DELAY TEST SCAN FLIP-FLOP (DTSFF) DESIGN AND ITS APPLICATIONS FOR

SCAN BASED DELAY TESTING

Except where reference is made to the work of others, the work described in this dissertation is my own or was done in collaboration with my advisory committee. This dissertation does not include proprietary or classified information.

Gefu Xu

Certificate of Approval:

Vishwani D. Agrawal James J. Danaher Professor Electrical & Computer Engineering	Adit D. Singh, Chair James B. Davis Professor Electrical & Computer Engineering
Charles E. Stroud Professor Electrical & Computer Engineering	Victor P. Nelson Professor Electrical & Computer Engineering

George T. Flowers
Interim Dean
Graduate School

DELAY TEST SCAN FLIP-FLOP (DTSFF) DESIGN AND ITS APPLICATIONS FOR SCAN BASED DELAY TESTING

Gefu Xu

A Dissertation

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Doctor of Philosophy

Auburn, Alabama
December 17, 2007

DELAY TEST SCAN FLIP-FLOP (DTSFF) DESIGN AND ITS APPLICATIONS FOR SCAN BASED DELAY TESTING

Gefu Xu

Signature of Author

Date of Graduation

VITA

Gefu Xu, son of Qimin Xu and Pinjie Liu, was born on November 12, 1978 in Wuxi, P. R. China. He entered Southeast University in 1996 and graduating with a Bachelor of Engineering degree in 2000 and a Master of Science degree in 2003. In 2003, he joined Ph.D. program at the Department of Electrical and Computer Engineering, Auburn University.

DISSERTATION ABSTRACT

DELAY TEST SCAN FLIP-FLOP (DTSFF) DESIGN AND ITS APPLICATIONS FOR SCAN BASED DELAY TESTING

Gefu Xu

Doctor of Philosophy, December 17, 2007
(M.S., Southeast University, 2003)
(B.S., Southeast University, 2000)

125 Typed Pages

Directed by Adit D. Singh

Scan based delay testing is currently mostly implemented using launch-on-capture (LOC) delay tests. Launch-on-shift (LOS) tests are generally more effective, achieving higher fault coverage with significantly fewer test vectors, but require a fast scan enable signal, which is not supported by most designs. A low cost solution is presented for implementing LOS tests by adding a small amount of logic (six transistors) in each flip-flop to align the slow scan enable signal to the clock edge. This new scan cell design which is called Delay Test Scan Flip-flop (DTSFF) can support full LOS and LOC testing, achieving an average TDF (Transition Delay Fault) coverage of 95.78% in this combined mode for the ISCAS89 benchmarks. Mixed LOC/LOS tests can be further applied to increase coverage for ISCAS89 benchmarks. In addition, a partial DTSFF scheme, which replaces only 20-40% carefully chosen scan flip-flops in the scan chain with the new DTSFF can achieve most of the coverage benefits of a full DTSFF design while minimizing area overhead. Our partial scan scheme for modified scan flip-flops can also be applied to enhanced scan designs that support high coverage TDF testing but with significant overhead. A flip-flop selection strategy presented for partial enhanced scan designs shows a very favorable trade-off between coverage and overhead. Experimental results using commercial ATPG tools show that 60-90% of the TDF coverage benefits of enhanced scan can be achieved using only 10-30% enhanced flip-flops.

The architectural restrictions of scan further greatly limit the effectiveness of traditional scan based delay tests. It has been recently shown that additional testing for delays on short paths using fast clocks can significantly lower DPM (Defect Per Million). However, accurately obtaining the needed timing information for such tests from simulation is extremely difficult. The simulations must not only accurately account for the effects of process parameter variations, but also power supply noise and crosstalk from the excessive switching activity of scan tests. We present a methodology for learning signal timing information on silicon to "calibrate" such tests which can be much more accurate and cost effective. Such an approach requires that the outputs of the applied tests be hazard-free to avoid learning incorrect timing due to a glitch at the output. Simulation results presented here indicate that such output hazard-free tests can be obtained with an average coverage only about 10 % below the transition delay fault coverage for both launch-on-shift and launch-on-capture modes.

ACKNOWLEDGMENTS

I would like to express my appreciation and sincere thanks to my advisor, Dr. Adit D. Singh, who guided and encouraged me throughout my studies. His advice and research attitude have provided me with a model for my entire future career. I also wish to thank my advisory committee members, Dr. Vishwani D. Agrawal, Dr. Charles E. Stroud and Dr. Victor P. Nelson, for their guidance and advices on this work.

Appreciation is also expressed to Dr. Haihua Yan, who is the Ph.D. student of Dr. Adit D. Singh and graduated in 2005, for his support and help during my Ph. D. study in Auburn University.

At the same time, I would like to thank, although this is too weak a word, my mother Pinjie Liu, my father Qimin Xu, my wife Ning Ge and all my family members for their continual encouragement and support throughout this work.

Finally, I would like to dedicate this dissertation to the memory of my grandfather Yunhu Liu. His love for me and thirst for knowledge continue to inspire me today.

This work is supported in part by National Science Foundation Grant 0325426.

Style manual or journal used: IEEE Journal on Solid State Circuits

Computer software used: Microsoft Word 2003

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1 INTRODUCTION

1.1 Scan Based Delay Testing

With the increase in gate count and operating frequency of integrated circuits (ICs) in nanometer technologies, manufacturing defects that cause timing errors have become a serious concern. Conventional stuck-at tests alone cannot fully detect defects inside a chip because the stuck-at fault model itself is not able to evaluate the circuit timing. Therefore, it becomes more and more important to employ delay tests on fabricated chips before they are shipped to meet required low DPM (Defect Per Million) level.

Typically there are two kinds of delay testing. One is functional delay testing and the other is structural delay testing [1]. For functional delay testing, tests are developed according to the functionality of the circuit. However, for products with short TTM (Time to Market) requirement and for products whose scales are in millions of gates, those functional tests suffer from unacceptable test development costs. In addition, in SOC (System-on-Chip) designs, limited test access to internal cores makes application of at-speed functional tests impractical. As a result, structural delay testing, typically scan-based delay testing, which can significantly improve the controllability and observability of internal signals in SOCs, appears to be the most promising approach for the delay testing of large SOCs.

Scan based structural delay testing involves applications of two test patterns <V1,V2> via the scan chain. The first vector V1, which is used to initialize the internal logic values of the CUT (Circuit under Test), is first scanned into the scan chain, typically using a slow scan clock. A 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 corresponding to an appropriate launch to capture window reflecting the desired operational frequency. Finally, the response captured in the scan chain is scanned out of the CUT and compared with the expected correct test response.

1.2 Transition Delay Testing and Path Delay Testing

Scan based structural delay testing can be classified into lumped transition delay testing and path delay testing in terms of the delay fault model. There are many fault models, such as Gate delay model, Transistor delay model, Segment delay model and In-line resistive delay model. In the lumped transition (gate) delay fault model, a delay defect is assumed to make the fault site charge or discharge more slowly than normal. These are defined as slow-to-rise and slow-to-fall transition delay faults. In the path delay fault model, the delay defect in the circuit is assumed to cause the cumulative delay of a combinational path to exceed some specified duration, which normally is one nominal clock period [2].

A non-robust path delay test for a path is a test which guarantees detection of a path-delay fault on the path when no other path delay fault is present in the circuit. A robust path delay test guarantees to produce an incorrect state at the observation points if the delay size exceeds the nominal clock period, irrespective of the delay distribution in the circuit [2].

For transition delay fault model, one advantage is that the total fault number is measurable and is twice total faulty sites. Besides, tests are easy to generate and a stuck-at fault test generator can be easily modified to generate transition fault tests.

Compared with transition delay fault testing, path delay fault testing can in theory detect more delay faults, because in the transition delay testing the delay on the faulty gate may be compensated for by the delay on other faster gates in the path which is used to propagate transition. However, the number of possible paths in the circuit grows up exponentially with the increase of the number of gates. Hence, it is impractical to detect all path delays in a circuit, especially for large SOCs.

