])$

(c) Optimal C-Test Adaptive Scheme :

- Number of vectors $\lceil log_2(n+2)\rceil + C$
- Test vectors to be reformatted
- Simulation to obtain expected response
- Expected response storage requirement $N.(\lceil log_2(n+2)\rceil + C)$ bits
- Diagnoses every short and SA
- Diagnosis Inet structure dependent
- ATE required
- Time complexity $O(N[logn + C])$

3.5 Proposed Schemes

This dissertation approaches the development of interconnect test schemes differently. It addresses the interconnect testing problem in the context of boundary scan and develops a family of schemes to exploit the advantages of the boundary scan environment. The schemes address various issues including test generation complexity, test vector loading and application, response analysis and diagnostic resolution. The existing schemes have used the boundary scan framework simply to get electronic access to the interconnect test points. In the proposed approach, the boundary scan architecture is treated as a framework for board level BIST. Boundary scan architecture is shown to have the capability to generate time-efficient and Inet structure independent test vectors. Response compaction within the boundary scan chain is introduced to reduce the shift out time as well as to simplify detection and diagnosis. The detection and diagnosis processes in the proposed schemes are made simple so that on-board BIST becomes feasible. This approach emphasizes the BIST implementation of the proposed test schemes. Test vector generation and response analysis can be done on the board under test. This will remove the need for ATE. However, the schemes are not limited to an on-board BIST environment. They can also be implemented with ATEs, like the existing external schemes. Various detection and diagnosis schemes are proposed in the following.

3.5.1 Detection of Shorts and SAs

Three different schemes are developed in the following for detection of shorts and SAs. The first scheme is a simple modification of the minimal-size test set scheme. It shows that the proposed modification removes the need for the restructuring of test vectors. The second and third schemes use a very time-efficient test vector set called the walking sequence. In one of these two walking sequence based schemes, output responses are compacted within each output scan cell to reduce the shift out time. In the other scheme, the responses are compacted outside the scan chain so that no extension to the standard boundary scan cells [JTAG 88] is necessary. The detection in both of these schemes is independent of the structural complexity of the Inets.

3.5.1.1 Order Independent Vector Set Scheme

As mentioned in Section 3.3, using the minimal size test set leads to the padding problem.

Minimal-size test set scheme requires knowledge about the order of the scan chain. A simple but elegant solution to this problem is to use some additional test vectors. To be specific, one could generate $\lceil log_2(N+2)\rceil$ vectors with each vector N bits long (instead of n bits) for a scan chain N cells long. N bits of each vector are loaded through the scan chain. The detection process is the same as in the minimal-size test set scheme, but no reformatting or structural information is required for test vector generation and loading. Since no information related to the order of the I/O cells in the scan chain is required, the scheme will be called the order independent test set scheme. This test set is not minimal size any more. $\lceil log_2(N+2)\rceil$ vectors are required instead of $\lceil log_2(n+2)\rceil$ vectors. The test time required is $N.\lceil log_2(N+2)\rceil$ compared to the test time of $N.\lceil log_2(n+2)\rceil$ for the minimal size test set scheme. Since, $n = O(N)$, the time complexity is $O(NlogN)$ in both schemes. A comparison of the test times of minimal-size test set scheme and order independent vector set scheme for different values of n and N are shown in Table 3.6. The order independent test vector set is suitable for BIST implementation because of its simple test generation and loading property.

N	n	$N\lceil logn\rceil$	$N\lceil logN\rceil$
1,000	100	7,000	10,000
1,000	500	9,000	10,000
4,000	400	36,000	48,000
4,000	2,000	44,000	48,000
10,000	1,000	100,000	140,000
10,000	5,000	130,000	140,000

Table 3.6 Comparison of test times.

Test Generation Hardware :

In the PCB level BIST environment test vectors are generated on the TDC. Therefore, the test generation hardware must be simple and small in size to keep the BIST overhead reasonable. The test generation hardware to generate $\lceil log_2(N+2)\rceil$ vectors is shown in Figure 3.1. In this scheme, a $\lceil log_2(N+2)\rceil$ bit counter generates the $\lceil log_2(N+2)\rceil$ test vectors. Notice that each test vector consists of (N+2) bits and is generated serially from one of the stages of this counter. A $\lceil log_2(N+2)\rceil$: 1 MUX is then used to select which test vector should be applied during one scan cycle. The MUX is controlled by a $\lceil log_2(log_2(N+2))\rceil$ bit counter. The state of this counter is changed after counting through $(N+2)$ in the $\lceil log_2(N+2)\rceil$ bit counter. The control bits are appended with the data bits in the counter. Therefore, the hardware is a $\lceil log_2(N+2)\rceil + \lceil log_2(log_2(N+2))\rceil$ bit counter where the $\lceil log_2(log_2(N+2))\rceil$ MSBs are the control bits and the $\lceil log_2(N+2)\rceil$ LSBs are the data bits. N output bits (excepting the first and the last bits) coming from the first LSB are chosen as the first vector by the $\lceil log_2(N+2)\rceil$:1 MUX. It can be shown that these bits form the first vector in the set of $\lceil log_2(N+2)\rceil$ vectors. In the same way the outputs of the second LSB register form the second vector and so on.

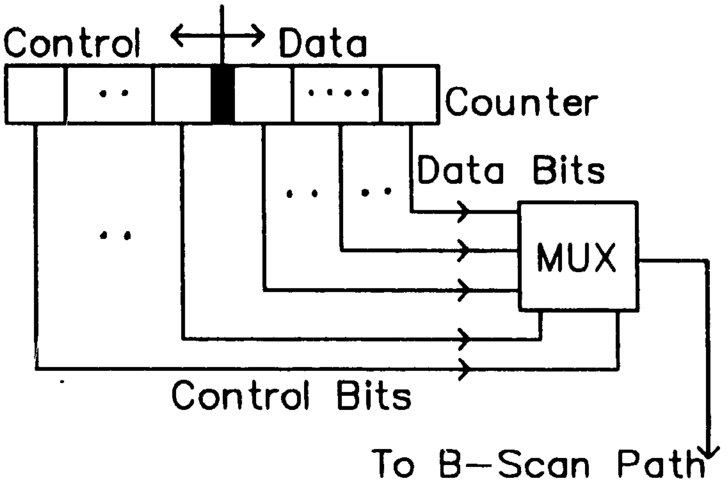


Figure 3.1 Hardware to generate $\lceil log_2(N+2)\rceil$ vectors.

Response Analysis :

After loading and application of each test vector, the response is shifted out for detection of faults. The obtained response is compared with the expected response. The expected response can be determined by fault-free simulation of the system of Inets under test. This requires structural information about these Inets. Therefore, fault detection, in this scheme, is Inet structure dependent, as it was in the minimal-size test set scheme. Moreover, the minimal-size test set scheme requires $N\lceil log_2 n\rceil$ bits of storage for the expected responses and $N\lceil log_2 N\rceil$ bits of expected responses are stored for this scheme.

The main advantage of the proposed scheme is boundary scan chain order independent test vector generation and loading.

3.5.1.2 Walking Sequence

To overcome the disadvantages of the existing schemes and order independent test set scheme, a different type of vector set is considered here. Consider a bit stream of a single '1' followed by all '0's shown below :

1 0 0 0 0 0 0 0 ...

This bit stream can be loaded through the boundary scan chain as the first test vector. Then, by gradually shifting this vector through the boundary scan chain, the rest of the vectors can be obtained. Since the vector set is obtained by gradually shifting the single '1' along the stream, this sequence is called a *walking one sequence.* Such a sequence is shown in Table 3.7.

Similarly, a walking zero sequence can be obtained where a bit stream (N-bit long) of single '0' followed by $(N-1)$ '1's, i.e., 0111... , is shifted $(N-1)$ times to get N vectors. Table 3.8 shows a walking zero sequence.

Scan Cells	Vectors				
	V_1	V_2	V_3	...	V_N
N_1	1	0	0	...	0
N_2	0	1	0	...	0
N_3	0	0	1	...	0
...	...	...	...	...	...
N_N	0	0	0	...	1

Table 3.7 Walking one sequence.

Test Generation Hardware :

A walking one (or, a walking zero) sequence is easy to generate. The outputs from the flip-flops of N-bit counter are fed to an NOR (OR) gate (see Figure 3.2) The resulting vector becomes 1000... (0111...).

Scan Cells	Vectors				
	V_1	V_2	V_3	...	V_N
N_1	0	1	1	...	1
N_2	1	0	1	..	1
N_3	1	1	0	...	1
...	...	...	...	...	...
N_N	1	1	1	...	0

Table 3.8 Walking zero sequence

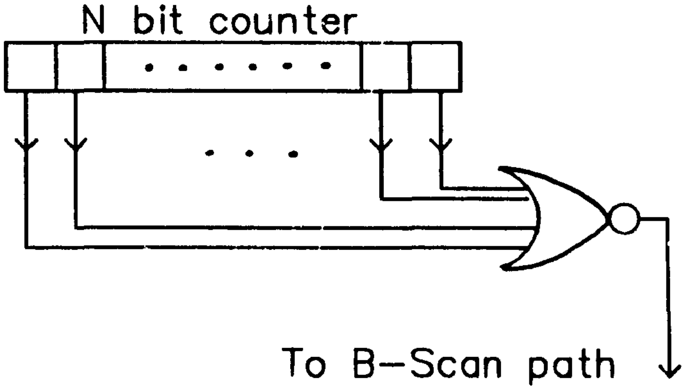


Figure 3.2 Hardware to generate walking sequence.

Theorem 3.1 *N vectors of a walking one/zero sequence are sufficient to detect every possible short and SA fault in a system of n Inets.*

Proof : It can be seen from any row in Table 3.7 that for a walking one sequence, the IBS corresponding to every boundary scan cell consists of a single '1' and $(N-1)$ '0's. It was mentioned in Chapter 2 that test vectors are applied from only one selected input scan cell of every Inet. It is easy to see that the IBS applied to every Inet consists of a single '1' and $(N-1)$ '0's. Therefore, in the fault-free case the RBS obtained at each output scan cell of every Inet consists of a single '1' and $(N-1)$ '0's. In the case of a SA fault, the number of '1's in the RBS is either increased to N '1's (for SA-1) or decreased to zero '1's (for SA-0).

Notice that every input vector in the walking one sequence contains a single '1'. Therefore, '1's can be applied to no two Inets simultaneously. At most, only one Inet is applied a '1' and the remaining Inets are applied '0's due to each input vector. IBSs applied to two Inets n_i and n_j look like this :

$$n_i \quad ...00..010..00...$$

$$n_j \quad ...01..000..00...$$

If two such lnets are shorted together, the number of '1's in the faulty RBSs is either increased (for OR short) or decreased (for AND short).

Therefore, any short or SA fault can be detected by applying a walking one sequence.

[QED].

An identical proof can be made for walking zero sequence.

In the following, two different schemes are proposed for detection of shorts and SAs using the walking sequences. In the first scheme, output responses are compacted outside the boundary scan chain. The responses are compacted within each output scan cell in the second scheme, in order to utilize the shifting property of the walking sequence.

(i) Walking Sequence Scheme using External Response Compaction

In this scheme every input vector is shifted in through the scan chain, then applied and the response is shifted out. Walking one sequence is used in the following to describe the scheme. However, walking zero sequence can also be used as the input sequence. The response is taken out and fed to a response compactor which is a 1's counter in this scheme. The procedure is as follows :

```
Begin
    Begin
    Repeat $N$ times
        Begin
        Load a vector (which is not previously loaded) from $N$ vectors
          of walking one sequence in the $N$ bit scan path
        Apply loaded vector
        Collect response at each output scan cell
        Unload the $N$ bit scan path with output response in 1's counter
        End
    End
Detect based on the count of the 1's counter
    If count $= N$ Then no single fault
    Else faulty
End
```

Lemma 3.1 *In the fault-free case, total number of '1's shifted out of the scan chain due to application of N vectors of the walking one sequence is N.*

Proof : Every vector in the walking one sequence contains a single '1' and (N - 1) '0's. The single '1' can be loaded in one of the five different positions as shown in Figure 3.3 :

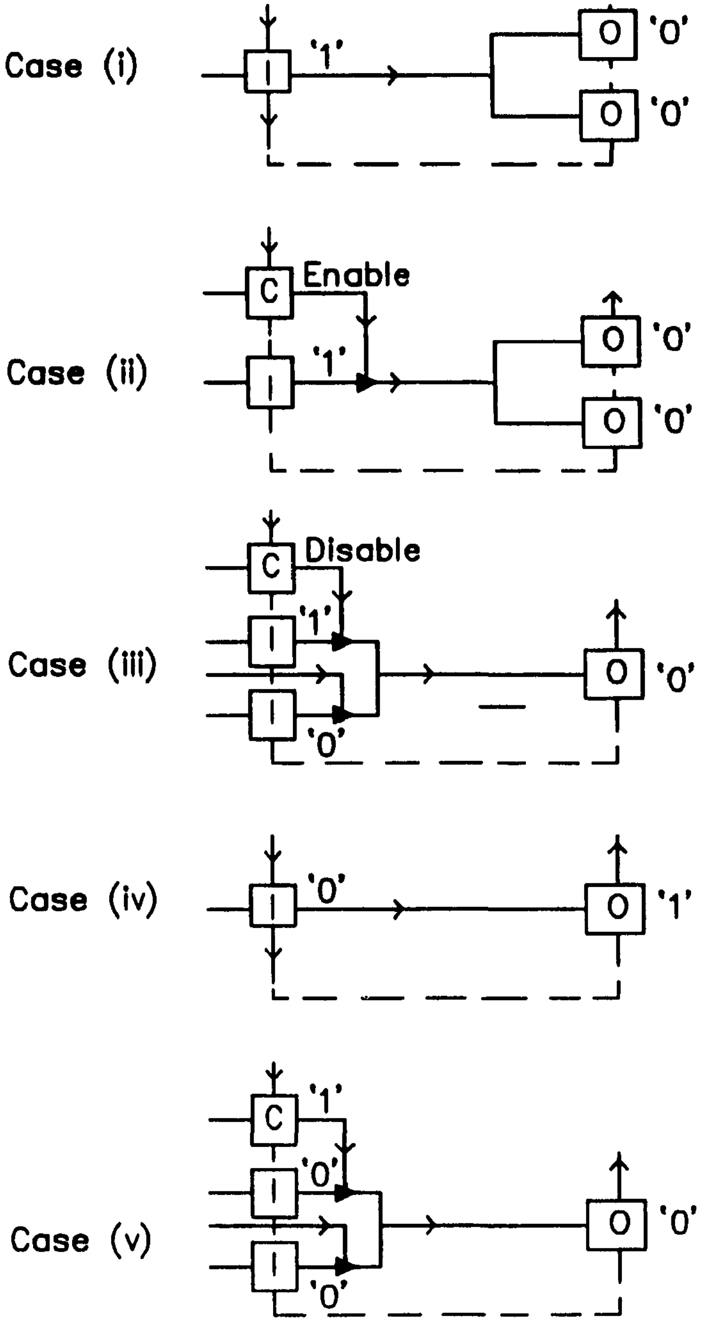


Figure 3.3 Walking sequence loading.

(i) In an input scan cell with no tri-state capability.

(ii) In an input scan cell (with tri-state capability) which is enabled.

(iii) In an input scan cell (with tri-state capability) which is disabled.

(iv) In an output scan cell.

(v) In a control cell.

Number of '1's received due to applying this vector and shifting out the response in the above five cases are as follows :

Case (i) : $m + 1$. where m is the number of output scan cells connected to the Inet fed by the input scan cell loaded with '1'.

Case (ii) : $m + 1$. where m is the number of output scan cells connected to the Inet fed by the input scan cell loaded with '1'.

Case (iii) : 1. The input scan cell is disabled. Thus. the loaded '1' will not be applied to any Inet. The single '1' shifted in will be simply shifted out.

Case (iv) : 0. The '1' loaded in the output scan cell will be 'lost' due to the '0' applied from the input scan cell connected to it.

Case (iv) : 1. The loaded '1' in the control cell will not be applied to any Inet. Thus. the '1' will be shifted out.

Therefore. total number of 1's shifted out due to application of N vectors of the walking one sequence

$$= \sum_{i=1}^{n}(m_i + 1) + t + c$$

where.

m_i = number of output scan cells connected to i-th Inet

n = number of Inets under test

t = number of disabled input scan cells

c = number of control cells

$$= \sum_{i=1}^{n} m_i + (n + t) + c$$

= (number of output scan cells) + (number of input scan cells) + (number of control cells)
$= N$

In the case of a faulty Inet in the system, the total number of '1's is increased or decreased depending upon the type of fault. SA-1 and OR short increase the count whereas SA-0 and AND short decrease the count. This scheme can detect any single short or SA fault in the system by counting the number of '1's shifted out, as proved in the following.

Theorem 3.2 *Every single short and SA fault in a system of n Inets can be detected by counting the number of '1's shifted out of the boundary scan chain due to the application of a walking one sequence.*

Proof : Let us suppose the i-th Inet has x_i input scan cells and y_i output scan cells connected to it. Therefore, the total number of scan cells in the i-th Inet are

$$v_i = x_i + y_i$$

In the 1's counting response compaction scheme, the scan cells which contain '1' after the application of each vector are counted.

It follows from Lemma 3.1 that the number of scan cells counted in the fault-free case are

$$CNT = N = \sum_{i=1}^{n} v_i + c$$

Faulty Conditions :

(i) For SA-1 :

The output scan cells corresponding to the faulty Inet are counted for every vector applied. Thus, when counted N times for N vectors, the total number of scan cells counted corresponding to the i-th faulty Inet are

$$v_i^F = x_i + N.y_i$$

For each faulty Inet.

$$v_i^F - v_i = (N-1)y_i$$

more scan cells are counted. Thus, for j Inets SA-1, the total count,

$$CNT_{F1} = \sum_{i=1}^{n} v_i + \sum_{q=1}^{j}(N-1)y_q + c$$

Therefore, $CNT_{F1} > CNT$.

(ii) For SA-0 :

The output scan cells corresponding to the faulty Inets are never counted. Thus, for each faulty Inet, y_i scan cells are never counted. For j Inets SA-0,

$$CNT_{F2} = \sum_{i=1}^{n} v_i - \sum_{q=1}^{j} y_q + c$$

Therefore, $CNT_{F2} < CNT$.

(iii) For OR short :

In this scheme, when m Inets are shorted together, each output scan cell of these faulty Inets is counted m times instead of once.

Number of input scan cells in m faulty Inets

$$X = \sum_{l=1}^{m} x_l$$

Number of output scan cells in m faulty Inets

$$Y = \sum_{l=1}^{m} y_l$$

In the fault-free case, the number of scan cells counted in these m Inets

$$= (x_1 + y_1) + (x_2 + y_2) + ... + (x_m + y_m)$$

$$= X + Y$$

In the faulty case, the count of input scan cells remain the same. The count of output scan cells

$$= my_1 + my_2 + ... + my_m = mY$$

Thus, the count is increased by

$$(m - 1)Y$$

Thus, for r OR shorts,

$$CNT_{F3} = \sum_{i=1}^{n} v_i + \sum_{s=1}^{r} (m_s - 1)Y_s + c$$

Therefore, $CNT_{F3} > CNT$.

