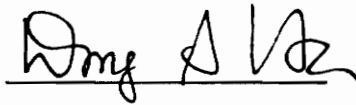


**SELECTION OF FLIP-FLOPS FOR PARTIAL SCAN PATHS BY USE OF A
STATISTICAL TESTABILITY MEASURE**

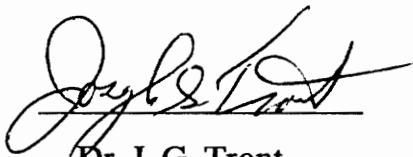
by
David B. Jett

Thesis submitted to the Faculty of the
Virginia Polytechnic Institute and State university
in partial fulfillment of the requirements for the degree of
Master of Science
in
Electrical Engineering

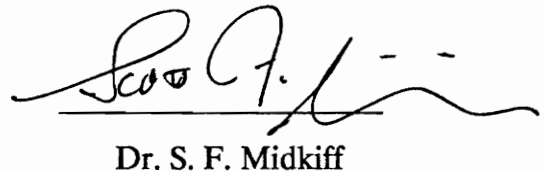
APPROVED:



Dr. D. S. Ha, Chairman



Dr. J. G. Tront



Dr. S. F. Midkiff

September 1992
Blacksburg, Virginia

C.2

LD
5655
V855
1992
J498
C.2

SELECTION OF FLIP-FLOPS FOR PARTIAL SCAN PATHS BY USE OF A STATISTICAL TESTABILITY MEASURE

by

D. B. Jett

Virginia Polytechnic Institute and State University

Blacksburg, VA

(ABSTRACT)

Partial scan paths improve the testability of digital circuits, and incur minimal costs in IC area overhead and test application time. Design constraints may require that a partial scan path include only those flip-flops that provide the greatest improvements in circuit testability. STAFFS, a tool that identifies such flip-flops, has been developed. It uses a statistical testability measure to acquire quantitative data for the controllabilities and observabilities of the nodes of a circuit. It predicts the changes that would occur in the data due to the scanning of specific flip-flops, and uses those predictions to select flip-flops. STAFFS weights the observability data versus the controllability data when selecting flip-flops, and it can efficiently select alternative scan designs for different weights. Experimental results for thirteen sequential benchmark circuits reveal that STAFFS consistently selects scan designs with fault coverages that are significantly higher than those of arbitrarily selected scan designs.

ACKNOWLEDGEMENTS

Researching and producing this thesis has been a long, difficult, but rewarding project, and a long list of friends have, through their support of my efforts, earned a piece of it. Simple thanks can only begin to reward these people for what they have given me. My parents and family have given me support and patience far above and beyond the call of duty. J. S. and Amy Brown and Julie Chadburn have been exceptionally generous and hospitable during my stays in Blacksburg. Hyung Lee provided me with his fault simulator HOPE, a fundamental tool for this research. Dr. W. F. Fabrycky and Dinesh Verma have generously supplied me with word processing facilities. My committee members have offered their time and efforts to reviewing my work. And of course, my brother Dave and my sister Sharon, having been through it themselves, have given me the advice, the sympathy, and occasionally the backbone that I've needed.

THANK YOU ALL.

Table of Contents

1.0	Introduction.....	1
2.0	Background.....	5
2.1	Scan Path Architecture.....	5
2.2	Partial Scan Paths.....	10
2.3	Flip-flop Selection Methods.....	13
2.3.1	Ad-hoc Methods.....	13
2.3.2	Local Circuit Topology Methods.....	14
2.3.3	Test Set Optimization Methods.....	14
2.3.4	Graph Theoretic Methods.....	15
2.3.5	Methods Using Testability Measures.....	16
2.4	Review of STAFAN.....	17
2.5	Summary.....	21
3.0	STAFFS: STatistics Aided Flip-Flop Selection.....	23
3.1	STAFFS Architecture.....	23
3.2	Data Collection.....	25
3.3	Testability Predictions.....	28
3.4	Selection of Flip-flops.....	31
3.5	Summary.....	34
4.0	Tests and Experimental Results.....	35
4.1	Evaluation Method.....	35
4.2	Tools Developed.....	37
4.2.1	STAFFS.....	37
4.2.2	CV.....	38

4.3	Experimental Environment.....	40
4.3.1	Test Circuits.....	40
4.3.2	Scan Path Length.....	41
4.3.3	Number of Test Patterns.....	42
4.3.4	Size of Fault Coverage Database.....	45
4.3.5	Weighting Factor.....	45
4.4	Experimental Results.....	46
4.4.1	Overall Effectiveness of STAFFS.....	46
4.4.2	Weighting Factor α	49
4.5	Concluding Remarks.....	55
5.0	Conclusion.....	57
	REFERENCES.....	60
	APPENDICES.....	63
	VITA.....	68

Figures

Figure 2.1(a).	Sequential circuit.....	7
Figure 2.1(b).	Sequential circuit with dashed scan path..	7
Figure 3.1.	Circuit fragment with loop.....	27
Figure 3.2.	STAFFS C'0 equations.....	30
Figure 4.1.	Random pattern fault coverage profiles for test circuits.....	44
Figure 4.2.	Fault coverages of scan designs.....	51

Tables

Table 4.1.	Test circuits.....	40
Table 4.2.	Fault coverages of selections, $\alpha = 0.5$	47
Table 4.3.	Fault coverages of scan designs.....	50

CHAPTER 1: INTRODUCTION

Testing of integrated circuits is relatively simple up to the LSI level. Generation of test patterns for a circuit, and then application of the set of test patterns to the circuit, can be performed in a reasonable amount of time. However, for circuits at the VLSI and ULSI levels, several problems arise in testing (Abramovici, et. al. [1990]). First, test pattern generation time is long due to the high complexity of the circuits. It is common for test pattern generation for VLSI circuits to take more than one day even for today's most powerful workstations. Second, the high circuit complexity makes it difficult to achieve high fault coverage. Low fault coverage poses serious problems for applications that require high reliability such as military and space applications. Third, the test application time is large due to a high volume of test patterns. The number of functional test patterns usually supplied by logic designers often exceeds 10,000 patterns.

One approach that alleviates these problems is to employ design-for-testability techniques at the early design stage. Such techniques enhance the controllability and observability of circuits through modification of the existing circuit blocks, or by the addition of extra circuits and pins. There are two different approaches for

design-for-testability techniques: ad-hoc methods and structured methods. Ad-hoc methods target specific testability problems and solve them locally. For example, critical nodes such as fanout points can be connected directly to unused circuit pins to enhance their controllabilities and observabilities, or hard-to-test circuit blocks can be partitioned into semi-independent subblocks that are easier to test. Ad-hoc methods suffer several drawbacks. Adding test points at critical nodes is limited by the number of spare circuit pins, and the benefit to be gained by partitioning a circuit depends on the skill and intuition of the designer. Structured methods, on the other hand, address testability problems globally and are not targeted at specific circuit structures. The most widely accepted structured methods are scan designs.

Scan designs modify and rewire the flip-flops in a circuit so that, with the circuit in a test mode, they are reconfigured to form a scan path, a shift register with its ends connected to circuit pins. During testing, test patterns are shifted into the scan path and applied to internal circuit nodes, and test results are captured by the scan path flip-flops and shifted out for observation. The technique has three major drawbacks, however: high area overhead, problems with long testing times, and potential system performance degradation. In full scan design every

flip-flop is included in a scan path. This simplifies test generation for the circuit, since in test mode it is transformed into a combinational circuit. An alternative is partial scan design, in which only a subset of the circuit flip-flops are scanned. Partial scan design alleviates the three drawbacks of full scan design. It can provide fault coverage comparable to that of complete scan design. The main problem with partial scan design is selecting the flip-flops to include in the scan path. A set of scan flip-flops that does not rank among the most optimal of the possible sets incurs costs in IC area and fault coverage over sets that are optimal or near-optimal.

This thesis investigates the problem of selecting flip-flops for partial scan designs. We studied the use of a testability measure as a basis for selecting flip-flops. We propose a method that identifies and selects flip-flops that maximize the global testability of a circuit. We developed a CAD tool called STAFFS (*STatistics Aided Flip-Flop Selection*) that employs the proposed method. Our experimental results demonstrate that the proposed method is effective in scan design.

The thesis is organized as follows. In Chapter 2 we describe the scan path architecture and discuss critical issues related to partial scan design. We review previous investigations on selecting scan flip-flops. We also review

STAFAN, the testability measure that is the basis for our work. In Chapter 3 we describe our method of selecting the flip-flops for a partial scan path. In Chapter 4 we describe the procedure and the environment for the experiments we conducted to test the effectiveness of the proposed method. Chapter 5 concludes the thesis.

CHAPTER 2: BACKGROUND

Scan design is a design-for-testability technique that alleviates difficulties in testing large sequential integrated circuits. In scan design, flip-flops are modified so they can be reconfigured to form a shift register during a test mode. This improves the testability of internal circuit nodes. In Section 2.1 we describe the scan path architecture, and we describe its advantages and disadvantages. In Section 2.2 we present partial scan design, a technique that alleviates the drawbacks of full scan design. We also introduce the main difficulty with the technique: how to identify the flip-flops that can provide the greatest improvements in testability for the circuit. Failure to identify and select such flip-flops for the scan path may result in a circuit that fails to meet testability specifications. In Section 2.3 we describe various selection methods that have been previously investigated, and we introduce our approach of using a testability measure as a guide to flip-flop selection. In Section 2.4 we briefly review STAFAN, the testability measure that is the basis of our work.

2.1 SCAN PATH ARCHITECTURE

In the basic scan design architecture, circuit flip-

flops are modified so that with the circuit in test mode, they are reconfigured as a shift register, or scan path, whose inputs and outputs are connected to external pins. The improved access to the flip-flops improves the controllability and observability, and hence the testability, of nodes near the scan path. Drawbacks of the technique are the additional circuit area needed for the flip-flop modifications, the longer testing time needed for shifting operations, and the potential degradation of system performance due to additional signal propagation delays.