1.3 Scan Elements for Scan Based Delay Testing

There are two kinds of commonly used scan elements for scan based structural delay testing. One is MUX (multiplexer) based scan flip-flop (SFF); the other is Level Sensitive Scan Design (LSSD) [3].

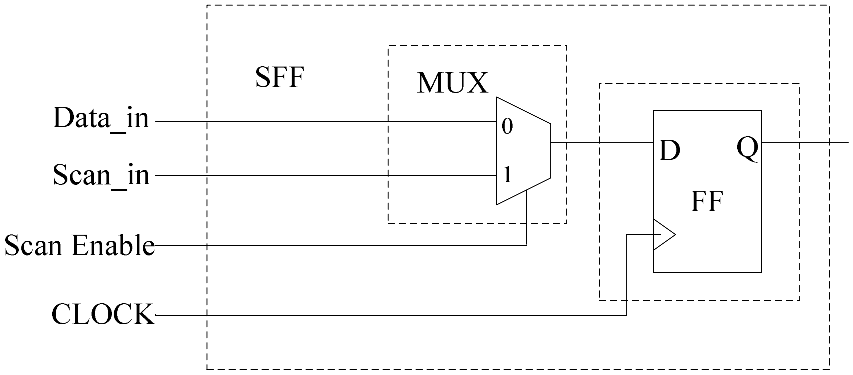

Figure 1-1: Multiplexer based scan flip-flop

To build a scan flip-flop, a multiplexer (MUX) is added on the data path of a regular D flip-flop, as shown in Figure 1-1. When Scan Enable signal is set to logic ‘0’, DFF accepts data from the input “Data_in”. When Scan Enable signal is set to logic ‘1’, DFF accepts data from the input “Scan_in”. For scan based structural delay testing, a number of SFFs are serialized into a scan chain, as shown in Figure 1-2. When Scan_enable (in Figure 1-2) is set to ‘1’, each SFF captures and stores data from the primary input (“Scan_in”) or the output (“Q”) of its preceding SFF. Then, data can be scanned into or out of SFFs through the scan chain. Therefore, when Scan_enable is set to ‘1’, SFFs operate under “shift mode” or “scan mode”. When Scan_enable is set to ‘0’, each SFF captures and stores data from outputs of Combinational Logic. After stimuli employed on the Combinational Logic, the responses of Combinational Logic can be captured in the scan chain. Therefore, when Scan_enable is set to ‘0’, SFFs operate under “function mode” or “capture mode”.

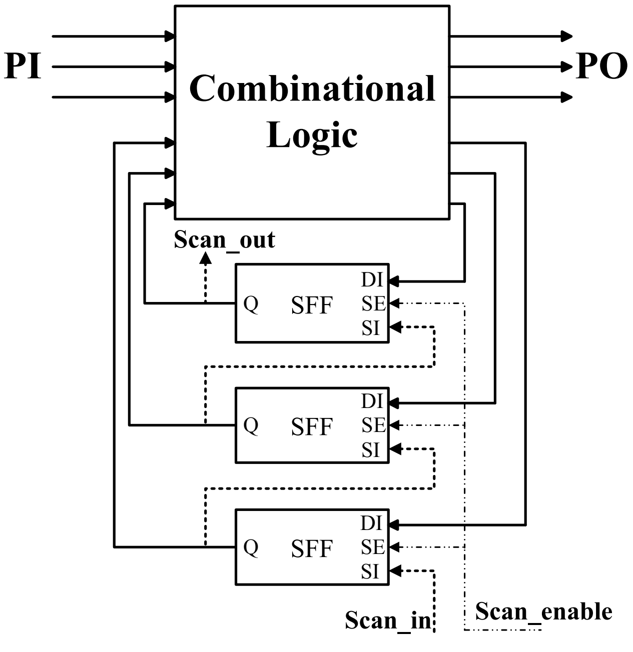

Figure 1-2: Scan chain design using SFF

Level Sensitive Scan Design (LSSD) developed by IBM [2] is another type of frequently used scan element for scan based structural delay testing. As shown in Figure 1-3, in the normal operation mode, when the test clock (TCK) is set to '0', the master clock (MCK) and the slave clock (SCK) are set to '1' sequentially to latch the data from the input "D" into Master latch and Slave latch. In the scan mode, when MCK is set to '0', TCK and SCK are set to '1' sequentially to latch the data from the input "SD" into Master latch and Slave latch.

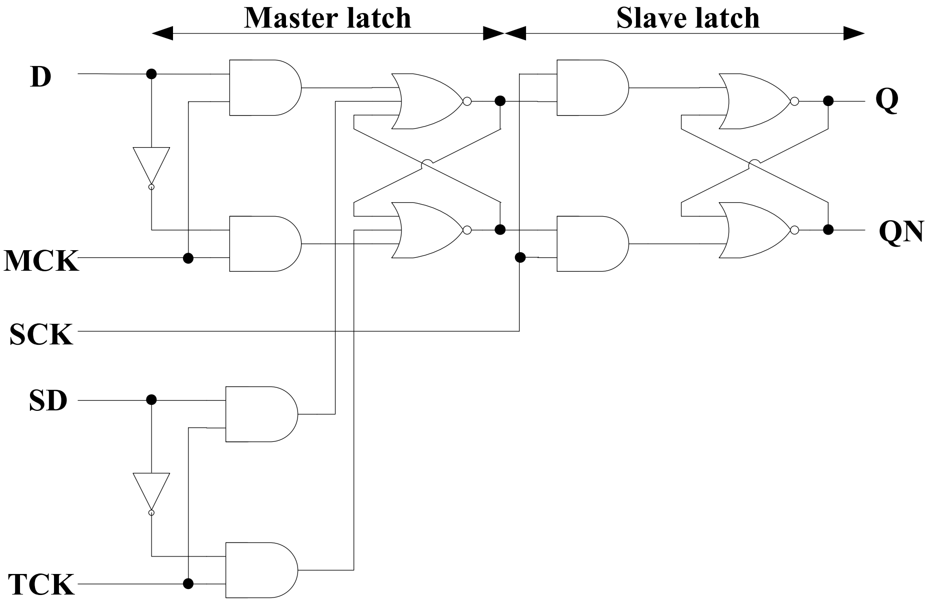

Figure 1-3: Level Sensitive Scan Design (LSSD) [2]

Compared with SFF and LSSD, SFF demonstrates the advantage of lower DFT overhead since LSSD needs two non-overlapping clock signals plus a test clock. However, SFF introduces gate delay on the data path since an extra MUX is added on the functional path.

1.4 LOS and LOC Scan Based Delay Testing Using Scan Flip-flops

Figure 1-4 shows a conceptual overview of scan based delay testing using two-vector test patterns <V1, V2>. The first vector V1 is scanned into the flip-flops and used to initialize the logic values at the input of the combinational logic block, which is the circuit under test (CUT). A second vector V2 is then used to launch transitions at theses inputs and propagate these transitions to the outputs of the CUT, which are then captured back in the scan chains.

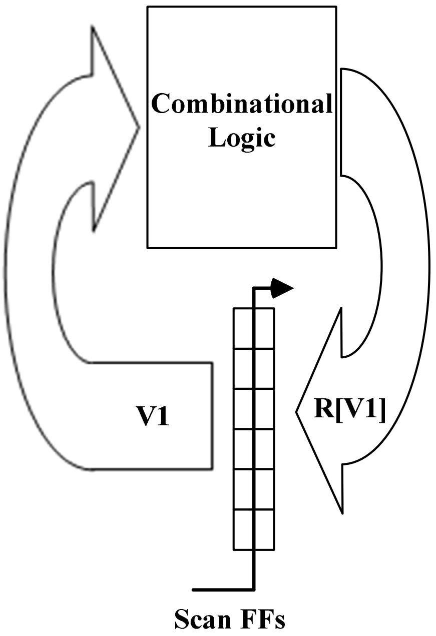

Figure 1-4 : Overview of scan based delay testing

According to how the second vector V2 is obtained, a transition delay test can be classified as a skewed-load delay test[4-6] or a broad-side delay test[7, 8]. For a skewed-load delay test, which is now more commonly called a launch-on-shift (LOS) delay test, the second vector V2 is one bit shift of the first vector V1. For the broad-side delay test, also called as launch-on-capture (LOC) delay test, the second vector V2 is the CUT's response to V1 ($R[V1]$ in Figure 1-4) captured in the scan chain.