Similarly, it can be shown that for AND short,

$$CNT_{F4} = \sum_{i=1}^{n} v_i - \sum_{s=1}^{r} Y_s + c$$

Therefore, $CNT_{F4} < CNT$.

Thus, it is shown that

$$CNT_F^- < CNT < CNT_F^+$$

where

CNT_F^- is the decrease in count

CNT_F^+ is the increase in count.

Therefore, any single fault is guaranteed to be detected.

[QED].

Multiple faults in this scheme can mask each other. For example, if the count increased by an OR short is exactly equal to the count decreased by an AND short, these two shorts mask each other. Similarly, a SA-1 fault can be masked by a SA-0 fault. In general, there are two types of faults. OR short and SA-1 are the increasing count type whereas AND short and SA-0 belong to the decreasing count type. As long as multiple faults belong to the same type, there is no masking. But multiple faults from different types can mask each other.

This scheme requires N bits to be shifted in and shifted out for each vector. Thus, for N vectors, the time complexity of the scheme is $O(N^2)$. The response analyzer is a $\lceil log_2 N \rceil$ bit 1's counter.

In this scheme, test vector generation and loading are independent of the order of the I/O scan cells in the boundary scan chain. Moreover, structural information about the Inets and expected response storage are not required for fault detection. Fault detection is as simple as counting the number of 1's shifted out of the scan chain. The simplicity of test vector generation and fault detection makes this scheme suitable for on-board BIST implementation. This scheme guarantees to detect every possible single short and SA fault. However, for multiple faults there is a chance of masking.

(ii) Walking Sequence Scheme using Local Response Compaction

It was mentioned earlier in this section that one main advantage of using a walking sequence is that only a single vector is generated and loaded through the boundary scan chain. By gradually shifting this vector within the boundary scan chain, the remaining vectors can be obtained. However, the scheme described above does not utilize this property of the walking sequence. Consequently, the time complexity of the scheme is $O(N^2)$. In the following, a detection scheme is proposed using the walking sequence which utilizes the internal shifting property of the walking sequence to improve the test time. As a result, the scheme requires local response compaction capability in each output scan cell to improve response analysis time.

This scheme requires the application of both walking one and zero sequences. The boundary scan chain is loaded with an N bit vector $W1(W0)$. [$W1$: 100...0 where '1' is loaded in the boundary scan cell closest to the boundary scan in edge of the PCB under test and $W0$: 011...1 where '0' is loaded in the boundary scan cell closest to the boundary scan in edge of the PCB under test.] It is assumed that the number of output scan cell in the boundary scan chain is M. The scheme is described in the following.

```
Begin
    Begin
    Repeat for $W1$ and $W0$
        Begin
        Load $W1(W0)$ in the $N$ bit boundary scan path
```

```
                Apply $W1(W0)$
                Collect and compact output at each output scan cell
                    Begin
                    Repeat $(N-1)$ times
                        Begin
                        Shift $W1(W0)$ one bit to the right (move a '0'/'1' in the leftmost cell)
                        Apply shifted $W1(W0)$
                        Collect and compact response at each output scan cell
                        End
                    End
                Unload the $N$ bit scan path with compacted response in a 1's counter
                End
        End
    Detect based on count in the 1's counter
        If count $= N + M$ Then fault-free
        Else faulty
    End
```

Example Inet structures are shown in Figure 3.4. Test vectors required for these structures are given in Table 3.9.

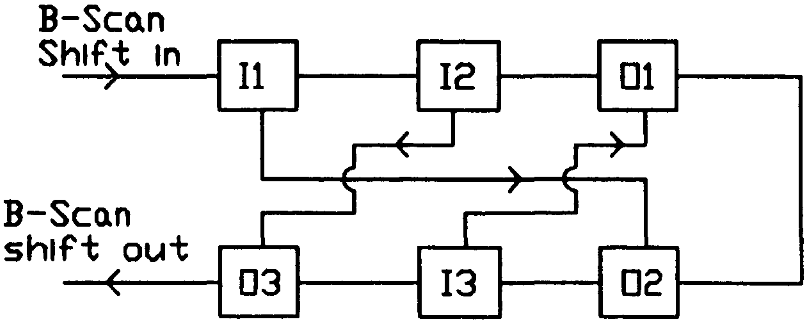


Figure 3.4 Inets under test.

Response Compaction :

In this scheme, output responses are compacted locally within each output scan cell. Response compaction circuitry needs to be added to the output scan cell proposed by JTAG [JTAG 88] for this purpose. Such an extended output scan cell with walking sequence response compaction capability is shown in Figure 3.5. Block diagram of Figure 3.5 is given in Figure 3.6

Scan Cells	Vectors					
	V_1	V_2	V_3	V_4	V_5	V_6
I_1	1	0	0	0	0	0
I_2	0	1	0	0	0	0
O_1	0	0	1	0	0	0
O_2	0	0	0	1	0	0
I_3	0	0	0	0	1	0
O_3	0	0	0	0	0	1

Table 3.9 Walking one sequence.

for the convenience of discussion. In this extended cell, the single bit register $R1$ is used only for loading and shifting of the walking sequence. The second single bit register $R2$ together with the two-input XOR gate form a parity checking circuitry for response compaction.

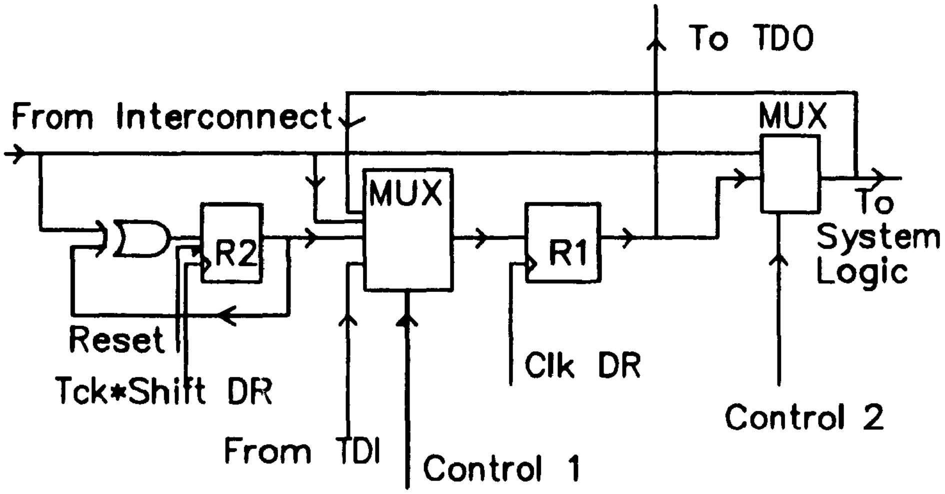


Figure 3.5 Extended output scan cell for walking sequence response compaction.

Register $R2$ is initialized to '0'. The first output bit coming from an Inet due to the application of the first vector of the walking sequence is XORed with the content of $R2$. The result is stored back in the register $R2$ and XORed with the next coming bit. This procedure is repeated N times for the N vectors and finally a one-bit compacted response is obtained in $R2$.

This response compaction process can be represented by a two-input XOR tree, as shown in Figure 3.7. Two inputs are fed to each level of this tree. One input coming from outside

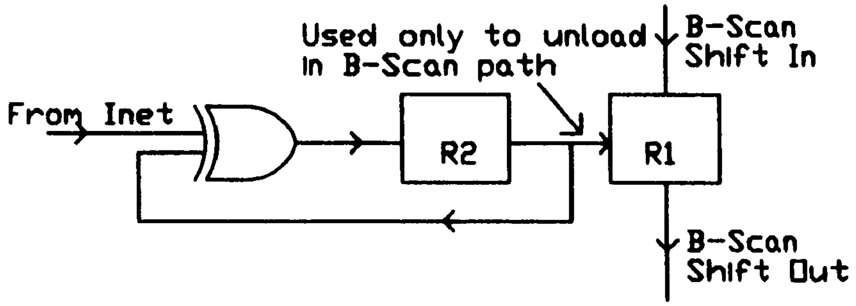


Figure 3.6 Block diagram of extended output scan cell for walking sequence response compaction.

and the second input coming from the previous level. This XOR tree represents the process taking place in every output scan cell. There are N inputs to this XOR tree. It is assumed in the following analysis that N is even. The final response ('1'/'0') depends upon the number of '1's fed to the tree. For an odd number of '1's the response is '1', and for an even number of '1's the response is '0'.

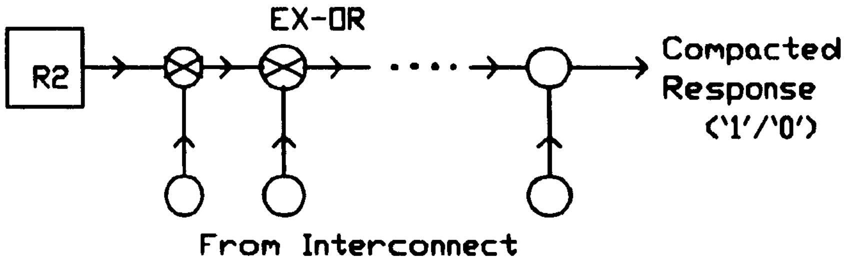


Figure 3.7 XOR tree

For a walking one sequence with N bits, each Inet is applied a single '1'. For a walking zero sequence with N bits, each Inet is applied $(N - 1)$ '1's. In both cases, the XOR tree is fed an odd number of '1's. Therefore, the fault free compacted response is '1' for both walking sequences.

For a walking one sequence, SA fault feeds an even number of '1's to the tree (zero '1's for SA-0 and N '1's for SA-1). Also, any AND short and OR short among the even number of Inets feed an even number of '1's to the tree. For all these faults, the compacted response is '0'. However, for OR short among an odd number of Inets, an odd number of '1's are fed to the XOR tree of each faulty Inet. Therefore, the compacted response is '1' (same as fault-free compacted response). But, in this case the walking zero sequence feeds N '1's to the XOR tree. Therefore, the compacted response for walking zero sequence is '0'.

Table 3.10 shows the compacted responses obtained by applying both the walking one and zero sequences. It can be seen from this table that the fault-free compacted bit-pair (for walking one and zero sequences) is different from any faulty compacted bit-pair.

Type of Fault	Compacted response for	
	Walking one	Walking zero
Fault-free	1	1
OR short (odd no. of Inets)	1	0
OR short (even no. of Inets)	0	0
AND short (odd no. of Inets)	0	1
AND short (even no. of Inets)	0	0
SA-1	0	0
SA-0	0	0

Table 3.10 Compacted responses due to walking sequences.

Detection :

For the detection of shorts and SAs, compacted responses in the boundary scan chain are shifted out and compacted once more. The response compactor in this stage is a 1's counter. After applying N vectors of the walking one sequence, the single-bit compacted response from each output scan cell is shifted out into the response compactor. The same procedure is repeated for the walking zero sequence.

Theorem 3.3 : *Every single and multiple short and SA fault in a system of n Inets can be detected by counting the number of 1's shifted out of the scan chain due to the application of both the walking one and zero sequences and compaction of the responses in each output scan cell for each of these two sequences.*

Proof : Let us suppose there are M output scan cells in the boundary scan chain of N scan cells.

Therefore, in the fault-free case, the expected count in the 1's counter

$$CNT = 2M + r$$

where $r =$ (number of '1's in the input and control scan cells due to walking one sequence after $N-1$ internal shifts) + (number of '1's in the input and the control scan cells due to walking zero sequence after $N-1$ internal shifts).

In this scheme, the control cells use by-pass capability so that test vectors can be generated by internal shifting of the walking sequences. The '1'('0') contained in every control cell after $(N-1)$ internal shifts of the walking sequences are shifted out in the response compactor.

Note that a single '1' and $(N-1)$ '0's are loaded in the scan chain because of the walking one sequence. After $(N-1)$ internal shifts, the single '1' reaches the last scan cell (i.e. the scan cell closest to the scan-out edge of the PCB under test). If this last scan cell is an input (or, control) scan cell, the '1' in this cell is shifted out of the scan chain. Each of the remaining input (control) scan cells contains a '0'. If this last scan cell is an output scan cell, the '1' in this cell is 'destroyed' by the compacted response contained in this cell. In this case, every input (control) scan cell contains a '0'.

For a walking zero sequence, a single '0' and $(N-1)$ '1's are loaded in the scan chain The '0' reaches the last scan cell after $(N-1)$ internal shifts. If this cell is an input (control) scan cell, then each of the $(N-M-1)$ remaining input (and control) scan cells contains a '1'. These '1's are shifted out of the scan chain. If the last cell is an output scan cell then each of the $(N-M)$ input and control scan cells contains a '1'.

Therefore, following the argument above, foı the last scan cell to be input (control) scan cell,

$$r = 1 + (N - M - 1)$$

$$= (N - M).$$

For the last scan cell to be output scan cell,

$$r = 0 + (N - M)$$

$$= (N - M).$$

Therefore, in both the cases, total count,

$$CNT = 2M + (N - M)$$

$$= (N + M).$$

It can be seen from Table 3.10 that the number of '1's obtained from each output scan cell of every Inet is two (one for the walking one sequence and one for the walking zero sequence) in the fault-free case. But, for any short or SA fault on any Inet, as shown in Table 3.10, the number of '1's from each output scan cell of the faulty Inet is reduced to a single '1' or zero '1'. In this response compaction scheme, the faults are unidirectional. Every fault reduces the total count. Thus, the faults cannot mask each other.

Therefore, for any type and number of faults, the faulty count

$CNT_F < N + M$

Thus,

$CNT_F < CNT$

Therefore, every short and SA fault is guaranteed to be detected.

[QED].

In this scheme only the first vector of each sequence (walking one and walking zero) is shifted into the boundary scan chain. This requires N bits to be shifted in for each sequence. Also, after the response compaction of each sequence, N bits are shifted out for detection. Therefore, the time complexity of this scheme is $O(N)$.

Like the previous walking sequence scheme, test vector generation and loading in this scheme are independent of the order of the I/O scan cells in the scan chain. Moreover, this scheme utilizes the shifting property of the walking sequence. The boundary scan chain is used in this scheme for the loading and generation of test vectors. Test vector generation is time-efficient since each internal shift within the boundary scan chain generates a new vector. The output responses are collected and compacted locally at each output scan cell. After compaction the number of '1's coming out of the boundary scan chain are counted for fault detection. Therefore, detection is simple and requires no information about the topology of the Inets. Also, multiple faults do not mask each other. These properties make the scheme very attractive for BIST implementation.

3.5.2 Diagnosis of Shorts and SAs

Three different schemes are proposed in the following regarding the diagnosis of shorts and SAs. Two of these schemes are based on order independent vector set described in Section 3.5.1.1. Output responses are compacted in both schemes to simplify the diagnosis process. However, in one of these schemes compaction is done outside the boundary scan chain and in the other scheme response compaction is local to each output scan cell. The third scheme proposed here is based on the walking sequence and is similar to the walking sequence detection scheme. All three schemes are suitable for BIST implementation. However, the walking sequence scheme is the most attractive because of its internal shifting property.

3.5.2.1 Order Independent Vector Set Scheme

As described in section 3.2.1.1, True/Complement Vector Set Scheme proposed by Wagner [Wagner 87] uses $2\lceil log_2(n+2)\rceil$ vectors to diagnose every possible short and SA fault in n Inets. However, for a scan chain N cells long, $2\lceil log_2(n+2)\rceil$ vectors are to be reformatted as described in Section 3.3. The diagnosis scheme can be made boundary scan chain order independent and the need for reformatting can be avoided by using $2\lceil log_2 N\rceil$ vectors. The $\lceil log_2 N\rceil$ vectors are similar to $\lceil log_2(N+2)\rceil$ vectors described in Section 3.5.1.1. For diagnosis, all '0' and all '1' IBSs can be included as valid assignments. Thus, $\lceil log_2 N\rceil$ vectors are required instead of $\lceil log_2(N+2)\rceil$ vectors. These $\lceil log_2 N\rceil$ vectors and their complements form the complete vector set for the proposed diagnosis scheme.

Table 3.11 shows the six vectors for $N = 6$. These vectors are applied in pairs. One vector (say V_1) is applied from the first set of $\lceil log_2 N\rceil$ vectors followed by its complement (say V_1') from the complementary set. This is repeated until all the vectors are applied. IBSs also are treated in pairs. In each pair there is a '0' and a '1' (because the components of the pair are coming from two complementary vectors). So in every non-faulty bit-pair, there is always a '0' followed by a '1' or a '1' followed by a '0'. For a SA-1 (or, a SA-0), the bit-pair are changed to two '1's (or, two '0's).

The effect of a short on such a bit-pair is discussed in the following. The input bit-pairs applied to two Inets can have four possible combinations, shown in Table 3.12. In the first two cases, C1 and C2, the inputs to the two Inets are the same. Therefore, no change can be observed due to a short. For set C3, if the two Inets are shorted, the outputs are changed to

Scan Cells	True set			Complement set		
	V_3	V_2	V_1	V_3'	V_2'	V_1'
N_1	0	0	0	1	1	1
N_2	0	0	1	1	1	0
N_3	0	1	0	1	0	1
N_4	0	1	1	1	0	0
N_5	1	0	0	0	1	1
N_6	1	0	1	0	1	0

Table 3.11 $2log_2N$ vectors for shorts and SAs diagnosis.

either '00' or '11'. The same is true for set C4. Therefore, if any two Inets differ in input bit-pair combination (and they do differ according to the Minimal-size Test Set Scheme described in Section 3.1.1), the short can be diagnosed as a pair of '0's (or '1's) at the output.

Inets	C1	C2	C3	C4
n_1	01	10	01	10
n_2	01	10	10	01

Table 3.12 Possible combinations of input bit-pairs.

Implementation Issues :

In this scheme, response compaction can be done locally or externally. As mentioned in Section 2.3.2, local response compaction is done by comparing, for each pair of input vectors, the pair of output bits obtained from each Inet within the associated output scan cell(s). For external response compaction, the output bits are shifted out of the scan chain, stored in an external register and compared outside the scan chain. Notice, however, that both techniques can be implemented in a board level BIST environment.

(i) Local Response Compaction :

In this scheme, response compaction takes place within each output scan cell. Therefore, response compaction circuitry must to be added to each output scan cell. Such an extended cell is shown in Figure 3.8. Figure 3.9 shows a block diagram of Figure 3.8. Two one-bit registers $R1$ and $R2$ are needed to store the output bit-pair. A two-input comparator is added to this register pair to compare the bit-pairs. Finally, a single bit register $R3$ is used to store and shift out the comparator output.

