Strategies for Efficient and Effective Scan Delay Testing

by

Chao Han

A thesis submitted to the Graduate Faculty of
Auburn University
in partial fulfillment of the
requirements for the Degree of Master

Auburn, Alabama
Aug 6, 2011

Keywords: Delay test, X values, transition delay fault,
output compression, partial enhanced scan

Approved by

Adit Singh, Professor of Electrical and Computer Engineering
Vishwani Agrawal, Professor of Electrical and Computer Engineering
Victor Nelson, Professor of Electrical and Computer Engineering

Abstract

Aggressive timing requirements in today's high-speed designs have introduced the need to test for small delay defects and distributed timing faults caused by statistical process variations. Faster-than-rated clock delay tests aimed at targeting small delay defects can generate a large number of unknown X values because the test response for all paths longer than the (over clocked) test clock period must be marked X. Unknown output values prevent the use of efficient test compression techniques. We propose and evaluate a simple multiplexing scheme for output test data compression which avoids any compaction of the test response. In addition, high delay fault coverage is required to ensure that the design meets the desired performance specifications, and the architectural limitations of traditional scan structure restrict the two pattern delay tests that can be applied to a design, resulting in degraded delay test coverage. The use of enhanced scan flip-flops can alleviate this problem by supporting arbitrary delay test vector pairs, but at very high area overhead. We present a new, computationally efficient method for selecting the enhanced scan flip-flops which have most benefits of full enhanced scan at the cost of only 10-20% enhanced scan flip-flops. Thus the proposed techniques contribute to improving the efficiency and effectiveness of scan delay testing.

Table of Contents

List of Tables

List of Figures

1 INTRODUCTION

Continuing advances in design techniques and fabrication process technology are resulting in the design and manufacture of very high speed digital systems. Digital systems operation at high clock speeds does not allow for much design margin, so these circuits have to be designed under very tight timing constraints. In addition, the reduction in feature size increases the probability that a manufacturing defect in the IC will result in a faulty chip. In such a scenario, it is important to test each fabricated chip to ensure that the circuit indeed performs correctly at the specified clock speed.

The objective of delay testing is to detect timing defects and ensure that the design meets the desired performance specifications. Traditionally, there are two types of methods for delay testing: at speed functional testing and scan delay testing. Functional tests, including some created for design verification, are applied at system operational speed to screen out parts with delay defects. However, applying functional tests is becoming very expensive, given the need for a high-speed tester to apply such tests. This approach is still used extensively for high performance parts, such as microprocessors and digital signal processors (DSPs) for which the functional tests can be loaded into on-chip caches and applied with low-cost testers. Another problem with using functional tests is the lack of assurance for high test quality. Several industrial experiments have shown that tests not specifically targeting delay faults have limited success in detecting timing defects [1]. On the other hand, ATPG

(automatic test pattern generation) based scan delay tests target specific delay fault models and can be applied using low-cost testers. Scan testing has been widely used in industry for cost effective stuck-at IC testing for many decades. Efforts are now being made to extend its effectiveness to timing testing.

In full-scan design, all storage elements are replaced with scan cells, which are the configured as one or more shift registers (also call scan chains) during the shift operation. The main advantage of full-scan design is that it converts the difficult problem of sequential ATPG into the simpler problem of combinational ATPG. However, Scan based delay testing involve the application of two test vectors <V1, V2> via the scan chains, and because of the structural limitation of full scan design, arbitrary vector pairs (V2 is not fully controllable) cannot be applied during the test, thus limiting the delay fault coverage.

In the thesis, we focus on scan based delay testing. As mentioned above, delay fault coverage is limited by the full scan structure, and enhanced scan design has been proposed to solve the problem. However, due to the duplication of flip-flops in enhanced scan, area overhead might become a serious problem if the flip-flops are not selected carefully to convert into enhanced scan.

In the early days of delay testing, most defects affecting performance could be detected using tests for gross delay defects. Aggressive timing requirements of today's high-speed designs have introduced the need to test small delay defects and distributed faults caused by statistical process variations [2]. One way of detecting small delay defects is using faster-than-rated-clock testing. However, in faster-than-rated-clock testing, the test response from all other paths with nominal delay greater than the test clock period is unpredictable and

must be assigned an unknown X value. Thus a scan chain may capture a large amount of X-states. Current test compression techniques have not been designed to handle this problem, and cannot be used, leading to highly inefficient faster-than-rated clock timing tests.

Small distributed delay defects can be modeled using a path-delay fault model; however, practical designs have a very large number of paths, and only a small fraction of them can be tested in a scan environment. The selection of paths for delay testing is especially difficult in performance-optimized designs because they often have a large number of paths with long propagation delays [1]. In addition, selection of critical paths for testing requires accurate timing information for the design, which is not readily available. Process variation makes this problem even more complex.

This thesis addresses the first two challenges mentioned above: flip-flop selection for high coverage transition fault testing and an output compression technique for handling X-states from over-clocked-delay tests. The remainder of this thesis is organized as follows: Chapter 2 presents the background for transition delay fault detecting and current test compression techniques. Chapter 3 describes our idea for an output compression scheme. Chapter 4 introduces the flip-flop selection approach for enhanced scan. Finally, Chapter 5 presents a summary and conclusion of the thesis, along with suggestions for future work.

2 BACKGROUND

2.1 Current Test Compression Techniques

Test compression techniques have provided a major advance to IC test methodology over the past decade by offering better than an order of magnitude reduction in test data volumes and scan test application time. Their adoption by industry has been remarkably rapid. Significant compression of test inputs is possible because of the relatively low percentage of "care bits" in ATPG generated input test vectors; only information regarding these few bits (in some coded form) needs to be supplied by the tester to the circuit under test (CUT). The remaining bits making up the test inputs can be generated on-chip by dedicated test decompression hardware, either as "random fill", or to meet desired profiles to minimize power dissipation during scan, etc. Traditionally, the compaction of test output signals from the CUT has generally been viewed as a much simpler problem. Linear feedback shift register (LFSR) based multiple input signature registers (MISRs) have been in use for several decades [3]. Relatively small (20-32bit) MISR registers can compress test response data by several orders of magnitude, with only a minimal probability of aliasing, where a faulty test response results in the fault free signature. Although such MISR based output compression continues to be used, EXOR based combinational output compactors are also gaining increasing popularity [4]. Here EXOR trees are used to compress subsets of the scan chain outputs into a single bit. While such combinational compaction typically generates more test result bits

(output data bits) to be compared against the expected response, this approach has the ability to tolerate an occasional undefined X-state in the scan outputs. Observe that in the event of an X-state captured as a test response in the scan chain, only those scan output bits compacted in the EXOR tree containing the X are invalidated. Test results at the other compacted outputs are still valid and can be observed for faults. In contrast, a single X input into a MISR can generally invalidate the resulting signature. (An innovative X-Canceling MISR scheme has been recently presented [5] that can tolerate a small number of X-values.)

EXOR based output compaction, along with various degrees of additional masking capability to filter out some X-states from the EXOR trees, is now widely used in industrial designs. It has proved to be very effective for traditional scan based stuck-at testing, where the number of X-states captured in the test response is generally quite small, typically well below 5%. However, if the number of X-states in the output response grows to 20-50% or more, the current methodology fails because every EXOR tree is likely to receive at least one (unmasked) X-input, invalidating the resulting test output. Such a scenario is encountered in scan based testing for small (fine) delay defects on short paths using faster-than-rated clocks; the test response from all other paths with nominal delay greater than the test clock period is unpredictable and must be assigned an unknown X value. Since often half or more of all circuit paths can be longer than the short paths being targeted by the faster test clocks, the number of X-states in the delay test response captured in the scan chains can become large and unbounded. Current test compression techniques have not been designed to efficiently handle such a large number of X-states. As a result, aggressive faster-than-rated clock delay tests must be applied without any test compression, making them at least an order of

magnitude more expensive to run on a production tester when compared to stuck-at tests that can routinely achieve 10-25X compression. This prohibitive increase in (per vector) test application cost is an important factor limiting the viability of faster-than-rated clock tests as a solution to the difficult IC quality and reliability challenge posed by *small delay defects*.

2.2 Transition Fault Model

Transition fault model is a popular fault model for detecting delay defects in a circuit. It assumes that only one gate is affected by a delay fault in the circuit. There are two transition faults associated with each gate: a slow-to-rise fault and a slow-to-fall fault. It is assumed that in the fault-free circuit each gate has nominal delay. Delay faults result in an increase of this delay. Under the transition fault model, the extra delay caused by the fault is assumed to be large enough to prevent the transition from reaching any primary output at the time of observation. In other words, the delay fault can be observed independent of whether the transition propagates through a long or short path to any primary output [1]. Therefore, it is a gross delay fault model.

To detect a transition fault in a combinational circuit it is necessary to apply two input vectors, V=(V1,V2). The first vector, V1, initializes the circuit, while the second vector, V2, activates the fault and propagates its effect to some primary output. Vector V2 can be found using stuck-at fault test generation tools. For example, for testing a slow-to-rise transition, the first vector initializes the fault site at 0, and the second vector is a test for stuck-at-0 fault at the fault site. A transition fault is considered detected if a transition occurs at the fault site and a sensitized path extends from the fault site to some primary output [1].

The main advantage of the transition fault model is that the number of faults in the circuit is relatively small (linear in terms of the number of gates). Also, the stuck-at fault test generation and fault simulation tools can be easily modified for handling transition faults.

2.3 Full-scan Based Transition Delay Testing

In full-scan design, all storage elements are replaced with scan cells, which are configured as one or more shift registers (also called scan chains) during the shift operation. As a result, all inputs to the combinational logic, including those driven by scan cells, can be controlled and all outputs from the combinational logic, including those driving scan cells, can be observed. The main advantage of full-scan design is that it converts the difficult problem of sequential ATPG into the simpler problem of combinational ATPG. Figure 2.1 shows the full scan design scheme.