The schematic waveforms in Figure 1-5 illustrate the timing associated with of executing LOS and LOC delay tests. Notice from Figure 1-1 that the scan enable signal is low (0) for the functional mode and high (1) in the scan shift mode. Therefore, the scan enable must be held high for the duration when the first test vector V1 is scanned into the scan chain. This is typically done using a slow scan clock. The waveforms in Figure 1-5 assume positive edge triggered flip-flops, and show the last scan clock pulse, which makes the V1 vector available at the CUT inputs following the positive clock edge. For the LOC test, the scan enable is then made low and enough time is allowed to elapse to allow the change in this slow global signal to take effect throughout the chip before two timed high speed clock pulses are applied to launch V2 and capture the CUT's response to this input change. Because scan enable is low (functional mode) at the first high speed launch clock edge, the V2 vector captured in the flip-flops and applied to the CUT is the circuit's response to V1, corresponding to a launch-on-capture (LOC) test. The time between the two fast clock edges must match the operational clock rate to ensure that the delay test checks that the CUT outputs reach the correct logic values within the functional clock period. These captured test results are again scanned out at a slow scan rate.

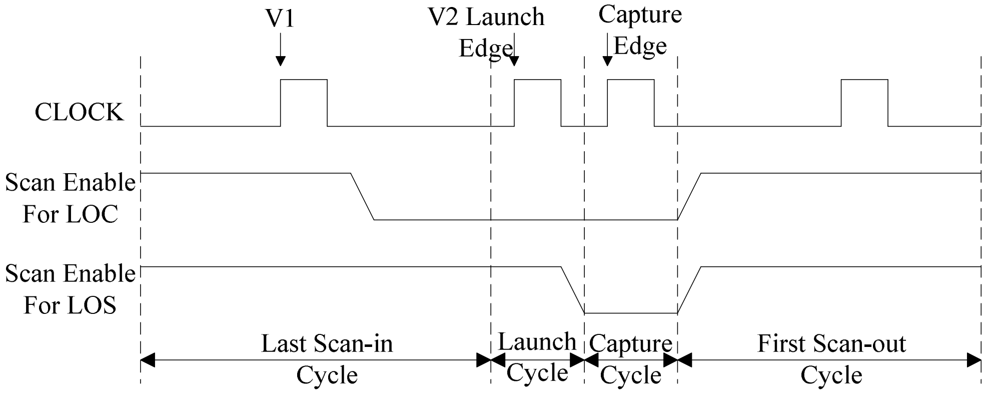

Figure 1-5: Waveforms for LOC and LOS delay test

In the LOS delay test, the second delay test vector V2 is obtained by shifting one bit from V1. In this case, the scan enable signal must remain high (in the scan shift mode) for one more active clock edge after V1 is shifted in, until V2 is launched at the flip-flop outputs on the positive edge of the first fast clock. Scan enable must then be quickly switched low (to the functional mode) so that the CUT's response to V2 can be captured back in the flip-flop. This is also illustrated in Figure 1-5. Because the scan enable must switch within the timed fast clock interval in this case, it is important for this global signal to reach all the flip-flops in the design within tight timing constraints. This requires that the scan enable signal for LOS testing be routed as a timing critical signal, just like a clock signal. However, this is very expensive, and not supported in most scan designs. Therefore, scan based delay tests today mostly employ the LOC mode.

1.5 Problems with LOS and LOC Delay Testing Using Scan Flip-flops

Generally speaking, compared with LOC delay tests, LOS delay tests display better TDF (Transition Delay Fault) [9] coverage [7, 10, 11] and typically reach this coverage with fewer test patterns [8, 10, 12]. Besides, the complexity of LOS ATPG (Automatic Test Pattern Generation) is lower than LOC ATPG because the former is a combinational ATPG and the latter is a sequential one. However, in practice only LOC can actually be applied to most circuits because LOS requires a high speed global scan enable signal. In large SOCs, it is difficult to design scan enable signals with sufficient drivability to drive all scan flip-flops of the circuit within the timing constraints. Scan enable signals must also synchronize at the input port of each scan flip-flop with minimal timing skew. This requires routing the scan enable signals as additional clock signals, which is expensive to implement and is not currently supported in most scan-based designs. Consequently, there is considerable interest in developing low cost designs to support LOS scan based delay tests. Such a capability can potentially also allow combining LOS and LOC tests for even higher TDF coverage.

1.6 Contribution of This Dissertation

A novel Delay Test Scan Flip-flop (DTSFF) is presented in this dissertation. Being different from classic scan flip-flop which can only support launch-on-capture (LOC) delay test, with a low scan enable signal, DTSFF supports both launch-on-shift (LOS) delay test and launch-on-capture (LOC) delay test. Delay test fault coverage of the DTSFF based scan design is dramatically improved from LOC test fault coverage up to nearly perfect transition delay fault coverage.

A partial DTSFF design and a partial enhanced scan design are further introduced to reduce the DFT (Design for Test) cost compared to the full DTSFF design (and also for enhanced scan designs). Scan units in a scan chain, which will be replaced with DTSFFs or enhanced scan flip-flops for partial scan design, can be screened by using controllability analysis method. And this scan unit selection can be further refined by using an Interchange Procedure. Experimental results show that, only 20-40% carefully chosen scan flip-flops in the scan chain with the new DTSFF can achieve most of the coverage benefits of a full DTSFF design; only 10-30% carefully chosen scan flip-flops in the scan chain with the enhanced scan flip-flop can achieve most of the coverage benefits of a full enhanced scan design.

For high quality delay testing, delays must be tested along worst case paths so that timing faults do not remain hidden in circuit timing slack. For practical test application, “real” timing information in a circuit is very difficult to be obtained due to variations in circuit process parameters and test conditions, such as power supply droop, different chip temperatures etc. Therefore, to accurately obtain the needed timing information for tests from simulation alone is extremely difficult. In this dissertation, a method to learn signal timing information on silicon to "calibrate" tests is presented to accurately and cost-effectively profile the “real” timing information in a circuit for tests. With this novel silicon calibrated test method and the high TDF coverage offered by the DTSFF, effective scan based delay testing can become practical for complex ICs and SOCs.

CHAPTER 2 PRIOR WORK ON SCAN BASED DELAY TESTING FOR IMPROVING FAULT COVERAGE AND REDUCING DFT OVERHEAD

2.1 Approaches Supporting LOS Delay Testing

Two broad approaches have been proposed to reduce the costs associated with distributing the high speed scan enable signal to support LOS tests. The first approach employs some form of pipelining [13, 14] and is now being supported by commercial tools. The scan enable signal is first distributed, in a pipelined manner, to scan control cells which are evenly distributed over the chip. These scan control cells are then only required to broadcast the fast scan enable in local regions of the chip within the single clock timing constraint, as shown in Figure 2-1.