The basic scan design architecture (Williams and Angell [1973]) is depicted in Figure 2.1(b). Every flip-flop in the scan path is modified to have a 2-input multiplexer at the flip-flop input. One multiplexer input is connected to the original input node of the flip-flop. The other multiplexer input is connected to the output of another scan flip-flop. All flip-flops are connected in this way to form a shift register. The extra multiplexer input of the first flip-flop in the scan path is connected to a circuit pin SI. The output of the last flip-flop in the scan path is connected to a circuit pin SO. SI and SO are then the serial input and the serial output, respectively, of the shift register. The additional signal TEST controls all multiplexers. If TEST is low, the circuit is configured in its normal mode of operation in which the flip-flops operate

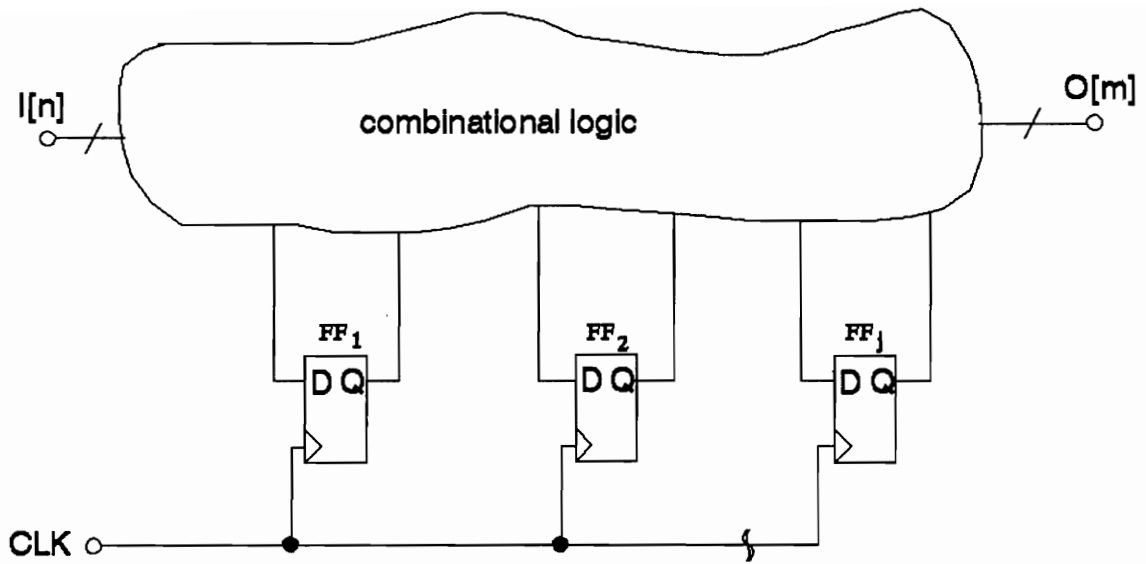


Figure 2.1(a). Sequential circuit.

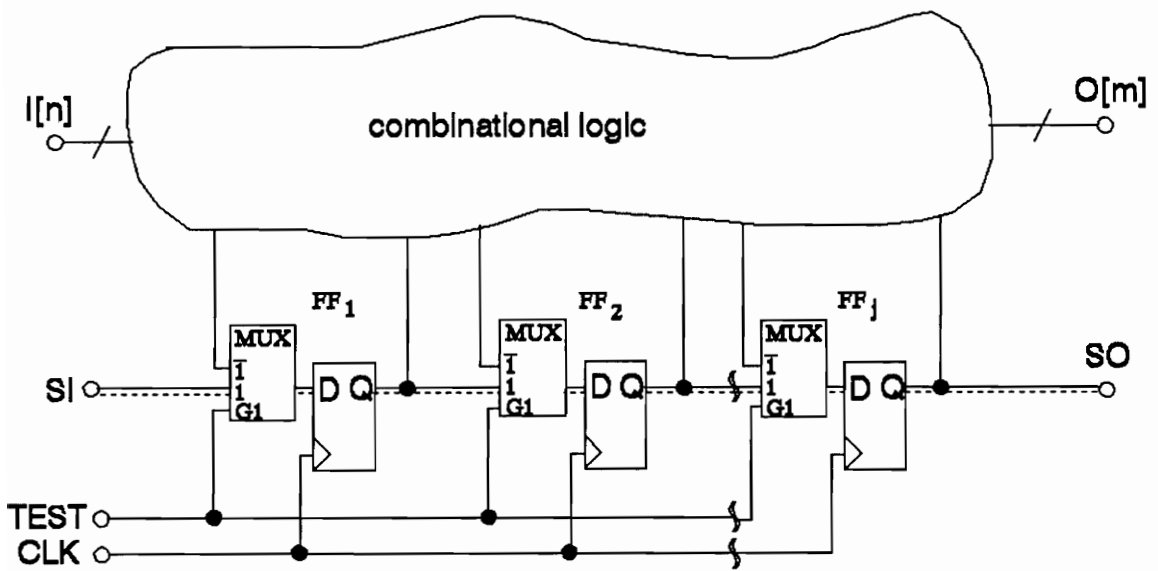


Figure 2.1(b). Sequential circuit with (dashed) scan path.

as in the original unmodified circuit. If TEST is high, the circuit is configured in its test mode, and the shift register is active. A single test is conducted with the following procedure.

1. Set TEST = 1.
2. Apply j bits of the test pattern to SI and shift them into the scan path with j pulses on CLK.
3. Set TEST = 0.
4. Apply n bits of the test pattern to the regular circuit inputs.
5. Observe m bits of the test result at the circuit outputs.
6. Apply one pulse to CLK.

At this point j bits of the test result are stored in the flip-flops. Those j bits are shifted out and observed as the subsequent test pattern is shifted into the scan path.

By improving the controllability and observability of internal nodes, a scan path improves the overall circuit testability. It thus improves fault coverage and/or reduces the number of test patterns. It also eases the workload of Automatic Test Pattern Generators (ATPGs), reducing test generation times. The technique has three major drawbacks, however. First, the time needed to apply a single test becomes longer due to the shifting operations. Second, the

area of the circuit increases due to the added multiplexers and interconnections. A full scan path implementation may increase the area of an IC by 30 percent (Agrawal, et. al. [1987]). Third, the performance of the circuit may degrade, since the multiplexers increase propagation delays at the flip-flop inputs. The added delays on critical timing paths may degrade the circuit's performance.

Variations on the basic scan path architecture have been proposed. Level Sensitive Scan Design (LSSD), proposed by Eichelberger and Williams (1977), is applicable to latch-based systems. It replaces each system latch with a two-latch element. It requires a two-phase clock design and strict operational rules to prevent races for the level-sensitive latches. The Scan/Set architecture proposed by Stewart (1977) also applies to latch-based systems. It implements the scan path as a completely separate shift register. The register outputs are connected in parallel to the multiplexed-input system latches. Scan/Set provides greater operational flexibility but requires more circuit area. The Multiple Scan Path technique (Williams and Angell [1973]) replaces a single scan path with many shorter ones. This reduces the time required for shifting patterns into the scan path. It may moderately reduce the area required for the scan path interconnections.

All of these variations have advantages and

disadvantages compared with the basic scan architecture. Our research focusses specifically on the basic architecture as described, but it is applicable to all of these variations in a straightforward manner.

2.2 PARTIAL SCAN PATHS

In full scan design, all of the flip-flops of a circuit are included in the scan path. Full scan design improves a circuit's testability, reduces the test pattern set size, and reduces test generation time. It has the drawbacks of increasing application time for test patterns, increasing the size of the circuit, and degrading the circuit's performance.

In partial scan design (Trischler [1980]), not all of the flip-flops in the circuit are included in the scan path. The disadvantages of the full scan architecture are consequently alleviated. The fewer flip-flop modifications and interconnections impose a smaller cost in circuit area. The shorter scan path incurs a smaller increase in test application time. And, if flip-flops on critical timing paths are excluded from the scan path, degradation of circuit performance can be avoided. Furthermore, the greatest marginal improvements in testability are obtained with the first flip-flops selected for a scan path. Trischler (1983) reported that scan paths comprising 15 to

25 percent of the circuit flip-flops usually provide testability improvements comparable to those obtained with full scan paths. Therefore, short partial scan paths can provide most of the testability benefits of full scan paths while incurring only a small fraction of their costs.

The most difficult problem with implementing a partial scan path is selecting the specific scan flip-flops. Since all scan flip-flops influence circuit testability by different amounts, every distinct set of scan flip-flops provides a characteristic level of improvement in circuit testability. These sets and their corresponding scan paths can be qualitatively ordered. Scan paths of a given length can be ordered by the testability improvements they provide. Alternatively, scan paths that provide a given improvement in testability can be ranked by their lengths or their corresponding costs in circuit area overhead. Such ordering implies that an optimal scan path exists for any partial scan path design problem. If a design is constrained by the allowable cost in circuit area, i.e. the scan path length is fixed, then the optimal scan path of that length provides the greatest improvement in circuit testability of all possible ones. If the design is constrained by the required level of testability, then the optimal scan path is the shortest one that provides that testability.

The variability in the testability improvements

provided by all possible scan paths of a given length is important. If the variability for a circuit is small, then a randomly selected set of flip-flops would be likely to give testability benefits close to those of the optimal scan path, at very little cost in design effort. Little benefit would be gained by searching for an optimal or near-optimal scan path. Most circuits, however, exhibit a significant variability in the testability improvements provided by the possible scan paths. For IC designs with demanding constraints on both fault coverage and area overhead, an optimal or near-optimal scan path may be required for the circuit to satisfy these constraints.

Finding a satisfactory scan path by checking all alternatives is impractical. Selecting m flip-flops out of n total circuit flip-flops yields

$$n! / [m! * (n-m)!]$$

distinct sets. For even a small MSI-sized circuit, for example, with $n = 21$ total flip-flops, designing a scan path with $m = 4$ flip-flops involves choosing among 5985 distinct sets. A guide is needed for identifying the flip-flops that provide the best testability improvements.

2.3 FLIP-FLOP SELECTION METHODS

Many methods for selecting scan flip-flops have been previously investigated. In this section we present a broad outline of existing approaches.

2.3.1 AD-HOC METHODS

Ad-hoc methods (Abramovici, et. al. [1990], Pradhan, D. K. [1990], McCluskey [1986]) for selecting or rejecting flip-flops identify them using easy, "by-inspection" criteria. Flip-flops with high fanin or fanout can be selected for the large number of nodes they affect. Flip-flops on critical timing paths can be rejected if their effects on signal timing are severe enough that the circuit fails to meet design specifications. Flip-flops on datapaths and buses, and ones at circuit inputs and outputs, can be rejected. Such flip-flops are inherently accessible, and so they can provide little improvement to the controllabilities and observabilities of nearby nodes.

These techniques apply to all scan path designs. They do not usually offer a complete solution to the scan flip-flop selection problem. Instead, they represent a first phase of the design process. By using these "by-inspection" methods first, a designer may substantially shrink the workload for a subsequent, more formal method.

2.3.2 LOCAL CIRCUIT TOPOLOGY METHODS

Local circuit topology methods rely on the testability characteristics of the specific topologic structures encountered in a circuit. Pradhan, et. al. [1988] proposed two methods. One method ranked each flip-flop by the number of other flip-flops at the base of its combinational input cone. It selected flip-flops that ranked highest. The second method considered reconvergent fanout pathways between flip-flop pairs. It ranked each flip-flop by the number of such fanout pathways involving the flip-flop, and selected flip-flops that ranked highest. Abramovici, et. al. [1991] perform a logic analysis on a circuit to determine, for each flip-flop, the number of faults that become uncontrollable or unobservable if the flip-flop is not scanned. They select flip-flops with the largest values for this metric, and that therefore should yield the highest fault coverage.