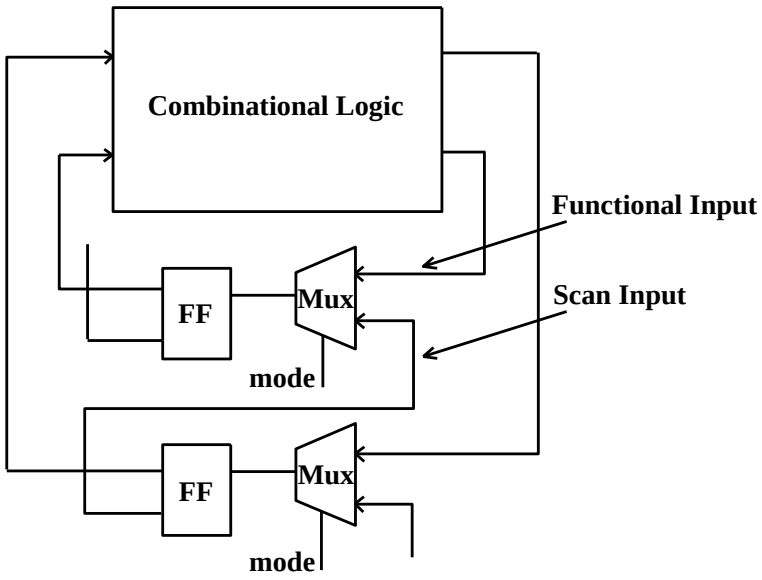

Figure 2.1 Full Scan Design Scheme

Scan based delay testing involve the application of two test vectors <V1, V2> via the scan chains. The first vector V1, which is used to initialize the internal logic values of CUT (circuit under test), is first scanned into the scan chain, typically using a slow scan clock. The

second vector V2 is then used to launch transitions at the inputs of the combinational part of the circuit. These transitions propagate to the outputs of the logic block and are then captured back in the scan chain by a fast capture clock pulse, reflecting operational frequency. Finally, the response captured in the scan chain is scanned out of the CUT and compared with the expected correct test response.

Unfortunately, because of the architectural limitations of scan, not all <V1, V2> combinations can be applied by a scan delay test. Depending on how the V2 vector is generated, scan delay tests are classified as Launch-on-Shift (LOS)[6,7], or Launch-on-Capture (LOC)[8, 9]. For the Launch-on-Shift test, the V2 vector is restricted to a one-bit shift from V1. For the Launch-on-Capture test, V2 is the response of the CUT to vector V1. In practice, LOS tests are not always supported because they require the scan enable signal to transit at-speed between the shift mode required to launch the test, and the functional mode required to capture the response of the timing test, all safely within a functional clock period. Such high speed scan enable signals are expensive to implement, although several CAD vendors now offer tools to support such an implementation. Nevertheless, low cost LOC test are generally preferred. Unfortunately, these restrictions on the V2 vector generally limit the transition delay fault (TDF) coverage achievable using both LOC and LOS scan delay tests. Achieving the very high TDF needed for high quality delay testing requires greater flexibility in choosing the V2 vector.

2.4 Enhanced Scan Transition Delay Testing

The enhanced scan approach was introduced to address this problem of low scan delay test coverage by removing the restrictions on the V2 vector and thereby allowing arbitrary

<V1, V2> combinations for high coverage delay testing. In the simplest enhanced scan schemes, one additional redundant flip-flop is interleaved with each of the functional flip-flops in the design, doubling the length of the scan chain, as shown in Figure 2.2. The V1 and V2 vectors can now be simultaneously scanned in and loaded into the scan chain, in an interleaved manner. At the initialization stage of the test, bits of the V1 vector are located in the functional flip-flops, while bits of the V2 vector located in the corresponding redundant flip-flop following each functional flip-flop. The delay test is applied in the LOS (launch-on-shift) mode, with the bits in the redundant flip-flops, which can now be chosen arbitrarily without any constraints, forming V2.

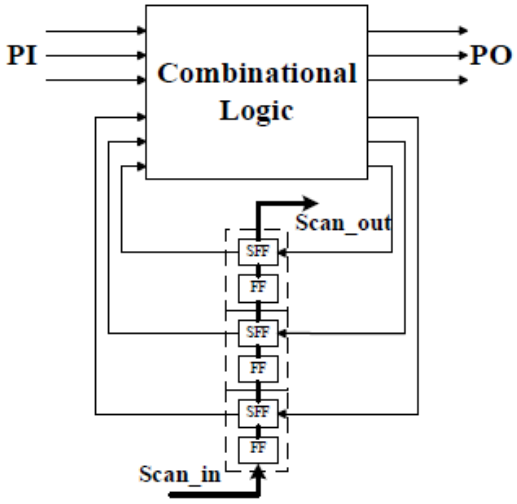

Figure 2.2 Classical enhanced scan with alternating regular and scan FFs[10]

Since the cost of duplicating all flip-flops in the design can be very high, a number of alternate enhanced scan approaches have been suggested to save some hardware costs. One design uses an extra “hold” latch (with an additional control line) at the output of each scan flip-flop. The idea here is to hold the V1 initialization pattern in these latches while an arbitrary V2 is being shifted into the scan chain [11]. Once the V2 vector is in place, the test can be launched by deactivating the hold control to make the latches transparent, thereby switching the inputs to the combinational logic from V1 to V2. An obvious disadvantage of this alternate enhanced scan design is the extra delays introduced on the signal paths. This is

addressed in the different enhanced scan design presented in [12]. Here the extra “hold” latch is implemented in parallel with the slave latch of the scan flip flop by using transmission gates to demultiplex the signal paths. Yet another technique, called First Level Hold, uses supply gating at the first level of logic gates to hold the state of a combinational circuit, instead of using an extra latch as in the other enhanced scan methods. This is claimed to reduce the area overhead for applying arbitrary two pattern tests [13, 14].

2.5 Partial Enhanced Scan

Although enhanced scan techniques have been around for several decades, they have rarely been used in practice so far because of the prohibitive area overhead. However, recent interest in achieving high delay test coverage from low cost LOC scan based tests, beyond what is possible from traditional LOC tests, has revived interest in such schemes [15]. The goal is to avoid the need for expensive at-speed functional tests, while achieving comparable

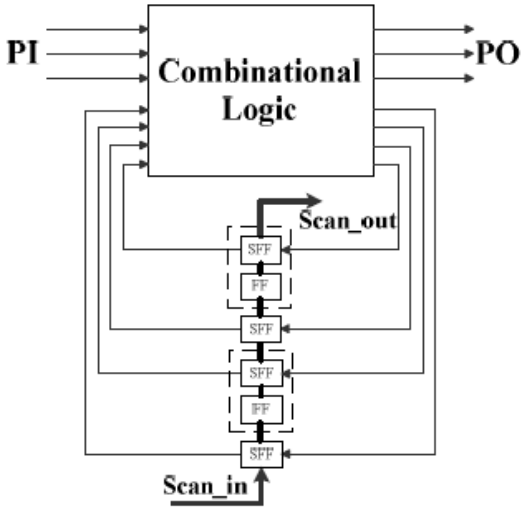

Figure 2.3 Partial Enhanced Scan (enhanced scan flip-flop pairs are enclosed in the dashed boxes)[10]

test quality. A particularly promising idea is the use of partial enhanced scan where the use of only a few carefully chosen enhanced flip-flops in the scan chain can deliver most of the benefits of a full enhanced scan design. This is illustrated in Figure 2.3.

2.6 Circuit Output Analysis

In a LOC scan delay test, the first vector V1 can be arbitrarily selected since V1 is scanned into the CUT from the tester. We can therefore say that V1 is completely un-biased by the structural limitations of scan design. The second vector V2, depends on V1 and cannot be arbitrarily selected. For a LOC test, each bit in V2 does not always have an equal probability of being '0' or '1' since V2 now is the response of V1, and is "conditioned" by the combinational logic. Some V2 bits captured from low controllability nodes at the output of the combinational logic (only single cycle combinational is considered here) are very frequently found to be '0' (poor 1 controllability) or '1'(poor 0 controllability). This bias degrades the TDF fault coverage because <V1V2> test patterns requiring V2 to take the poor controllability value are often impossible to apply. Thus we need to analyse the output controllability of the combinational part of a circuit in order to modify the circuit into enhanced scan design.

2.6.1 Monte Carlo Probability Simulation

By applying large number of random vectors to the inputs of the combinational circuit and counting the 0 and 1 of the outputs we can get output 0 and 1 probability information. The partial enhanced scan methodology presented in [10] attempts to identify and rank order the biased flip-flops in a design using Monte Carlo simulations. It then uses this ranked list of flip-flops as candidates for changing into enhanced scan flip-flops. Starting with the least controllable flip-flop input, this ordering is used to incrementally find the next flip-flop to convert to an enhanced scan flip-flop for the best TDF coverage improvement, as an increasing number of enhanced scan flip-flops are introduced in the partial enhanced scan

design.

However, in addition to the prohibitive cost of Monte Carlo simulation for large circuits, it was observed in [10] that these simple controllability estimates do not capture all the complex interactions between the inputs required to activate and propagate delay fault effects through the logic; controllability based rank ordering appears only somewhat loosely related to increasing TDF coverage with more enhanced flip-flops.

2.6.2 SCOAP Controllability Analysis

The *Sandia Controllability/Observability Analysis Program* (SCOAP) was developed by Goldstein [32] for testability analysis applications. In Glodstein's method of calculating controllabilities, the first step is to set the difficulty of controlling each primary input (PI) to 0 (called CC0) to the value 1 and the difficulty of controlling each PI to 1 (called CC1) to the value 1. We progress through the circuit in a forward pass, in level order. The level of a logic gate is the maximum of the distances (in logic gates) of its various inputs from the PIs. Thus if we calculate controllabilities of logic gates in order of increasing level number, then we will only process logic gates whose input signal controllabilities (CC0 and CC1) have already been determined.

For each logic gate that we traverse, we add 1 to the controllability. This accounts for the logic depth. If a logic gate output is produced by setting only one input to a controlling value, then:

Output controllability = min (input controllabilities) + 1

If a logic gate output can only be produced by setting all inputs to a non-controlling value, then:

Output controllability = $\sum$(input controllabilities) + 1

If an output can be controlled by multiple input sets (e.g., a two-input XOR gate where "01" and "10" input sets will both cause a 1 output), then:

Output controllability = min (controllabilities of input sets) + 1

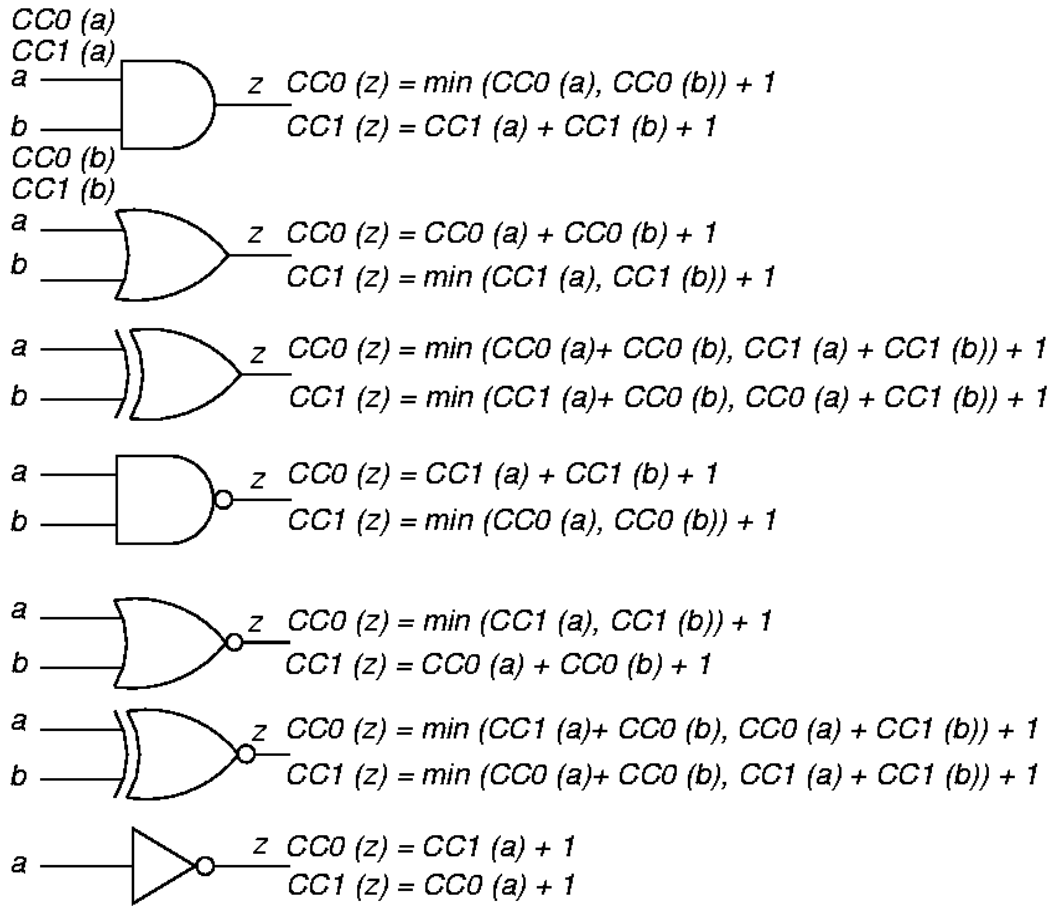

Figure 2.4 output controllability calculation of basic digital logic gates [11]

Figure 2.4 shows the output controllability calculation of all the basic digital logic gates. Errors arise in the controllability calculation due to reconvergent fanout where the reconverging signals may correlate, and therefore the controllability becomes inaccurate at the reconvergence point. Goldstein's procedure may overestimate or underestimate the controllability difficulty by assuming that reconverging signals are independent. Figure 2.5 shows an example of an output controllability calculation.

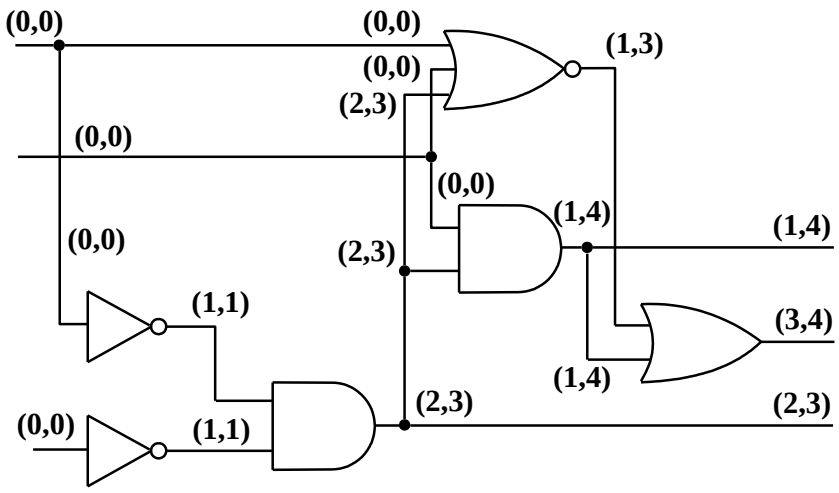

Figure 2.5 Example of Controllability Calculation

3 X – TOLERANT OUTPUT COMPRESSION SCHEME

3.1 Motivation and Prior Research

Recent technology generations, particularly since the move to copper interconnects in the 180-130 nm nodes, display a noticeable increase in delay defects that impact circuit timing. Recent research interest has specially focused on small (fine) delay defects, which can often remain hidden within circuit timing slacks, or clock timing margins, during testing. While it is sometimes argued that such defects which are not detectable at the rated clock speed during test are functionally benign and can be ignored, there is an emerging consensus that they must be detected to ensure acceptable product quality in high end ICs.

Because many such small timing defects can remain hidden in circuit timing slacks, particularly for shorter paths, if tested using the functional clock rate, there is growing interest in new test methods that detect excessive switching delays on signals within the slack interval, even when all the circuit outputs meet the nominal clock specification [16-20]. This requires strobing the circuit outputs (latching outputs in the scan chains) at frequencies faster than the nominal clock to observe transition times within the timing slack, as illustrated in Figure 3.1. If the expected switching time for each output signal is available from simulation, an observed switching delay during testing that is significantly in excess of that predicted for the line by the simulation indicates a delay defect. Small delay defects on short paths can be detected in this way, even when they cannot be observed at the rated clock. The

True-Time simulator in the Encounter Test system from Cadence is an example of a simulation tool being marketed to support such a test [20]. More recently, other faster-than-rated clock test methodologies have been presented that do not rely on accurate timing simulation and are more robust to process variations [16-19].

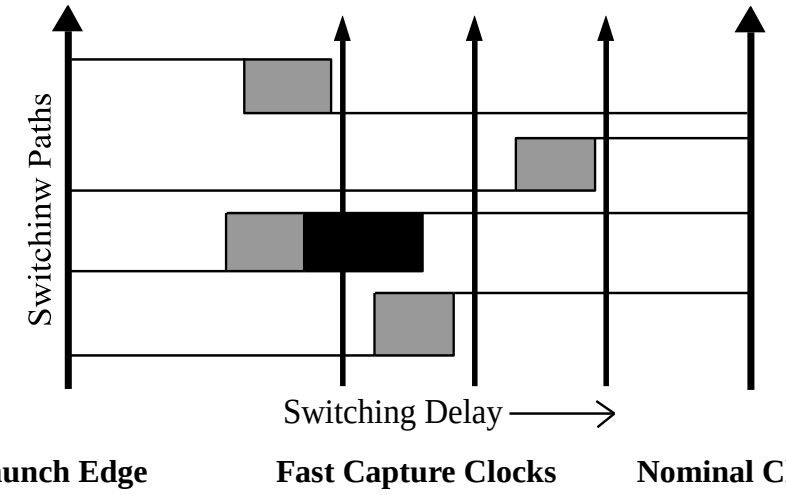

Figure3.1 Detecting Small Delay Defects on Short Paths using Multiple Fast Clock

Observe in Figure 3.1 that each switching transition can be tested by capturing timing information in the scan chains using the tightest possible clock, while allowing for normal process related timing variations (shown in grey). Note however, that only paths with delay shorter that the capture clock period are guaranteed to reach their final stable logic values. Because of possible switching hazards, the values captured in the scan chains for all longer paths are unpredictable, and must be treated as X (unknown) in interpreting the test response. This can result in a large and unbounded number of X-states in the scan out data during such delay tests when using aggressive fast capture clocks. Without a test compression scheme capable of handling such a large number of X-states, faster-than-rated clock tests must be applied without any output compression, which can make (per vector) test application costs at least an order of magnitude more expensive, greatly restricting their practical use in

targeting small delay defects.

A number of schemes for handling X values in test response compression have already been developed [3,21,22,23-25]. X-masking schemes prevent the X-values from propagating into the output compactor by using masking hardware controlled by additional control inputs, (coded as part of the test inputs as mask data) and applied to the CUT along with the test inputs. X-tolerant compactors can reliably compact data even in the presence of a few X-values [26,27,28,29]. More recently, X-Cancelling MISR based output compaction has been used[5]. However, most of these schemes are aimed at managing the small (or moderate) number (< 10%) of X-states that that can occur during stuck-at testing from sources such as uninitialized memory elements, signal contention, floating tri-state signals, etc. Even if the best of these techniques are combined together, e.g. by first masking out as many X-states as possible from entering the compactor and then applying X-Canceling on those that get through[30], only about 5-8% X-values can be handled without a significant fall-off in the attainable output compression. Such techniques will be completely overwhelmed by the 20-50% or higher number of X-states that can be generated during faster-than-rated clock timing tests as discussed above.

A recent paper [31] extends combinational X masking output compression schemes to include a “direct observation” mode that allows any scan cell to be directly connected to some test output, thereby guaranteeing full X-tolerance. While this direct observation is conceptually similar to the multiplexing capability required by our proposed approach, as presented in [31] it is only incorporated in an EXOR compaction environment to handle statistically unlikely combinations of X-states that might defeat the traditional X-tolerant

capability of the design for an occasional input vector. The application addressed is traditional stuck-at testing with a low to moderate number of X-states. The use of a multiplexing capability to efficiently handle unbounded X-states in over clocked delay tests is not explored in that paper, or any other prior work.

3.2 The Proposed Output Compression Scheme

The proposed output compression schemes takes advantage of the fact that for any test set, only a very small percentage of the output response bits in the scan out data need be observed to achieve the required test coverage for the targeted faults. Observing the other outputs only provides additional detection of faults that may already have been detected dozens, if not hundreds or even thousands of times at other observed outputs during the application of the test set. Observe that this idea is very similar to the underlying concept that allows significant compression of the test input vectors: only a very small number of bits in the input vectors are “care” bits, the rest can be randomly filled. However, significantly limiting the observed test response bits has not been used for output test data compression to date. In the thesis we present and evaluate a simple output data volume compression structure based on this idea.