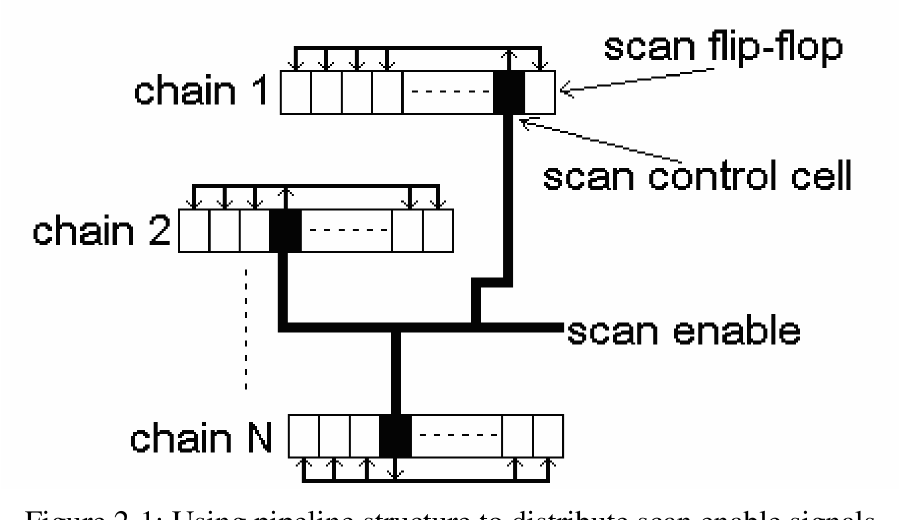

Figure 2-1: Using pipeline structure to distribute scan enable signals

Note that, although this approach reduces the drive requirements on the scan control cells, it does not altogether eliminate the problem. A fast scan enable signal for local regions is still needed, and if 20-100 flip-flops are driven, the drivability requirements for this local fast scan enable remain considerable. Furthermore, these timing critical signals can greatly complicate layout and timing closure. For regular ASIC design flow, the layout of the functional logic is made before scan flip-flops are serialized into a scan chain, since the layout of the functional logic is set to a higher priority during layout stage to meet the timing closure. Then the layout of the scan chain is made by using unused channels and tracks in the chip. When pipeline approach is adopted, the distribution of scan flip-flops needs to be considered at the time of the layout of the functional logic. These restrictions of the layout of scan flip-flops complicate layout and timing closure.

In some cases, where scan enable control paths interact with clock skew, flip-flop hold time violation or flip-flop setup time violation may also be a problem. Figure 2-2 demonstrates a pipeline structure. Scan Control Cells are distributed in a pipeline manner and generates locate fast scan enable signals to control flip-flops in local regions. CK1, CK2, CK3 and CK4 are different leaves of a clock tree.

When global scan enable signal slowly switches from high to low before the first rising edge of CK, as shown in Figure 2-2, the local fast scan enable signal (SEa), which controls SFF1 and SFF2, switches from high to low as well. Because the drivability of “Scan Control Cell a” is low, SEa switches just after the first rising edges of CK1. However, SFF1 is clocked by CK2, which is later than CK1 due to the clock skew. Therefore the shift operation does not occur in SFF1 on the first rising edge of CK2 because the local fast scan enable signal (SEa) is already low when the first rising edge of CK2 triggers the shift operation. Similarly, SFF2, which is clocked by CK1, might also miss the shift operation due to the delay on the clock signal (CK1) path, as shown in Figure 2-2.

When global scan enable signal slowly switches from high to low before the first rising edge of CK, as shown in Figure 2-2, the local fast scan enable signal (SEb), which controls SFF3 and SFF4, switches from high to low as well. Because the drivability of "Scan Control Cell b" is high, SEb switches just before the second rising edges of CK3. However, SFF3 is clocked by CK4, which is earlier than CK3 due to the clock skew. Therefore the capture operation does not occur in SFF3 on the second rising edge of CK4 because the local scan enable signal (SEb) remains high when the second rising edge of CK4 triggers the capture operation. Similarly, SFF4, which is clocked by CK3, might also miss the capture operation due to the delay on the scan enable signal (SEb) path.

Figure 2-2 demonstrates fours extreme cases, where scan flip-flops clearly miss the shift operations or capture operations. However, in reality, the clock skew might not be as obvious as demonstrated in Figure 2-2. Therefore, local scan enable signal might switch just inside the hold time window or set up time window of the scan flip-flops and cause violations.

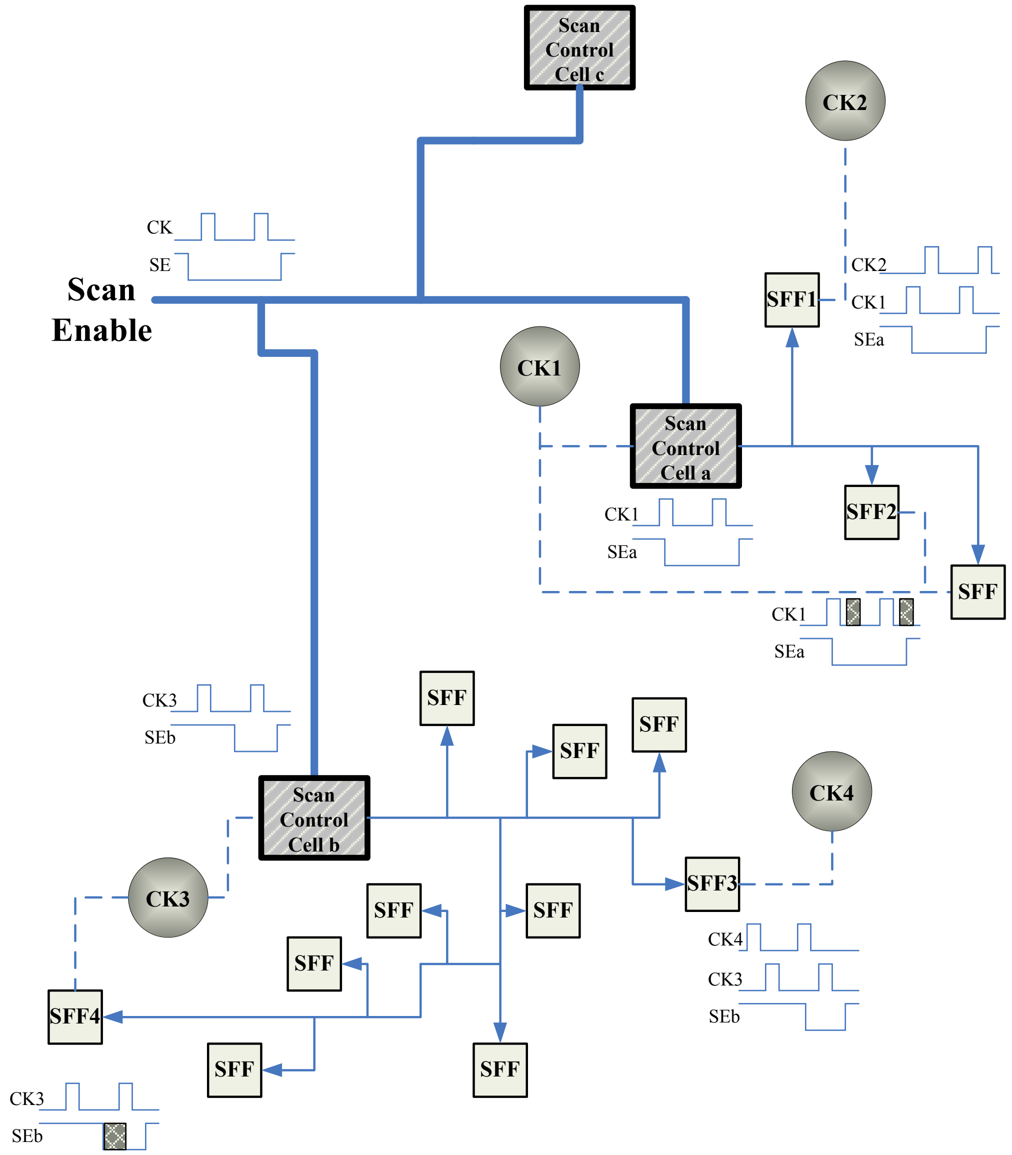

Figure 2-2: Clock Skew issues of LOS testing using pipeline structure

A second approach avoids using a fast scan enable signal altogether by not requiring the scan flip-flops operating in the LOS mode to switch between the scan shift mode (at launch) to functional mode (so as to capture the test response). The flip-flop remains in the scan shift mode after launching the delay test, so no test response is captured in that flip-flop. Clearly such a strategy cannot be applied to all the scan flip-flops because the test response would not be captured at all then. Therefore, this technique only allows LOS for a subset of the flip-flops. However, in such a mixed mode test with some flip-flops operating in the LOC mode and some in this modified LOS mode, richer two-pattern scan tests can be applied than normal LOC tests, while the observability of the test response is not excessively compromised if the majority of the flip-flops operate in the LOC mode and can capture the test response. It has been shown in [15-18] that that such an approach can improve TDF coverage beyond that achievable from LOC tests alone, although because of the loss of observability, the coverage remains well below that achievable from traditional LOS tests. Note that the overhead of these designs includes an additional (slow speed) global control signal.