2.3.3 TEST SET OPTIMIZATION METHODS

Test set optimization methods (Agrawal, et. al. [1987], Ma, et. al. [1988]) collect sets of possible test patterns that are associated with specific scan flip-flops. Flip-flop selections are then made so as to optimize the test pattern set. Agrawal, et. al. [1987] proposed a "frequency

method" that generates all single pattern tests for a set of faults in a fully scanned version of a circuit. Every such test requires control of a specific set of flip-flops and circuit inputs. The method selects flip-flops that maximize the number of tests made possible by the selections, and hence maximize the fault coverage as well. Ma, et. al. [1988] proposed a similar technique that encompasses multiple-pattern tests.

2.3.4 GRAPH THEORETIC METHODS

Graph theoretic methods (Cheng and Agrawal [1989], Gupta, et. al. [1990], Wunderlich [1989], Jou and Cheng [1991], Chickermane [1991], Bertrand, et. al [1992], Lee and Reddy [1990]) represent circuits as graphs in which, for example, arcs represent flip-flops, and selecting a flip-flop for a scan path corresponds to removing its arc from the graph. Graph theory is used to identify flip-flops whose selection would produce a residual circuit graph with certain desired properties. Cheng and Agrawal [1989] proposed a method that removes all directed cycles from the residual graph, and that minimizes the lengths of the longest directed paths. The benefit is a greatly reduced workload for sequential ATPGs. Gupta, et. al. [1990] proposed a method called BALLAST. BALLAST attempts to produce a residual graph that is a balanced sequential

structure. Such a structure is easily initialized to any sequential machine state, and combinational test generation techniques can be used.

2.3.5 METHODS USING TESTABILITY MEASURES

A more deterministic approach is to use a testability measure to identify hard-to-detect faults (Trischler [1983], Kim and Kime [1990]). A testability measure generates metrics that quantify the controllabilities and observabilities of a circuit's nodes. This data specifically identifies hard-to-test nodes, and can provide more accurate guidance for alleviating specific testability problems than many of the methods discussed above.

SCOAP (Goldstein [1979]) is the most commonly encountered testability measure. Its metrics correspond to the number of distinct nodal values or clock pulses that are required for controlling or observing circuit nodes. Its metrics thus tend to measure the difficulty of generating tests for specific faults rather than absolute nodal testabilities. Trischler [1983] used SCOAP metrics to identify and select flip-flops with poor controllabilities and observabilities.

A similar testability metric is the execution time of a test pattern generator. This approach effectively optimizes the testability measure for the specific tool, i.e. the

ATPG, being used for the design task. It is useful for reducing the time necessary for generating tests for a circuit. Kim and Kime [1990] collect data on ATPG execution times for all faults in a circuit, and then use heuristic methods to estimate how that data would change in response to the scanning of individual flip-flops. These predictions guide their flip-flop selections.

For this thesis we developed a selection method of this category. We based our method on the testability measure STAFAN (Jain and Agrawal [1984]), which we describe in the next section. STAFAN produces probabilities that reflect the absolute controllabilities and observabilities of circuit nodes. From this data we calculate predictions of the quantitative changes in the controllabilities and observabilities of the circuit that would result from the selection of specific flip-flops for a scan path. This provides a quantitative guide to the influence of the individual flip-flops on the testability of the circuit. We describe our method in detail in the next chapter.

2.4 REVIEW OF STAFAN

STAFAN (*ST*atistical *F*ault *A*nalysis) (Jain and Agrawal [1984]) exercises a circuit with a series of pseudorandom patterns and derives probabilistic data that express the absolute controllabilities and observabilities of the

circuit nodes. It generates a four-element vector {C0, C1, B0, B1} for each node. The C0 (C1) element represents the probability that the node will be controlled to a logic 0 (1). It is measured statistically during fault-free simulation of the circuit. The B0 (B1) element represents the probability that, if the node is set to logic 0 (1), it will be sensitized to an output. It is calculated from the controllability data. The node's testabilities {T0, T1}, or the probabilities that faults on the node will be detected by the pseudorandom series, are then computed as

$$T0 = C1 * B1 \quad \text{for a stuck-at 0 fault}$$

$$T1 = C0 * B0 \quad \text{for a stuck-at 1 fault}$$

All elements have values in the range [0.0, 1.0], consistent with their definitions as probabilities.

STAFAN performs fault-free simulation of the circuit for a series of pseudorandom patterns. It maintains a collection of counters during this phase: a 0-controlled and 1-controlled counter for every node, a sensitized counter for every input to a gate, and a loop sensitized counter for every topologic loop in the circuit. The counters are updated for every pattern. Each controlled counter is incremented if its node holds the corresponding logic value. Each sensitized counter is incremented if its

gate input is sensitized to the gate output. Each loop counter is incremented if the pathway around the loop is sensitized by all loop inputs. After all patterns are processed, the counter values represent the absolute probabilities of their corresponding events.

STAFAN then calculates the observability data. For every gate the observabilities of each input are calculated from the gate output's observabilities, and from the input's controllability and sensitizability. Calculations are propagated gate-to-gate from the circuit outputs toward the circuit inputs. The 0- and 1-observabilities of all circuit outputs are first set to 1.0, i.e. always observable. Observabilities are then calculated for the inputs of the gates that drive these outputs, and so on.

Fanout nodes and loops provide multiple pathways over which nodes can be observed. They are processed as follows.

a) Fanout nodes. The observability of a fanout stem depends on the independence of the fanout branches, i.e. the correlation between successful sensitizations along different branches. It is bounded by two cases. The upper bound occurs when all branches exhibit complete independence, i.e. there is no correlation between sensitizations along separate branches. It equals the joint probabilities of observations along the individual branches. The lower bound occurs in the opposite case, when

observations along one branch cover observations along all others. The observability of this branch is greater than that of all other branches. The lower bound equals the observability of this branch. STAFAN computes the fanout stem observability as the weighted sum

$$B0(1) = (1 - \alpha) * (\text{lower bound}) + \alpha * (\text{upper bound})$$

α is an empirically determined constant. Experimentation revealed that "the results were rather insensitive to the actual value of α " (Jain and Agrawal [1984]), so STAFAN assigned a value of $\alpha = 1$. $\alpha = 1$ implies that sensitizations along all fanout branches are completely independent. This yields optimistic estimates for fanout stems whose branches are not completely independent.

b) Loops. STAFAN uses two methods for handling loops. The first method derives an analytical expression for the observability of a node in the loop. The expression involves terms for the observabilities of output nodes, and for the loop sensitization probability. It is applicable only to isolated loops. In the second method, when the propagation of observability calculations encounters a loop, observabilities of the loop nodes are initially assumed to equal 0.0. Calculations propagate around the loop until observability values are obtained for all loop nodes.

Propagation of the calculations around the loop is then repeated using the updated values. The process repeats until the values for the loop nodes converge.

2.5 SUMMARY

Partial scan paths provide a practical means of improving the testability of VLSI circuits. With the IC operating in test mode, partial scan paths allow access to internal circuit flip-flops, and thus improve the controllability and observability of otherwise hard-to-test nodes. By involving a small fraction of the circuit flip-flops, a partial scan path can provide improvements in testability that are comparable to those of a full scan path, but at a much lower cost in circuit area, circuit performance, and test application time.

Most circuits exhibit significant variability in the improvements in testability provided by the many possible distinct sets of scan flip-flops. Sets of flip-flops that are selected arbitrarily are likely to provide smaller benefits than optimal and near-optimal ones, and may fail to satisfy fault coverage and area overhead specifications for circuits with severe design constraints. "By inspection" flip-flop selection methods can identify some of the most- and least-optimal scan flip-flops, but more sophisticated selection methods are usually needed for a usable solution.

In this thesis we developed one such selection method. Our approach is based on using a testability measure to quantify the testabilities of circuit nodes. In the next chapter we describe our method.

CHAPTER 3: STAFFS: STATISTICS AIDED FLIP-FLOP SELECTION

We have developed a method for selecting circuit flip-flops for partial scan paths. Using a statistical testability measure as a source of testability data for a circuit, our method calculates predictions of how that data would change in response to the selection of specific scan flip-flops. We developed a computer-aided design tool called **STAFFS** that employs the proposed method. In Section 3.1 we briefly describe **STAFFS**. Its three major operations are described in detail in the remaining three sections. In Section 3.2 we describe the methods used by **STAFFS** to collect controllability and observability data. In Section 3.3 we describe the transformation of that data into predictions of how the data would change with the scanning of specific flip-flops. In Section 3.4 we discuss our proposed method for selecting flip-flops based on the collected testability data.

3.1 STAFFS ARCHITECTURE

STAFFS (**ST**atistics **A**ided **F**lip-**F**lop **S**election) is a computer-aided tool that selects circuit flip-flops for a partial scan path. Its three main processes are as follows. First, statistical data for the circuit's nodal controllabilities and observabilities are collected using

simplifications of the techniques used with STAFAN. Next, the collected data is used to calculate quantitative predictions of how each flip-flop's selection for a scan path would change the statistical data for all circuit nodes. Finally, the predictions are used to calculate a single metric for each flip-flop. These single metrics express the influence of the flip-flops on overall circuit testability, and they guide the flip-flop selections.

Our objective in this research has been to develop STAFFS to be a practical method of selecting scan flip-flops. To be practical, it must consistently select an optimal or near-optimal set of scan flip-flops, and must do so in a reasonable amount of time. Exact calculations of testability data would require complex processing that would incur prohibitively long execution times for VLSI circuits. Hence we adopted procedural simplifications at the cost of some inaccuracy in the testability data. In particular, the simplifications tend to make the observability data pessimistic. Such inaccuracies certainly affect the reliability of flip-flop selections. Furthermore, the observability of a node depends on the controllabilities of the nodes that control its sensitization paths to circuit outputs. The observability data incorporates the information represented by the controllability data. For these two reasons we designed STAFFS so that it can be used

as a tool for investigating the relative importance of the controllability data and the observability data in selecting flip-flops.

3.2 DATA COLLECTION

In the first main task, statistical data for the controllabilities and observabilities of the circuit nodes are collected, using simplifications of the techniques used with STAFAN. STAFFS' deviations from STAFAN's procedures are as follows:

a) Circuit restrictions

Circuits must be pure synchronous and must have a global set or reset for all flip-flops. Pure synchronous circuits simplify fault-free simulation, the propagation of observability calculations, and the propagation of the deltas calculations to be described in Section 3.3. The global set or reset guarantees that no node is at an unknown initial state. Thus $C0 + C1 = 1.0$, and only one controllability counter is needed for every node. The restriction does not affect the accuracy of the testability data.

b) Fanout nodes

The empirical constant α was found, in general, to

have little impact on the accuracy of observability values (Jain and Agrawal [1984]). STAFFS sets $\alpha = 0$. The observability of the fanout stem is then equal to that of the fanout branch with the greatest observability. This is the lower bound of the range of possible values for a fanout stem. Hence, this simplification yields pessimistic values for the observabilities of fanout stems.

c) Loops

The analytical method used with STAFAN to determine loop observabilities applies only to isolated loops. The iterative method is computation-intensive for complex loop structures. To simplify calculations for loops, the following procedure is used for STAFFS. When the propagation of observability calculations encounters a loop as shown in Figure 3.1, the observabilities of the feedback stem **a** are set to equal those of the nonfeedback branch **b**. Propagation of the calculations then continues from **a** toward the circuit inputs. Propagation will eventually encounter the feedback branch **c**. Propagation along that path is then discontinued.