Figure 3.2 shows a simple multiplexer based output compression scheme. In this example, the scan chains are partitioned into sets of 16, such that each set of 16 scan chain outputs are connected to a single test output pin through a 16-way multiplexer. Thus in each scan out cycle, only 1 out of every 16 scan outputs is available for observation at the tester; the other test response bits are ignroed. Also, the maximum output data volume compression

attainable is 16. While other, more efficient possibilities exist, let us first assume for

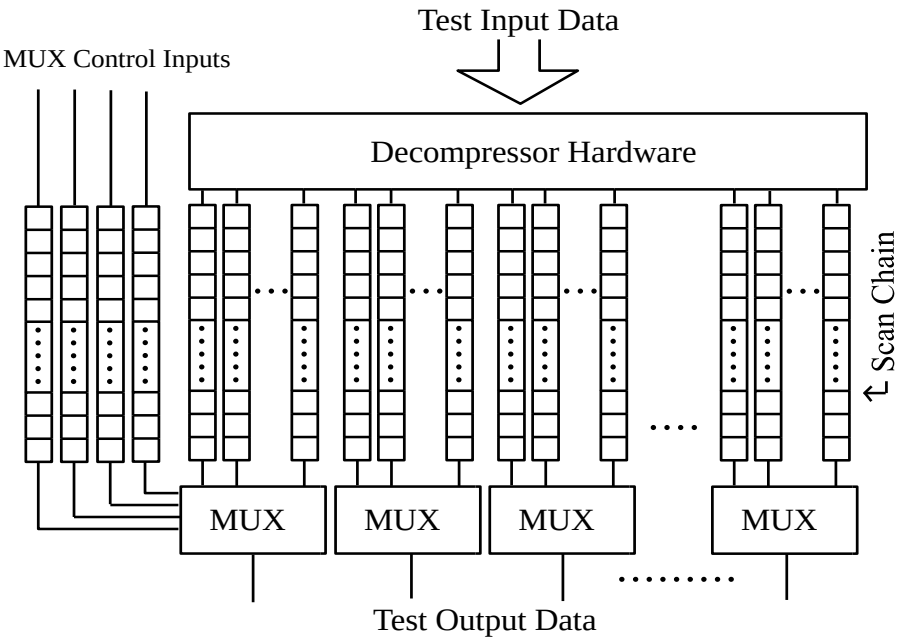

Figure 3.2 MUX based Output Compression Architecture

simplicity that all multiplexers receive the same control inputs during any scan out cycle, and the four control bits required are provided by four extra "scan" chains (actually four shift registers of the same length as the scan chains) that are loaded by the tester, in parallel with the loading of the scan vectors, using four additional dedicated input channels.

Observe in Figure 3.2 that no output compaction is performed by combining test outputs in EXOR gates, so X-values in the test response do not invalidate other valid test response bits. Both valid test response values and X-values appear unaltered at the multiplexed test output pins to be compared against the expected response in the tester. Thus this design can handle an unbounded number of X-states captured in the scan chains.

In evaluating the viability of the proposed approach, the first important question to be studied is whether the proposed output compression technique can achieve the same delay test coverage as a fully observable design, without an overwhelming increase in the required test set. In our example above that uses 16-way multiplexers, if the required size of the test

set needed is doubled, the effective output test time/test data volume compression drops from 16X to 8X. As indicated by the experimental results in the next section, the above numbers are typical when 16-way multiplexers are used at the output pins; better results are possible with larger multiplexers. Although this 5-10X test application time/test data compression may appear modest at first sight, compared to what is achievable for stuck-at tests, because of the large and unbounded number of X states, output compression for aggressive delay tests is currently not supported at all. The proposed approach can speed up such tests by up to an order of magnitude.

Note that our focus in this paper is on test result (output) compression. Scan delay test input patterns are very similar to stuck-at tests; only a single (initializing) pattern is scanned in while the second pattern (for the two pattern delay test) is generated by the on chip circuitry either through a scan shift (LOS: launch-on-shift) or from a functional response (LOC: launch-on-capture). Therefore, on the input side, the delay test set can take advantage of the impressive (20-50X or better [14]) compression factors achievable from available commercial test compression techniques. Any increase in the test set size because of the limited observability of the test response will of course degrade the effective input compression. The multiplexer control bits also add 10-20% to the input test data volume. For example, 4 multiplexer control chains introduced in a design with 20 (compressed) scan inputs implies a 20% overhead. However, because of the much higher input compression factors generally achievable, in practice test application time and test data volume will be mostly determined by the much more limited output compression possible for over-clocked delay tests. This is studied in the experiments in the next Section.

3.3 Experimental Results

To evaluate the effectiveness of the simple multiplexed output scheme presented in Figure 3.2, we performed simulations on the seven largest ISCAS 89 benchmark circuits; the smaller circuits do not have a sufficient number of flip-flops to form a meaningful number of multiple scan chains. The selected circuits have more than 179 flip-flops, to form at least 32 scan chains that are convenient to group into 8:1, 16:1 and 32:1 output multiplexers. Observe in Figure 3.2 that all the multiplexers receive the same control inputs during any scan out cycle; however these controls can be individually set for each scan cycle.

The ATPG challenge in our experiments is to generate compact scan delay test sets for the circuits when only one of the multiplexed scan chain outputs is observable during any scan out cycle as shown in Figure 3.2. The ATPG needs to be aware of this multiplexed architecture so that it can optimally assign the multiplexer control signals for each scan out cycle to find an efficient test set. To achieve this using commercial ATPG tools, we simulated a modified circuit that mimics the observability limitation of Figure 3.2 for LOS tests. This is shown in Figure 3.4. Compare the inputs for original non multiplexed chains, as shown in figure 3.3, to their counterparts in Figure 3.4, which illustrates our modified circuit used for ATPG. We feed the multiplexer inputs with signals that were originally going into flip-flops at the same level in every chain, and connect the fanned out output of multiplexer to inputs of all such flip-flops. Such a structure is duplicated k times to feed each of the k flip-flops in all the scan chains. The result is that in the modified structure all flip flops connected to the output of each multiplexer get the same value during response capture. The multiplexer input selected for capture depends on the multiplexer control values. These control signals to

every multiplexer are treated as (pseudo) primary inputs by APTG. Thus only one of the test response bits from each multiplexer is observable for every scan cycle, mimicking the structure of Figure 3.2 for ATPG.

Clearly, after the application of the first vector in functional mode, the values captured into flip-flops of the modified structure are no longer the full response from the combinational logic; instead the scan chains capture replicated copies of a subset of the response bits selected by the multiplexer control settings. This is why LOC delay tests cannot be simulated by our modified circuits. However, LOS delay tests launch from the shift mode and capture

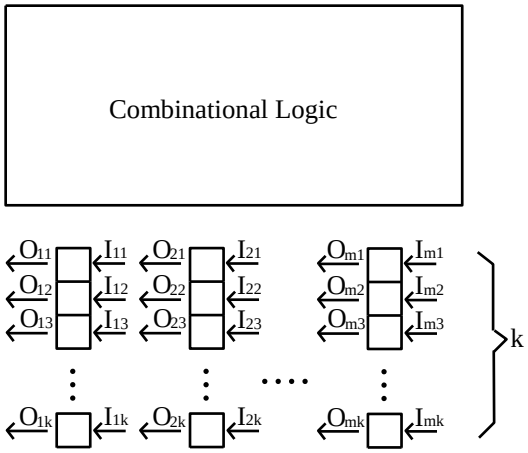

Figure 3.3 Flip-flop inputs/outputs from/to Combinational Logic

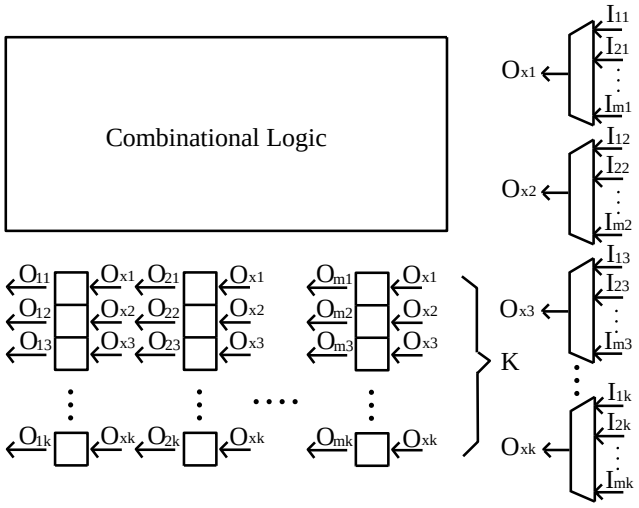

Figure 3.4 MUX based Output Compression Architecture

the functional response only once; these are correctly handled for ATPG purposes by the structure in Figure 3.4 with the scan out observability limitations of Figure 3.3.

In our experiments a commercial ATPG is used to generate LOS delay test input vector pairs for the combinational logic in the design and appropriate multiplexer control inputs, so as to achieve appropriate test coverage for the "multiplexed" scan-out structure. During any applied test pattern, only 1/16 (in the case of the 16:1 multiplexers shown in Figure 3.2) of the (pseudo) primary outputs of the combinational logic block are observed; the specific outputs observed are decided by the applied multiplexer controls. As the same value is always fed into 16 flip-flops because of the structure shown in Figure 3.4, these scan out values are always the same. This can be viewed as an output compression factor of 16.

We used DFTAdvisor to generate the expected number of scan chains for selected ISCAS89 benchmark circuits, and hand-modified them into the models shown in figure 3.4. We then generated LOS delay test sets using FastScan for fault coverage as near as possible to that obtainable for the unmodified original circuits. For each circuit, we used three different multiplexer sizes: 8:1, 16:1 and 32:1, with a fixed scan chain number of 32. The results are tabulated in Table 3.1. (Note that any faults inside the multiplexers are not included in our coverage reporting because our combinational model of the sequential scan structure requires replication of the multiplexers for each scan out cycle.)

Table 3.1 shows results for ISCAS89 benchmark circuits for the case where the total number of scan chains in the design is 32. This number was chosen to allow sufficient scan chain length for the smaller circuits. Four multiplexers are used when 8:1 multiplexers are employed, two are used when 16:1 multiplexers are employed, and a single 32:1 multiplexer

Table 3.1 Results for different Multiplexer sizes with 32 Scan Chains

Bench-mark Circuit	# FFs	# Gates	W/O Mux		With 8:1 Mux		
			# Patterns	Delay FC	# Patterns	Delay FC	Effective Output Compression
s5378	179	2779	114	58.3%	128	58.9%	7.13
s9234	211	5597	276	74.2%	369	75.0%	5.98
s13207	638	7951	532	91.2%	593	91.8%	7.18
s15850	534	9772	344	89.2%	428	89.7%	6.43
s35932	1728	16065	81	86.3%	224	85.9%	2.89
s38417	1636	22179	298	96.8%	320	95.5%	7.45
s38584	1426	19253	319	91.8%	416	91.0%	6.13

Bench-mark Circuit	With 16:1 Mux			With 32:1 Mux		
	#Patterns	Delay FC	Effective Output Compression	#Patterns	Delay FC	Effective Output Compression
s5378	167	59.1%	10.92	191	59.0%	19.10
s9234	397	75.1%	11.12	434	75.2%	20.35
s13207	649	91.8%	13.12	639	91.8%	26.64
s15850	473	89.7%	11.64	512	89.8%	21.50
s35932	320	86.3%	4.05	352	84.5%	7.36
s38417	378	95.8%	12.61	432	95.8%	22.07
s38584	512	91.0%	9.97	640	91.5%	15.95

is used in the last instance. The table shows the LOS transition delay fault coverage, and the number of test patterns needed when all scan-outputs are observable (no compression), and the fault coverage and number of test patterns needed where the only the multiplexed outputs are observed during each scan cycle. Notice that the number of test patterns needed to achieve roughly the same coverage increases when only a few multiplexed outputs are observed. The effective output data/test time compression is the multiplexing factor discounted by the fractional increase in test set size to achieve the same coverage. Observe that for most of the circuits, impressive output test data compression can be achieved, particularly for large multiplexer sizes. The only exception is s35932, for which the results are more modest. This appears to be because the requisite coverage with full output visibility

(without output multiplexing) is achieved for this case with very few (81) test patterns as compared to similarly sized circuits. This suggests that each test pattern detects many faults, and most likely on many different outputs. Clearly, if only a few outputs are observed through the multiplexers, many of the faults will be missed, requiring additional test patterns. However, most circuits are not likely to display such high testability with compact test sets, and can therefore be expected to display better output compression factors with the multiplexed architecture.