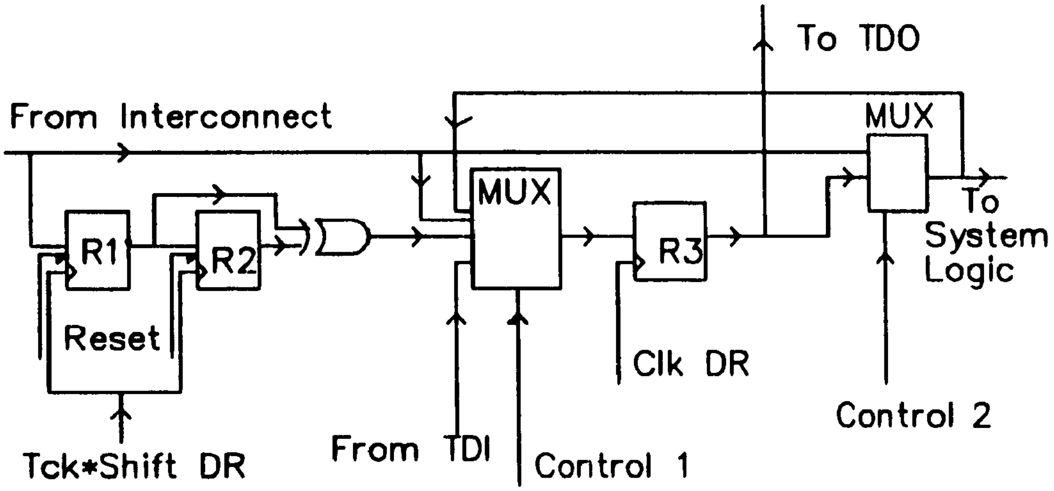


Figure 3.8 Extended output scan cell for local response compaction

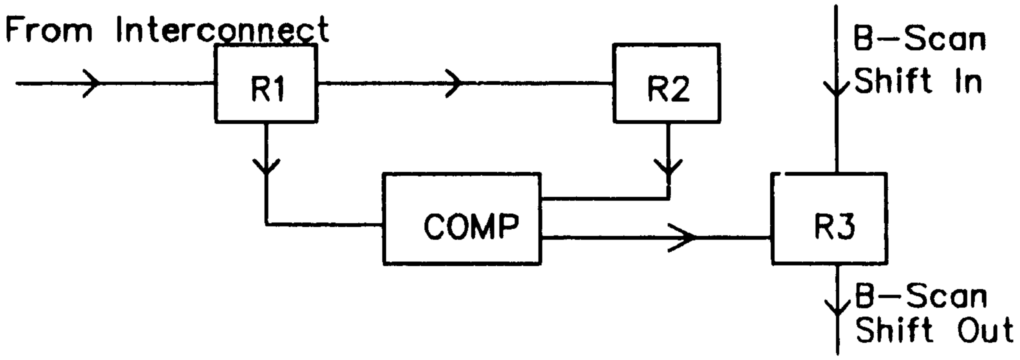


Figure 3.9 Block diagram of extended output scan cell for local response compaction

The first input vector is loaded through the scan chain and applied to the Inets. The output response of each Inet is stored in the one-bit register of the associated output scan cell. Then the complementary input vector is loaded and applied, and the responses are stored in the second one-bit registers. These two bits are then compared in the comparator, and the output of the comparator is stored in $R3$. These outputs are shifted out for diagnosis. This procedure is repeated for all the $\lceil log_2N \rceil$ pairs of vectors in the set.

The procedure is as follows :

```
Begin
Repeat logN times
    Begin
    Load a vector (which is not previously loaded) from $logN$ vectors in $N$ bit scan path
    Apply loaded vector
    Collect response at each output scan cell
    Repeat above three steps for the corresponding complement vector
    Compare responses at each output scan cell
    Unload the $N$ bit scan path with compared response
    Diagnose based on compared output
        If comparison 'matches' Then faulty
        Else non-faulty
    End
End
```

In this scheme, the N bits are shifted in through the scan chain for each vector and the N bits of comparator results are shifted out for each pair of vectors.

Thus, for local response compaction,

Loading Time, $T_l = N.2\lceil log_2N \rceil = O(N.log_2N)$

Shift-Out Time, $T_o = N.\lceil log_2N \rceil = O(N.log_2N)$

(ii) External Response Compaction :

The arrangement for external response compaction is shown in Figure 3.10. The output response of the first vector is shifted out of the scan chain and loaded in an N bit shift registe. The complementaiy vector is then applied and while this response is shifted out, it is compared with the response stored in the external register, bit by bit, through a two-input comparator. This procedure is repeated for every pair of input vectors. External response compaction requires an N bit shift register and a two-input comparator external to the scan chain. No extension of the I/O scan cells is necessary.

The scheme is described as follows :

```
Begin
Repeat logN times
    Begin
```

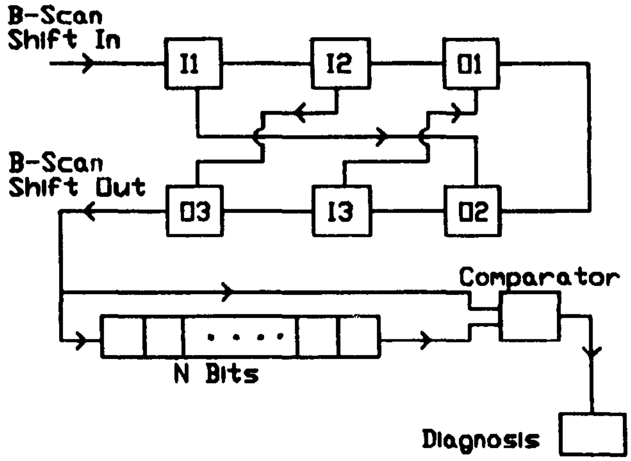


Figure 3.10 Arrangements for external diagnosis.

```
        Load a vector (which is not previously loaded) from $logN$ vectors in $N$ bit scan path
        Apply loaded vector
        Collect response at each output scan cell
        Unload the $N$ bit scan path in $N$ bit external register
        Load complementary vector
        Apply loaded vector
        Collect response at each output scan cell
        Shift and compare response with contents of the $N$ bit register
        Diagnose based on comparison
            If comparison 'matches' Then faulty
            Else non-faulty
        End
    End
```

For each input vector, N bits are loaded and N bits of responses are shifted out through the scan chain.

Thus, for external response compaction,

Loading Time, $T_l = N.2\lceil log_2 N\rceil = O(N.log_2 N)$.

Shift-out Time, $T_o = N.2\lceil log_2 N\rceil = O(N.log_2 N)$.

The order of complexity is the same for local and external response compaction. For local response compaction, when the comparator output bits are coming out of the scan chain, one has to distinguish between the bits coming from input scan cells and those coming from output scan cells. Therefore, the order of I/O scan cells should be known. External response compaction does not require any such information. The fault-free comparator output is a '1' independent of whether the bits are coming from input scan cells or from output scan cells. However, an N bit external register and a two-input comparator are required for external response compaction.

In order independent vector set scheme, test generation and loading are independent of the order of the I/O scan cells. However, each vector must be loaded individually through the boundary scan chain like the existing external schemes. The scheme can diagnose every possible short and SA fault. Diagnosis is independent of the structural complexity of the Inets and is local for each Inet, which means that the compacted bits obtained from each Inet are sufficient to identify that Inet as fault-free or faulty. Response compaction can be done locally or externally. Local response compaction requires the addition of response compaction circuitry to each of the output scan cells. External response compaction requires no such extension. Therefore, between the two implementations, the external diagnosis scheme is more advantageous.

The order independent vector set scheme is suitable for BIST implementation, for properties like simple test vector generation and for structure independent fault diagnosis.

3.5.2.2 Walking Sequence Scheme

The walking sequence detection scheme described in Section 3.5.1.2 can also be used for the diagnosis of shorts and SAs. For diagnosis, however, the second stage of response compaction in the 1's counter is not necessary. From the first stage of response compaction in the output scan cells, a pair of compacted response bits corresponding to each of these cells are obtained. One of these bits corresponds to the walking one sequence response compaction and the other bit corresponds to the walking zero sequence response compaction. It is shown in Table 3 10 that the fault-free compacted bit-pair is '11'. However, for any short or SA fault, the response bit pair is shown to be changed to '01', '10' or '00'. Thus, based on the compacted responses shifted out, Inet corresponding to each output scan cell can be declared faulty or fault-free.

The scheme is described in the following :

```
Begin
    Begin
    Repeat for $W1$ and $W0$
        Begin
        Load $W1(W0)$ in the $N$ bit boundary scan path
        Apply $W1(W0)$
        Collect and compact output at each output scan cell
            Begin
            Repeat $(N - 1)$ times
                Begin
                Shift $W1(W0)$ one bit to the right (move a '0'('1') in the leftmost cell)
                Apply shifted $W1(W0)$
                Collect and compact response at each output scan cell
                End
            End
        Unload the $N$ bit scan path with compacted response
        End
    End
Diagnose based on compacted bit-pair from each output scan cell
    If bit-pair = '11' Then fault-free
    Else faulty
End
```

W1 and W0 are as defined in Section 3.5.1.2.

Time Requirement :

In this scheme. complete diagnosis requires $2N$ vectors. For N vectors of the walking one sequence. only the first vector is loaded and shifted $(N - 1)$ times. This requires N bits of loading and $(N - 1)$ shifts. The walking zero sequence requires the same operations Therefore. altogether. there are $2N$ bits to be loaded and $(2N-2)$ internal shifts. For response compaction. N bits of compacted responses are shifted out twice (once after every N vectors are applied). So. $2N$ shift-outs are necessary. The time complexity is $O(N)$

A comparison of the time requirements of the order independent vector set scheme and of the walking sequence scheme is shown in the following.

For the order independent vector set scheme (local response compaction) ·

Loading time $T_l = 2Nlog_2N$

Test application time $T_a = 2log_2N$

Shifting-out time $T_o = Nlog_2N$

For walking sequence scheme :

$T_l = 2N$

$T_a = 2N$

$T_o = 2N$

$2Nlog_2N + 2log_2N + Nlog_2N > 2N + 2N + 2N$

$\approx$ for, $N > 4$

Therefore, although the number of vectors applied is fewer in the order independent vector set diagnosis scheme, the walking sequence diagnosis scheme (for $N > 4$) requires less time.

The walking sequence diagnosis scheme does local response compaction with the time complexity of $O(N)$. An external response compaction implementation is possible with the time complexity of $O(N^2)$. In this latter implementation, every vector must be shifted in and every response must be shifted out individually. The response should be compacted outside the boundary scan chain for diagnosis. This implementation requires no extension of the boundary scan cells. However, the time requirement is greater than the local response compaction scheme.

The walking sequence diagnosis scheme can diagnose every possible short and SA fault. Test generation is independent of the order of the I/O scan cells. Diagnosis is independent of the structural complexity of the Inets and is local for each Inet like the order independent vector set scheme. But the walking sequence scheme (with local response compaction) is time efficient compared to the order independent vector set scheme for any practical purpose.

Similar to the walking sequence detection scheme, the walking sequence diagnosis scheme is the most suitable for on-board BIST implementation among the existing and the proposed diagnosis schemes.

3.6 Summary of Proposed Detection and Diagnosis Schemes

(i) Detection Schemes

(a) Order Independent Vector Set Scheme :

- Number of vectors applied $\lceil log_2(N+2)\rceil$
- Vector loading scan chain order independent
- Simulation to obtain expected response
- Expected response storage requirement $N.\lceil log_2(N+2)\rceil$ bits
- Detects every short and SA
- Detection Inet structure dependent
- Time complexity $O(NlogN)$

(b) Walking Sequence Scheme using External Response Compaction :

- Number of vectors applied N
- Vector loading scan chain order independent
- No simulation to obtain expected response
- No expected response storage requirement
- Detects every single short and SA. Chances of masking for multiple faults
- Detection Inet structure dependent
- Time complexity $O(N^2)$

(c) Walking Sequence Scheme using Local Response Compaction :

- Number of vectors applied $2N$
- Vector loading scan chain order independent
- No simulation to obtain expected response

- No expected response storage requirement
- Detects every short and SA (no masking)
- Detection Inet structure dependent
- Extension of output scan cells for response compaction. Extra hardware required : one single-bit register and one two-input XOR gate per output scan cell
- Time complexity $O(N)$

(ii) Diagnosis Schemes

(a) Order Independent Vector Set Scheme (Local Response Compaction) :

- Number of vectors applied $2.\lceil log_2 N\rceil$
- Vector loading scan chain order independent
- No simulation to obtain expected response
- No expected response storage requirement
- Diagnoses every short and SA (no masking)
- Diagnosis Inet structure dependent
- Extension of output scan cells for response compaction. Extra hardware required : Two single-bit registers and a two-input comparator per input scan cell
- Time complexity $O(NlogN)$

(b) Order Independent Vector Set Scheme (External Response Compaction) :

- Number of vectors applied $2.\lceil log_2 N\rceil$
- Vector loading scan chain order independent
- No simulation to obtain expected response
- No expected response storage requirement
- Diagnoses every short and SA (no masking)

- Diagnosis Inet structure dependent
- No extension of output scan cells for response compaction
- Time complexity $O(NlogN)$

(c) Walking Sequence Scheme using Local Response Compaction :

- Number of vectors applied $2N$
- Vector loading scan chain order independent
- No simulation to obtain expected response
- No expected response storage requirement
- Diagnoses every short and SA (no masking)
- Diagnosis Inet structure dependent
- Extension of output scan cells for response compaction. Extra hardware required : One single-bit register and a two-input XOR gate per output scan cell
- Time complexity $O(N)$

3.7 Comparison of External Schemes and BIST Schemes

(i) Test Vector Generation :

In external schemes, test vectors are generated based on the number of Inets. Vectors are reformatted according to the order of the I/O scan cells before loading. In the proposed BIST schemes, test vector generation and loading are independent of the order of the I/O scan cells. Test generation hardware is simple so that on-board BIST is feasible.

(ii) Test Vector Loading :

Each vector is loaded individually through the scan chain in the existing schemes. In the proposed schemes, test vectors are generated by internal shifting of the first vector loaded. This technique leads to time-efficient test schemes. However, in some of the proposed schemes, test vectors are loaded one at a time. For those schemes, the time requirement is similar to the external schemes' time requirement.

(iii) Output Response Analysis :

In the external schemes, each response is shifted out and compared with the stored expected response for detection and diagnosis. Expected response depends upon the structure of the Inets under test. Thus, detection and diagnosis are structure or circuit dependent. Also, storage is required for expected response.

Detection and diagnosis are independent of the structural complexity of the Inets in the proposed schemes (except for the order independent vector set detection scheme). The fault-free compacted response is the same, irrespective of the structure of the Inet and what is applied from the input end of the Inet. Therefore, simulation is not required to obtain circuit independent expected response. This also removes the need for space to store the expected response. Arrangement, like counting the number of '1's('0's) coming out of the scan chain or monitoring the shifted out contents of the scan chain in the TDC, perform detection and diagnosis. These simple techniques are suitable for on-board BIST implementation.

(iv) Extension of Scan Cells :

The boundary scan chain is used to load test vectors and shift out responses in the external test schemes. These operations do not require any extension of the scan cells proposed in the JTAG design [JTAG 88]. In some of the proposed BIST schemes, output responses are compacted within each output scan cell to reduce shift out time. Response compaction circuitry is added to each output scan cell. Moreover, the control cells are extended to have by-pass capability. Therefore, the advantages of those BIST schemes are obtained at the cost of additional hardware. However, some implementations of the BIST schemes do not require local response compaction, hence, extensions of the I/O scan cells is not necessary.

Chapter 4 Testing of Open Faults

As mentioned in Chapter 2. open faults affect the Inets differently than shorts and SA faults. That is why testing open faults is treated in a separate chapter in this dissertation.

Testing of open faults is divided into two categories. The first category includes the detection of Inet open faults. In these tests. Inets are treated as units under test. The second category involves the diagnosis of open faults. Again. there are two types of diagnosis. The first type. called *Inet Open Fault Diagnosis*. treats the Inets as the units under test. All the faulty Inets are identified in this test. The other type of diagnosis is called *Inet Conductor Open Fault Diagnosis*. These tests are performed to diagnose the faulty conductors within each Inet. Inet conductors are the units under test in this latter approach. These different types of open faults testing are treated in detail in the following.

Looking back. it was mentioned in Chapter 2 that input vectors must be applied from every input scan cell (by enabling it with appropriate control settings) of a multi-driver Inet. one at a time. This is necessary in order to sensitize every conductor in the Inet. Therefore. if the maximum degree of an Inet. among all the Inets under test. is P. the input vector set should be applied P times with P different control settings. This is applicable to all the detection and diagnosis schemes proposed in the following.

4.1 Detection of Inet Open Faults

Three different schemes for detecting Inet open faults are proposed in the following. In the first scheme, test vectors are generated based on the order of the I/O scan cells. Detection is as simple as counting the '1's ('0's) coming out of the scan chain. The second and third schemes are the order independent test set scheme and the walking sequence scheme proposed in Chapter 3, respectively.

4.1.1 LI/CLI Vector Scheme

Open faults are tested by checking for a conducting path from each input scan cell to all the output scan cells in an Inet. To do this, each output scan cell is initialized to a known logic value. The opposite logic value is applied from the input scan cells. In the fault-free case, the values in output scan cells are changed through the conducting paths.

For the detection of open faults, a vector is shifted in through the scan chain. Every output scan cell of every Inet under test is initialized to '0' and every input cell is loaded with '1' using this vector. This vector initializes the output cells and loads the input cells at the same time. Such a vector is a *load-initialize* (LI) vector. This LI vector is then applied and the response is shifted out.

In the fault-free case, the content of every output scan cell is changed from the initialized value of '0' to '1'. Every input scan cell is loaded with '1'. Also, it is assumed that the control cells have by-pass capability and the by-pass bit in each control cell is '1'. Therefore, the number of '1's shifted out of the scan chain is exactly N in the fault-free case. In the case of an open fault in any Inet, the '1' applied from its input scan cell is not received at the output scan cell(s). The output end scan cell(s) corresponding to the faulty Inet retains the initialized value ('0'). Therefore, the number of '1's shifted out of the scan chain is less than N in the presence of an open fault.

As mentioned in Chapter 2, open test requires that each input scan cell driver should drive the Inet with both '0' and '1' and that these values are correctly received by the output scan cell receiver(s). The LI vector described above tests for only the logic value of '1'. Therefore, the same procedure must be repeated with the complementary LI (i.e., CLI) vector where the input scan cells are loaded with '0's and the output scan cells are initialized with '1's. Also, the LI and CLI vectors must be applied P times, with P different control settings, as mentioned at the beginning of this chapter.