This procedure is based on the assumption that pathways through feedback branches contribute a negligible amount to the observabilities of nodes in the input cone of a feedback loop stem. Even if negligible, however, their contributions

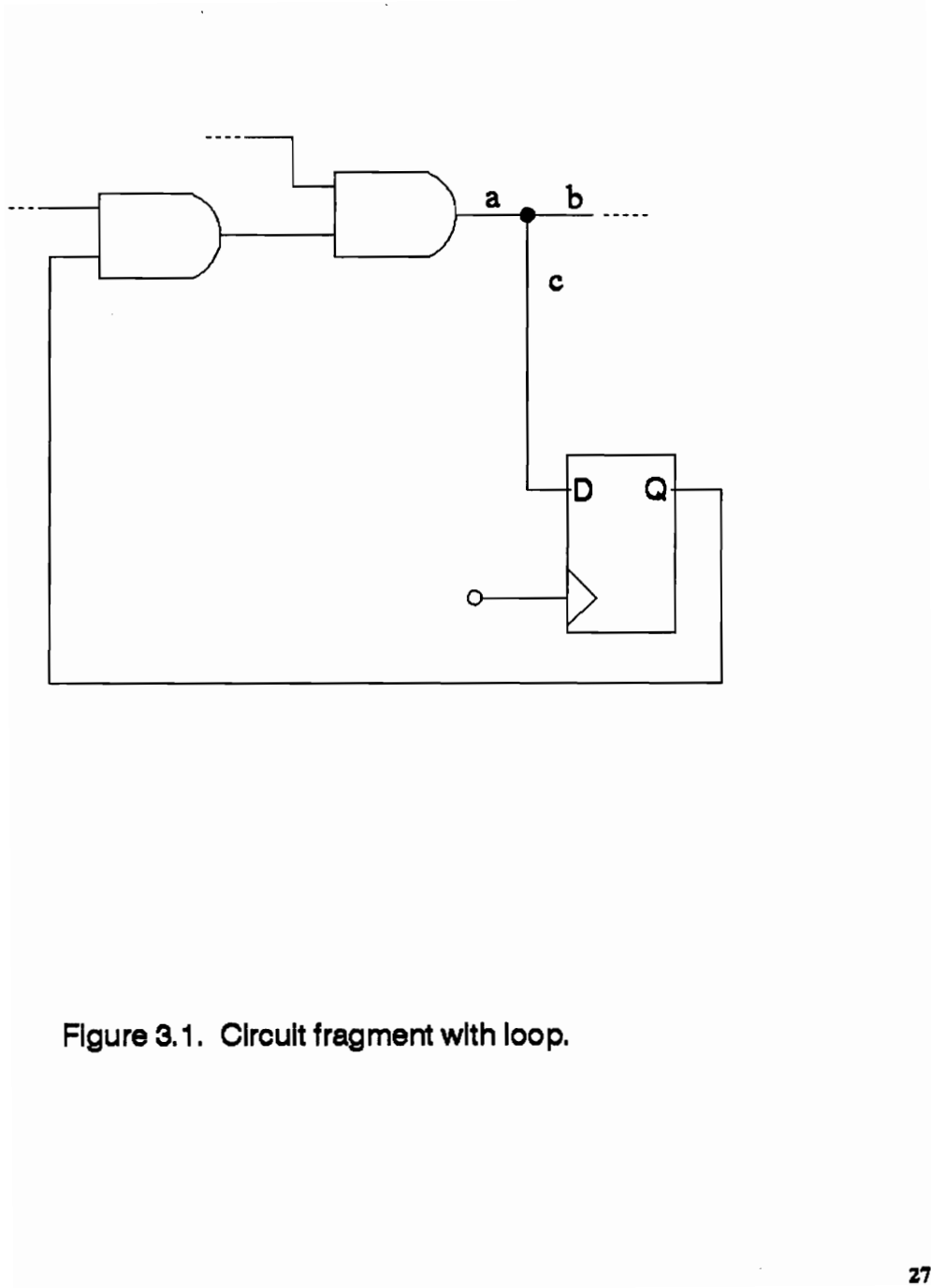


Figure 3.1. Circuit fragment with loop.

may be nonzero, and the computed observabilities will be too low. Hence, this simplification yields pessimistic values for the observabilities of the inputs of feedback loop stems.

In summary, the observabilities calculated by STAFFS for fanout nodes and feedback loop stems are the lower bounds of their possible values. Values for these nodes, therefore, are pessimistic. Furthermore, these pessimistic values are used to calculate the observabilities of nodes in the input cones of fanout nodes and feedback loop stems. In general, observabilities of nodes that lie in the input cones of a fanout node or feedback loop stem will tend to have pessimistic values.

3.3 TESTABILITY PREDICTIONS

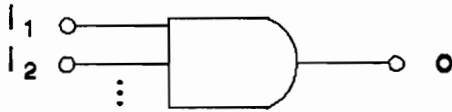
After the statistical controllability and observability data are obtained, predictions of the changes in the statistical data that would result from the selection of individual flip-flops are calculated. The new controllability and observability values for node n that result if flip-flop i is selected for a scan path are defined as:

$C_n'0(i)$: new 0 controllability
 $C_n'1(i)$: new 1 controllability
 $B_n'0(i)$: new 0 observability
 $B_n'1(i)$: new 1 observability

Since $C_n0 + C_n1 = 1.0$ for any node, $C_n'0(i) + C_n'1(i) = 1$. This enables STAFFS to calculate only the $C_n'0(i)$ values for each node.

If flip-flop i is selected for a scan path, its output node is directly controllable, and $C_n'0(i) = 0.5$ for the node. The controllabilities of all nodes in its output cone change as well, and so must be recomputed. Processing begins at the output node of the flip-flop and propagates along circuit pathways toward the circuit outputs. For every node processed, $C_n'0(i)$ is calculated from the controllabilities of the inputs to the gate that drives the node. The gate-dependent equations used are tabulated in Figure 3.2. $C_n'0(i)$ values for the inputs are used if they are available; otherwise, the original $C0$ values are used.

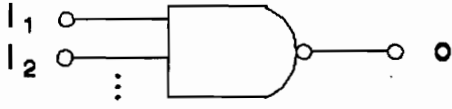
The term F in each equation compensates for reconvergent fanout at a gate. If there is no correlation between the controllabilities of the inputs of a gate, then the expected controllability of its output is a simple product of the controllabilities of the inputs. If reconvergent fanout occurs at the gate then the correlation



$$C'_x 0(o) = (1 - C'_x 1(o)) * F_{AND}$$

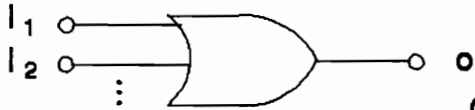
$$= (1 - X1(I_1) * X1(I_2) * \dots) * F_{AND}$$

$$F_{AND} = C'_x 0(o) / (1 - C'_x 1(I_1) * C'_x 1(I_2) * \dots)$$



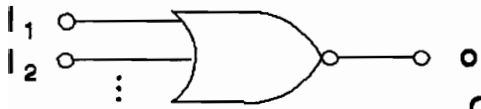
$$C'_x 0(o) = (X1(I_1) * X1(I_2) * \dots) * F_{NAND}$$

$$F_{NAND} = C'_x 0(o) / (C'_x 1(I_1) * C'_x 1(I_2) * \dots)$$



$$C'_x 0(o) = (X0(I_1) * X0(I_2) * \dots) * F_{OR}$$

$$F_{OR} = C'_x 0(o) / (C'_x 0(I_1) * C'_x 0(I_2) * \dots)$$



$$C'_x 0(o) = (1 - C'_x 1(o)) * F_{NOR}$$

$$= (1 - X0(I_1) * X0(I_2) * \dots) * F_{NOR}$$

$$F_{NOR} = C'_x 0(o) / (1 - C'_x 0(I_1) * C'_x 0(I_2) * \dots)$$

$$\text{For all inputs } I_m, X0(I_m) = \begin{cases} C'_x 0(I_n), & \text{if available} \\ C'_x 0(I_n), & \text{otherwise} \end{cases}$$

$$X1(I_m) = \begin{cases} C'_x 1(I_n), & \text{if available} \\ C'_x 1(I_n), & \text{otherwise} \end{cases}$$

Figure 3.2. STAFFS C'0 equations.

is nonzero between the inputs that are part of the reconvergent paths. The controllability of the output then varies from its expected value. The term F compensates for this variation. STAFFS assumes that the variation is a scale factor that remains constant for all scanned versions of the circuit. Using the data of the unscanned circuit, it calculates F for a node to be the actual controllability divided by the expected controllability.

New controllabilities at a node and in its output cone affect the observabilities along all pathways that pass through the cone. Therefore, $B_n'0(i)$ and $B_n'1(i)$ values are calculated after $C_n'0(i)$ processing is complete. When flip-flop i is included in a scan path, its input node is directly observable, and hence its observabilities are set to $B_n'0(i) = B_n'1(i) = 1.0$. To determine $B_n'0(i)$ and $B_n'1(i)$ for all other nodes the initial observability calculations are then repeated, with the input of flip-flop i added to the initial frontier. $C_n'0(i)$ values are used when they are available.

3.4 SELECTION OF FLIP-FLOPS

The controllability and observability predictions estimate the influence of a flip-flop on the controllabilities and observabilities of the circuit nodes. To obtain guidance to selecting scan flip-flops, the

predictions for each flip-flop should be combined into a single metric.

B_0 , the 0-observability of a node, is the probability that the node is sensitized to an output when the node is set to logic 0. The testability of a s-a-1 fault at a node is therefore $C_0 * B_0$. Likewise, the testability of a s-a-0 fault at a node is $C_1 * B_1$. Thus, the change of the testability of a node due to the selection of flip-flop i is expressed as:

$$dT_0(i) = C'_1(i) * B'_1(i) - C_1 * B_1$$

$$dT_1(i) = C'_0(i) * B'_0(i) - C_0 * B_0$$

As discussed in Section 3.1, part of our research is to investigate the relative importance of the observability data and the controllability data in selecting flip-flops. Hence, instead of using the above equations for the changes in a node's testability, we use the following weighted equations. The change in the testabilities of stuck-at faults at node n due to the selection of flip-flop i are expressed as:

$$dT_{0n}(i) = \mathbf{x} * dC_n(i) + (1-\mathbf{x}) * (B_n'_1(i) - B_{n1})$$

$$dT_{1n}(i) = \mathbf{x} * dC_n(i) + (1-\mathbf{x}) * (B_n'_0(i) - B_{n0})$$

where

$$dC_n(i) = |C_n'0(i) - 0.5| - |C_n0 - 0.5|$$

and

$$\mathbf{x} \text{ is a constant, } 0.0 \leq \mathbf{x} \leq 1.0$$

The absolute change in the testability of the circuit due to the selection of flip-flop i is expressed as:

$$\begin{aligned} dT(i) &= \sum_{n=1}^N [dT_{0n}(i) + dT_{1n}(i)] \\ &= 2\mathbf{x} * \sum_{n=1}^N dC_n(i) + \\ &\quad (1-\mathbf{x}) * \sum_{n=1}^N [(B_n'1(i) - B_n1) + (B_n'0(i) - B_n0)] \end{aligned}$$

where N is the number of nodes in the circuit.