The experiments above assumed 32 scan chains because of the modest number of flip flops in the designs. This resulted in only a single 32:1 multiplexer being used in designs using the largest multiplexer. Unfortunately, this does not capture the impact of a key architectural limitation in Figure 3.2 which forces the same control signals on all the multiplexers during a scan cycle. When all scan chains are fed into a single multiplexer, there is no conflict in setting the multiplexer control values. To study this issue further, we considered the two largest benchmarks circuits, which have a sufficient number of flip flops (about 1500) to allow up to 128 scan chains. Results for designs using 32:1 multiplexers are presented in Table 3.2. Notice the increase in the number of required delay test patterns, and the corresponding decrease in the compression factor, as the number of scan chains (and therefore multiplexers) in the design increases. This is because fewer desired scan out bits from a test response can be observed without potential multiplexer control conflicts in a design with 128 scan chains and four 32:1 multiplexers than for 32 scan chains which need only a single 32:1 multiplexer. Nevertheless, the results show that a 10X compression factor or better is still achievable. Designs with still larger number of scan chains will optimally

employ 64:1 or even larger multiplexers; the results in Table 3.1 show that the compression factor improves with the size of the multiplexers used.

Table 3.2 Application of 32:1 Mux on Different Number of Scan Chains

Scan Chains	Effective Output Compression & Fault Coverage									
	s38548					s34817				
	#pattern w/o mux	Delay FC w/o mux	#pattern w/ mux	Delay FC w/ mux	Comp factor	#pattern w/o mux	Delay FC w/o mux	# pattern w/ mux	Delay FC w/ mux	Comp factor
32	319	91.80%	640	91.49%	15.95	298	96.80%	432	95.79%	22.07
64	311	90.67%	760	90.44%	13.09	312	93.89%	487	93.05%	20.50
128	302	87.68%	960	87.26%	10.07	301	91.10%	576	89.49%	16.72

One option to partially relax this limitation on all output multiplexers receiving the same control inputs is to use a few additional control inputs (but less than the number needed for controlling all the multiplexers independently –which would be prohibitively expensive) and generate the multiplexer control signals using phase shifters. Unfortunately, unless the number of inputs becomes close to what is needed for independent control, such a strategy allows too few different control patterns at the multiplexer inputs to easily allow simultaneous multiple fault detection at different multiplexer outputs. Exploiting such a design will require an ATPG engine that fully understands the compression architecture and the phase shifter design. To estimate the best achievable results in the latter case, Table 3.3 presents the comparison of compression factors if all output multiplexers could in fact be independently controlled, instead of receiving common control inputs.

3.4 Conclusion and Discussion

In conclusion, while the 10-15X overall test compression for the larger circuits presented

here is less impressive than commonly quoted for industrial strength compression methodologies, it is important to note that these compression results can support an

Table 3.3 Compression Factors with Common and Independent Mux Control

#Scan Chains	Effective Output Compression Factor							
	s38548				s34817			
	16:1 mux comm ctrl	16:1 mux indpdt ctrl	32:1 mux comm ctrl	32:1 mux indpdt ctrl	16:1 mux comm ctrl	16:1 mux indpdt ctrl	32:1 mux comm ctrl	32:1 mux indpdt ctrl
64	9.11	11.66	13.09	16.18	10.05	12.59	20.50	23.60
128	6.99	11.03	10.07	16.24	9.01	12.49	16.72	24.38

unbounded number of X-states.

Many improvements and optimizations to the basic approach presented here are possible, and will be the subject of future research. For example, the multiplexer control inputs can also be compressed, although this will result in loss of full control on the scanned out bit in each scan cycle. This problem can be partially alleviated by using phase shifter circuits to form the multiplexer controls from a larger number of bits in the multiplexer control chains for enhanced flexibility in setting the multiplexer controls. Selecting an appropriate aspect ratio for the scan chains may also be a factor in optimizing output compression; fewer scan chains provide greater flexibility in selecting the output bits observed, but increase test application time. Finally, a good dynamic compaction capability is essential to developing compact test sets for the proposed approach, where output observability is limited. These possibilities will be explored in future work.

While the proposed output test data compression approach appears to be vitally useful when a very large number of X-values are expected in the test response, as in aggressive delay testing, traditional compaction based approaches are still likely to provide significantly

better compression for slow speed tests where the number of X-values can be controlled or bounded. Such tests that observe virtually all scan outputs are also essential for targeting unmodeled defects. This suggests a hybrid test response compression architecture that can allow both approaches to be used as appropriate for a given test set.

4 PARTIAL ENHANCED SCAN FLIP-FLOP SELECTION APPROACH

While earlier research [10] has established the potential of a partial enhanced scan approach in achieving high delay test coverage with low overhead, the goal of this work is a practical and computationally efficient flip-flop selection methodology that does not require extensive Monte Carlo simulations; at the same time it should deliver significantly better results in terms of area overhead and delay test coverage. For this purpose, we aim to take advantage of the analytically derived testability measures available with most commercial test suites. Circuit "controllability" and "observability" testability measures are quickly and easily estimated for the nodes in a circuit by tracing circuit paths and considering the input-output relationship of the gates along the path. No random vector simulation or back tracking is involved as in ATPG. In this paper we specifically work with the SCOAP testability measures [11] and Mentor Graphics tools.

4.1 Input Constraint Fault Influence

By using SCOAP analysis we have the controllability information of each state output of a circuit. Some of the state outputs are extremely hard to set to 0 or 1, but some are very easy to set up. However this information alone cannot help us to decide which scan flip-flop should be modified to enhanced scan flip-flop. Assume one state output has very large 1 controllability measure (CC1 >>1) from SCOAP, but only few faults require a 1 at that state

input in their launch time frame, then even if CC1 >>1, inserting an enhanced scan flip-flop at the state input cannot help us detect significantly more faults. We can only select 10% to 20% of the scan flip-flops and so we need to select those which can help us detect as many faults as possible.

Recall that in a two vector delay test, V1 is the initializing vector, while V2 launches the transition at the target node and also sets up conditions along the signal path so that a slow to rise/fall transition is observable at some signal output. Unfortunately, in a launch-on-capture test, only V1 can be scanned in; the key V2 patterns are generated by the logic and cannot be independently controlled. The V2 vector is actually a stuck at fault detection vector and by finding the number of gates in the circuit affected by an input constraint to 0 and 1 on each flip-flop (V2) output, we can determine how many stuck at faults are affected by the input constraint to 0 and 1. If the number is small, then it is less important for the corresponding flip-flop to be made fully controllable, allowing it to be dropped from further consideration to

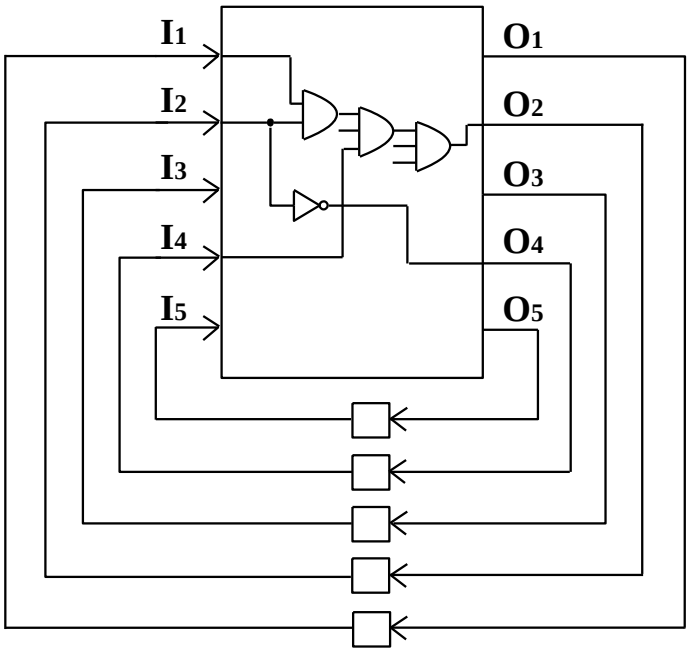

Fig. 4.1: Circuit example showing TDF degradation for LOC tests because of V2 dependency on V1

yield a more compact selection of the most effective partial enhanced scan flip-flops. We explain the reason behind heuristic next with the help of a simple example.

Figure 4.1 helps illustrate the above observation. First note in Figure 4.1 that the V1 vector of the LOC test is the input scanned in and applied to the combinational logic to generate outputs that are captured as V2. Therefore in the Figure, V1 = I1,I2,I3,I4,I5 and V2 = O1,O2,O3,O4,O5. Now consider I1 constrained to 0. This results in controlling values on the inputs of the three AND gates. Therefore, poor 1-controllability of O1, which is captured and applied as I1 in V2, will appear as a 0 constraint on I1 in the second cycle, and force 0s at the outputs of the three AND gates. (Since the signal has poor 1 controllability, it will mostly take a zero value.) This can result in significant lost TDF coverage since a large number of affected signal values in the circuit due to the constrained input can block propagation of fault effects. On the other hand, a 1 constraint on I1 does not affect many circuit gates. Consequently, poor 0-controllability of O2 will not have much impact on TDF coverage. Thus, a measure that evaluates the number of gates affected by a low controllability flip-flop can effectively help prioritize the flip-flops that need to be enhanced. The "Gate Reporting" function in DFT Advisor can be used to get this information (number of gates affected by input constraint) easily so we don't need to write another program to do that.

4.2 The Flip-Flop Selection Procedure

We now describe our flip-flop selection procedure. First we use SCOAP in DFT Advisor to generate the '0' and '1' controllability values for each flip-flop input (or equivalently, the outputs of the combinational logic block in Figure 4.1). SCOAP controllability measures are

positive integers, and are not scaled. Very small controllability values such as 0 or 1 suggest highly controllable signals that can be easily set to the target logic value. Larger controllability values, such as 20 or 35, indicate increasing difficulty in setting the signal to the desired value.

Next we use the Gate Reporting option in DFT Advisor to report the number of gates affected by constraining each flip-flop output (input to the combinational block) to 0 and also to 1. This constrains an input to 0 (or 1), and then traces this input forward to all possible outputs, counting the number of gate outputs that are forced to 0 or 1 values by this input constraint. This measure is an indication of the number of TDF that may go undetected as a result of the input constraint. (The input, part of the V2 vector is, of course, constrained because of possible logic "bias" as it is generated from V1 as explained earlier.)