Lemma 4.1 *Every open fault in a system of Inets (with maximum degree P) under test can be detected by applying a LI vector and a CLI vector, P times each, with P different control settings.*

Proof : Open fault in every Inet can be tested in parallel since an open fault affects only a single Inet. It is shown in the above discussion that, if an LI vector is applied P times with P different control settings, then every conductor in every Inet is sensitized with a logic '1'. Similarly, a CLI vector applied P times sensitizes these conductors with logic '0'. These applications of the LI and CLI vectors are sufficient to sensitize every conductor in every Inet with both logic values '1' and '0'. These sensitized values are observable from one or more of the output scan cells of each Inet. Therefore, P applications of LI and CLI vectors are sufficient to detect every open fault in every Inet (with maximum degree P) under test.

[QED].

The scheme is described as follows :

```
Begin
Repeat for LI and CLI vectors
    Begin
    Repeat $P$ times for $P$ different control settings
        Begin
        Load the LI(CLI) vector in the $N$ bit scan path
        Apply loaded vector
        Collect response at each output scan cell
        Unload the $N$ bit scan path with output responses in 1's(0's) counter
        End
    End
Detect based on count
    If count $= PN$ Then fault-free
    Else faulty
End
```

The LI(CLI) vector in this scheme is generated based on the order of the I/O scan cells Each vector is loaded individually - P times with P different control settings - through the boundary scan chain. Therefore, the time complexity of the scheme is $O(PN)$. Test vector generation and loading depend upon the structural information of the Inets This scheme can detect every open fault in Inets. Detection is done outside the boundary scan chain. Therefore, no extension of the standard output scan cell is necessary to implement the scheme. Detection is based on a simple count of 1's(0's) coming out of the boundary scan chain and is independent of the structural complexity of the Inets.

4.1.2 Order Independent Vector Set Scheme

The scheme described in Section 3.5.1.1 for shorts and SAs detection is also applicable for the detection of open faults. In this scheme, $\lceil log_2(N+2)\rceil$ vectors are applied such that the IBS applied to each Inet contains both '1's and '0's. If any Inet has an open fault, the '1's ('0's) applied from the input scan cell of this Inet do not reach the output scan cell(s). The output scan cell(s) retains the initialized value. Therefore, by observing the contents of each output scan cell, Inet open faults can be detected.

4.1.3 Walking Sequence Scheme

Two different schemes are proposed in Section 3.5.1.2 for detection of shorts and SAs using the walking sequence. One scheme uses external response compaction technique, the other local response compaction technique. Both schemes have the capability to detect open faults. In these schemes, each Inet is applied a single '1' and $(N-1)$ '0's due to the N vectors of the walking one sequence. The Inet conductors are sensitized for both logic '0' and '1'. In the case of an open fault in an Inet, the inputs applied from the input scan cell of this Inet do not reach the output scan cell(s). The output scan cell(s) retains the initialized value. Thus, for a walking one sequence, the output scan cell(s) of each faulty Inet contains N '1's('0's). Therefore, in the presence of an open fault, the count in the 1's counter is increased (decreased) for the walking sequence scheme with external response compaction. For the walking sequence scheme with local response compaction, compacted response in each output scan cell is '1' in the fault-free case. In the case of an open fault in an Inet, the corresponding output scan cell(s) receives N '1's('0's). Therefore, the faulty compacted response is '0' in each output scan cell. Thus, in both schemes any open fault can be detected. The scheme can also be described using a walking zero sequence.

4.2 Diagnosis of Open Faults

As mentioned earlier in this chapter, open fault diagnosis is treated on two levels. At one level, Inets are considered to be units under test. Diagnosis implies identifying the faulty Inet and not the faulty conductor within the faulty Inet. In the following, three schemes are proposed for diagnosing Inet open faults. The other level of diagnosis considers the Inet conductors to be units under test. This type of testing requires detailed structural information about the Inets. A scheme is proposed in the following for Inet conductor open fault diagnosis. This is an adaptive scheme to be used in conjunction with one of the Inet open fault diagnosis schemes.

4.2.1 Diagnosis of Inet Open Faults

4.2.1.1 LI/CLI Vector Scheme

Inet open faults can be diagnosed using the LI/CLI vectors introduced in Section 4.1.1. Each input scan cell is loaded with '1'('0') and each output scan cell is initialized to '0'('1') using the LI(CLI) vector. In the fault-free case, the content of each output scan cell is changed from '0'('1') to '1'('0') once the LI(CLI) vector is applied. For an open fault, the output scan cell(s) corresponding to the faulty Inet retains the initialized value. Thus, for open fault diagnosis, the contents of the scan cells are shifted out and monitored after applying the LI(CLI) vector. Each shifted out bit is a '1'('0') in the fault-free case. A '0'('1') shifted out of a scan cell diagnoses the corresponding Inet as faulty.

The procedure is described in the following :

```
Begin
Repeat for LI and CLI vectors
    Begin
    Repeat $P$ times for $P$ different control settings
        Begin
        Load the LI(CLI) vector in the $N$ bit scan path
        Apply loaded vector
        Collect response at each output scan cell
        Unload the $N$ bit scan path with output responses in 1's(0's) counter
        Diagnose based on shifted out response
            If response bit = '1'('0') Then fault-free
```

```
                    Else faulty
                End
            End
        End
```

4.2.1.2 Order Independent Vector Set Scheme

The scheme described in Section 3.5.2.1 for short and SA faults diagnosis can also be used for open fault diagnosis. In this scheme $2.\lceil log_2N \rceil$ vectors are applied with N bits each. It is shown that every bit-pair applied from each input scan cell and received at each output scan cell is '01'/'10' for every pair of complementary vectors. The same is true for open fault diagnosis. The fault-free bit-pair received at every output scan cell is '01'/'10'. But in the presence of an open fault in an Inet, the corresponding bit-pair is changed to '00'/'11'. Therefore, open faults can be diagnosed by bit-pair comparison. Bit-pairs can be compared in each output scan cell (local response compaction) or outside the scan cell (external response compaction) as described in Section 3.5.2.1.

4.2.1.3 Walking Sequence Scheme

The walking sequence diagnosis scheme proposed in Section 3.5.2.2 is also applicable for the diagnosis of open faults. It is shown that, for a walking one sequence, every output scan cell receives a single '1' and $(N-1)$ '0's in the fault-free case. In the case of an open fault in an Inet, the corresponding output scan cell(s) receives N '1's('0's) due to the application of N vectors of the walking one sequence as described in Section 4.1.3. The faulty compacted response in this cell is '0', whereas the fault-free compacted response is '1'. Therefore, an open fault can be diagnosed from the compacted response of the output scan cell. Similarly, a walking zero sequence can be used for open fault diagnosis.

4.3 Diagnosis of Inet Conductor Open Fault

As mentioned in Chapter 2, an open fault may affect only part of the faulty Inet. Therefore, it is possible that a certain part of the faulty Inet may be functioning properly in the presence of an open fault in another part of the Inet. If this faulty part, or more specifically, the faulty conductor in the Inet can be identified, then it can be repaired or it may be possible to use the 'good' portion of the Inet. This section proposes a scheme for the diagnosis of such open faults in Inet conductors.

This is an adaptive scheme to be used in conjunction with any of the three diagnosis schemes proposed above. The Inet open fault diagnosis scheme is applied to diagnose faulty Inets. Once the faulty Inets are identified, the conductor open fault diagnosis scheme is applied to locate the faulty conductor in each of the identified Inets. It is assumed that only one conductor in every Inet can be faulty at any given time.

Suppose the Inet open fault diagnosis scheme has identified an Inet with p input scan cells as faulty. In the proposed scheme, one vector is applied from each of the input scan cells of the faulty Inet, one at a time. This vector loads a '1' in every input scan cell of the faulty Inet. Also, every output scan cell of this Inet is initialized to '0' by this vector. Therefore, this vector is similar to the LI vector introduced in Section 4.1.1. Such a vector is called an *Inet-load-initialize* (ILI) vector. The ILI vector is applied to the faulty Inet p times with p different control settings, so that exactly one vector is applied from each input scan cell. Each output scan cell receives p bit RBS, due to the application of the ILI vector p times. This RBS is shifted out and compared with the stored RBSs for diagnosis of the faulty conductor. The procedure is repeated with complementary ILI (i.e. CILI) vector so that the complete set of ILI and CILI vectors sensitize every conductor of the faulty Inet with both logic '0' and '1'.

ILI(CILI) vector can be applied to all the faulty Inets in parallel. Therefore, P control settings are sufficient to apply these vectors to all the faulty Inets under test, where P is the maximum degree of any Inet.

The scheme is described in the following.

```
Begin
Repeat for ILI and CILI vector
    Begin
    Repeat $P$ times for $P$ different control settings
        Begin
        Load the ILI(CILI) vector in the $N$ bit scan path
        Apply loaded vector
        Collect response at each output scan cell
        Unload the output response in $N$ bit external register
        End
    End
Diagnose by comparing $P$ bit RBS from each output scan cell with stored RBS
End
```

Theorem 4.1 : *Every conductor open fault in a multi-driver Inet with P input scan cells and a single output scan cell can be diagnosed by applying an ILI vector and a CILI vector, P times each, once from each input scan cell.*

Proof : The output scan cells can receive 2^P possible combinations of RBSs, each P bits long, for the ILI(CILI) vector applied P times, once from each input scan cell of the Inet. One of these 2^P RBSs corresponds to the non-faulty condition. The remaining $2^P - 1$ sequences are available to diagnose Inet conductor open fault.

The concepts of path and cpath are introduced in Section 2.1. To recall, a path is defined as the collection of conductors connecting one input scan cell to one output scan cell in the Inet. A cpath corresponding to a conductor is the collection of all the paths which include this conductor as a member.

In the P input scan cell and a single output scan cell Inet, there are P paths, one corresponding to each input scan cell. Therefore, each bit of the P bit RBS, obtained through the application of the ILI(CILI) vector, corresponds to one of the P paths in the Inet. For a faulty conductor in the Inet, the paths going through that conductor (i.e. the cpath of that conductor) are faulty. Consequently, the bits in the RBS corresponding to the cpath of this conductor are erroneous. The cpaths of every two conductors in an Inet are different from each other. Therefore, the set of erroneous bits in the RBS is unique for each faulty conductor.

Therefore, the P bit RBS, obtained through the application of the ILI(CILI) vector, is sufficient to diagnose every conductor open fault.

[QED].

The proof can be easily extended for multiple output scan cell Inets.

An example of this diagnosis procedure is shown in the following. $ABCDE$, shown in Figure 4.1, is a multi-driver Inet with four input scan cells A, B, C, D and a single output scan cell E. Each of the four input scan cells is enabled/disabled by the content of the corresponding control cell. The input scan cells are loaded with '1'('0') and the output scan cells are initialized to '0'('1') with the ILI(CILI) vector. The LI and the CLI vectors for the Inet $ABCDE$ are shown in Table 4.1. A is enabled from $C1$ to apply the first vector. B, C and D are kept disabled during this application. Similarly, three other vectors are applied from B, C and D, one at a time. RBSs corresponding to the non-faulty Inet and different faulty conductors in the Inet are shown in Table 4.2. Every faulty conductor is uniquely diagnosable from the corresponding erroneous RBS given in this table.

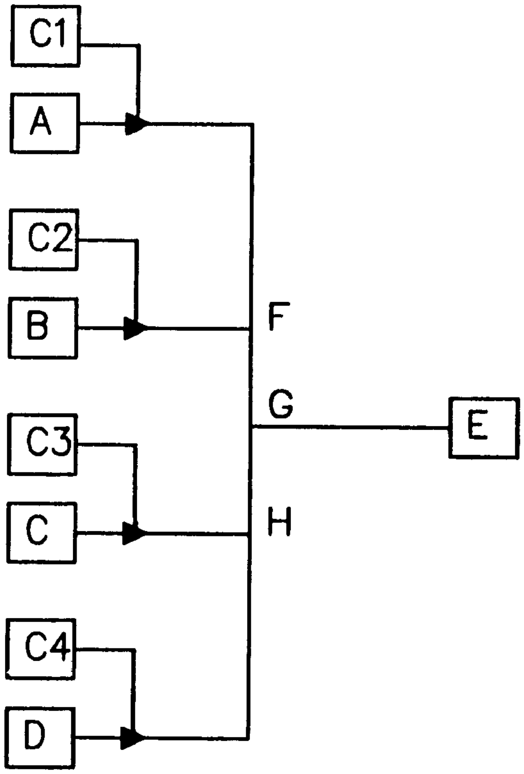


Figure 4.1 Inet conductor open fault diagnosis.

Scan Cell	ILI Vector	CILI Vector
A	1	0
B	1	0
C	1	0
D	1	0
E	0	1

Table 4.1 Input vectors.

Faulty Conductor	RBS for	
	ILI Vectors	CILI Vectors
No-fault	1111	0000
AF	0111	1000
BF	1011	0100
CH	1101	0010
DH	1110	0001
FG	0011	1100
GH	1100	0011
GE	0000	1111

Table 4.2 Output responses for diagnosis.

In this scheme, test vector generation is dependent upon the structure of the faulty Inets. Diagnosis is done by comparing the obtained RBS with the expected RBSs. Expected RBSs are to be computed based on the Inet structures. Therefore, the scheme is not as convenient as the other proposed schemes from the BIST point view. However, this is an adaptive scheme to be applied only if the Inets are found faulty as an outcome of the Inet open fault testing. Moreover, only the faulty Inets, and not the whole set of Inets, need to be tested to diagnose the faulty conductors.

4.4 Summary of Proposed Schemes

Different schemes are proposed in this chapter for detection and diagnosis of open faults. Some of these schemes are extensions of the order independent vector set based schemes and the walking sequence based schemes developed in Chapter 3 for detection and diagnosis of shorts and SA faults. The advantages of these schemes - ease of test vector generation and loading, time-efficient vector generation by internal shifting, and structure independent detection and diagnosis - are also available in the proposed extensions of the schemes for open testing.

LI/CLI vector schemes proposed here deal specifically with open testing. It is shown that detection and diagnosis are independent of the structural complexity of the Inets, hence, these schemes are suitable for BIST implementation. However, test vector generation depends upon the Inet structural information.

An adaptive scheme is also proposed to locate the faulty conductor in the faulty Inet. This scheme can be used together with one of the Inet open fault diagnosis schemes. Detailed structural information about the Inets is required in this scheme, because it deals with the diagnosis of faults in Inet conductors.

Chapter 5 Testing of Glue Logic Interconnects

In Chapters 3 and 4 schemes have been developed for testing different types of interconnect faults. It has been assumed that all the components on a PCB have the boundary scan architecture. The Inets under test can be completely controlled and observed from the boundary scan chain. In reality, however, it is quite possible that some of the components on a PCB may not have boundary scan facility around their periphery. In particular, smaller components (for example, 74-series TTL components) might not have boundary scan. These smaller non-boundary scan components, called *Glue Logic components*, add more complexity to interconnect testing. Interconnects connected to the Glue Logic (GL) cannot be completely accessed from the boundary scan chain. Therefore, test application and response collection for detection and diagnosis become difficult. Also, functional circuitry in the GL components must be taken into account while developing test schemes for these interconnects.

This chapter discusses the testing of interconnects in a partial boundary scan environment described above. In the following, different test schemes are proposed for these interconnects. The schemes proposed are based on the walking sequence scheme and the order independent vector set scheme in Chapter 3. Test vectors are generated for an example Glue Logic and an example PCB using the developed test schemes at the end of this chapter.

5.1 Preliminaries

Some useful terms are defined in the following.

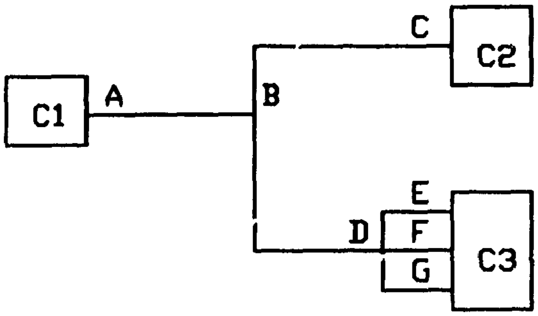


Figure 5.1 Interconnects connecting IC components

GL (Glue Logic) Block : A small digital component on a PCB under test without boundary scan facility around its periphery.

Simple non-boundary scan components like Demultiplexer, Multiplexer, Decoder etc. are considered to be GL blocks. Major functional ICs like ASICs are assumed to be boundary scan components. It is also assumed that every GL block contains only combinational circuitry. Components with sequential circuits are not treated as GL blocks in this dissertation.

GL (Glue Logic) : Set of all the GL blocks on a PCB under test.

GL Function : Logic function realized by glue logic.

Interconnect : An interconnect is a group of electrical conductors physically connected together and bounded by two or more I/O pins. Interconnects connect different IC components (boundary scan component, GL block) together for the desired function on the PCB.

Branch : A branch is an electrical conductor in an interconnect. A branch is an interconnect component.

Every branch has two branch-ends. Two branches can be connected by joining one branch-end of each branch together. A branch is *internal* if none of its two branch-ends is directly (not through another branch) accessible from IC components or PCB edge connectors. A branch is *peripheral* if one or both of its branch-ends are directly accessible. A directly accessible branch-end is a *peripheral branch-end.* Figure 5.1 shows an interconnect $ABCDEFG$ connecting the components $C1$, $C2$ and $C3$. BD is an internal branch. AB is a peripheral branch accessible from $C1$, with A as its peripheral branch-end. Similarly, BC, DE, DF and DG are the other peripheral branches in this example.

By definition, a peripheral branch-end accessible from a boundary scan component terminates into a boundary scan cell, whereas a peripheral branch-end terminating into a GL block is not connected to a boundary scan cell.

Inet : An interconnect is an Inet if all its peripheral branch-ends terminate into boundary scan cells. $ABCDEFG$, in Figure 5.1, is an Inet if $C1$, $C2$ and $C3$ are boundary scan components.

GLI (Glue Logic Interconnect) : An interconnect is a GLI if one or more of its peripheral branch-ends terminate into GL. $ABCDEFG$, in Figure 5.1, is a GLI if any one of $C1$, $C2$ or $C3$ is a GL block.

Figure 5.2 shows a board with boundary scan components and GL connected by Inets and GLIs. AB and CD are Inets connecting two boundary scan components $C1$ and $C2$. Ee is a GLI going from $C1$ to GL and jJ is a GLI going from GL to $C2$. Ff, Gg, Hh etc. are other GLIs in this figure. Lower-case letters like e, f, g etc. are used to represent the peripheral branch-ends of the GLIs terminating into GL.