The flip-flop corresponding to the greatest value of $dT(i)$ should provide the greatest increase in overall circuit testability, and hence is the one selected. The relative importance of the controllability data and the observability data can be investigated by comparing the scan designs that are selected for different values of the weighting factor \mathbf{x} . The cost of selecting scan designs for different values of \mathbf{x} is very low. The processing time needed to perform the

weighting is thus negligible compared to the time needed to perform the summations.

3.5 SUMMARY

In this chapter we described STAFFS, a computer-aided tool that uses a statistical testability measure as a basis for selecting scan path flip-flops. STAFFS calculates estimates of the changes that would occur in the statistical controllability and observability data due to the selection of specific flip-flops for a scan path. It then calculates a single metric for each flip-flop that expresses the influence of the flip-flop on the overall testability of the circuit. The metrics guide the flip-flop selections. In order to facilitate investigation of the relative importance of the controllability data and the observability data in selecting flip-flops, the calculation of the metrics includes a weighting factor α on the controllability and observability data.

CHAPTER 4: TESTS AND EXPERIMENTAL RESULTS

To evaluate STAFFS we conducted experiments using 13 ISCAS89 benchmark circuits (Brglez, et. al. [1989]). In this chapter we describe our test procedure and present the experimental results. In Section 4.1 we describe our evaluation methods. In Section 4.2 we present STAFFS and CV, another computer program that we developed for conducting the research. In Section 4.3 we present the test circuits and discuss the values that we used for the operational parameters of STAFFS and CV. In Section 4.4 we present and analyze the experimental results.

4.1 EVALUATION METHOD

In this section we describe our method of evaluating the performance of STAFFS. We have two goals. The first goal is to assess the effectiveness of STAFFS in selecting scan flip-flops. STAFFS must reliably select sets of scan flip-flops that are significantly better than arbitrarily selected ones. The second goal is to determine a recommended value or range of values for the weighting factor α . These values will represent the most favorable combinations of the observability and controllability data based on their relative importance in selecting flip-flops.

The three primary goals of design-for-test techniques

are to minimize the test set size, to minimize test generation time, and to maximize fault coverage. Partial scan design techniques emphasize one or more of these goals. Our method emphasizes improvements in testability for a given circuit. Therefore, our metric for the performance of a given partial scan design is its random pattern fault coverage. To assess the performance of our method, we measure the improvement in random pattern fault coverage provided by the partial scan designs that are selected for the test circuits.

Any partial scan design improves the testability of a circuit, and hence its random pattern fault coverage. For a partial scan design to be optimal or near-optimal, the improvement in fault coverage that it provides must be significantly higher than that provided by an arbitrarily selected partial scan design. For this measurement we obtained the fault coverages of a large random sample of possible partial scan designs for each circuit under test. We refer to this collection of fault coverages as the database associated with the test circuit. The partial scan designs represented in the database have the same scan path length as the selected partial scan design, and their fault coverages are measured with the same test patterns that are used to measure the fault coverage of the selected partial scan design. We define the mean of the fault coverages in

the database to be the expected value of the fault coverage of an arbitrarily selected partial scan design for the circuit.

In our research, we do not consider the order of the flip-flops in the scan paths. Although the order of the flip-flops in a scan path influences test pattern generation, the testability of the scan design is independent of the order of the flip-flops.

4.2 TOOLS DEVELOPED

We developed two tools, called STAFFS and CV, for this research. STAFFS implements the flip-flop selection method described in Chapter 3. CV is a fault simulator that we used to determine the fault coverages of circuits. We describe these tools in detail in the following subsections.

4.2.1 STAFFS

This program implements the flip-flop selection method described in Chapter 3. The program reads the description of a circuit under test in the ISCAS89 netlist format. It applies a specified number of pseudorandom test patterns to the circuit, performs fault-free simulation of the circuit and collects the statistical testability data. It increments the value of the weighting factor x from 0.0 to 1.0 by a specified amount, and for each value it selects a

set of scan flip-flops.

Command line options control the operational parameters of STAFFS. Options include the number of test patterns, the number of flip-flops to be selected, and the increment for the weighting factor α . A complete description of the options is available in Appendix A, the STAFFS User's Guide. For reference, the ISCAS89 netlist of an example circuit is given in Appendix B.

4.2.2 CV

CV reads the description of a circuit under test in the ISCAS89 netlist format and performs fault simulation on the circuit. If a list of specific scan flip-flops is identified, it will perform fault simulation on that scanned version of the circuit. CV is constructed from HOPE, an existing synchronous sequential circuit fault simulator (Lee and Ha [1992]). CV can operate in two modes, single circuit mode and database mode, as described below.

In its single circuit mode of operation, CV performs fault simulation for a single circuit. A particular scan design, identified by the flip-flops it contains, can be specified. In this case CV treats the outputs of the scan flip-flops as primary inputs, and it treats the inputs of the flip-flops as primary outputs. If no scan flip-flops are specified, CV performs fault simulation for the original

unscanned circuit. Command-line options for this mode include the number of test patterns, the number of scan flip-flops, if any, and their circuit labels.

In its database mode of operation, CV performs fault simulation for many possible partial scan designs of the given circuit and collects the results in a single file, or database. For each partial scan design in the sample, CV generates a set of scan flip-flops automatically and performs fault simulation for that scan design. A command line option controls whether CV generates the sets of flip-flops systematically or randomly. Systematic generation is used for small test circuits. These circuits have a small number of possible partial scan designs, and fault simulation of all of them is feasible. Random generation is used for large test circuits. These circuits have a large number of possible partial scan designs. Fault simulation of all of them is not feasible, so it is performed for a random sample of the possible partial scan designs. The number of partial scan designs sampled for each test circuit is tabulated in Table 4.1. Command-line options for this mode include the number of pseudorandom test patterns, the length of the scan paths, the number of partial scan designs to be sampled for the database, and the choice between systematic and random generation of the sample partial scan paths. A complete description of the options is available

Table 4.1. Test circuits.

circuit	# inputs	# outputs	# gates	# flip-flops	# scan flip-flops	# possible scan designs	# scan designs sampled for database	# test patterns
s208	11	2	96	8	1	8	8	5000
s288	3	6	119	14	3	384	384	5000
s349	9	11	161	15	3	455	455	5000
s386	7	7	159	6	1	6	6	10000
s400	3	6	162	21	4	5,985	500	5000
s420	19	2	196	16	3	560	500	5000
s444	3	6	181	21	4	5,985	500	5000
s526	3	6	193	21	4	5,985	500	5000
s713	35	23	383	19	4	3,876	500	5000
s832	18	19	287	5	1	5	5	10000
s838	35	2	380	32	6	906,192	500	5000
s1238	14	14	508	18	4	3,060	500	10000
s1494	8	19	647	6	1	6	6	10000

in Appendix C, the CV User's Guide.

4.3 EXPERIMENTAL ENVIRONMENT

In this section we identify the circuits that we used for testing. We then address the operational parameters that must be set for STAFFS and CV, and we justify the values that we used for these parameters in testing.

4.3.1 TEST CIRCUITS

We conducted our experiments using ISCAS89 sequential benchmark circuits. Of the 31 ISCAS89 circuits, we selected the 13 circuits shown in Table 4.1. The remaining 18 circuits were too easily tested, too large, or were resynthesized from other circuits in the set. Circuits that are too easily tested achieve 100 percent random pattern fault coverage without scan design. Circuits that are too large require prohibitive processing times for STAFFS. A resynthesized circuit is almost identical to its corresponding original circuit, and thus has almost the same testability characteristics.

4.3.2 SCAN PATH LENGTH

The length of the scan path in a partial scan design is determined by two design goals: the desired fault coverage and the tolerable overhead in chip area. If the desired

fault coverage is the dominating constraint for a design, then test generation or fault simulation is necessary after the selection of small sets of flip-flops. The design process could be lengthy for large circuits. If the tolerable area overhead is the dominating constraint, scan path length is determined easily. The number of flip-flops to be selected is given as the tolerable area overhead divided by the area overhead per scan flip-flop.

For our research the constraint used to determine the scan path length is irrelevant. For simplicity, we adopted the constraint on tolerable area overhead for all test circuits. In all cases, we specified scan path lengths equal to 20 percent of the circuit's flip-flops. The length of the scan path for each test circuit is tabulated in Table 4.1.

4.3.3 NUMBER OF TEST PATTERNS

The number of test patterns must be determined for two parts of our test procedure. In the first part, STAFFS collects statistics on nodal controllabilities with the application of a series of pseudorandom test patterns. The number of test patterns determines the precision, but not the accuracy, of the statistics. This number does not depend on the specific circuit under test. We used 500 patterns for all test circuits.

In the second part, pseudorandom patterns are used to determine fault coverages of circuits. In this case the required number of patterns depends on the circuit under test. The test length must be long enough to detect all easy-to-detect faults. However, unduly long tests waste processing time. The random pattern fault coverage of a circuit rises rapidly at first as easy-to-detect faults are detected. The fault coverage rises more slowly thereafter, and eventually saturates when only hard-to-detect faults remain undetected. The saturated fault coverage of a circuit reflects the number of hard-to-detect faults in the circuit. Since the goal of scan design is to improve the testability of hard-to-detect faults, the testability improvement provided by a scan design is accurately represented by its saturated fault coverage. Furthermore, we evaluate the performance of a selected partial scan design by comparing its fault coverage with those of a large random sample of other possible ones. A meaningful comparison requires saturated fault coverages.