The key idea in Hybrid Delay Scan [10] is to use a fast scan enable to only drive a small fraction of the scan flip-flops, which are selected to maximize controllability measures for internal circuit nodes. The rest of the flip-flops still operate in the LOC mode. This limits the cost of distributing the fast scan enable, but again at the expense of significantly reducing coverage when compared to full LOS testing.

LOS TDF coverage can be further improved by detecting multiple-cycle activation detectable faults [11, 12]. However, commonly occurring hazards can significantly degrade this improvement in practice. In addition, according to the assumptions of the multiple-cycle activation detectable fault model, the fault size of extra detectable faults must be larger than two or more clock periods, which indicates the resolution of the fault size in this method is looser than the resolution of the fault size in the traditional TDF delay test. Recall that in a classic TDF delay test, only those detectable faults whose size is smaller than one clock period may escape the test.

Notice that, other than multiple-cycle activation method detecting extra delay faults whose delay size are larger than one clock period, there are other delay testing methods focusing on detecting fine delay defects (the size of the delay fault to be detected is smaller than one nominal clock period) such as DDSI (Delay Detecting in the Slack Interval) test method [19-25].

2.2 Approaches Supporting Enhanced Scan Delay Test

Enhanced scan delay test methods were introduced to remove the restrictions on the V2 vector and allow arbitrary <V1, V2> combinations for high coverage delay testing. After adding an extra latch and an extra clock signal, a modified LSSD design [26] can be used for enhanced scan delay testing. However, to implement two non-overlapping clock signals and two test clocks for this modified LSSD design is very expensive.

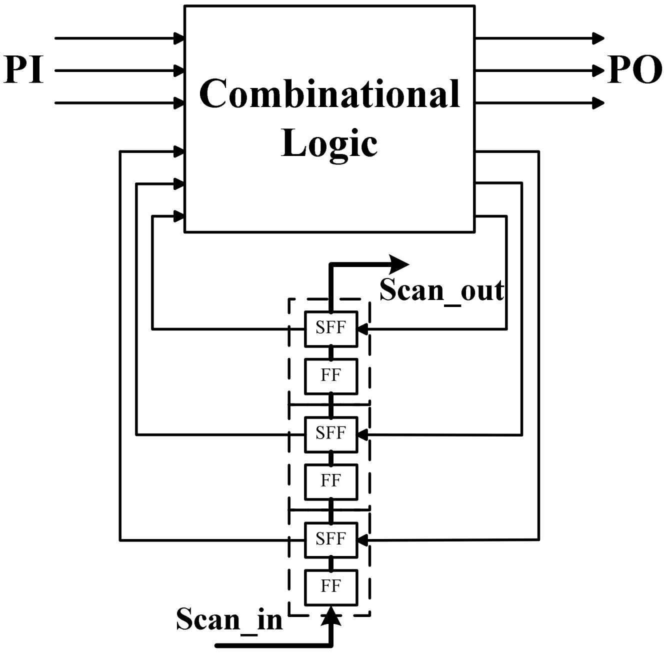

Figure 2-3: Classical enhanced scan with alternating regular and scan FFs

In another simple enhanced scan scheme, one additional redundant flip-flop is interleaved with each of the functional flip-flops in the design, which doubles the length of the scan chain, as shown in Figure 2-3. 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 are 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 forms V2 and can now be chosen arbitrarily without any constraints.

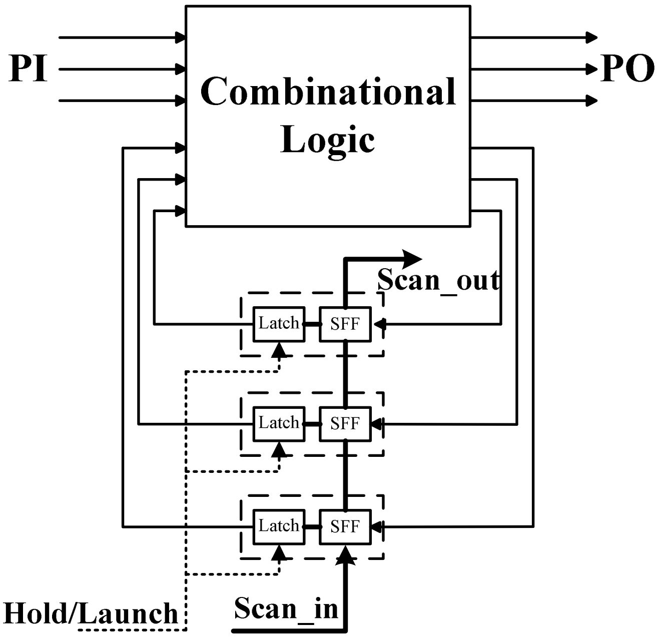

Figure 2-4: Enhanced scan with hold latches

Since the cost of duplicating all flip-flops in the design can be very high, an alternate enhanced scan approach uses an extra “hold” latch (with an additional control line) at the output of each scan flip-flop as shown in Figure 2-4. The idea here is to hold the V1 initialization pattern in these latches while an arbitrary V2 pattern is being shifted into the scan chain [2]. 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 combinational logic inputs from V1 to V2.

An obvious disadvantage of this alternate enhanced scan design is the extra delays introduced on the signal paths. This problem is addressed in a different enhanced scan design presented in [27]. 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, which is 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 of applying arbitrary two-pattern tests [28, 29].

Unfortunately, in all of the enhanced scan designs discussed so far, control signals capable of switching at operational clock speeds are needed to ensure proper test timing. For example, it is well understood that the scan enable signal must be capable of at-speed switching to support the LOS tests needed by the design in Figure 2-3. Similarly the individual hold control signals at each latch in Figure 2-4 must also switch in a timed and synchronized manner throughout the IC to launch the V1 to V2 transition at the same instant at all the inputs. This is not practical using a slow global broadcast signal which can be subject to substantial timing skews at different locations on the chip. Implementing high speed control signals is very expensive, loosely comparable in cost to an extra clock signal. Such signals must be avoided in any low cost design which attempts cost savings from a partial enhanced scan methodology.

The need for a high speed scan enable is alleviated in the dual flip-flop enhanced scan design in [16], where the enhanced scan flip-flop comprises two cascaded standard scan flip-flops as shown in Figure 2-5. Here, while Scan Enable 1 is in the shift mode (high) to shift and launch V2 at the launch clock edge, Scan Enable 2 is set to the functional mode (low) to capture the response in FF2. This can be achieved using slow speed scan control signals, without any need for high speed switching within the launch to capture window. Note, however, that this approach needs an extra (slow) global scan enable control signal.

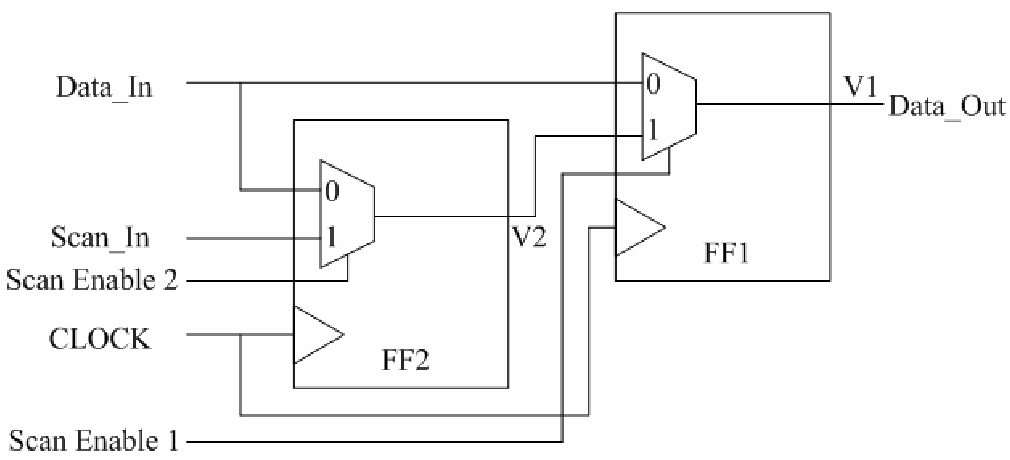

Figure 2-5: Dual Enhanced Scan Flip-flop with slow Scan Enable [16]

CHAPTER 3 DELAY TEST SCAN FLIP-FLOP

3.1 The Structure of Delay Test Scan Flip-Flop

The basic structure of Delay Test Scan flip-flop (DTSFF), first presented in [30], is shown in Figure 3-1. Observe that the DTSFF has identical pinouts when compared with the traditional scan flip-flop, and is therefore fully compatible with industry standard design tools. Notice in Figure 3-1 that an extra clock alignment logic block is introduced in the circuitry of a conventional scan cell to realize the DTSFF. In the schematic, this is shown as three logic gates for ease of presentation. This simple clock alignment logic translates the incoming slow scan enable signal into a properly timed signal that makes the MUX transition in a timely manner during the launch cycle to support LOS tests. Note that the functional paths through the multiplexer are completely unaffected by this logic, therefore the DTSFF has no performance impact.