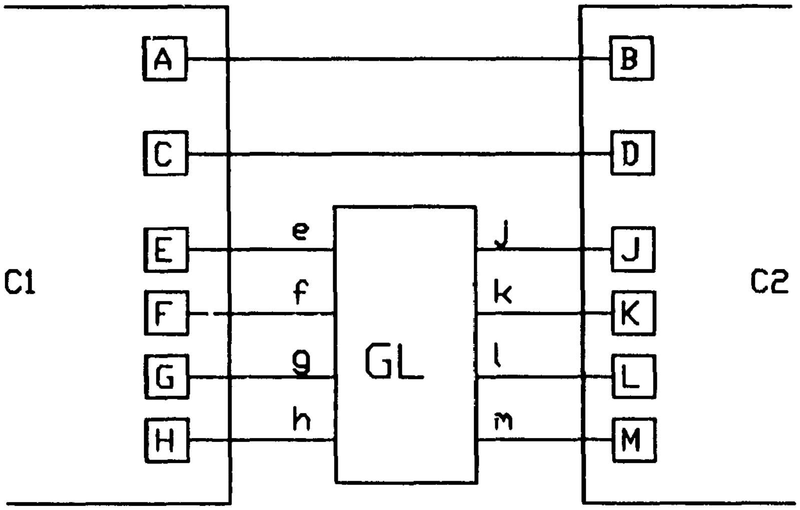


Figure 5.2 Interconnects under test

GLIs fall into two categories ·

GLI_{IN} : A GLI with at least one peripheral branch-end connected to a GL block input, at least one peripheral branch-end connected to an input scan cell and no branch-end connected to GL output. Note that a GLI_{IN} could have branch-end(s) connected to an output scan cell(s).

There are four types of $GLI_{IN}s$:

Type 1 : a GLI_{IN} which has exactly one peripheral branch-end coming from an input scan cell and exactly one peripheral branch-end going to a GL block, such as Dd in Figure 5.3.

Type 2 : a GLI_{IN} which has exactly one peripheral branch-end coming from an input scan cell but fans out to one or more GL blocks and output scan cells, such as $ABbC$ in Figure 5.3.

Type 3 : a GLI_{IN} which has more than one peripheral branch-end connected to input scan cells and fans out to one or more GL blocks, such as $EFGg$ in Figure 5.3.

Type 4 : a GLI_{IN} which has exactly one peripheral branch-end connected to an input scan cell and fans out to one or more GL blocks, such as $HIhi$ in Figure 5.3

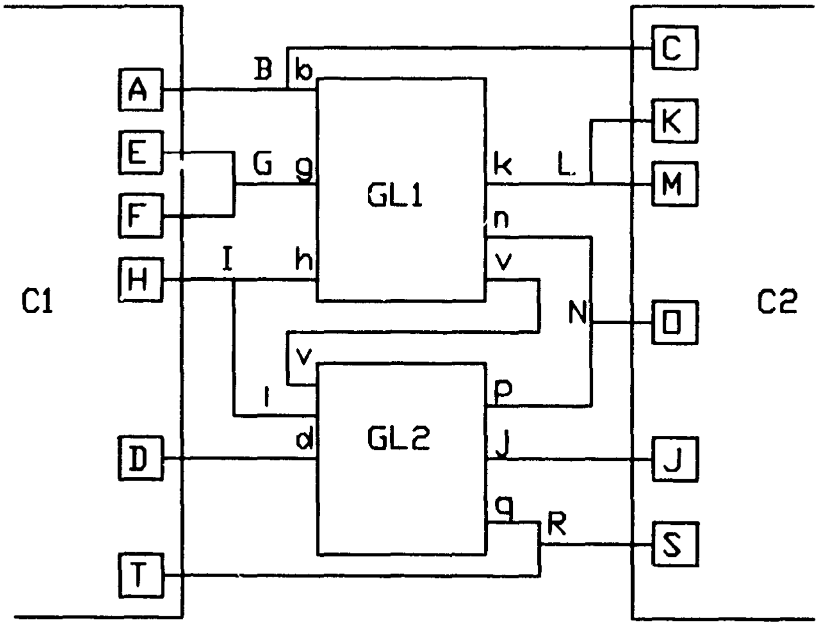


Figure 5.3 GLI structures.

GLI_{OUT} : A GLI with at least one peripheral branch-end connected to a GL block output, at least one peripheral branch-end connected to an output scan cell and no branch-end connected to GL input Note that a CLI_{OUT} could have a branch-end connected to an input scan cell.

There are four types of $GLI_{OUT}s$:

Type 1 : a GLI_{OUT} with exactly one peripheral branch-end coming from a GL block and exactly one peripheral branch-end going to an output scan cell, such as jJ in Figure 5.3.

Type 2 : a GLI_{OUT} which has exactly one peripheral branch-end coming from a GL block but fans out to two or more output scan cells, such as $kLKM$ in Figure 5.3.

Type 3 : a GLI_{OUT} which has more than one peripheral branch-ends coming from GL blocks and can fanout to one or more output scan cells, such as $nNOp$ in Figure 5.3.

Type 4 : a GLI_{OUT} which has exactly one peripheral branch-end coming from a GL block, one or more peripheral branch-ends coming from input scan cells and fai s out to one or more output scan cells, such as $qRST$ in Figure 5.3.

Besides these different types of GLIs, there are other possible GLI structures which satisfy the definition of neither GLI_{IN} nor GLI_{OUT}. Such structures are shown in Figure 5.4. In general, these GLIs feed the output of one GL block as the input of other GL blocks. These can also be connected to input scan cells and/or output scan cells. In these cases, GL blocks are combined to form a bigger block, such that some of the GLI structures become internal to the new GL block The remaining GLI structures can be redefined as $GLI_{IN}s$ and/or $GLI_{OUT}s$. This is shown in Figure 5.5. The faults in the GLI structures internal to the new GL block are not considered in the GLI fault model. Faults in the redefined GLI structures are considered as valid GLI faults. For example, in Figure 5.5, the faults in Gg and dD are considered for GLI testing. The faults in various interconnects (shown in dotted lines) internal to the GL block are not considered for GLI testing.

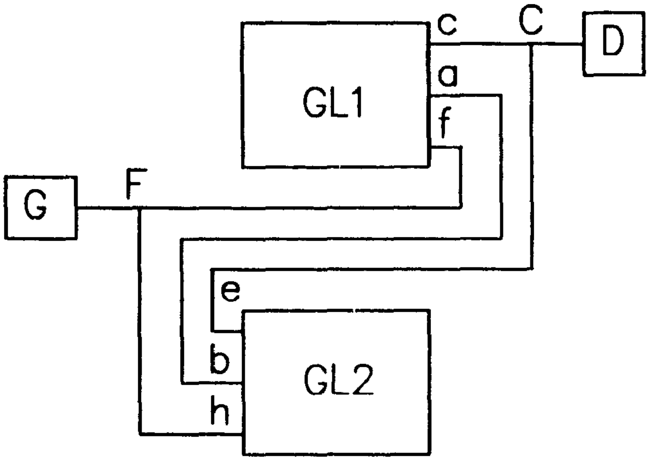


Figure 5.4 Special GLI Structures.

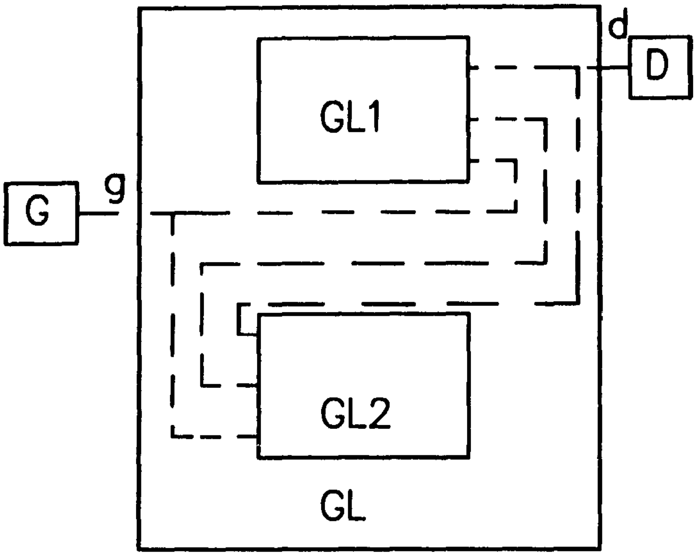


Figure 5.5 Special GLI structures redefined.

These structures pose different accessibility problems from interconnect testing point of view, as discussed in Section 5.3.

A peripheral branch-end defined previously can be one of two types : input end type or output end type. A peripheral branch end of a GLI_{IN} is an input end if connected to an input scan cell and it is an output end if connected to a GL or to an output scan cell. An input end of a GLI_{OUT} is attached to a GL or to an input scan cell whereas an output end is terminated in an output scan cell. For instance, in Figure 5.3, A is the input end and b and C are output ends of GLI_{IN} $ABbC$. Similarly, n and p are the input ends and O is the output end of GLI_{OUT} $nNOp$.

5.2 Fault Model

The fault model is the same one discussed in Section 2.2. Therefore, short, SA and open faults are addressed in different test schemes.

However, shorts can be divided into the following categories depending upon the types of interconnects shorted together.

- Short between Inets.
- Short between Inets and $GLI_{IN}s$.
- Short between Inets and $GLI_{OUT}s$.

- Short between $GLI_{IN}s$.
- Short between $GLI_{OUT}s$.
- Short between $GLI_{IN}s$ and $GLI_{OUT}s$.

As regards open faults testing, test schemes are developed for Inet open faults. Inet conductor open faults are not addressed because of reduced accessibility.

5.3 GLI Test Problems

It was mentioned at the beginning of this chapter that GLIs cannot be accessed completely from the boundary scan chain. The $GLI_{IN}s$ cannot be observed directly and the $GLI_{OUT}s$ cannot be controlled directly from the boundary scan chain. These problems are discussed in detail in the following. An extension of the input scan cell design is proposed to make the $GLI_{IN}s$ observable from the input scan cells. Finally, application of test vectors is discussed in the context of various GLI_{OUT} structures.

5.3.1 Controllability and Observability of Interconnects

An interconnect should be directly or indirectly accessible for test. First of all, an interconnect should be controllable, which involves applying test stimuli with proper test controls. Moreover, the interconnect should be observable, i.e., the effect of the applied stimulus can be observed or monitored. Controllability and observability can be direct or indirect. With the boundary scan framework, direct controllability means input stimulus can be applied to an interconnect from an input scan cell or a PCB edge connector. Indirect controllability refers to the interconnect stimulus coming from some GL block. Similarly, an interconnect is directly observable if the response is observed at an output scan cell connected to it. Indirect observability means the response passes through a GL block before it can be observed.

Every interconnect, as defined in Section 5.1, has two or more peripheral branch-ends. For shorts and SAs testing, an interconnect should be controllable from at least one branch-end and observable from at least one branch-end. Since every peripheral branch-end of each Inet is connected to a boundary scan cell, every Inet is both directly controllable and observable from these boundary scan cells.

Since by definition, at least one peripheral branch-end of a GLI_{IN} is connected to an input scan cell, test stimulus can be applied from this scan cell. Therefore, every GLI_{IN}

is directly controllable. A GLI_{IN} is directly observable only if one peripheral branch-end is connected to an output scan cell. For example, $ABbC$ in Figure 5.3 is directly observable from the output scan cell connected to the branch-end C. However, Dd is not directly observable. Thus, for shorts and SAs testing, every GLI_{IN} is directly controllable and may or may not be directly observable.

A GLI_{OUT} is connected to at least one output scan cell. Therefore, it is directly observable from this output scan cell. A GLI_{OUT} may or may not be directly controllable depending on whether it is connected to an input scan cell or not.

For open testing, every peripheral branch-end must be directly controllable or observable. In Inets, this condition is satisfied because every peripheral branch-end is connected to a boundary scan cell. However, for every GLI_{IN}, the open faults in those branches which terminate into GL cannot be directly observed. Similarly, those branches in $GLI_{OUT}s$ which come out of GL are not directly controllable. Thus, for open faults, each GLI_{IN} is directly controllable, but not completely directly observable; and GLI_{OUT} is directly observable, but not completely directly controllable.

Moreover, multi-driver $GLI_{OUT}s$ ($nNOp$ in Figure 5.3) are to be controlled from within GL. This adds to the controllability problem in GLI testing. It is assumed that multi-drivers can be enabled/disabled from within each GL block during testing.

5.3.2 Increasing Observability of GLI_{IN}

It is shown in the previous section that $GLI_{IN}s$ may or may not have direct observability. Every GLI_{IN} has at least one input scan cell connected to it. Input stimulus is applied from this cell to test the GLI. The JTAG design [JTAG 88] of this cell is such that the stimulus applied from the cell can be fed back for monitoring. An observation or a response compaction block can be added to this monitoring facility so that the applied stimulus can be observed and/or compacted at the same input scan cell. Such an extended input scan cell is shown in Figure 5.6. Input applied from the register $R2$ to the GLI_{IN} is fed back to the response compaction block RC for compaction. Therefore, the same scan cell is used to control and to observe a GLI_{IN}. Actual circuitry in RC depends on the particular response compaction scheme. Compacted response is eventually shifted to $R1$ and taken out for diagnosis. This addition to the input scan cell of GLI_{IN} provides complete controllability and observability for shorts and SAs testing. However, for opens testing, most of the GLI_{IN} branch faults cannot be observed from the extended input scan cell. To be specific, any open fault beyond the feedback point (F in Figure 5.6) cannot be observed from this extended cell. Notice that this extended design of the input scan cell is compatible with the corresponding JTAG design. If necessary, the response compaction block can be by-passed so that the cell will function as a regular input scan cell.

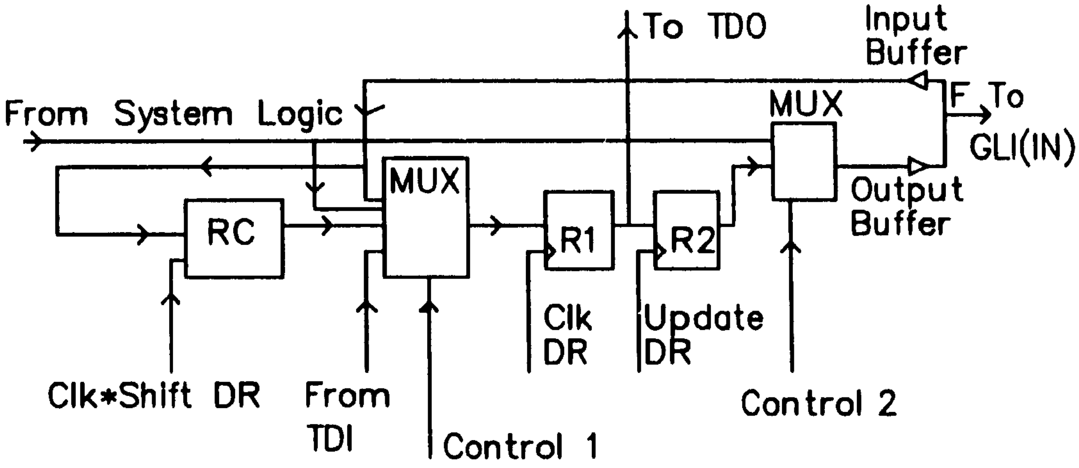


Figure 5.6 Input scan cell with response compaction capability.

5.3.3 Application of Test Vectors on $GLI_{OUT}s$

Since $GLI_{OUT}s$ are mostly fed by inputs coming from GL blocks, one has to be careful in defining test schemes for $GLI_{OUT}s$. For instance, the walking sequence applied to $GLI_{OUT}s$ would require that certain input vectors, which are GL function dependent, be applied in a specific order such that a walking one (zero) can be seen on all peripheral branch-ends of all $GLI_{OUT}s$. However, since there are four types of $GLI_{OUT}s$ as defined in Section 5.1, we need to establish rules about what a walking sequence corresponds to on $GLI_{OUT}s$. For type 1 and type 2 $GLI_{OUT}s$, there is no problem because each type has exactly one peripheral branch end coming from a GL block. For type 3 and type 4, it is assumed that the input vectors applied to GL are such that exactly one input end is enabled during the entire walking sequence.

Similar considerations on type 3 and type 4 $GLI_{OUT}s$ are inherent in all other test schemes to be described in the following.

5.4 Fault Detection and Diagnosis

Inets and GLIs defined in Section 5.1 should be tested using the boundary scan architecture available on the PCB. Boundary scan cells are the control points and observation points. GLIs have reduced controllability and observability. These must be controlled and observed using the limited control and observation points as well as the associated GL function. In the following, different detection and diagnosis schemes are proposed to test the Inets and the GLIs. The schemes place emphasis on PCB level BIST implementation and provide tradeoffs like ease of test vector generation and loading, simple response compaction, structure independent detection and diagnosis and time-efficient test application.

5.4.1 Walking Sequence based Scheme

The walking sequence diagnosis scheme for Inets is described in Section 3.5.2.2. It has been shown that the scheme is attractive for PCB level BIST implementation because of the internal shifting property of the walking sequence and for the response compaction technique used in the scheme as well. A scheme proposed in the following, tests the Inet and the GLI faults, based on this walking sequence diagnosis scheme. The proposed scheme retains most of the attractive properties of the original scheme for PCB level BIST implementation.

This scheme is divided into two parts. Faults involving $GLI_{IN}s$ are diagnosed in Part 1, and faults involving GLI_{OUT}'s are diagnosed in Part 2. Test vector generation and application are independent of GL function in Part 1, and are GL function dependent in Part 2 of the scheme.

Part 1 :

The walking sequence diagnosis scheme described in Section 3.5.2.2 is applied in this part of the scheme. As mentioned in Chapter 2, test vectors are applied from one pre-selected input scan cell of each Inet/GLI_{IN}. The responses are observed and compacted at the output scan cell(s) of each Inet and GLI_{OUT}. Also, as shown in Figure 5.6, the input stimulus applied is fed back for compaction in $GLI_{IN}s$.

The following faults are diagnosed in this part of the scheme :

(i) Every short and SA fault in Inets

(ii) Every short and SA fault in $GLI_{IN}s$

(iii) Every short between Inets and $GLI_{IN}s$

Notice that both the walking one and the zero sequences must be applied to diagnose all types of OR and AND shorts, as mentioned in Sections 3.5.1.2 and 3.5.2.2.

For opens testing, the walking one and the zero sequences both must be applied P times, where P is the maximum degree of any Inet or GLI_{IN} among all the Inets and $GLI_{IN}s$ under test. This guarantees complete coverage of all open faults in Inets and of those open faults in $GLI_{IN}s$ which are observable at corresponding input scan cells. In this testing procedure, notice that when the compacted output bits are scanned out for detection and diagnosis of the aforementioned faults, the information contained in output scan cells of $GLI_{OUT}s$ is of no particular interest.

Part 2 :

The faults involving $GLI_{OUT}s$ and Inets are tested in this part of the scheme. Any one of two different methods described in the following can be used in this part, depending on the implementation efficiency, the functionality of GL blocks and the coverage required.

(i) Pseudo Walking Sequence Scheme

In this scheme a first set of input vectors is generated and applied in a specific order such that the expected outputs become a walking one sequence only at $GLI_{OUT}s$, but produce a '0' value at all the Inets. This expected output set is called a *pseudo walking one sequence*. If there are 'm' $GLI_{OUT}s$ under test, then 'm' vectors are generated such that for each vector expected output at only one GLI_{OUT} is '1' and the expected outputs at the remaining $GLI_{OUT}s$ and Inets are '0's. A complete test set is obtained by generating two additional test vectors at the most. An $(m+1)$-st vector is required which produces a '1' at every Inet output and '0' at every GLI_{OUT}. Finally, an $(m+2)$-nd input vector is required if m is even (to make the number of applied vectors even). This vector produces a '0' in all $GLI_{OUT}s$ and Inets. Notice that each of these $(m+2)$ vectors must be loaded through the boundary scan chain, one at a time. These $(m+2)$ vectors ensure that each output scan cell receives a '1' and an odd number of '0's.