To determine the test length necessary to achieve saturated fault coverages for the test circuits, we performed fault simulation on the original unscanned circuits using 10,000 test patterns. The profiles of the random pattern fault coverages versus the test length for the circuits are shown in Figure 4.1. From the figure, the

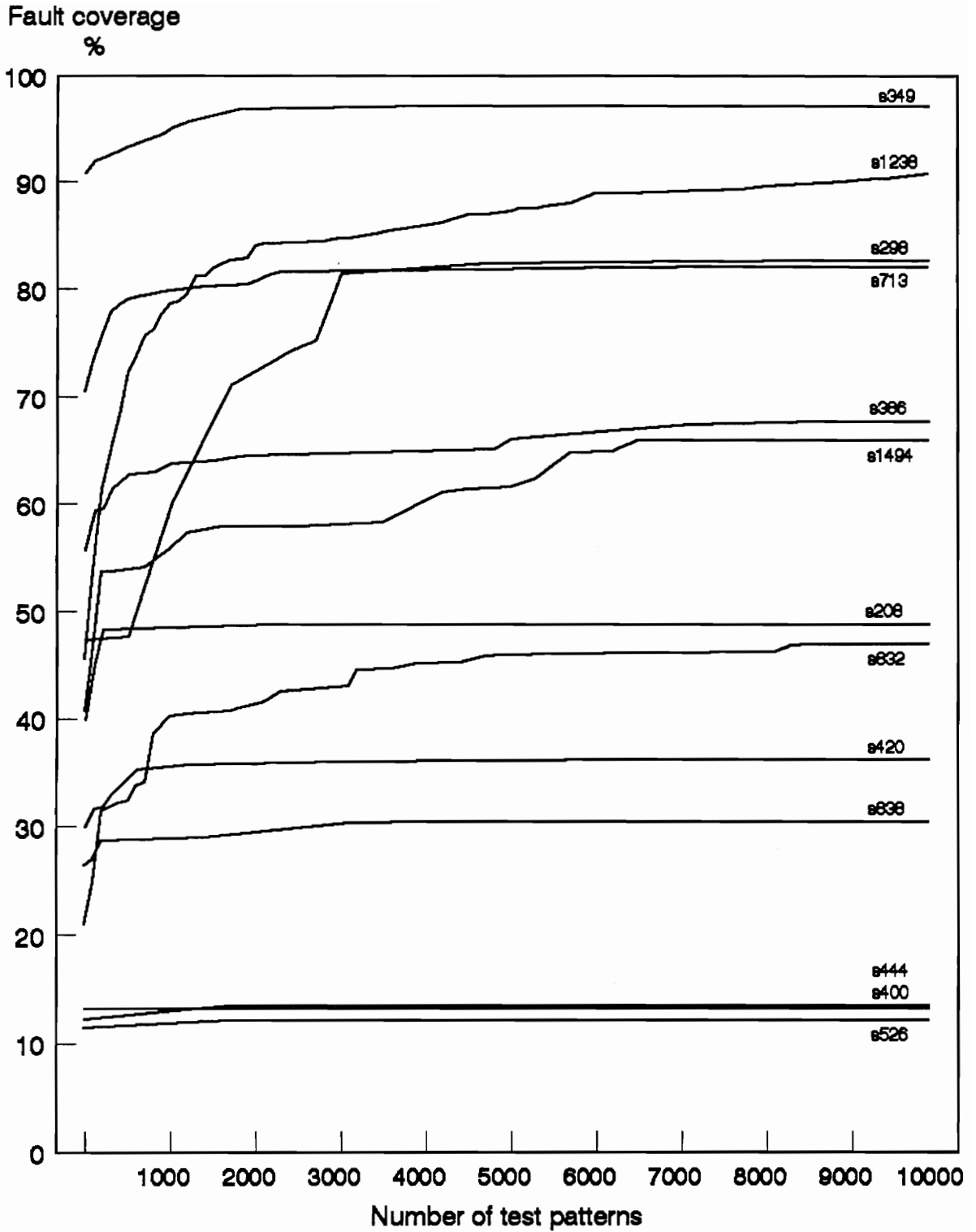


Figure 4.1. Random pattern fault coverage profiles for test circuits.

fault coverage for the circuits s386, s832, s1238, and s1494 is saturated for 10,000 test patterns. The fault coverage for the remaining circuits is saturated for 5,000 test patterns. Hence in measuring fault coverages for scan designs, we applied 10,000 test patterns to the circuits s386, s832, s1238, and s1494, and we applied 5,000 test patterns for all others. The test length used for each circuit is tabulated in Table 4.1.

4.3.4 SIZE OF FAULT COVERAGE DATABASE

As described in Section 4.1, we measure the performance of a selected partial scan design by comparing its random pattern fault coverage with the fault coverages in a database. The database represents a large random sample of the possible partial scan designs for the circuit under test. To construct the databases for the test circuits, we measured the fault coverages of 500 partial scan designs for circuits that have at least 500 possible ones. For circuits that have fewer than 500 possible partial scan designs, we measured the fault coverages of all of them. For this task we used the program CV in its database mode of operation.

4.3.5 WEIGHTING FACTOR

We want to determine the best value or range of values for the weighting factor x . As described in Section 3.4,

the relative importance of the controllability and observability data to the selection of the flip-flops can be studied by comparing the partial scan designs selected for different values of α . In our experiments, we varied the value of α from 0.0 to 1.0 in increments of 0.1. We selected a set of scan flip-flops for each value of α . In total, we examined 11 partial scan designs for each test circuit.

4.4 EXPERIMENTAL RESULTS

4.4.1 OVERALL EFFECTIVENESS OF STAFFS

The first goal of our experiments is to demonstrate the effectiveness of STAFFS in partial scan design. This can be achieved if we show that the partial scan designs selected by STAFFS achieve significantly higher fault coverages than those of partial scan designs whose flip-flops are arbitrarily selected. We define the fault coverage of an arbitrarily selected partial scan design to be the mean of the fault coverages in the database for the circuit under test.

For every test circuit STAFFS selected a partial scan design with the weighting factor $\alpha = 0.5$. With this value, equal weight is given to the changes in controllabilities and the changes in observabilities. The fault coverages of the selected partial scan designs are shown in Table 4.2.

**Table 4.2. Fault coverages of selections,
x = 0.5.**

circuit	original fault coverage	mean of database fault coverages	std. dev. of database fault coverages	fault coverage of selected scan design
s208	48.84	56.67	1.95	54.38
s298	82.79	92.20	4.98	92.67
s349	97.14	98.16	0.73	97.47
s366	67.71	73.88	8.83	82.64
s400	13.21	58.36	13.52	84.72
s420	36.28	50.03	1.95	51.38
s444	13.50	57.22	13.48	74.27
s526	10.63	42.81	11.89	58.11
s713	81.93	84.20	3.56	89.98
s832	46.32	61.95	10.16	53.10
s838	29.40	45.49	1.23	47.18
s1238	91.14	91.80	0.43	92.37
s1494	65.74	90.62	6.73	95.62

The table also includes the fault coverages of the original unscanned circuits, and the means and standard deviations of the fault coverages in each database.

The data shows that the fault coverages of all of the arbitrarily selected partial scan designs are higher than the fault coverages of the original circuits. The average improvement in fault coverage for the 13 circuits is 17.14 percentage points. This result is expected, since any scan design improves the testability of a circuit.

The fault coverages of the partial scan designs selected by STAFFS are higher than the fault coverages of the arbitrarily selected partial scan designs for 10 out of the 13 test circuits. The average improvement in fault coverage for these 10 circuits is 7.82 percentage points. This increase may be significant, since the higher fault coverages are mostly due to improvements in the testability of hard-to-detect faults. A test pattern generator spends a significant portion of its processing time generating tests for hard-to-detect faults. A small increase in the coverage of hard-to-detect faults may enable a large reduction of processing time for test pattern generation.

If the variability of the fault coverages of the possible scan designs for a circuit is small, then the use of a sophisticated method of selecting scan flip-flops is not justified. An arbitrary selection of scan flip-flops

would be likely to provide similar improvements in testability with much less effort. Therefore, the performance of a scan design method is an important concern only for circuits with a large variability. The variability for a circuit is represented by the standard deviation of the fault coverages in the database for the circuit. Eight of the test circuits have standard deviations above 3.00. For seven of these circuits, the partial scan designs selected by STAFFS have higher fault coverages than the arbitrarily selected partial scan designs. The average improvement in fault coverage for these circuits is 10.67 percentage points. Hence, we conclude that STAFFS is effective for selecting flip-flops for scan design.

4.4.2 WEIGHTING FACTOR α

The second goal of our experiments is to determine the best value or range of values for the weighting factor α . These values would result in partial scan designs with the highest fault coverages for the circuits under test.

For each test circuit, we varied the value of α from 0.0 to 1.0 in increments of 0.1, and selected a scan design for each value. The fault coverages of the selected scan designs are shown in Table 4.3 and Figure 4.2. In the last column of Table 4.3, the values of α are those that correspond to the selected scan designs that have the

Table 4.3. Fault coverages of scan designs.

circuit	mean of database fault coverages	weighting factor $\alpha =$										range of α for best fault coverage			
		0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9		1.0		
s208	56.67	54.38	54.38	54.38	54.38	54.38	54.38	54.38	58.53	58.53	58.53	58.53	58.53	58.53	0.7 - 1.0
s298	92.20	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	92.67	1.0
s349	98.16	97.47	97.47	97.47	97.47	97.47	97.47	97.47	97.47	97.47	97.47	97.47	97.47	98.03	1.0
s386	79.88	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	82.64	0.0 - 0.9
s400	56.36	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	64.72	0.0 - 0.9
s420	50.03	51.38	51.38	51.38	51.38	51.38	51.38	51.38	51.38	51.38	51.38	51.38	51.38	50.92	0.0 - 0.9
s444	57.22	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	74.27	0.0 - 1.0
s526	42.81	58.11	58.11	58.11	58.11	58.11	58.11	58.11	58.11	58.11	58.11	58.11	58.11	54.53	0.0 - 0.9
s713	64.20	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	90.83	0.0 - 0.4
s832	61.95	53.10	53.10	53.10	53.10	53.10	53.10	53.10	53.10	53.10	53.10	53.10	53.10	75.94	1.0
s838	45.49	47.18	47.18	47.18	47.18	47.18	47.18	47.18	47.18	47.18	47.07	47.87	47.87	47.87	0.7 - 1.0
s1238	91.80	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	92.37	0.0 - 1.0
s1494	90.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	95.62	0.0 - 1.0