We now need to combine the above two measures to obtain a single prioritization parameter for each flip-flop. This requires re-scaling of the controllability values because the values generated by SCOAP are not proportional to their actual impact. We use a general formula of the form:

Scale value = (SCOAP controllability value – a)*b +1.

Here "a" is the smallest SCOAP controllability value that can require a flip-flop being converted to enhanced scan, and "b" is a re-scaling parameter. We subtract "a" from the controllability value because the state outputs which have the value less than "a" are considered easily controllable and we do not need to consider them for enhanced scan. For different circuits, these two values may vary to generate the most accurate scale value. From experimentation with the ISCAS89 benchmark circuits, we found the best values to be a = 2

and b = 0.2. This implies that a SCOAP controllability values of 2, 3, 4 etc. are re-scaled to 1, 1.2, 1.4. Those state outputs which have 1 and 0 controllability values are not considered for modification into enhanced scan. Clearly other scaling formula can developed to be equally effective.

After rescaling the controllability values, we multiply these values by the number of gates affected by the 0 or 1 input constraint implied by the lack of controllability of the corresponding signal. For example, if a flip-flop has 0 controllability of 10 (the larger the number the harder it is to force the output to 0), then the re-scaled controllability value is (10-2)*0.2+1 = 2.6. Now if the number of gates affected by that input constrained to 1 is 30, then the prioritizing value for that flip-flop is computed to be 2.6*30=78. We repeat the same calculation again starting with the 1 controllability value for that flip flop. Since a significant TDF impact from a constraint to either 0 or 1 value suggests that the flip-flop should be enhanced, we choose the larger resulting value to be the final prioritizing value for that flip flop.

We perform this procedure for every flip flop in the circuit, and then order them by putting the flip flops with the largest prioritizing values in the front of the list. Thus we have an ordered list which gives us the relative importance of flip-flops to be made into enhanced scan. We pick flip-flops in order from the front of the list to convert into enhanced scan until we get satisfactory fault coverage.

While our new flip-flop ordering procedure performs significantly better than the approach described in [10], it is still a heuristic approach and obviously not optimum. In fact, evaluating a single flip-flop at a time may not capture all the complex interactions between

the inputs required to activate and propagate delay fault effects through the logic. For example, in some cases two (or more) flip-flops may not significantly improve coverage when only one of the two is enhanced at a time, but may have a much greater impact on TDF when both are enhanced together. Such complex interactions will be missed by our procedure. Therefore, an interchange procedure can still further improve our results. However, as will be observed in the next section, it can usually be limited to a relatively small subset of flip-flops.

4.3 Interchange Procedure

After the flip-flop selection, we do the interchange procedure within the relatively small subset of flip-flops (10%-30%). Assume that the resolution of the Interchange Procedure is 5%. Then the small subset of flip-flops in the circuit are assigned into 20 groups (G_1, G_2… G_{20}). G_1 contains the first 5% of the flip-flops, G_2 contains the first 10% of the flip-flops, and so on. G_0 is also defined as a group which contains zero flip-flops.

FC_N (N=0, 1, 2 …20) is defined as the fault coverage when flip-flops in groups indexed £N (G_0, G_1…G_N) are implemented with enhanced scan flip-flops.

$Slope_N$ (N=1, 2… 20) is defined as FC_N -FC_{N-1}. (The goal of idea FF selection is to ensure that $Slope_N$ decreases monotonically when N increases)

$Delta\_Slope_N$ (N=2, 3…20) is defined as $Slope_{N-1}$-$Slope_N$. (For the ideal FF selection ordering, $Delta\_Slope_N$ is always non negative.)

In the Interchange Procedure for the experiments reported in the thesis, we arbitrarily choose the Interchange_allowable_times to be 10. A larger number can also be chosen which will provide more accurate results at the cost of greater computational effort. Similarly, we

choose the Delta_Slope_allowable_value to be -0.03 instead of zero in order to reduce the number of interchange iterations. Interchanges between very small differences in slopes will not yield a meaningful difference in the fault coverage versus Enhanced Scan FF percentage trade-off. Note that each time when we update FC_{M-1}, we must resynthesize a new partial enhanced scan circuit and re-run Fastscan (Menter Graphics tools for testing).

```
Calculate FC_N (N=0, 1, 2…20);
Interchange_times=0;
Delta_slope_allowable_value=-0.03;
Interchange_allowable_times=10;
While (minimum {Delta_Slope_N} < Delta_slope_allowable_value)
        && (Interchange_times<Interchange_allowable_times)
{
        Find M, where Delta_Slope_M = minimum {Delta_Slope_N};
        Exchange Flip-flops in group G_M with flip-flops in group G_M-1;
        Update FC_M-1;
        Interchange_times = Interchange_times +1;
}
```

4.4. Experimental Results

In order to study the effectiveness of the new partial enhanced scan methodology presented in this paper, we again investigated five of the larger ISCAS89 benchmark circuits studied in [10]. These contain from 74 to 638 flip-flops; the smaller benchmark circuits have too few flip-flops to provide meaningful results. Several of the larger benchmarks displayed high nineties LOC TDF coverage without the use of enhanced scan; given the limited headroom for TDF coverage improvement in these circuits, these were not investigated in [10]. However, to see how our methodology would work with larger circuits, we also applied it on s38584 (1426 flip flops) which has a 97.4% TDF coverage without any enhanced scan. For each design, we obtained an ordering on the flip-flops as explained earlier, and then

evaluated TDF coverage as the number enhanced scan flip-flops was increased, 5% at a time, based on the ordering in the prioritized list. Finally, the interchange procedure from [10] was again used to obtain the final plots in Figure 4.2.

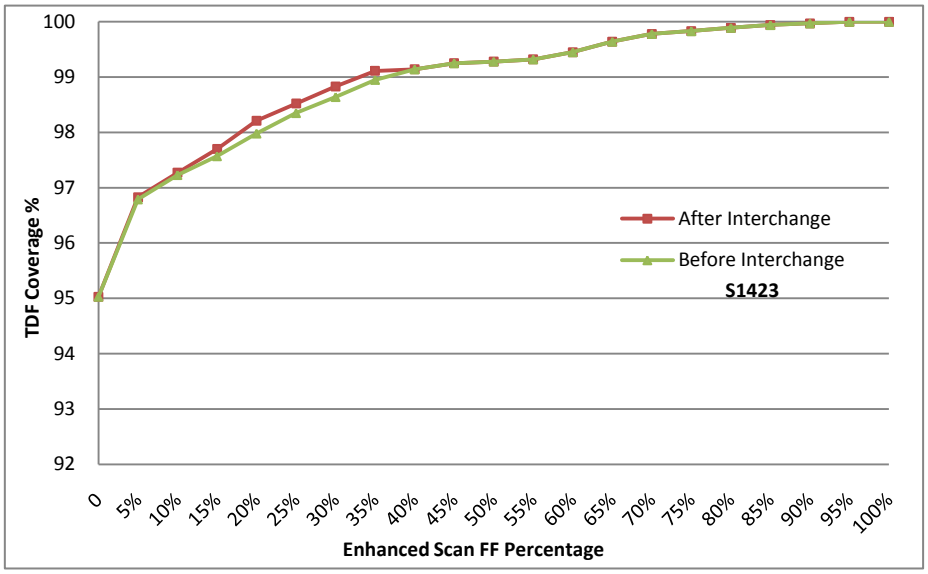

Figure 4.2(a): Benchmark s1423

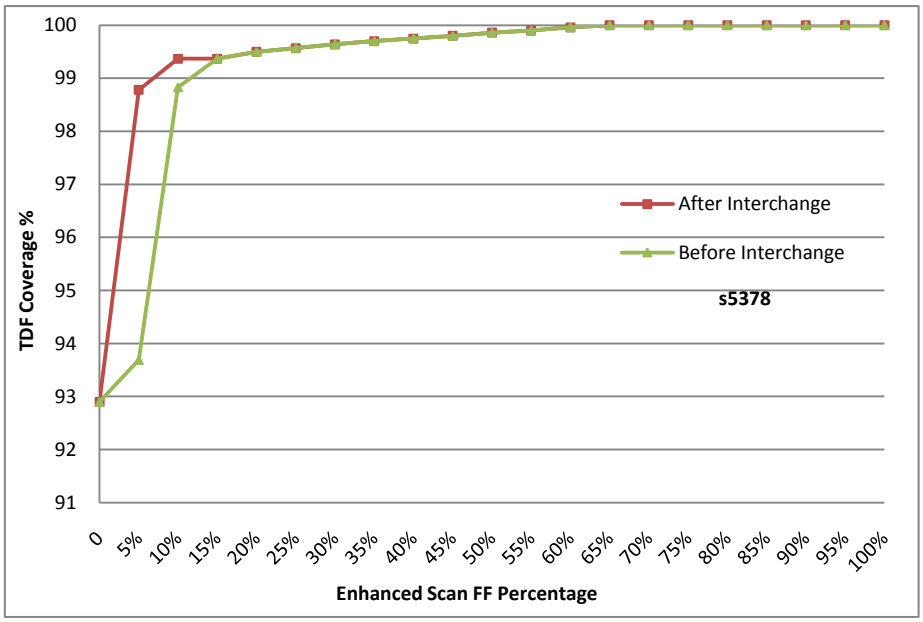

Figure 4.2 (b): Benchmark s5378

The plots in Figure 4.2 show the results. Note that the fault coverage we reported here does not count undetectable faults, so with full enhanced scan, TDF coverage reaches 100%.

The plots all display an attractive fault coverage versus percentage enhanced scan trade-off, except perhaps the smallest s1423 in Figure 4.2(a) which requires 35% enhanced flip-flops to get TDF coverage up to 99%. In all the other cases, a relatively small percentage of enhanced scan flip-flops provides most of the benefit of full enhanced scan.