The same procedure is to be repeated for a pseudo walking zero sequence. Like the pseudo walking one sequence, first m vectors of a pseudo walking zero sequence are such, that the expected outputs become a walking zero sequence only at $GLI_{OUT}s$, but produce a '1' value at all the Inets. The $(m + 1)$-st vector produces a '0' at every Inet output and '1' at every GLI_{OUT}. An $(m+2)$-nd vector is required if m is even. This vector produces a '0' in all the Inets and the $GLI_{OUT}s$.

The response compaction scheme is the same as in Section 3.5.2.2. Therefore, using the same reasoning, it can be shown that the following faults can be detected and diagnosed

(i) Every short between Inets and $GLI_{OUT}s$

(ii) Every short and SA fault in $GLI_{OUT}s$

(iii) All open faults in $GLI_{OUT}s$ except on the disabled branches of type 3 and type 4 $GLI_{OUT}s$ (see Section 5.1)

The scheme is described as follows :

```
Begin
    Begin
    Repeat for pseudo walking one and zero sequences
        Begin
        Repeat $(m+1)/(m+2)$ times for $(m+1)/(m+2)$ vectors
            Begin
            Load a vector from $(m+1)$ [for odd $m$]/ $(m+2)$ [for even $m$]
               vectors which is not loaded before in the $N$ bit scan path
            Apply loaded vector
            Collect and compact response at each output scan cell
            End
        End
    Unload the $N$ bit scan path with compacted responses
    End
Diagnose based on compacted bit-pairs from each output scan cell
    If bit-pair = '11' Then fault-free
    Else faulty
End
```

This scheme requires that both the pseudo walking one and the zero sequences should be applied for testing of the different types of faults mentioned above. However, since test generation is GL function dependent in this case, it may be possible that the GL function does not allow the application of pseudo walking one(zero) sequence at the $GLI_{OUT}s$. Therefore, fault coverage depends on the feasibility of generation of pseudo walking sequences. An example of the application of this scheme is presented in Section 5.5.

Time Complexity

In the scheme described above, since all $GLI_{OUT}s$ are tested in series, the time complexity is simply $O(mN)$. However, this time complexity can be reduced by testing all GL blocks in parallel, but at the expense of a somewhat reduced fault coverage. In the parallel approach, each GL block is applied a pseudo walking one sequence of its own, in parallel with all other GL blocks. Here, one must ensure the proper disabling of a multi-driver $GLI_{OUT}s$ of type 3 and type 4 as described in Section 5.3.3. Here, the time complexity is $O(qN)$, where q is the maximum number of $GLI_{OUT}s$ from any GL block. In this parallel approach, some $GLI_{OUT}s$ in different GL blocks receive the same RBSs. Shorts among these $GLI_{OUT}s$ can never be detected. This leads to loss in fault coverage.

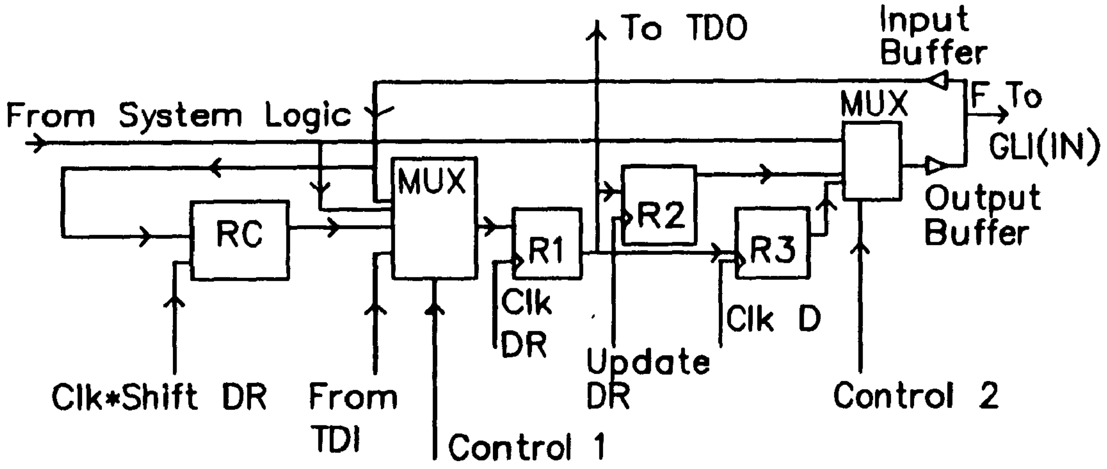
(ii) Partial Walking Sequence Scheme

In this scheme, walking one(zero) sequences are applied to the Inets and constant logic values (all '0's or all '1's) are applied to each of the $GLI_{IN}s$. Therefore, the scheme is called a *partial walking sequence scheme.* The main advantage of this scheme is time efficiency. Like the walking sequence (Section 3.5.2.2), the partial walking sequence is loaded through the scan chain only once. Thus, for N bit scan chain, time complexity is $O(N)$. However, compared to the pseudo walking sequence scheme coverage is reduced. Shorts only between Inets and $GLI_{OUT}s$ are diagnosed. GLI_{OUT} shorts, SAs and open faults cannot be tested using this scheme.

Detection

For detection, a walking one sequence is applied only to the Inets and a constant logic value ('0') is applied to each of the $GLI_{IN}s$. Input scan cell of each GLI_{IN} is extended to load and to apply this partial walking one sequence. Figure 5.7 shows such an extended cell.

Register $R3$, in this cell, is initialized to a constant logic value ('0'). This value is applied N times to the GLI_{IN}. Register $R1$ is used as the by-pass path for the walking one sequence. Each input bit from the walking one sequence is shifted in $R1$, held there, and shifted forward to the next scan cell. Since a constant logic value is applied to each GLI_{IN}, expected values at each GLI_{OUT} will also be a constant logic value (N '0's or N '1's for N vectors applied), irrespective of the GL block function.



Clk D = Update DR * Inst

where Inst is the bit from instruction register to load Register R3.

Figure 5.7 Extended input scan cell for partial walking sequence.

Response Compaction :

Responses are compacted at each output scan cell and extended input scan cell. A single bit compacted response is obtained from each of these cells after applying N vectors. A walking one sequence is applied to the Inets. Therefore, the fault-free compacted response at each Inet output scan cell is '1'. However, since all '0's (or, all '1's) are expected at each GLI_{OUT}, fault-free compacted response is '0' at each output scan cell of $GLI_{OUT}s$. Notice that the compacted responses at input scan cells of $GLI_{IN}s$ are of no interest here. Now, for an OR short between Inet and GLI_{OUT}, where the expected outputs at the GLI_{OUT} are all '1's, the number of '1's at the Inet changes from odd to even and the number of '1's at the GLI_{OUT} remains unchanged. Therefore, the faulty compacted response at the Inet changes from '1' to '0' and the compacted response at the GLI_{OUT} remains unchanged at '0'. Similarly, for an AND short between the Inet and the GLI_{OUT} mentioned above, the compacted response at the Inet remains unchanged at '1' and the compacted response at the GLI_{OUT} changes from '0' to '1'. The same type of changes can be observed in the faulty compacted responses if the expected outputs at GLI_{OUT} are all '0's instead of all '1's. Thus, for any short between Inet and GLI_{OUT}, compacted response at the Inet or GLI_{OUT} (but not both) are changed from the corresponding fault-free value. Detection is done by shifting out these cc,npacted responses and observing the bits coming out of each output scan cell. Notice that the compaction scheme is such, that the fault-free (faulty) compacted responses at the Inet and GLI_{OUT} are complementary to each other. Thus, for detection, one should know the correspondence between the scan cells and the output bits.

The partial walking sequence detection scheme is described in the following. $W1$ and $W0$ are as defined in Section 3.5.1.2.

```
Begin
Load constant values '0'('1') in $GLI_{IN}$ input scan cells
  and $W1(W0)$ in the $N$ bit boundary scan path
Apply loaded vector
Collect and compact response at each output scan cell
    Begin
    Repeat $(N - 1)$ times
        Begin
        Shift $W1(W0)$ one bit to the right (move a '0'('1') in the leftmost cell)
        Apply shifted $W1(W0)$ to the Inets and constant
          values (loaded previously) to the $GLI_{IN}s$
        Collect and compact responses in each output scan cell
        End
    End
Unload the $N$ bit scan path with compacted responses
Detect based on compacted responses
End
```

The partial walking one sequence applied to Inets and $GLI_{IN}s$ of Figure 5.2 is shown in Table 5.1. Expected outputs and compacted responses at the Inets and $GLI_{OUT}s$ of Figure 5.2 are given in Tables 5.2 and 5.3 respectively. Table 5.3 shows that for a short between an lnet and GLI_{OUT}, one (but not both) of the faulty compacted responses is changed from its fault-free value.

	V_1	V_2	V_3	...	V_N
AB	1	0	0	...	0
CD	0	1	0	...	0
Ee	0	0	0	...	0
Ff	0	0	0	...	0
Gg	0	0	0	...	0
Hh	0	0	0	...	0

Table 5.1 Partial walking one sequence.

	Y_1	Y_2	Y_3	...	Y_N
AB	1	0	0	...	0
CD	0	1	0	...	0
jJ	0	0	0	...	0
kK	0	0	0	...	0
lL	0	0	0	...	0
mM	1	1	1	...	1

Table 5.2 Expected outputs due to partial walking one sequence.

AB	CD	jJ	kK	lL	mM	Remarks
1	1	0	0	0	0	NF
1	1	0	0	*1*	0	OR-short (AB-lL)
1	*0*	0	0	0	0	OR-short (CD-mM)
0	1	0	0	0	0	AND-short (AB-lL)
1	1	0	0	0	*1*	AND-short (CD-mM)

Table 5.3 Compacted responses due to partial walking one sequence

The detection scheme is described above with a walking one sequence. Also, it is assumed that '0' is applied from each GLI_{IN} input scan cell. However, the scheme works with a walking zero sequence as well, and any constant logic value can be applied from any of the GLI_{IN}

input scan cells, as long as the same value is applied N times for the N vectors of the walking sequence.

Diagnosis

Diagnosis requires that the compacted responses at both the Inet and the GLI_{OUT} output scan cells be changed from the corresponding fault-free value. It was shown above that the compacted response at either Inet or GLI_{OUT} is changed due to a short by applying a walking one sequence to the Inets and constant logic values to the $GLI_{OUT}s$. Now, the other faulty compacted response can also be changed from its fault-free value by applying a walking zero sequence to the Inets and complementary constant logic values to the $GLI_{OUT}s$ Tables 5.2 and 5.4 show the expected outputs at the Inets and $GLI_{OUT}s$ of Figure 5.2, because of the partial walking one and zero sequences respectively. Notice that the expected outputs at each GLI_{OUT} are made complementary to each other for the two partial walking sequences Table 5.5 shows the compacted responses along each Inet and GLI_{OUT}. In each entry of this table, there is a pair of compacted bits. The first bit corresponds to the partial walking one sequence and the second bit corresponds to the partial walking zero sequence. It is shown that the fault-free compacted bit-pair is '11'/'00'. However, in case of a fault, this bit-pair changes to '10'/'01'. Therefore, faulty Inets and $GLI_{OUT}s$ can be diagnosed by observing the compacted bit-pairs.

	Y_1	Y_2	Y_3	...	Y_N
AB	0	1	1	...	1
CD	1	0	1	...	1
jJ	1	1	1	...	1
kK	1	1	1	...	1
lL	1	1	1	...	1
mM	0	0	0	...	0

Table 5.4 Expected outputs due to partial walking zero sequence

Diagnosis requires that both the walking one and zero sequences are applied to the Inets. Expected outputs along each GLI_{OUT} should be constant complementary logic values corresponding to the two walking sequences. Based on these expected outputs and GL function,

AB	CD	jJ	kK	IL	mM	Remarks
11	11	00	00	00	00	NF
10				10		OR-short (AB-IL)
	01				01	OR-short (CD-mM)
01				01		AND-short (AB-IL)
	10				10	AND-short (CD-mM)

Table 5.5 Compacted responses due to partial walking one and zero sequences.

constant logic values to be applied from each GLI_{IN} are determined. Therefore, test vector generation for diagnosis depends upon the GL function.

The scheme is described in the following. Here, true sequence refers to loading a constant value in each GLI_{IN} input scan cells and walking one sequence in the N bit boundary scan path. Thus, W1 : 1000... for true sequence. Complementary sequence refers to loading a constant value in each GLI_{IN} input scan cells such that the outputs along each GLI_{OUT} are complementary to the outputs along that GLI_{OUT} because of true sequence. W0 : 0111... for complementary sequence.

```
Begin
    Begin
    Repeat for true and complementary sequences
        Begin
        Load constant values '0'('1') in $GLI_{IN}$ input scan cells
            and $W1(W0)$ in the $N$ bit boundary scan path
        Apply loaded vector
        Collect and compact response at each output scan cell
            Begin
            Repeat $(N - 1)$ times
                Begin
                Shift $W1(W0)$ one bit to the right (move a '0'('1') in the leftmost cell)
                Apply shifted $W1(W0)$ to the Inets and constant
                    values (loaded previously) to the $GLI_{IN}s$
                Collect and compact responses in each output scan cell
                End
            End
        Unload the $N$ bit scan path with compacted responses
        End
    End
Diagnose based on compacted bit-pair from each output scan cell
    If bit-pair = '00'/'11' Then fault-free
    Else faulty
End
```

5.4.2 GLFI (Glue Logic Function Independent) Scheme

This scheme is completely independent of GL function. It is divided into two parts. Part 1 tests the faults in Inets and $GLI_{IN}s$, and Part 2 tests the faults between Inets and $GLI_{OUT}s$.

Part 1 :

This part of the scheme uses the order independent vector set diagnosis scheme described in Section 3.5.2.1. $2\lceil log_2 N\rceil$ vectors are applied where N is the number of scan cells in the boundary scan chain. Response compaction is done by comparing bit-pairs. Responses are compacted at each output scan cell and at the extended input scan cell of each GLI_{IN} The following faults are diagnosed :

(i) Every short and SA fault in Inets

(ii) Every short and SA fault in $GLI_{IN}s$

(iii) Every short between Inets and $GLI_{IN}s$

Like the walking sequence based scheme for opens testing, $2\lceil log_2 N\rceil$ vectors are applied P times, where P is the maximum degree of any Inet or GLI_{IN} among all the Inets and $GLI_{IN}s$ under test. This can diagnose every open fault in Inets and only those open faults in $GLI_{IN}s$ which are observable at the corresponding input scan cells.

An example is given in Section 5.5.

Part 2 :

Every short between Inets and $GLI_{OUT}s$ are diagnosed in this part of the scheme. GLI_{OUT} shorts, SAs and open faults cannot be diagnosed by this scheme. The main advantage of this scheme is that test vector generation needs no information about the GL function.

In this part, an exhaustive set of 2^i vectors is generated for each GL block where i is the number of $GLI_{IN}s$ connected to that block. Each of these 2^i vectors must be applied twice, consecutively. If there are more than one GL blocks, i is chosen such that i is the maximum number of $GLI_{IN}s$ connected to any GL block under test. Therefore, 2.2^i vectors are generated for every GL block under test. However, for any GL block which has j $GLI_{IN}s$ connected to it, where $j < i$, 2.2^j vectors are generated based on its exhaustive set. The remaining vectors to complete the set of 2.2^i vectors are all '0's (all '1's) for this GL block. A '01' bit-pair is generated for each Inet corresponding to each of the 2^i vector pairs for each GL block. These vectors for individual GL blocks and Inets are stacked together to form the complete set of 2.2^i vectors for this part of the scheme. Test vector loading and application, and response compaction are the same as described in Section 3.5.2.1.

In this scheme of 2.2^i vectors, all possible combinations of inputs are applied to each GL block. This guarantees that every GLI_{OUT} should take each value pair '00' and '11' at least once. For a pair of the same input vector applied, the fault-free expected outputs at each GLI_{OUT} are '00'/'11'. However, notice that for every such vector pair, fault-free expected outputs at each Inet are '01'/'10'. For an OR short between a GLI_{OUT} and Inet, the output values at the GLI_{OUT} are changed from '00' to '01'/'10', and the values at the Inet change from '01'/'10' to '11'. Similarly, for an AND short between GLI_{OUT} and Inet, faulty values at GLI_{OUT} and Inet are '01'/'10' and '00' respectively. The diagnosis procedure is the same as in the order independent vector set diagnosis scheme. After applying a pair of vectors, the responses are compared and the result is shifted out. Notice that the compaction scheme is such that the fault-free (faulty) comparator outputs at Inet and GLI_{OUT} are complementary to each other. Thus, for diagnosis, one should know the correspondence between the scan cells and output bits.

The scheme is described in the following.

```
Begin
Repeat $2^i$ times
    Begin
    Load a vector (which is not previously loaded) from $2^i$ vectors in the $N$ bit scan path
    Apply loaded vector
    Collect response at each output scan cell and
      at each extended input scan cell
    Repeat the above two steps with the same vector
```

```
        Unload the $N$ bit scan path with compared responses
        Diagnose based on compared output bits
        End
End
```

Interconnects	V_1	V'_1	V_2	V'_2	V_3	V'_3	...	V_{16}	V'_{16}
AB	0	1	0	1	0	1	...	0	1
CD	0	1	0	1	0	1	...	0	1
Ee	0	0	0	0	0	0	...	1	1
Ff	0	0	0	0	0	0	...	1	1
Gg	0	0	0	0	1	1	...	1	1
Hh	0	0	1	1	0	0	...	1	1

Table 5.6 2.2^t Input vectors for GLFI scheme.

Inet Input	GLI_{OUT} Output(FF)	Fault	Inet Output(F)	GLI_{OUT} Output(F)
01	11	OR	**11**	11
01	00	OR	01	**01**
01	00	AND	**00**	00
01	11	AND	01	**01**

Table 5.7 Faulty output bit-pairs.

Output Response	Inet	GLI_{OUT}
10	FF	Faulty
01	FF	Faulty
00	Faulty	FF
11	Faulty	FF

Table 5.8 Diagnostics table.

A test set of 2.2^i vectors for interconnects in Figure 5.2 is shown in Table 5.6. Different input combinations and corresponding faulty outputs for Inets and $GLI_{OUT}s$ are shown in Table 5.7. Table 5.8 shows the diagnostics for Table 5.7.

Time Complexity

Time complexity of Part 1 is $O(NlogN)$.