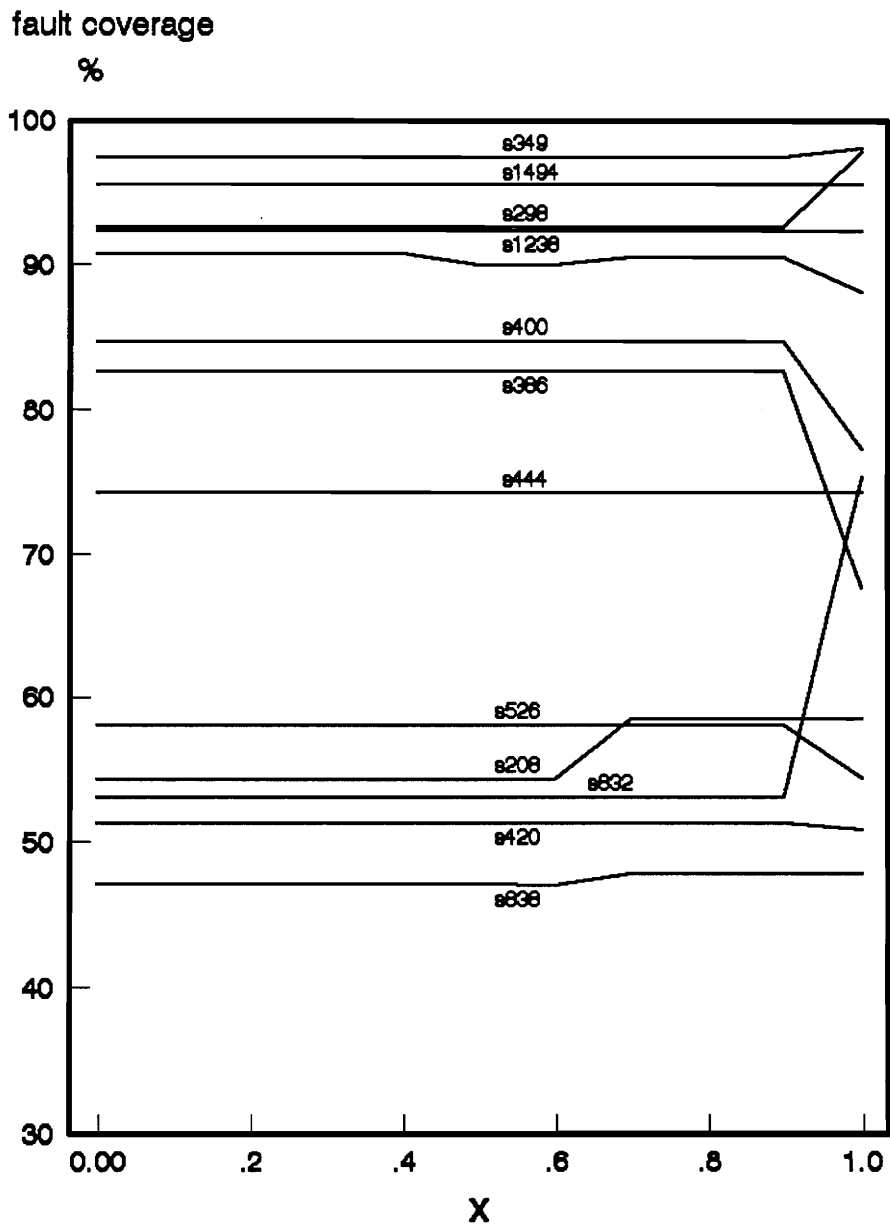


Figure 4.2. Fault coverages of scan designs.

highest fault coverage for the circuit.

We categorize the circuits into the following four groups, based on the the values of α that result in the highest fault coverages of the selected partial scan designs. Group 1 is the least restrictive on the range of values of α that it supports, and group 4 is the most restrictive.

Group 1. Fault coverage insensitive to α . For the three circuits (s444, s1238, and s1494) in this group, the fault coverages of the selected partial scan designs are independent of the value of α . Furthermore, they are all higher than the fault coverages of arbitrarily selected partial scan designs. Hence, any value of α is suitable for these circuits.

Group 2. Fault coverage highest for wide range of values of α : For the four circuits (s386, s400, s420, and s526) in this group, the fault coverages of the selected scan designs are uniform for $0.0 \leq \alpha \leq 0.9$, and achieve their highest values in this range. The fault coverages are lower for $\alpha = 1.0$. All of the scan designs selected for values in this range have higher fault coverages than those of arbitrarily selected partial scan paths. Therefore, values of α between 0.0 and 0.9 are best for these circuits.

Group 3. Fault coverage highest for narrow range of small values of α . Only one circuit, s713, is in this

group. The fault coverages of the selected partial scan designs are highest for $x \leq 0.4$. However, the fault coverage achieved for $0.7 \leq x \leq 0.9$ is only slightly lower. The fault coverages achieved for all values are higher than that of an arbitrarily selected circuit. Therefore, $x \leq 0.4$ gives the best results for this circuit, but larger values may be used.

Group 4. Fault coverage highest for narrow range of large values of x . For the five circuits (s208, s298, s349, s832, and s838) in this group, higher fault coverage for the selected partial scan designs is achieved for large values of x . For two of the circuits (s208 and s838), the partial scan path selections achieve their highest fault coverages for $0.7 \leq x \leq 1.0$. For the other three circuits (s298, s349, and s832), the fault coverages of the selected partial scan paths do not change for $0.0 \leq x \leq 0.9$, and achieve their highest only for $x = 1.0$. With this value for x , only the controllability data are used in selecting the flip-flops. Overall, $x = 1.0$ yields the best results for the circuits in this group.

In general, $x = 1.0$ provides the best overall results for the test circuits. With this value, only the controllability data is considered in selecting flip-flops. The five circuits in group 4 require $x = 1.0$ for their best performance. Of the five circuits in groups 2 and 3, four

achieve fault coverages that are above those of arbitrarily selected scan paths. The circuits in group 1 place no constraints on the value of α . Overall, 8 circuits achieve their highest fault coverages, and 11 circuits achieve fault coverages that are higher than those of arbitrarily selected scan parts. However, all circuits in groups 3 and 4 achieve better fault coverages for $\alpha \leq 0.9$, and significant improvements in the testabilities of these circuits are sacrificed with $\alpha = 1.0$. Therefore, no single value of α is ideal for all circuits.

We interpret these results to mean that the controllability data is accurate, and that selecting scan flip-flops using only the controllability data is effective. The contrast between the performances of the circuits in groups 1 and 3 indicates that the reliability of the observability data varies. For the circuits in group 1, use of the observability data yields significantly worse results, while the opposite is true for the circuits in group 3. Furthermore, only a small contribution of the observability data is necessary for it to affect flip-flop selection. Little change in the performance of the selections occurs for α less than 0.9. Most of the change occurs for α between 0.9 and 1.0.

Therefore, we conclude that the range of values for α between 0.9 and 1.0 provides the best overall performance

for STAFFS. Twelve of the thirteen circuits achieve their highest fault coverages for values in this range, and the other circuit, s713, achieves a fault coverage for $\alpha = 0.9$ that is only slightly below its highest value. All but one of the circuits achieve fault coverages that are significantly higher than those of arbitrarily selected scan designs.

As described in Section 3.4, when the single metrics are computed for each flip-flop, most of the processing time is spent performing the summations of the controllability and observability data. In a practical application of STAFFS, there is no significant cost in selecting scan designs for different values of α . Therefore, if fault simulation time is small or is not a concern, a recommended strategy is to select partial scan designs for a few different values of α , and to compare the fault coverages of the different selections to identify the one with the highest fault coverage.

4.5 CONCLUDING REMARKS

We have demonstrated that STAFFS is an effective tool for selecting circuit flip-flops for partial scan path designs. For most circuits, STAFFS selects partial scan designs that have fault coverages that are significantly higher than those of arbitrarily selected ones. Its

performance depends on the value of α , however, and the best value of α varies for different circuits. From our experimental results, $\alpha \geq 0.9$ provides the best overall results. This indicates that the most emphasis should be given to the controllability data in selecting flip-flops. If the time needed for fault simulation is not critical, it is practical to determine the best value of α for a given circuit by comparing the partial scan paths selected for several values of α .

CHAPTER 5. CONCLUSION

Scan design is an effective method of improving the testability of integrated circuits, but design constraints may require that the scan path include only some of the circuit flip-flops. In this thesis we developed a computer-aided design tool called STAFFS for identifying the scan flip-flops that are most likely to provide the greatest improvements in fault coverage. STAFFS uses statistical methods to measure the absolute controllabilities and observabilities of the circuit nodes. It predicts the quantitative changes that would occur in this data due to the selection of specific flip-flops for a scan path, and uses those predictions as a guide to selecting the scan flip-flops.

To avoid prohibitively long execution times, STAFFS uses approximate calculations for the observabilities and the predicted changes in the testability data. To compare the predicted influences of the scan flip-flops on overall circuit testability, STAFFS calculates a single metric for each flip-flop using an equation that applies a relative weight to the controllability and observability data. STAFFS can efficiently select different sets of scan flip-flops for different values of the weighting factor. Therefore, if the cost of fault simulation is not a

constraint, it is practical to use STAFFS to select several partial scan designs, and then to select one based on the actual fault coverages of the alternatives.

We tested STAFFS using 13 benchmark circuits. The experimental results demonstrate that STAFFS is effective in selecting scan flip-flops that provide significant improvements in circuit testability. With the weighting factor α equal to either 0.9 or 1.0, STAFFS achieves fault coverages that are significantly higher than the fault coverages of arbitrarily selected scan designs for 12 out of the 13 circuits. The improvements in fault coverage for the selections made by STAFFS averaged 8.63 percentage points, and were as high as 20.95 percentage points.

These results were obtained despite the computational approximations that were used. Further work to develop more sophisticated methods may provide more reliable data, and hence more optimal flip-flop selections. Three possible directions for further research are as follows. First, a more accurate, computationally practical method of calculating the observabilities of feedback loop nodes is needed. Second, a minor amount of topologic analysis may be used to identify inaccuracy-prone structures, e.g. fanout stems and fanout reconvergencies, for which additional processing would be justified. Third, our selection method does not account for the interdependency of the testability

improvements provided by different flip-flops. If the accuracy of the predicted changes in the testability data can be sufficiently improved, then these predictions may be used to update the testability data after flip-flop selections.

REFERENCES

- Abramovici, M., M. A. Breuer, and A. D. Friedman. Digital Systems Testing and Testable Design, New York, NY: Computer Science Press, 1990.
- Abramovici, M., J. J. Kulikowski, and R. K. Roy. "The Best Flip-flops to Scan," Proceedings of the International Test Conference, 1991, pp. 166-173.
- Agrawal, V., K. Cheng, D. D. Johnson and T. Lin. "A Complete Solution to the Partial Scan Problem," Proceedings of the International Test Conference, 1987, pp. 44-51.
- Bertrand, Y., F. Bancel and M. Renovell. "A DFT Technique Based on Graph Partitioning to Reduce the Test Generation Problem for Sequential Circuits," submitted to the 1992 International Test Conference.
- Brglez, F., D. Bryan and K. Kozminski. "Combinational Profiles of Sequential Benchmark Circuits," Proceedings of the International Symposium on Circuits and Systems, 1989, pp. 1929-1934.
- Cheng, K. and V. D. Agrawal. "A Partial Scan Method for Sequential Circuits with Feedback," IEEE Transactions on Computers, vol. 39, no. 4, April 1990, pp.544-548.
- Cheng, K. and V. Agrawal. "An Economical Scan Design for Sequential Logic Test Generation," Proceedings of the 19th International Symposium on Fault Tolerant Computing, 1989, pp. 28-35.
- Chickermane, V. and J. H. Patel. "A Fault Oriented Partial Scan Design Approach," Proceedings of the International Conference on CAD, 1991, pp. 400-403.
- Chickermane, V. and J. H. Patel. "An Optimization Based Approach to the Partial Scan Design Problem," Proceedings of the International Test Conference, 1990, pp. 377-386.
- Eichelberger, E. B. and T. W. Williams. "A Logic Design Structure for LSI Testability," Proceedings of the 14th Design Automation Conference, 1977, pp. 462-468.
- Goldstein, L. "Controllability/Observability Analysis of Digital Circuits," IEEE Trans. on Circuits and Systems, vol. CAS-26, no. 9, Sept. 1979, pp. 685-93.