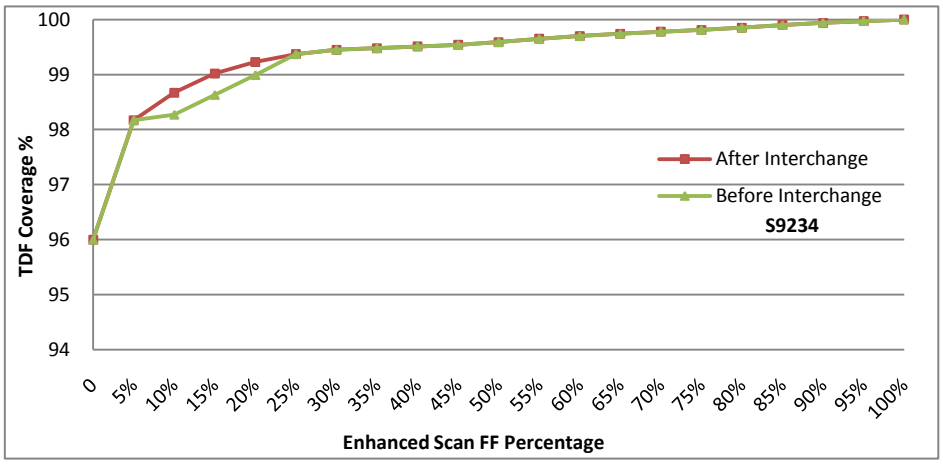

Figure 4.2 (c): Benchmark s9234

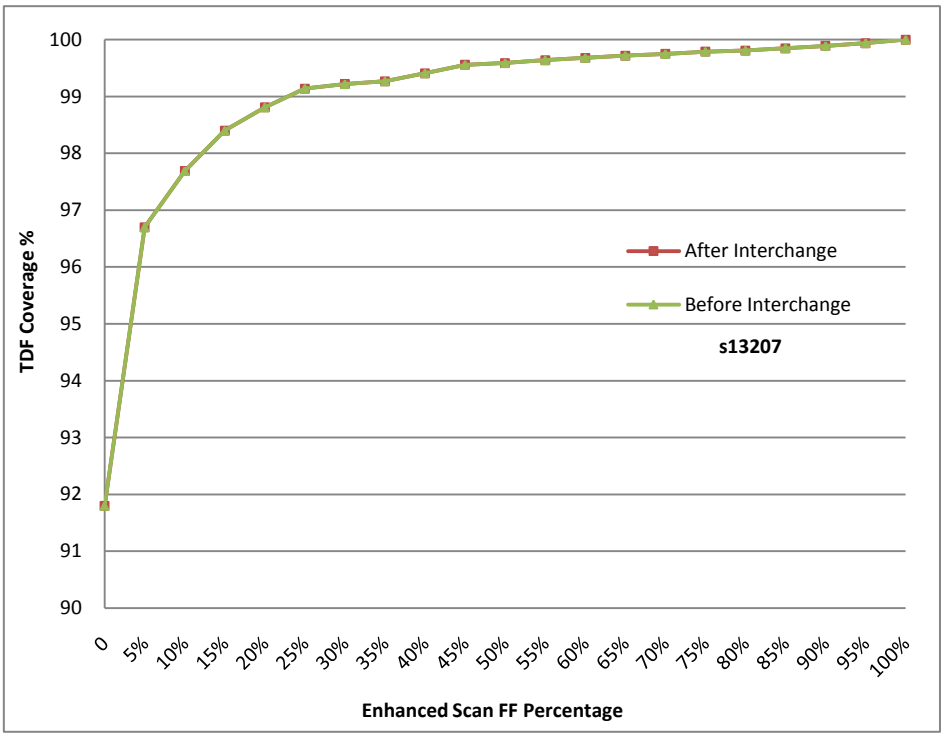

Figure 4.2 (d): Benchmark s13207

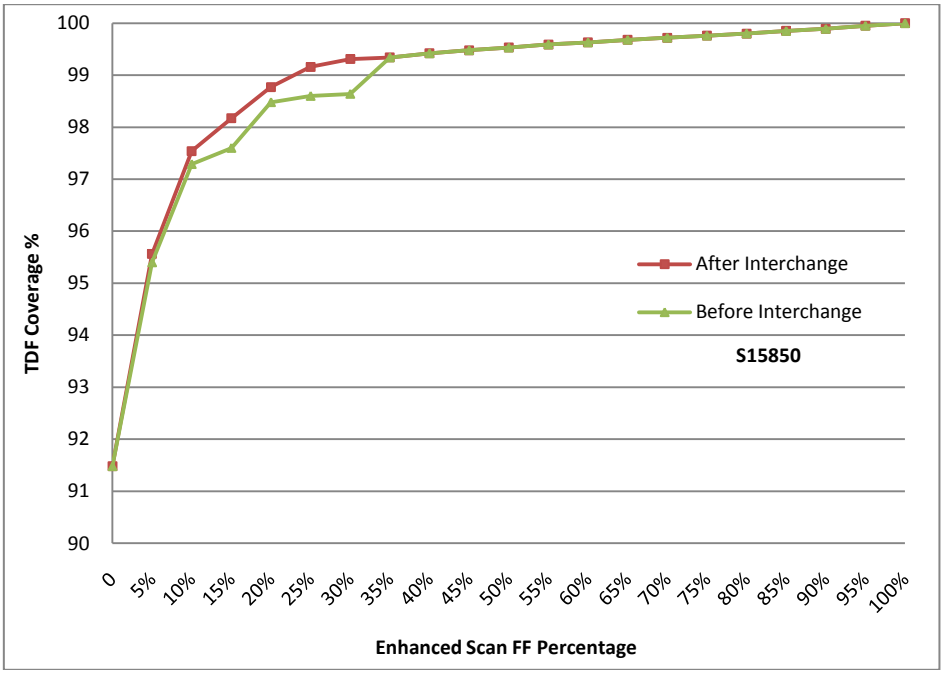

Figure 4.2 (e): Benchmark s15850

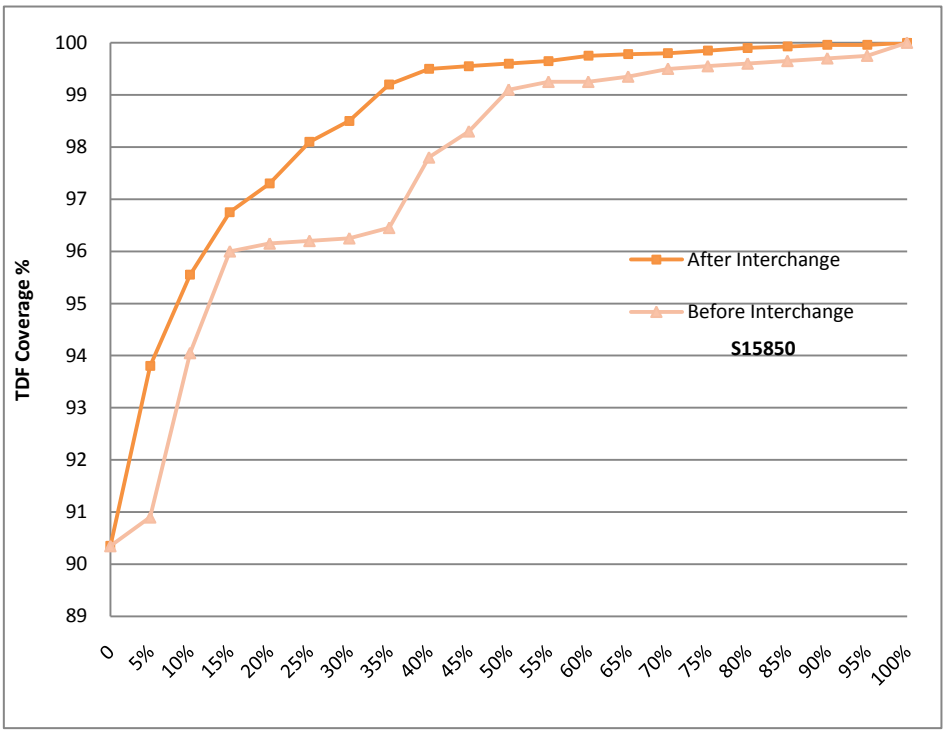

Figure 4.2 (f): Benchmark s15850 from [10]

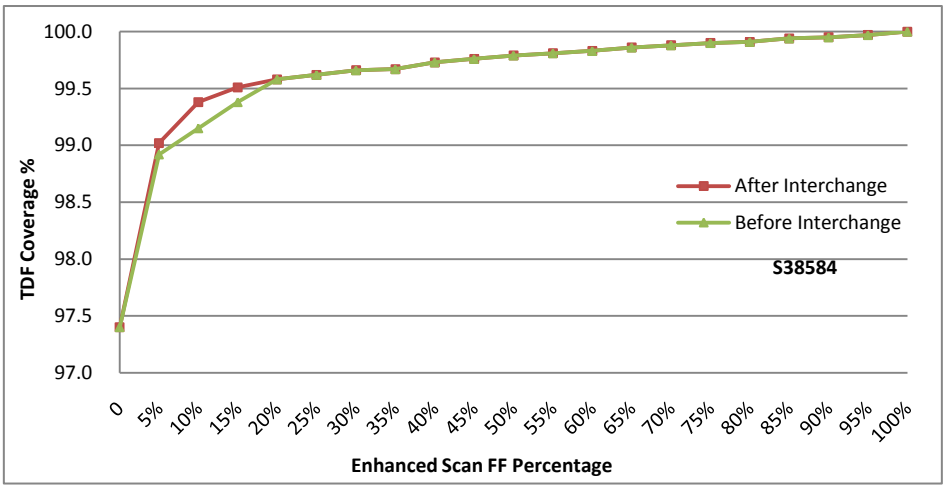

Figure 4.2 (g): Benchmark s38584

The effectiveness of our new methodology over earlier results can be seen by comparing Figure 4.2 (f), reproduced from [10], with Figure 4.2(e) which shows the new results for the same benchmark circuit s15850 (although [10] employed the Synopsis Test Suite). Notice the much improved flip-flop ordering by our new method by comparing the two plots before the interchange procedure. For example, with 20% enhanced flip-flops, Figure 4.2(e) shows a 98.5% TDF coverage while the coverage is only about 96% in Figure 4.2(f). In fact our results even without the interchange procedure are always better than the best results with interchange in prior research [10]. Additionally, our new method requires relatively few interchange iterations to achieve optimal results when compared to the earlier work.

To more comprehensively compare the results of the new flip-flop selection methodology for partial enhanced scan with earlier results, consider Table 4.1. For each of the benchmark circuits shown, the table lists the fraction of enhanced scan flip-flops that can achieve 70, 80, and 90% of the coverage improvement achievable with full enhanced scan design. For example, if for standard scan, a circuit has 90% TDF coverage, while it has 100% coverage with full enhanced scan, then 70% of the improvement (from full enhanced scan) would

correspond to a coverage of 97%, 80% improvement to 98%, and 90% improvement to 99%. Table 4.1 shows that on average the new approach can achieve 80% of the improvement in coverage from full enhanced scan using only 16.6% enhanced scan flip-flops. This is a significant improvement over the earlier approach which required almost 28%. For some circuits, such as s5378, very few flip-flops can provide nearly the same coverage as full enhanced scan.