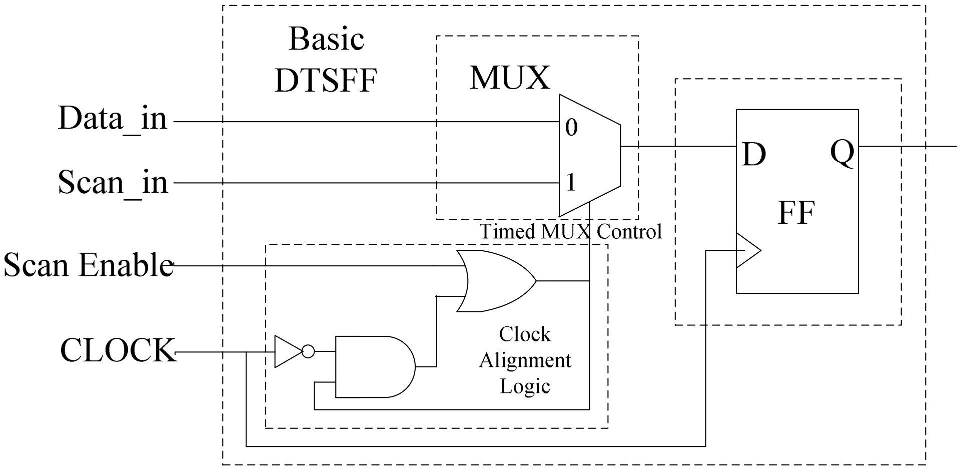

Figure 3-1: The basic Delay Test Scan Flip-flop (DTSFF)

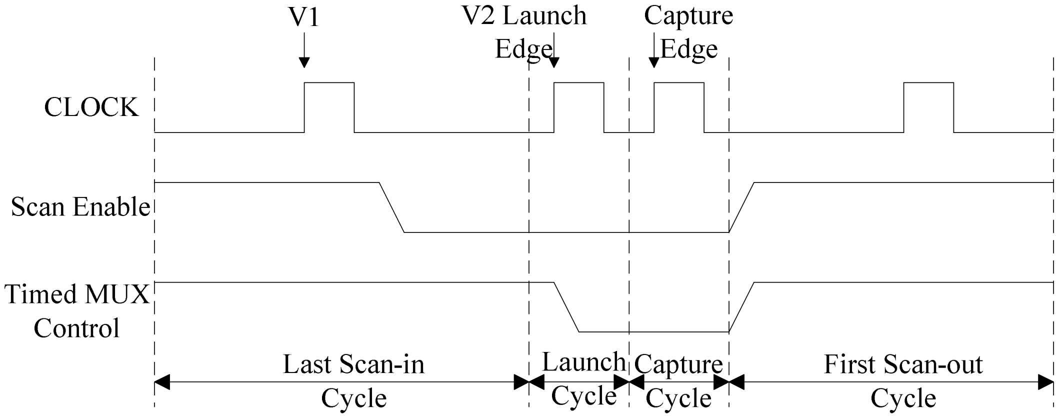

Figure 3-2: Timing diagram for LOS delay testing

Notice in Figure 3-1 that a high scan enable signal directly forces the (timed) control input of the MUX to be high through the OR gate, consistent with the scan shift mode. This is shown in the timing diagram in Figure 3-2. During the first part of the scan-in cycle, both the scan enable and the timed MUX control signals are high. Notice, however, that when the Scan Enable is switched low (to logic ‘0’), the timed MUX control signal does not respond immediately; instead it remains latched high while the clock is low. From Figure 3-1 it can be seen that this is because feedback from this initial high value, along with the high inverted clock signal, generates a high (logic ‘1’) at the AND gate output, which propagates through the OR gate, keeping the OR output latched high. This high timed MUX control signal only goes low after the clock goes high, forcing the AND gate output low. Thus, while the scan enable control signal switches asynchronously from high to low at the end of the last scan shift cycle, which loads V1 into the scan chain, the actual timed control signal sent to the MUX by the clock alignment block switches synchronously after the next active clock edge. This results in an additional shift operation in the scan chain, which launches V2, and then the MUX switches to activate the data input to capture the test response, precisely as required for a LOS test.

With proper timing of the scan enable signal, the basic DTSFF of Figure 3-1 can also be made to operate in the LOC mode without any modification [31]. For the LOC test, the trick is to switch the Scan Enable signal from the shift mode (high) to the functional mode (low) before the last scan shift. This is illustrated in Figure 3-3. Notice in the figure that Scan Enable goes low before the arrival of the final clock edge needed to shift in the V1 scan pattern. But the clock alignment block in the DTSFF delays activating the timed multiplexer control until after the clock edge so that this last desired shift of V1 does occur. However, following this shift operation, the multiplexer is now in the functional mode. Therefore, two fast clocks, which are the launch and capture cycles in the functional mode, realize the LOC test.

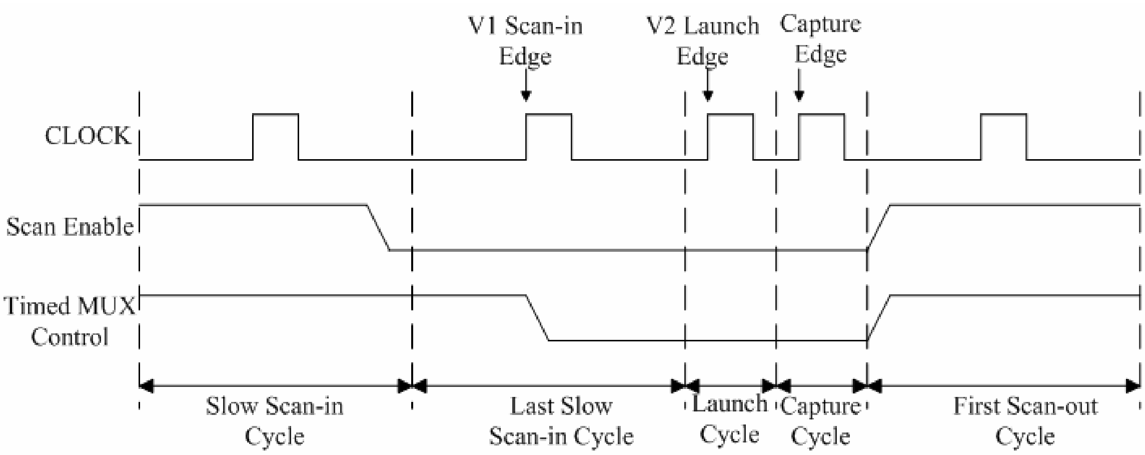

Figure 3-3: Timing diagram for LOC delay testing

To summarize, switching the Scan Enable signal low (to the functional mode) after the last scan shift clock edge achieves an LOS test as illustrated in Figure 3-2; switching the Scan Enable one scan cycle earlier achieves an LOC test (Figure 3-3). Notice again that there are no new timing constraints placed on the Scan Enable since in both cases it can switch any time during the low slow scan clock window; as in any traditional LOC test, this scan enable transition must be allowed enough time to settle before the next clock edge.