2.2^i vectors in Part 2 are applied to all GL blocks in parallel. Each vector is N bits long. Thus, for parallel application, time complexity is $O(N.2^i)$ where i is the maximum number of $GLI_{IN}s$ going to any GL block among all the GL blocks under test. This is reasonable, assuming that 2^i is small compared to N. However, it is possible to increase the fault coverage by applying test vectors to one GL block at a time. For this serial testing, time complexity increases to $O(N[2^{i}1 + 2^{i}2 + ...])$ where $i_1, i_2, ...$ are the numbers of $GLI_{IN}s$ corresponding to $GL_1, GL_2, ...$. In this case, 2.2^{ix} vectors are applied to any x-th GL block and '01' pairs are applied to the $GLI_{IN}s$ of all other GL blocks like the Inets. Therefore, shorts between $GLI_{OUT}s$ of any x-th block and $GLI_{IN}s$ of all other GL blocks can also be diagnosed by this serial approach.

5.5 Application of Test Schemes

This section discusses application of the test schemes developed in the previous section. In Section 5.5.1, test vectors are derived for an example IC (GL block). Section 5.5.2 describes the test vectors developed for an example PCB.

5.5.1 Test Vector Sets for Example IC

The example IC is a 1 of 8 Demultiplexer/Decoder (10162, Digital 10,000 series ECL). A block diagram of this GL block is shown in Figure 5.8.

The 10162 is a binary coded 3 line to 8 line decoder [Signetics 74]. Outputs are normally low with the selected output going high. Two enable inputs (E_0 and E_1) make it ideally suited for demultiplexer applications. One of the two enable inputs can be used as the data enable input. Either enable input when high, forces all outputs low.

As far as Glue Logic Interconnect testing is concerned, this GL block has five $GLI_{IN}s$ (E_1, E_0, A_2, A_1, A_0) and eight $GLI_{OUT}s$ (D_0, D_1, ... D_7). Different schemes developed

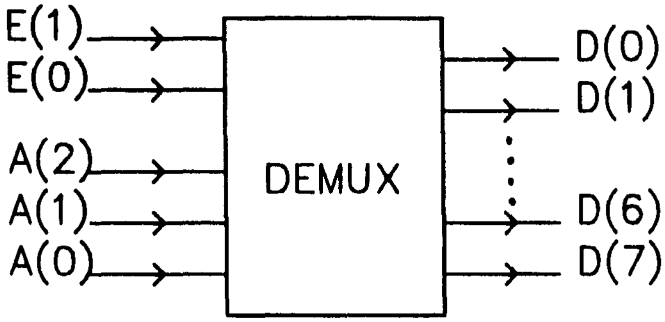


Figure 5.8 Demultiplexer under test.

in the previous section are used in the following to find test vectors for the GLIs of this GL block.

a) Walking Sequence based Scheme

Part 1

This part applies a simple walking one and a simple walking zero sequence as test vectors. A walking one sequence for the demultiplexer GLIs are shown in Table 5.9.

Interconnects	V_1	V_2	V_3	V_4	V_5
E_1	1	0	0	0	0
E_2	0	1	0	0	0
A_2	0	0	1	0	0
A_1	0	0	0	1	0
A_0	0	0	0	0	1

Table 5.9 Walking one sequence for demultiplexer.

Part 2

(i) Pseudo Walking Sequence Scheme

In this part it is required that the expected outputs at the $GLI_{OUT}s$ is a walking one sequence. Test vector generation begins with a list of expected outputs which is shown in

Table 5.10. A set of input vectors is then generated based on the function of the GL block and the expected output list. In this particular example, E_1 and E_0 are both set to '0'. Select lines (A_2, A_1, A_0) go through all the combinations of inputs (from '000' to '111'). For each combination, only one output line is selected (the output along that line is '1' and all other outputs are '0'). Therefore, a walking one sequence is available at the output side. The input vectors are shown in Table 5.11.

Test vector generation is function dependent in this part. In spite of that, test vector generation for this example IC is rather straightforward.

GLI_{OUTs}	Y_1	Y_2	...	Y_8
D_0	1	0	...	0
D_1	0	1	...	0
D_2	0	0	..	0
D_3	0	0	..	0
D_4	0	0	..	0
D_5	0	0	..	0
D_6	0	0	..	0
D_7	0	0	..	1

Table 5.10 Pseudo walking one sequence for demultiplexer.

GLI_{INs}	V_1	V_2	...	V_8
E_1	0	0	...	0
E_0	0	0	...	0
A_2	0	0	...	1
A_1	0	0	...	1
A_0	0	1	...	1

Table 5.11 Pseudo walking one sequence input set for demultiplexer.

(ii) Partial Walking Sequence Scheme

Detection

Detection requires that a walking one(zero) sequence be applied to the Inets and constant logic values be applied to each GLI_{IN}. Constant logic value of '0' is applied to each of the five $GLI_{IN}s$ in this example. This produces '0's at every GLI_{OUT} (except D_0). Input vectors and corresponding expected outputs for the Demultiplexer are shown in Tables 5.12 and 5.13 respectively.

$GLI_{IN}s$	V
E_1	0
E_0	0
A_2	0
A_1	0
A_0	0

Table 5.12 Input vector for detection.

$GLI_{OUT}s$	Y
D_0	1
D_1	0
...	...
D_6	0
D_7	0

Table 5.13 Expected output for detection.

Diagnosis

Complementary logic values must be applied to each GLI for partial walking one and zero sequences. In this example, '0' is applied to each GLI_{IN} in the first step. Expected outputs because of these inputs are '1' at D_0, and '0' at all the other $GLI_{OUT}s$. Diagnosis requires that '0' should be applied to D_0, and '1' to all the other $GLI_{OUT}s$ in the second step. All '1's cannot be produced at all the $GLI_{OUT}s$ (D_1, D_2, ... D_7) simultaneously for this demultiplexer function. Therefore, faults in these $GLI_{OUT}s$ cannot be diagnosed using this scheme.

b) GLFI Scheme

Part 1

2.$logN$ vectors are applied in this part of the GLFI scheme where N is the number of scan cells in the scan chain. This completely function independent vector set is shown in Table 5.14.

$GLI_{IN}s$	V_1	V_2	V_3	V_4	V_5	V_6
E_1	0	1	0	1	1	0
E_0	1	0	1	0	0	1
A_2	0	1	1	0	0	1
A_1	1	0	0	1	0	1
A_0	0	1	0	1	0	1

Table 5.14 2.$logN$ vectors for demultiplexer.

Part 2

For i $GLI_{IN}s$ 2.2^i test vectors are applied in this part. There are five $GLI_{IN}s$ in this example IC. Therefore, each of the 2^5 vectors (exhaustive set for five $GLI_{IN}s$) is applied twice. These 64 vectors are shown in Table 15.

$GLI_{IN}s$	V_1	V'_1	V_2	V'_2	...	V_{32}	V'_{32}
E_1	0	0	0	0	...	1	1
E_0	0	0	0	0	...	1	1
A_2	0	0	0	0	...	1	1
A_1	0	0	0	0	...	1	1
A_0	0	0	1	1	...	1	1

Table 5.15 2.2^i vectors for demultiplexer.

5.5.2 Test vectors for example PCB

Test vectors are developed for an example PCB containing GL blocks. The board has 14 ASICs connected by a number of Glue Logic ICs. In terms of gate counts, the GL part contains 900 AND, 502 OR, 14 NAND, 104 NOR and 34 XOR gates. There are 670 inputs and 1003 outputs. Some of these inputs/outputs are connected in the boundary scan chain and the rest must be externally probed for testing.

The scan cells of the ICs on this PCB do not have any extension for internal shifting, response compaction etc. The scan cells are based on JTAG designs. Therefore, the boundary scan chain must be used only to shift in the input vectors and shift out the responses.

Two sets of vectors are developed for testing the Inets and the GLIs on this PCB. One set contains a walking one sequence which diagnoses shorts and SAs in Inets. The second set of vectors diagnoses shorts between Inets and GLIs. This set of vectors is derived by using the method developed in Part 2 of the GLFI scheme. In the board netlist, only the gate level description, and not the functional boundary, of the Glue Logic ICs is available. Thus, pseudo functional blocks are defined from the netlist for the generation of test vectors. A pseudo functional block is a collection of gates and GLIs where every input (GLI_{IN}) to a block is connected to all other inputs to that block through some logic gates in the block. Also, the inputs to one block are not connected to the inputs to any other block. The largest of such blocks has 8 inputs. Therefore, using Part 2 of GLFI scheme $2.2^8 = 512$ vectors are developed for each block. Smaller blocks are combined together to form bigger blocks of input size 8 at the most. The vectors must be applied to all the blocks in parallel, and 128 pairs of '01' must be applied to each Inet during this second phase of testing.

The total number of inputs and outputs on this PCB is 1673. Some of these inputs/outputs can be directly accessed from the PCB edge connectors. The rest are connected in the boundary scan chain. Following, is an estimation of the test application time considering that the scan chain length 1600. Boundary scan cells of the ASICs on the PCB do not have any response compaction capability. Therefore, each vector is to be shifted in and the corresponding response is to be shifted out, one at a time. Each vector of the walking one sequence vector set requires 1600 shift ins for loading, and 1600 shift outs after response collection. Therefore, 3201 clock cycles (one cycle for applying the loaded vector) is required for each vector. With a 10MHz test clock, test application time is approximately 0.54 second for the walking one sequence. There are 512 vectors in the second set. Here also, each vector requires 3201 clock cycles. Therefore, with the sam est clock, approximately 0.17 second is required for this set of vectors. Therefore, the total test time for the interconnects on this PCB is approximately 0.71 second.

5.6 Summary of Proposed Schemes

Test schemes are proposed in this chapter for detection and diagnosis of the different types of interconnect faults in a partial boundary scan environment. These schemes are based on the order independent vector scheme and the walking sequence scheme introduced in Chapter 3.

In some of the schemes proposed here, test vector generation is independent of the GL function. Also, in some of these schemes test vector loading is independent of the order of the I/O scan cells and the structure of the interconnects. However, in schemes like the pseudo walking sequence scheme and the partial walking sequence diagnosis scheme, test vector generation is GL function independent. But these schemes provide increased fault coverage compared to the function independent schemes.

In general, the GLI_{IN} faults are easier to test. However, most of the GLI_{IN} open faults go undetected because the responses are observed from the input ends of the $GLI_{IN}s$. GLI_{OUT} faults are difficult to test because of their indirect controllability through the GL function.

Detection and diagnosis are independent of the structural complexity of the interconnects. Therefore, no expected response is required to be generated and stored, because the response compaction schemes developed in Chapter 3 are used in these detection and diagnosis procedures. Output scan cells and GLI_{IN} input scan cells are extended for local response compaction. GLI_{IN} input scan cells are also extended in the partial walking sequence scheme for by-pass opertion.

The schemes developed in this chapter are not as convenient as those proposed in Chapters 3 and 4 from the BIST point of view. Also, the fault coverage of these schemes is limited, because of the reduced accessibility of the partial boundary scan environment. However, the developed schemes are the extensions of the schemes proposed in Chapter 3. Thus, some of the desired properties, like function independent test generation, time-efficient test vector loading, structure independent detection and diagnosis, are available in one or the other scheme proposed in this chapter.

Chapter 6 Conclusion

The problems and the complexities of board level interconnect testing are addressed in this dissertation. With the increasing component density in ICs and the increasing number of ICs on printed circuit boards, the number of interconnects on the boards have increased tremendously. Apparently, interconnects are simple electrical wires running in parallel to each other. But, in reality, the problems in interconnect testing are no simpler than the problems of testing the circuitry inside IC components. The complexity of the interconnecting testing problem is discussed in detail in Chapter 1 of the dissertation. It is shown that one of the main problems in board level interconnect testing is that of test point accessibility. Boundary scan architecture proposed by Joint Test Action Group (JTAG) is an efficient framework to provide electronic access to the interconnect test points. This dissertation addresses the interconnect testing problem in the context of the boundary scan architecture. The basic elements of this design for testability framework are discussed in Chapter 1.

Several schemes are proposed in the literature for testing different types of interconnect faults. These schemes have advantages and disadvantages. This dissertation does not intend to develop more schemes along the same line. Rather, the approach taken here is motivated by the fact that the cost of introducing the design for testability framework of the boundary scan architecture can be justified only if the framework is exploited to its full extent for board level testing. Therefore, an attempt has been made to use the boundary scan architecture for every aspect of interconnect testing, starting from test vector generation to output response analysis. This approach results in a family of built-in self-test (BIST) schemes for testing of the board level interconnects. The board level BIST schemes have features similar to the component level BIST schemes. Test vector generation is simple and test vector loading, through the boundary scan chain, is time-efficient in some of these schemes. And this is true even though deterministic vector sets are used in these BIST schemes (whereas component level BIST schemes use pseudo-random vector sets in most cases). Several output response compaction techniques are introduced in the BIST schemes to reduce the output data shift out time and to simplify the process of detection and diagnosis. Output scan cells are used as single-bit response compactors for these response compaction techniques. Detection and diagnosis are independent of the topology and structural complexity of the interconnects under test. In most schemes, there is no need for expected response storage. Detection and diagnosis are as simple as observing and/or counting the contents shifted out of single-bit response compactors in the output scan cells. Therefore, the approach in this dissertation establishes that the boundary scan architecture has potential functions for interconnect testing. These functions of the boundary scan architecture make it more appealing as a board level DFT framework. Utilization of these properties of the framework results in a family of board level interconnect BIST schemes proposed in the work.

The complexities of interconnect structures are described in Chapter 2. The fault model and the test environment are also discussed in this chapter, so that test schemes can be developed based on these frameworks. Three types of faults, namely short, stuck-at and open faults, are treated as valid interconnect faults. Among these, shorts and stuck-at faults are shown to make the entire interconnect(s) functionally faulty. In the presence of these faults, faulty interconnects cannot function correctly. However, this is not true for open faults. Some conductors of an interconnect can function properly in the presence of open fault in some other conductors of the faulty interconnect. Thus, opens testing requires more structural information about the interconnects under test. Because these fault types behave differently, shorts and stuck-at faults testing are addressed in one chapter and open faults testing are discussed in another chapter.

Chapter 3 begins with an overview of the existing test schemes for shorts and SAs testing.

It is mentioned that the existing schemes are incomplete in the sense that most are test generation algorithms. Implementation details are missing in these schemes. Contrarily, each scheme proposed in this dissertation addresses the various aspects like test vector generation, test application environment, time complexity etc. In other words, each scheme is complete from an implementation point of view. All of the existing schemes treat the boundary scan chain as a means of electronic access to the interconnect test points only. Therefore, the role of the boundary scan chain is to shift in the input vectors and to shift out the responses. But in the proposed schemes, boundary scan chain is used to generate time efficient test vectors and to do time efficient response compaction. This dissertation proposes a number of extensions of the scan cells to introduce computational ability within the scan chain. Thus, the proposed schemes utilize the boundary scan architecture for various functions related to interconnect testing

Chapters 3 and 4 discuss test schemes for short, stuck-at and open fault testing, considering that the boards under test have one hundred percent boundary scan capability. Chapter 5 addresses the more complex but practical environment of interconnect testing where the boards under test have a partial boundary scan environment. Schemes are developed for testing the interconnect faults, assuming that some of the IC components on the board do not have boundary scan capability. The schemes proposed in this chapter are extensions and/or modifications of the schemes developed in Chapters 3 and 4. However, the schemes for glue logic (IC components without boundary scan) interconnect testing are not as efficient as those for simple interconnect testing, mainly because of reduced accessibility to the glue logic interconnects. In spite of that, the developed schemes have simple methods for test vectors generation. Detection and diagnosis, too, are based on local (single scan cell based) structure independent information.

It was mentioned several times that the work in this dissertation is inspired by the boundary scan architecture proposed by JTAG. The basic elements of this architecture and the design of different boundary scan cells are discussed in the introduction (Chapter 1) of this dissertation. However, in this work, boundary scan architecture is not only used as a mere vehicle for electronic access to the interconnect test points. This dissertation suggests a number of extensions of the boundary scan cells for an on-board BIST approach. Interestingly, the developed schemes are flexible enough. Suggested extensions are necessary only if the schemes are to be implemented with their full potential. For example, in the walking sequence diagnosis scheme for shorts and stuck-at faults diagnosis proposed in Chapter 3, output scan cells need to be extended only if response compaction is done locally in each output scan cell. But the scheme can be implemented without any extension of the JTAG proposed scan cells. In this case, every test vector is to be shifted in and every response is to be shifted out of the scan chain, one at a time, as described in the walking sequence diagnosis scheme without local response compaction (Chapter 3). However, the time complexity increases from $O(N)$ to $O(N^2)$. Even in the unmodified boundary scan environment, the schemes can be implemented with board level BIST capability. Test vectors can be generated using simple hardware and responses can be compacted outside the boundary scan chain in a test and diagnosis controller (TDC) on the board under test. Although this approach cannot utilize the full potential of the proposed schemes, an on-board BIST approach removes the need for ATE, and detection and diagnosis remain simple, local and structure independent. Thus, this dissertation shows the potentials of the boundary scan architecture in the context of interconnect testing. Schemes are proposed which exploit the potentials of the boundary scan environment. It is for the designer to consider the different trade-offs and to select the scheme which suits the needs and resources.

This dissertation addresses board level interconnect testing. But, the testing of interconnects at various levels like in large area ICs, wafer scale systems are as important problems as the problem of board level interconnect testing. Interestingly, the concepts developed here are equally applicable to those areas of interconnect testing. The basic idea is to isolate the interconnects from the components (functional blocks, ICs etc.) that they connect together. These isolated interconnects must be accessed to apply test vectors, to collect responses, etc. Once such a framework is available, the proposed schemes are applicable for testing at any level of interconnects under consideration.

The test schemes developed in this dissertation are complete in the sense that implementation issues are fully addressed. However, the issues are discussed at a conceptual level. Therefore, as a future extension, it would be interesting to design these schemes at the detailed circuit level and to implement into silicon.

Only one type of wired interconnect, the tri-state interconnect, is addressed in this dissertation. There are other types of wired interconnects like wired-OR and wired-AND interconnects. The behavior of these interconnects are more like OR and AND gates respectively, and not like the simple wires. Therefore, testing these wired interconnects is different from the testing of the interconnects discussed here. In future, the developed schemes could be extended to test all types of interconnects including the said wired interconnects.

To conclude, this dissertation has taken a completely new and novel approach towards testing board level interconnects in the context of the boundary scan architecture. A number of interconnect testing problems are elegantly solved and several efficient testing schemes are proposed using the board level BIST approach.

References

[Abazi 86] Z. Abazi and P. Theyenod-Foss, *Markov Models for the Random Testing Analysis of Cards*, Proc. of FTCS 16, July 1986, pp. 272-277.

[Agarwal 81] V.K. Agarwal and E. Cerny, *Store and Generate Built-In-Testing Approach*, Proc. of FTCS 11, June 1981, pp. 35-40.

[Ando 80] H Ando, *Testing VLSI with Random Access Scan*, Digest of Papers, COMPCON 1980, pp. 50-52.