TABLE 4.1 TDF Comparison of the new approach with [10]

Bench mark circuit	Enhanced Flip Flops Needed Improvement Relative to Full Enhanced Scan					
	Gefu Xu's result from[10]			Our new results		
	70%	80%	90%	70%	80%	90%
S1423	30%	36%	45%	25%	34%	44%
S5378	11.5%	14%	18%	3%	4%	8%
S9234	26%	35%	41%	13%	18%	35%
S13207	20%	29%	75%	9%	14%	26%
S15850	18%	25%	32%	10%	16%	24%
S38584	-	-	-	8%	14%	39%
Average	21.1%	27.8%	42.2%	11.3%	16.6%	29.3%

4.4 Conclusion and Discussion

The architectural limitations of traditional scan restrict the two pattern delay tests that can be applied to a design, resulting in degraded delay test coverage. The use of enhanced scan flip-flops can alleviate this problem by supporting arbitrary delay test vector pairs, but at very high area overhead. Earlier work[10], using a Monte-Carlo simulation based flip-flop selection procedure on the smaller benchmark circuits has shown that most of the TDF coverage benefits of full enhanced scan can be achieved by using only 20-30% enhanced scan flip-flops. However, Monte-Carlo simulation to obtain signal probabilities to identify

flip-flops that have poor controllability is not practical for large circuits. We present a new, computationally efficient method for selecting the enhanced scan flip-flops that leverages commercial testability tools by using easy to compute SCOAP testability measures. Furthermore, our method substantially improves on the earlier partial enhanced scan results by developing a methodology for eliminating some poor controllability flip-flops as candidates for enhanced scan through analysis of signal constraints due to the circuit structure. We also discover additional flip-flops that display strong dependencies between the V1 and V2 vectors during LOC tests that were missed in [10]. The result is a computationally efficient selection method that identifies only those flip-flops where the dependency between the V1 and V2 vectors in LOC tests limits TDF coverage. Our results show that the use of only 10-20% enhanced scan flip-flops can support high quality delay tests. This can make it viable to use low cost partial enhanced scan along with the slow scan enable designs discussed in this paper, particularly in applications where high quality delay testing is essential.

5 SUMMARY AND CONCLUSION

Aggressive timing requirements of today's high speed IC designs have introduced the need for small delay defects testing and very high delay fault coverage. Small delay defects can be targeted by faster-than-rated clock tests where unbounded X states can be generated. Current output test compression techniques have not been designed to efficiently handle large numbers of X-states. In this thesis, a multiplexing scheme for output data compression is proposed to deal with the problem. The compression scheme takes advantage of the fact that for any test set, only a very small number of the output response bits in the scan out data need to be observed to achieve the required test coverage for the targeted faults. Experimental results with ISCAS 89 benchmark circuits show that 10-15X overall test compression ratio for transition delay faults can be achieved.

To further improve the efficiency of the proposed output compression scheme, the multiplexer control can also be compressed, although this will result in loss of full control on the scanned out bit in each scan cycle. This problem can be partially alleviated by using a phase shifter circuit to form the multiplexer controls from a larger number of bits in the multiplexer control chains for enhanced flexibility in setting the multiplexer controls. Selecting an appropriate aspect ratio for the scan chains may also be a factor in optimizing output compression; fewer scan chains provide a greater flexibility in selecting the output bits observed, but increase test application time. Finally, a good dynamic compaction capability is

essential to developing compact test sets for the proposed approach, where output observability is limited. These possibilities will be explored in future work.

Delay fault coverage is also limited by the traditional full scan design structure. Prior research on partial enhanced scan flip-flop selection is not efficient and effective enough. A new computational efficient method for enhanced flip-flop selection is proposed in the thesis. The method combines SCOAP (Sandia Controllability/Observability Analysis Program) testability measures and input constraint gate influence analysis. The idea is that if some state inputs are difficult to control to “0” or “1” and when those state inputs are constrained to “1” or “0”, large numbers of gates from those state inputs would be constrained, which means large numbers of stuck at faults cannot be detected, then the flip-flops corresponding to these state inputs should be made into enhanced scan flip-flops. Experimental results on ISCAS 89 benchmark circuits show that 10-20% enhanced scan flip-flops can achieve more than 80% of the benefits of full enhanced scan design for transition delay faults.

In the thesis, the best combination values of “a” and “b” used in the formula for enhanced flip-flop selection is based on fault simulation. When circuits under test become larger, several fault simulations might be required to get the best values of “a” and “b”, and thus will result in increased enhanced flip-flop selection time. Future work includes looking for an efficient method to chose “a” and “b” values without using fault simulation. And more experiments based on larger benchmark circuits like ITC 99 and ITC 02 are needed to further testify the effectiveness of the enhanced flip-flop selection method.

REFERENCES

[1] Laung-Terng Wang, Cheng-Wen Wu, Xiaoqing Wen, “VLSI TEST PRINCIPLES AND ARCHITECTURES”, Morgan Kaufmann, 2006.

[2] Angela Krstic, Kwang-Ting Cheng, “Delay Fault Testing for VLSI Circuits”, KluwerAcademic, 1997.

[3] Barnhart, C., V. Brunkhorst, F. Distler, O. Farnsworth, B. Keller, and B. Koenemann, “OPMISR: the Foundation for Compressed ATPG Vectors,” Proc. of International Test Conference, pp. 748-757, 2001.

[4] Wohl, P., L. Huisman, “Analysis and Design of Optimal Combinational Compactors,” Proc. of VLSI Test Symposium, pp. 101-106, 2003.

[5] Touba, N.A., “X-Canceling MISR – An X-Tolerant Methodology for Compacting Output Responses with Unknowns Using a MISR,” Proc. of International Test Conference, Paper 6.2, 2007.

[6] S. Patil and J. Savir, "Skewed-Load Transition Test: Part II, Coverage", in Proc. International Test Conference, 1992, p. 714.

[7] J. Savir, "Skewed-Load Transition Test: Part I, Calculus", in Proc. International Test Conference, 1992, p.705.

[8] J. Savir and S. Patil, "On broad-side delay test", Very Large Scale Integration (VLSI) Systems, vol. 2, 1994, pp.368.

[9] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J. Campbell and J. Berech, "Scan-based transition fault testing - implementation and low cost test challenges", in Proc. International Test Conference, 2002, pp. 1120-1129.

[10] G. Xu and A. D. Singh, "Flip-Flop Selection to Maximize TDF Coverage with Partial Enhanced Scan", in Proc. Asian Test Symposium, 2007.

[11] M. L. Bushnell and V. D. Agrawal, "Essentials of Electronic Testing for Digital, Memory and Mixed-Signal VLSI Circuits", Springer, 2000.

[12] J. P. Hurst and N. Kanopoulos, "Flip-flop sharing in standard scan path to enhance delay fault testing of sequential circuits", in Proc. Asian Test Symposium, 1995, pp. 346-352.

[13] S. Bhunia, H. Mahmoodi, A. Raychowdhury and K. Roy, "First level hold: a novel low-overhead delay fault testing technique", in Proc. International Symposium on Defect and Fault Tolerance in VLSI Systems, 2004, pp. 314- 315.

[14] S. Bhunia, H. Mahmoodi, A. Raychowdhury and K. Roy, "A Novel Low-overhead Delay Testing Technique for Arbitrary Two-Pattern Test Application", in Proc. Design, Automation and Test in Europe, 2005, pp. 1136-1141.

[15] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy and I.Pomeranz, "Methods For Improving Transition Delay Fault Coverage Using Broadside Tests", in Proc. International Test Conference, 2005, pp. 256-265.

[16] A. D. Singh, “Scan Based Testing of Dual/Multi Core Processors for Small Delay Defects” in Proc. International Test Conference, 2008.

[17] A. D. Singh, "A self-timed structural test methodology for timing anomalies due to

defects and process variations", in Proc. International Test Conference, 2005.

[18] H. Yan and A. D. Singh, "A New Delay Test Based on Delay Defect Detection Within Slack Intervals (DDSI)" IEEE Transactions on Very Large Scale Integration Systems, vol. 14, 2006, pp. 1216-1226.

[19] H. Yan, A. D. Singh; "Experiments at Detecting Delay Faults using Multiple Higher Frequency Clocks and Results from Neighboring Die", Proceedings of the International Test Conference, 2003.

[20] C. Barnhart, "Delay Testing for Nanometer Chips , Chip Design, August/September 2004, pp 8-14.

[21] Pomeranz, I., S. Kundu, and S.M. Reddy, "On Output Response Compression in the Presense of Unknown Output Values," Proc. of Design Automation Conference, pp. 255-258, 2002.

[22] Rajski, J., J. Tyszer, G. Mrugalski, W.-T. Cheng, N. Mukherjee, and M. Kassab, "X-Press Compactor for 1000x Reduction of Test Data," Proc. of International Test Conference, Paper 18.1, 2006.

[23] Wohl, P., J.A. Waicukauski, and T.W. Williams, "Design of Compactors for Signature-Analyzers in Built-In Self-Test," Proc. of International Test Conference, pp. 5463, 2001.

[24] Wohl, P., J.A. Waicukauski, S. Patel, and M.B. Amin, "X-Tolerant Compression and Application of Scan-ATPG Patterns in a BIST Architecture," Proc. of International Test Conference, pp. 727-736, 2003.

[25] Wohl, P., L. Huisman, "Analysis and Design of Optimal Combinational Compactors,"

Proc. of VLSI Test Symposium, pp. 101-106, 2003.

[26] Mitra, S., and K.S. Kim, “X-Compact: An Efficient Response Compaction Scheme,” IEEE Trans. on Computer-Aided Design, Vol. 23, No. 3, pp. 421-432, Mar. 2004.

[27] Mitra, S., S.S. Lumetta, and M. Mitzenmacher, “X-Tolerant Signature Analysis,” Proc. of International Test Conference, pp. 432-441, 2004.

[28] Rajski, J., J. Tyszer, C. Wang, and S.M. Reddy, “Finite Memory Test Response Compactors for Embedded Test Applications,” IEEE Trans. on Computer-Aided Design, Vol. 24, No. 4, pp. 622-634, Apr. 2005.

[29] Sharma M. and W.-T. Cheng, “X-Filter: Filtering Unknowns from Compacted Test Responses,” Proc. of International Test Conference, Paper 42.1, 2005.

[30] R. Garg, R. Putman, and N. A. Touba, “Increasing Output Compaction in Presence of Unknowns using an X-Canceling MISR with Deterministic Observation” Proceedings 26th IEEE VLSI Test Symposium, May 2008.

[31] Wohl, P., J.A. Waicukauski, and S. Ramnath, “Fully X-Tolerant Combinational Scan Compression,” Proc. of International Test Conference, Paper 6.1, 2007.

[32] Lawrence H. Goldstein , Evelyn L. Thigpen, “SCOAP: Sandia Controllability/Observability Analysis Program,” Proc. of the 17th Design Automation Conference, pp. 190 – 196, 1980.