The basic DTSFF introduced above targets on positive edge-triggered clock signals. After minor modifications on the basic DTSFF, a negative edge-triggered DTSFF can be obtained from the basic version by removing the inverter gate out of the Clock Alignment Logic, as illustrated in Figure 3-4. Similar timing diagrams can be applied on this Negative Edge-triggered DTSFF for both LOS and LOC delay testing, as is demonstrated in Figure 3-5 and Figure 3-6.

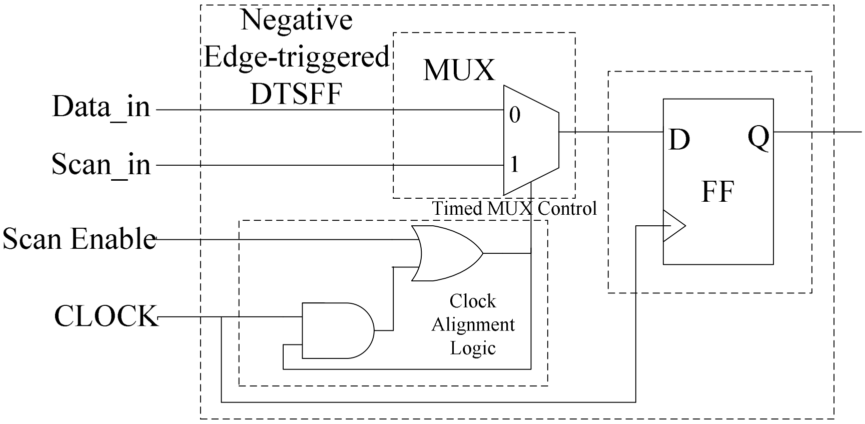

Figure 3-4 : Negative Edge-triggered Delay Test Scan Flip-flop

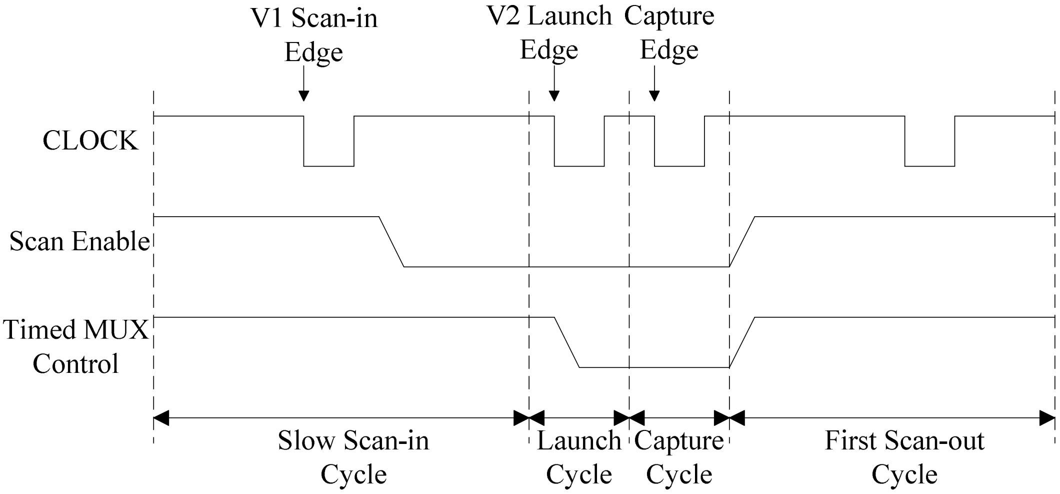

Figure 3-5 : Timing diagram for LOS delay testing (Negative Edge-triggered DTSFF)

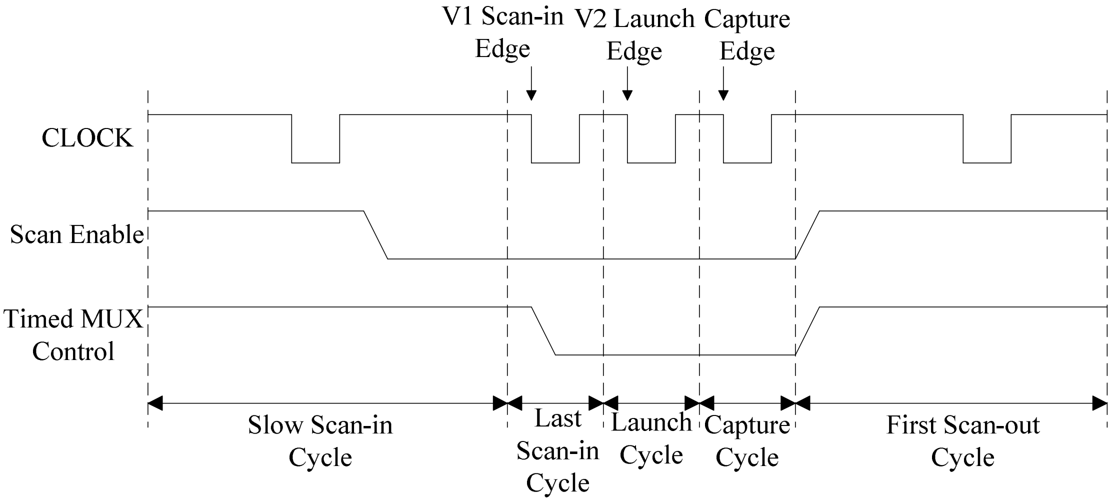

Figure 3-6 : Timing diagram for LOC delay testing (Negative Edge-triggered DTSFF)

3.2 Implementing Delay Test Scan Flip-Flop with Minimum Overhead

There are many different ways to map the Clock Alignment Logic into primitive logic gate combinations such as NAND-gate combination, NOR-gate combination and etc.

According to the Boolean function of the Clock Alignment Logic, which is $Timed\_Mux\_Ctl_n = Timed\_Mux\_Ctl_{n-1} \bullet \overline{CLK} + Scan\_Enable$ (Equation 3-1), it can be transformed into other functional equivalent Boolean equations such as:

$$Timed\_Mux\_Ctl_n = \overline{\overline{Timed\_Mux\_Ctl_{n-1} \bullet \overline{CLK}} \bullet \overline{Scan\_Enable}} \text{ (Equation 3-2)}$$

$$\overline{Timed\_Mux\_Ctl_n} = \overline{\overline{\overline{Timed\_Mux\_Ctl_{n-1}} + CLK} + Scan\_Enable} \text{(Equation 3-3)}$$

$$\overline{Timed\_Mux\_Ctl_n} = (\overline{Timed\_Mux\_Ctl_{n-1}} + CLK) \bullet \overline{Scan\_Enable} \text{(Equation 3-4)}$$