[Aranoff 81] S. Aranoff and Y Abulaffio, *Routing of Printed Circuit Boards*, Proc. of 18th Design Automation Conf., 1981, pp. 130-136.

[Armstrong 72] D.B. Armstrong, *A Deductive Method for Simulating Faults in Logic Circuits*, IEEE Trans. on Comp., Vol. C-21, No. 5, May 1972, pp. 464-471.

[Avra 82] L. Avra, *A VHSIC ETM-Bus Compatible Test and Maintenance Interface*, Proc. of Int'l. Test Conf. 1982, pp. 83-90.

[Ballew 89] W.D. Ballew and L.M. Streb, *Board-Level Boundary-Scan : Regaining Observability with an Additional IC*, Proc. of Int'l. Test Conf. 1989, pp. 182-189.

[Bardell 82] P.H. Bardell and W.H. McAnney, *Self-Testing of Multichip Logic Modules*, Proc. of Int'l. Test Conf. 1982, pp. 200-204.

[Bardell 87] P.H. Bardell, W.H. McAnney and J. Savir, *Built-In Self-Test for VLSI : Pseudorandom Techniques*, Publised by John Wiley & Sons Inc., 1987.

[Bateson 85] J. Bateson, *In-Circuit Testing*, Published by Van Nostrand Reinhald Company Inc. 1985.

[Beenker 85] F.M. Beenker, *Systematic and Structured Methods for Digital Board Testing*, Proc. of Int'l. Test Conf. 1985, pp. 380-385.

[Beenker 86] F.M. Beenker et. al., *Macro Testing : Unifying IC and Board Testing*, IEEE Design and Test of Computers, Dec. 1986, pp. 26-32.

[Bennetts 82] R.G. Bennetts, *Introduction to Digital Board Testing*, Published by Crane Russard & Company, Inc., 1982.

[Bose 82] A. K. Bose et.al., *A Fault Simulator for MOS LSI Circuits*, Proc. of 19th Design Automation Conf. 1982, pp. 400-409.

[Breuer 72] M.A. Breuer et. al., *Design Automation of Digital Systems*, Prentice Hall, 1972.

[Brglez 89] F. Brglez, C. Gloster and G. Kedem, *Hardware-Based Weighted Random Pattern Generation for Boundary Scan*, Version 1.0, Technical Report TR 89-11.

[Budde 88] W.O. Budde, *Modular Testprocessor for VLSI Chips and High-Density PC Boards*, IEEE Trans. on CAD of IC and Systems, Oct. 1988, Vol. 7, No. 10, pp. 1118-1124.

[Carter 82] W.C. Carter, *The Ubiquitous Parity Bit*, Proc. of 12th Int'l. Symp. on Fault-Tolerant Computing 1982, pp. 289-296.

[Chang 70] H.Y. Chang, E.C. Manning and G. Metze, *Fault Diagnosis of Digital Systems*, Published by Wiley-Interscience, N.Y., 1970.

[Dahbura 89] A.T. Dahbura, M.U. Uyar and C.W. Yau, *An Optimal Test Sequence for the JTAG/IEEE P1149.1 Test Access Port Controller*, Proc. of Int'l. Test Conf. 1989, pp. 55-62.

[DasGupta 78] S. DasGupta, E.B. Eichelberger and T.W. Williams, *LSI Chip Design for Testability*, Proc. of Int'l. Solid State Conf. 1978, pp. 216-217.

[DasGupta 81] S. DasGupta, R.G. Walther, T.W. Williams and E.B. Eichelberger, *An Enhancement to LSSD and some Applications of LSSD in Reliability, Availability and Serviceability*, Proc. of FTCS 11, June 1981, pp. 32-34.

[Daugherty 89] D.W. Daugherty, *The Sources of Connectivity Data for Bare-Board Testing*, Electronics Test Magz., April 1989, pp. 22-24.

[Design & Test 85] IEEE Design & Test Magazine, Vol. 2, No. 2, April 1985.

[Dion 87] J. Dion, *Fast Printed Circuit Board Routing*, Proc. of 24th Design Automation Conf. 1987, pp. 724-734.

[Eichelberger 78] E.B. Eichelberger and T.W. Williams, *A Logic Design Structure for LSI Testability*, Journal of Design Automation and Fault-Tolerant Computing, Vol. 2, No. 2, May 1978, pp. 165-178.

[Feugate 88] R.J. Feugate, Jr., and S.M. McIntyre, *Introduction to VLSI Testing*, Published by Prentice Hall, 1988.

[Frohwerk 77] R.A. Frohwerk, *Signature Analysis : A New Digital Field Service Method*, Hewlett-Packard Journal, May 1977, pp. 2-8.

[Fujiwara 83] H. Fujiwara and T. Shimono, *On the Acceleration of Test Generation Algorithms*, IEEE Trans. on Comp., Vol. C-32, No. 12, Dec. 1983, pp. 1137-1144.

[Funatsu 75] S. Funatsu, N. Wakatsuki and T. Arima, *Test Generation Systems in Japan*, Proc. of 12th Design Automation Symp. 1975, pp. 114-122.

[Ghate 87] P.B. Ghate, *Reliability of VLSI Interconnections*, Proc. of Int'l. Conf. on Computer Design 1987, pp. 240-243.

[Gloster 87] C. Gloster, Jr., and F. Brglez, *Integration of Boundary Scan with Cellular-Based Built-In Self-Test for Scan Based Architectures*, Version 1.0, Technical Report TR 87-18.

[Goel 80] P. Goel, *Test Cost Analysis and Projections*, Proc. of IEEE Design Automation Conf. 1980, pp. 77-82.

[Goel 81] P. Goel, *An Implicit Enumeration Algorithm to Generate Tests for Combinational Logic Circuits*, IEEE Trans. on Comp., Vol. C-30, No. 3, March 1981, pp. 215-222.

[Goel 82] P. Goel and M.T. MaMahon, *Electronic Chip-In-Place Test*, Proc. of Int't. Test Conf. 1982, pp. 83-90.

[Golomb 82] S.W. Golomb, *Shift Register Sequences*, Published by Aegean Park Press, 1982.

[Goshina 81] S. Goshina, *Diagnostic System for Large Scale Logic Cards and LSIs*, Proc. of 18th Design Automation Conf. 1981, pp. 256-259.

[Gruetzner 89] M. Gruetzner and H. Grabinski. *Simulation and Test of Faults in WSI Interconnect Systems*. Proc. of Int'l. Conf. on WSI 1989, pp. 345-354.

[Halliday 89] A. Halliday, G. Young and A. Crouch, *Prototype Testing Simplified by Scannable Buffers and Latches*, Proc. of Int'l. Test Conf. 1989, pp. 174-181.

[Hansen 89a] P. Hansen, *Tough Board Test Problem Solved with Boundary Scan*, Electronics Test Magz., June 1989.

[Hansen 89b] P. Hansen, *Testing Conventional Logic and Memory Clusters Using Boundary Scan Devices as Virtual ATE Channels*, Proc. of Int'l. Test Conf. 1989, pp. 166-173.

[Hassan 84] S.Z. Hassan and E.J. McCluskey, *Increased Fault Coverage Through Multiple Signatures*, Proc. of 14th Int'l. Symp. on Fault-Tolerant Computing 1984, pp. 354-359.

[Hassan 88] A. Hassan, J. Rajski and V.K. Agarwal, *Testing and Diagnosis of Interconnects using Boundary Scan Architecture*, Proc. of Int'l. Test Conf. 1988, pp. 126-137.

[Hassan 89] A. Hassan, V.K. Agarwal, J. Rajski and B. Nadeau-Dostie, *Testing of Glue Logic Interconnects using Boundary Scan Architecture*, Proc. of Int'l. Test Conf. 1989, pp. 700-711.

[Hayes 75] J.P. Hayes, *Transition Count Testing of Combinational Logic Circuits*, IEEE Trans. on Comp., June 1975, C-25, pp. 613-620.

[Hayes 76] J.P. Hayes, *Check Sum Test Methods*, Proc. of 6th Int'l. Symp. on Fault-Tolerant Computing 1976, pp. 114-119.

[Howells 88] M. Howells, R. Aitken, V.K. Agarwal, *Defect Tolerant Interconnects for VLSI*, Proceedings of International Workshop on Defect and Fault-Tolerance in VLSI Systems, Paper No. 2.2, Oct. 1988.

[Hsu 81] F. Hsu, P. Solecky and R. Beaudoin, *Structural Trace Diagnosis for LSSD Board Testing - An Alternative to Full Fault Simulated Diagnosis*, Proc. of 18th Design Automation Conf. 1981, pp. 891-897.

[Hurst 85] S.L. Hurst. D.M. Miller and J.C. Muzio. *Spectral Techniques in Digital Logic.* Published by Academic Press. London. 1985.

[Jarwala 89] N. Jarwala and C.W. Yau. *A New Framework for Analyzing Test Generation and Diagnosis Algorithms for Wiring Networks.* Proc. of Int'l. Test Conf. 1989. pp. 63-70.

[Joseph 85] R.L. Joseph. *An Expert Systems Approach to Completing Partially Routed Printed Circuit Boards.* Proc. of 22nd Design Automation Conf. 1985. pp. 523-528.

[JTAG 88] JTAG Boundary Scan Architecture Standard Proposal. Version 2.0. Published on 30 March 1988.

[Katevenis 85] M.G.H. Katevenis and M.G. Blatt. *Switch Design for Soft-Reconfigurable WSI Systems.* Proc. of MIT Chapel Hill Conf. on VLSI. 1985. pp. 197-219.

[Kautz 74] W.H. Kautz. *Testing of Faults in Wiring Networks.* IEEE Trans. on Comp.. Vol. C-23. No. 4. April 1974. pp. 358-363.

[Kitchen 74] W.J. Kitchen and A.R. Alvarez. *BI-CMOS for Computer Applications.* Proc. of IEEE Int'l. Conf. on Computer Design 1987. pp. 244-249.

[Koenemann 79] B. Koenemann. H. Mucha and G. Zwiehoff. *Built-In Logic Block Observation Techniques.* Proc. of Int.l. Test Conf. 1979. pp. 37-41.

[Kulkarni 78] K.G. Kulkarni and A. Prabhakar. *Automated Routing of Printed Circuit Boards.* Journal of Inst. for Elec. and Tele. Eng. 1978. Vol. 24. 158-163.

[Krasniewski 87] A. Krasniewski and S. Pilarski. *Circular Self-Test Path : A Low-Cost BIST Technique.* Proc. of 24th Design Automation Conf. 1987. pp. 407-415.

[Levendel 81] Y.H. Levendel and P.R. Menon. *Fault Simulation Methods - Extensions and Comparisons.* Bell System Tech. Journal. Vol. 60. No. 9. Nov. 1981. pp. 2235-2258.

[Maly 87] W. Maly. *Realistic Fault Modeling for VLSI Testing.* Proc. of 24th Design Automation Conf. 1987. pp. 173-180.

[Maamari 88] F. Maamari and J. Rajski. *A Fault Simulation Method Based on Stem Regions.* Proc. of Int'l. Conf. on Computer-Aided Design. 1988.

[Mangir 83] T. Mangir. *Design for Testability : An Integrated Approach to VLSI Testing.* Proc. of Int'l. Conf. on Comp. Aided Design 1983. pp. 68-70.

[Mann 80] W.R. Mann. *Microelectronic Device Test Strategies - A Manufacturer's Approach.* Proc. of Int'l. Test Conf. 1980. pp. 195-202.

[Maunder 87] C. Maunder and F. Beenker *Boundary Scan : A Framework for Structural Design for Test.* Proc. of Int'l. Test Conf. 1987. pp. 714-723.

[McCluskey 85] E.J. McCluskey. *Built-In Self Test Techniques*. IEEE Design & Test Magazine. Vol. 2. No. 2. April 1985. pp. 21-28.

[McCluskey 86] E.J. McCluskey. *Logic Design Principles with Emphasis on Testable Semicustom Circuits*. Published by Prentice Hall. 1986.

[Miller 84] D.M. Miller and J.C. Muzio. *Spectral Signatures Fault Signatures for Single Stuck-At Faults in Combinational Networks*. IEEE Trans. on Comp., Vol. C-33. No. 8. Aug. 1984. pp. 765-769.

[Muto 88] T.Muto. *Some Reliability Problems of Surface-Mounted Devices* . IEEE Circuits and Devices Magazine. July 1988. pp. 9 15.

[Muzio 83] J.C. Muzio and D.M. Miller. *Spectral Fault Signatures for Internally Unate Combinational Networks*. IEEE Trans. on Comp., Vol. C-32. 1983. pp. 382-395.

[Myers 83] M.A. Myers. *An Analysis of the Cost and Quality Impact of LSI-VLSI Technology on PCB Test Strategies*. Proc. of Int'l. Test Conf. 1983. pp. 382-395.

[Naveda 86] J.F. Naveda. K.C. Chang and H.C. Du. *A New Approach to Multi-Layer PCB Routing with Short Vias*. Proc. of 23rd Design Automation Conf. 1986. pp. 696-701.

[Nesbat 80] S. Bozorgui-Nesbat and E.J. McCluskey. *Structured Design for Testability to Eliminate Test Pattern Generation*. Proc. of FTCS 10. pp. 158-163.

[Palusinski 88] O.A. Palusinski. *Modeling and Simulation of Coupled Lossy Lines for VLSI Interconnections*. Proc. of Int'l. Conf. on Comp. Design 1988. pp. 82-86.

[Pardee 88] S. Pardee and T.P. Pennino. *Printed-Circuit Board Design*. IEEE Spectrum. Sept. 1988. pp. 40-44.

[Parker 87] K.P. Parker. *Integrating Design and Test : Using Tools for ATE Programming*. Published by Computer Society Press of IEEE 1987.

[Parker 89] K.P. Parker. *The Impact of Boundary Scan on Board Test*. IEEE Design and Test of Comp., Aug. 1989. pp. 18-30.

[Peterson 72] W.W. Peterson and E.J. Weldon. *Error Correcting Codes*. 2nd Edition. Published by MIT Press. Cambridge. 1972.

[Pradhan 87] M. Pradhan. R.E. Tullos. H. Bleeker and F. Beenker. *Developing a Standard for Boundary Scan Implementation*. Proc. of Int'l. Conf. on Comp. Design 1987. pp. 462-466.

[Prang 89] J.A. Prang. *Moving Testability Issues Forward in the Design Cycle*. Electronics Test Magz., April 1989. pp. 38-42.

[Ramachandran 86] V. Ramachandran. *On Driving Many Long Wires in a VLSI Layout*. Journal of ACM. Vol. 33. No. 4. Oct. 1986. pp. 687-701.

[Riessen 89] R.P. Van Riessen, H.G. Kerkhoff and A. Kloppenburg, *Design and Implementation of a Hierarchical Testable Architecture*, Proc. of European Test Conf 1989, pp. 112-118.

[Roth 67] J.P. Roth, W.G. Bouricuis and P.R. Schneider, *Programmed Algorithms to Compute Tests to Detect and Distinguish Between Failures in Logic Circuits*, IEEE Trans. on Comp., Vol. C-16, No. 10, Oct. 1967, pp. 567-589.

[Savir 80] J. Savir, *Syndrome Testable Design of Combinational Circuits*, IEEE Trans. on Comp., June 1980, C-29, No. 6, pp. 442-451.

[Saxena 86] N.R. Saxena and J.B. Robinson, *Accumulator Compression Testing*, IEEE Trans. on Comp., Vol. C-35, No. 4, April 1986, 317-321.

[Signetics 74] Signetics IC Manual, Signetics Corporation, 1974.

[Susskind 81] A.K. Susskind, *Testing by Verifying Walsh Coefficients*, Proc. of 11th Int'l. Symp. on Fault-Tolerant Computing 1981, pp. 206-208.

[Stewart 77] J.H. Stewart, *Future Testing of Large LSI Circuit Cards*, Proc. of 1977 Semiconductor Test Symposium, pp. 6-15.

[Tullos 83] R.E. Tullos, *Automatic Board Testing : Coping with Complex Circuits*, IEEE Spectrum, July 1983, pp. 38-43.

[Tullos 89] R.E. Tullos and C.W. Yau, *BIST and Boundary Scan for Board Level Test : Test Program Pseudocode*, Proc. of European Test Conf. 1989, pp. 106-111.

[Turino 88] J. Turino, *IEEE P1149 Proposed Standard Testability Bus - An Update with Case Histories*, Proc. of Int'l. Conf. on Comp. Design 1988, pp. 334-337.

[Vereen 89a] L. Vereen, *Component Test : Bigger Problems, Better Tools*, Electronics Test Magz., Jan. 1989, pp. 40-44.

[Vereen 89b] L. Vereen, *Board Test : Facing the Challenges of ASICs and Surface Mount*, Electronics Test Magz., Jan. 1989, pp. 46-59.

[Vereen 89c] L. Vereen, *Board Tester Supports Boundary Scan*, Electronics Test Magz., June 1989, pp. 30-31.

[Vining 89] S. Vining, *Trade-Off Decisions Made for P1149.1 Controller Design*, Proc. of Int'l. Test Conf. 1989, pp. 47-54.

[Wagner 87] P.T. Wagner, *Interconnect Testing with Boundary Scan*, Proc. of Int'l. Test Conf. 1987, pp. 52-57.

[Wang 89] L.T. Wang, M. Marhoefer and E.J. McCluskey, *A Self-Test and Self-Diagnosis Architecture for Boards using Boundary Scan*, Proc. of European Test Conf. 1989, pp. 119-126.

[Whetsel 88] L. Whetsel, *A Proposed Standard Test Bus and Boundary Scan Architecture*, Proc. of Int'l. Conf. on Comp. Design 1988, pp. 330-333.

[Wilkins 88] B.R. Wilkins, *HIT : A Standard Constructional System for Testability and Maintainability*, Proc. of Int'l. Conf. on Comp. Design 1988, pp. 338-343.

[Williams 79] T.W. Williams and K.P. Parker, *Testing Logic Networks and Designing for Testability*, Computer Magz., Oct. 1979, pp. 9-21.

[Williams 82] T.W. Williams and K.P. Parker, *Design for Testability - a Survey*, IEEE Trans. on Comp., Vol. C-31, No. 1, Jan. 1982, pp. 2-15.

[Yau 89] C.W. Yau and N. Jarwala, *A Unified Theory for Designing Optimal Test Generation and Diagnosis Algorithms for Board Interconnects*, Proc. of Int'l. Test Conf. 1989, pp. 71-77.

[Young 85] I.A. Young, R.A. Gasser, Jr., and G.H. Kawamoto and T.L. Deeter, *Interconnect Considerations for 1μm CMOS Technology*, Proc. of Int'l. Conf. on Comp. Design 1985, pp. 155-158.

[Zorian 86] Y. Zorian and V.K. Agarwal, *A General Scheme to Optimize Error Masking in Built-In Self-Testing*, Proc. of 18th Int'l. Symp. on Fault-Tolerant Computing 1986, pp. 410-415.