- Gupta, R., R. Gupta., and M. Breuer. "The BALLAST Methodology for Structured Partial Scan Design," IEEE Trans. on Computers, vol. 39, no. 4, April 1990, pp. 538-544.
- Hudli, R. V. and S. C. Seth. "Testability Analysis of Synchronous Sequential Circuits Based on Structural Data," Proceedings of the International Test Conference, 1989, pp. 364-372.
- Jain, S. and V. Agrawal. "STAFAN: an Alternative to Fault Simulation," Proceedings of the 21st Design Automation Conference, 1984, pp. 18-23.
- Jou, J.-Y. and K.-T. Cheng. "Timing-Driven Partial Scan," Proceedings of the International Conference on CAD, 1991, pp. 404-407.
- Kim, K. S. and C. R. Kime. "Partial Scan by Use of Empirical Testability," Proceedings of the International Conference on CAD, 1990, pp. 314-17.
- Lee, D. H. and S. M. Reddy. "On Determining Scan Flip-flops in Partial-Scan Designs," Proceedings of the International Conference on CAD, 1990, pp. 322-325.
- Lee, H. K. and D. S. Ha. "HOPE: An Efficient Parallel Fault Simulator for Synchronous Sequential Circuits," submitted to the 1992 Design Automation Conference.
- Ma, H-K. T., S. Devadas, A. R. Newton, and A. Sangiovanni-Vincentelli. "An Incomplete Scan Design Approach to Test Generation for Sequential Machines," Proceedings of the International Test Conference, 1988, pp. 730-34.
- McCluskey, E. J. Logic Design Principles with Emphasis on Testable Semicustom Circuits, Englewood Cliffs, NJ: Prentice-Hall, 1986.
- Pradhan, D. K., ed. Fault-Tolerant Computing: Theory and Techniques, Englewood Cliffs, NJ: Prentice-Hall, 1986.
- Pradhan, M. M., E. J. O'Brien and S. L. Lam. "Circular BIST with Partial Scan," Proceedings of the International Test Conference, 1988, pp. 719-729.
- Stewart, J. H. "Future Testing of Large LSI Circuit Cards," Digest of the 1977 Semiconductor Test Symposium, 1977, pp. 6-15.
- Trischler, E. "Incomplete Scan Path with an Automatic Test

Generation Methodology," Proceedings of the IEEE Test Conference, 1980, pp. 153-161.

Trischler, E. "Testability Analysis and Incomplete Scan Path," Proceedings of the International Conference on CAD, 1983, pp.38-39.

Williams, M. J. Y. and J. B. Angell. "Enhancing Testability of Large Scale Integrated Circuits via Test Points and Additional Logic," IEEE Trans. on Computers, vol. C-22, no. 1, Jan 1973, pp 46-60.

Wunderlich, H.-J. "The design of Random-Testable Sequential Circuits," Proceedings of the 19th International Symposium on Fault-Tolerant Computing, 1989, pp. 118-125.

APPENDIX A. STAFFS USER'S GUIDE

ONSCREEN HELP GUIDE:

Usage: `staffs [options] circuitname`

Option [default]:

`none` Display this guide.
`-f n` Select `n` flip-flops for a partial scan path [`-f 0`]
`-i fn` Read test patterns from the specified file.
[generate pseudorandomly patterns internally]
`-l fn` Write statistical controllability and
observability data into logfile `fn`, and write
deltas data into file `circuitname.del`.
`-n n` Report to screen every `n` patterns. [`-n 100`]
`-p` Write flip-flop selections into the file
`circuitname.stf`. [write to screen]
`-r n` Run `n` test patterns. [`-r 1000`]
`-s n` Initial seed for the random number generator. If
`n=0`, seed is derived from the day time of the
computer. [`-s 0`]
`-S` Initialize all circuit nodes to logic 1. [logic
0]
`-x n` Vary the weighting factor `x` from 0.0 to 1.0 in
increments of `n`. [`-x 0.1`]

APPENDIX B. ISCAS89 Netlist Example: Circuit S27

```

# s27
# 4 inputs
# 1 outputs
# 3 D-type flipflops
# 2 inverters
# 8 gates (2 ANDs + 1 NANDs + 2 ORs + 4 NORs)

```

```

INPUT(G0)
INPUT(G1)
INPUT(G2)
INPUT(G3)

```

```

OUTPUT(G17)

```

```

G5 = DFF(G10)
G6 = DFF(G11)
G7 = DFF(G13)

```

```

G14 = NOT(G0)
G17 = NOT(G11)

```

```

G8 = AND(G14, G6)

```

```

G15 = OR(G12, G8)
G16 = OR(G3, G8)

```

```

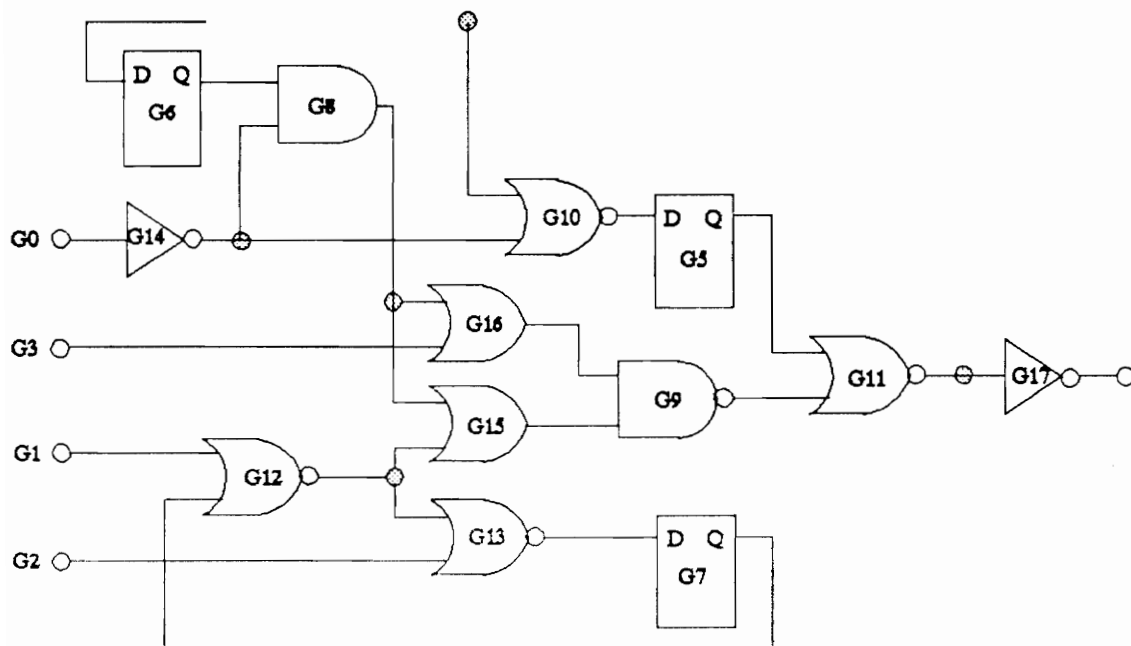
G9 = NAND(G16, G15)

```

```

G10 = NOR(G14, G11)
G11 = NOR(G5, G9)
G12 = NOR(G1, G7)
G13 = NOR(G2, G12)

```



APPENDIX C. CV USER'S GUIDE

ONSCREEN HELP GUIDE:

Usage: cv [options] circuitname

Option [default]:

none Display this guide.
-c Continue processing of an earlier interrupted run.
 The -l option must be used to identify the logfile
 of the previous run.
-f n Select n flip-flops for a partial scan path.
 [-f 0]
-F n FF1 ... FFn
 Get fault coverage for the circuit having scan
 flip-flops FF1 through FFn.
-i fn Read test patterns from the specified file.
 [generate pseudorandom patterns internally]
-l fn Write fault coverages into the specified file.
-n n Report fault coverage for every n test patterns.
 [-n 100]
-r n Run n test patterns. [-r 1000]
-R n Pick scan sets randomly instead of in sequence.
 Continue until n scan sets are processed. [in
 sequence]
-s n Initial seed for the random number generator. If
 n=0, seed is derived from the day time of the
 computer. [-s 0]
-S Initialize all circuit nodes to logic 1. [logic
 0]
-v Write to screen every time fault coverage is
 reported.

NOTES ON COMMAND OPTIONS:

- c** If a previous run of CV was interrupted, this option allows processing to resume at the point of interruption. The **-l** option must also be used to specify the logfile that holds the results of the interrupted run. CV will recover all active option settings from the incomplete logfile, so all other options can be omitted from the command line.
- f** If omitted, or if **n = 0**, CV will find the fault coverage of the unscanned circuit.
- F** With this option, CV finds the fault coverage for a single scanned circuit. {**FF1 ... FF_n**} is the list of specific flip-flops in the scan path.
- i** Directs CV to read input patterns from the specified file. If omitted, CV will generate a pseudorandom series of patterns using the seed specified with the **-s** option.
- l** Directs CV to write fault coverage results into the specified logfile. If omitted, CV will write results to the screen.
- n** Specifies the test pattern interval at which CV reports fault coverages.
- r** Specifies the number of test patterns CV runs for fault simulation. If fault coverage reaches 100% before **n** patterns are simulated, CV exits processing.
- R** When CV is to process multiple candidate circuits, this option directs CV to choose scan flip-flop sets randomly. If omitted CV will choose them systematically.
- s** No additional comment.
- S** No additional comment.
- v** If CV writes results to a logfile, no screen output is made. This option ensures that visual evidence of the program's progress is available.

EXAMPLES:

All examples use ISCAS89 circuit s208 and a random number generator seed of 1.

a) Find the fault coverage for the unscanned test circuit with 10000 test patterns. Write the results to the screen.

```
cv -s 1 -r 10000 s208
```

b) Find the fault coverage for a candidate circuit with scan flip-flops Y_1, Y_3 and Y_7. Specify 5000 test patterns. Write the results to the screen.

```
cv -s 1 -r 5000 -F 3 Y_1 Y_3 Y_7 s208
```

c) Establish a complete database for the circuit, i.e. get fault coverages for all possible candidate circuits. Specify 5000 test patterns and a scan path length of 3 flip-flops. Collect the fault coverages in a logfile.

```
cv -s 1 -r 5000 -f 3 -l s2083.log s208
```

d) Establish an incomplete database for the circuit, i.e. get fault coverages for a random sample of the possible candidate circuits. Specify 5000 test patterns, a scan path length of 3 flip-flops, and 500 candidate samples. Record fault coverages every 1000 test patterns. Collect the fault coverages in a logfile.

```
cv -s 1 -r 5000 -f 3 -n 1000 -R 500 -l s2083.log s208
```

e) Resume processing of an interrupted run of CV, e.g. the run of example d).

```
cv -c -l s2083.log
```

VITA

David B. Jett was born in Norfolk, Virginia in 1963. He earned his B.S. in Electrical Engineering from Virginia Tech in Blacksburg, Virginia in June, 1985. He then worked as a Design Engineer with the Martin Marietta Corp. in Glen Burnie, Maryland until 1988, when he returned to Virginia Tech to earn his M.S. in Electrical Engineering. He plans to resume work as a Design Engineer.