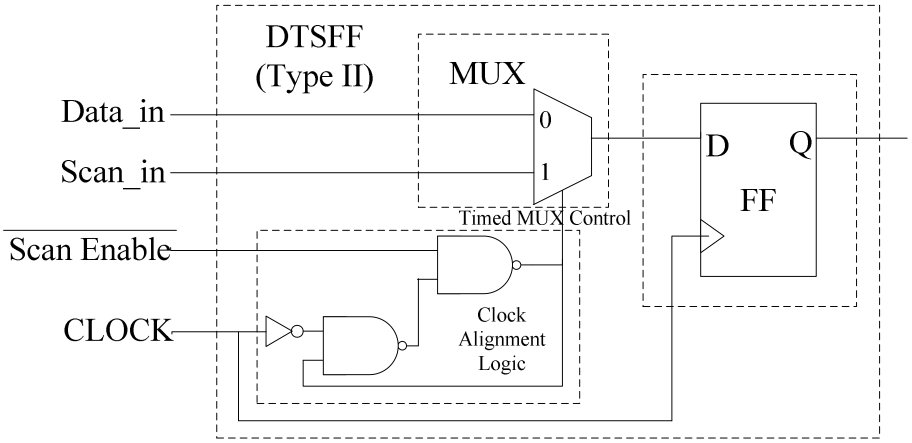

Figure 3-7 : DTSFF Type II

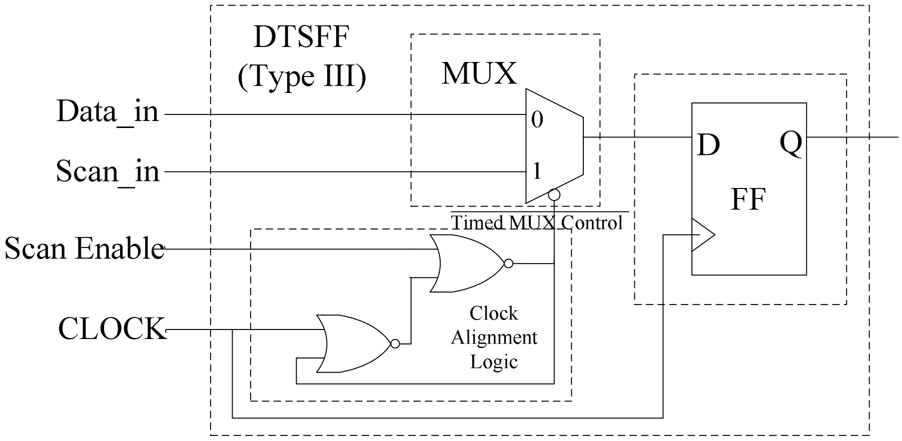

Figure 3-8 : DTSFF Type III

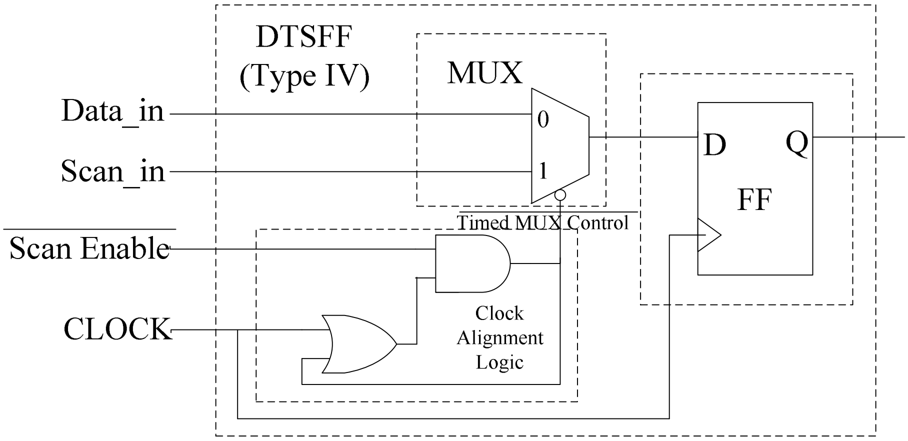

Figure 3-9 : DTSFF Type IV

Those functional equivalent Boolean equations can be further mapped into DTSFF (Type II, III and IV) designs, which are shown in Figure 3-7, Figure 3-8 and Figure 3-9 individually. Since every flip-flop in the design is going to be replaced with the DTSFF, it is important that the area overhead of the new design be small. Figure 3-10 shows one possible implementation of the basic DTSFF, where the Clock Alignment Logic block is implemented (with output inverted) by a single AND-OR-INVERT (AOI) gate. This implementation takes advantage of the fact that the complemented clock signal is readily available in a flip-flop cell. Also, the inverted output of the AOI(1,2) gate is not a problem because this signal serves as the Timed Control for the multiplexer. Any implementation of a multiplexer requires complementary control inputs; therefore an inverter to complement this control signal is already part of the multiplexer.

The DTSFF can thus be realized using only a single additional AOI(1,2) gate, which can be readily implemented using only six transistors as shown in Figure 3-11. This is only about half the transistor count of the multiplexer, and approximately 10%-20% of the total transistor count of a typical scan cell.

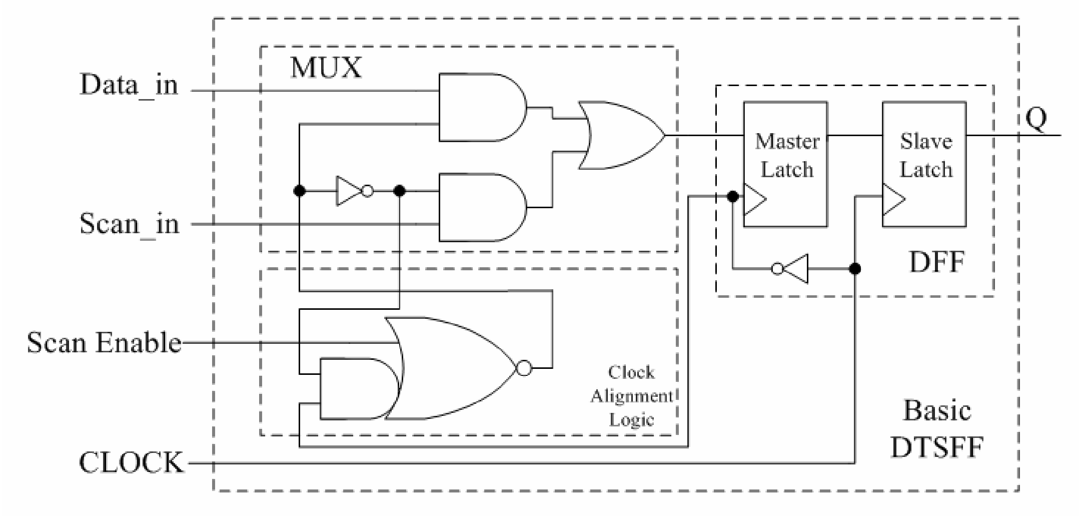

Figure 3-10: DTSFF implemented using an AOI(1,2)

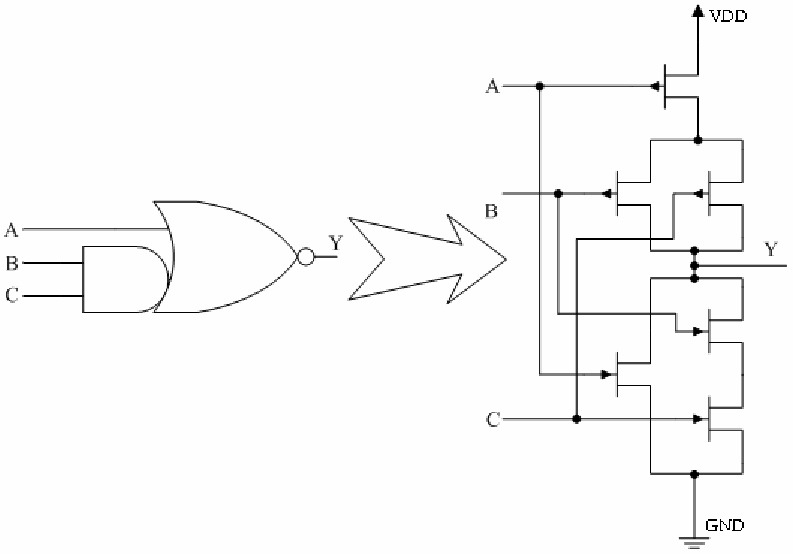

Figure 3-11: Six transistor AOI(1,2) gate

In order to compare the extra area overhead between DTSFF and SFF, as shown in Figure 3-12, a layout of a DTSFF, which contains one SFF and one AOI21 gate, is demonstrated in the above figure; a layout of a standard SFF is demonstrated in the bottom figure. All standard cells are chosen from the AMI (0.5) standard library. Notice that there are no layout optimizations applied on DTSFF unit to reduce area overhead in

Figure 3-12. As a matter of fact, given the fact that in a multistage standard cell, transistors driving local internal signals in the input stage can be made much smaller than those in the output stages, which drive much larger inter-cell interconnect capacitances, the increase in the scan cell area due to the extra six small transistors can be kept very small with careful cell design. Furthermore, all internal timing-related issues can be robustly addressed while designing this DTS flip-flop standard cell. Thus DTSFF based designs can be implemented with very minimal area overhead. Furthermore, as has already been pointed out, the pinout and I/O signals of DTSFF are identical to those of the standard scan cell, offering a transparent capability of integration into standard design flows.

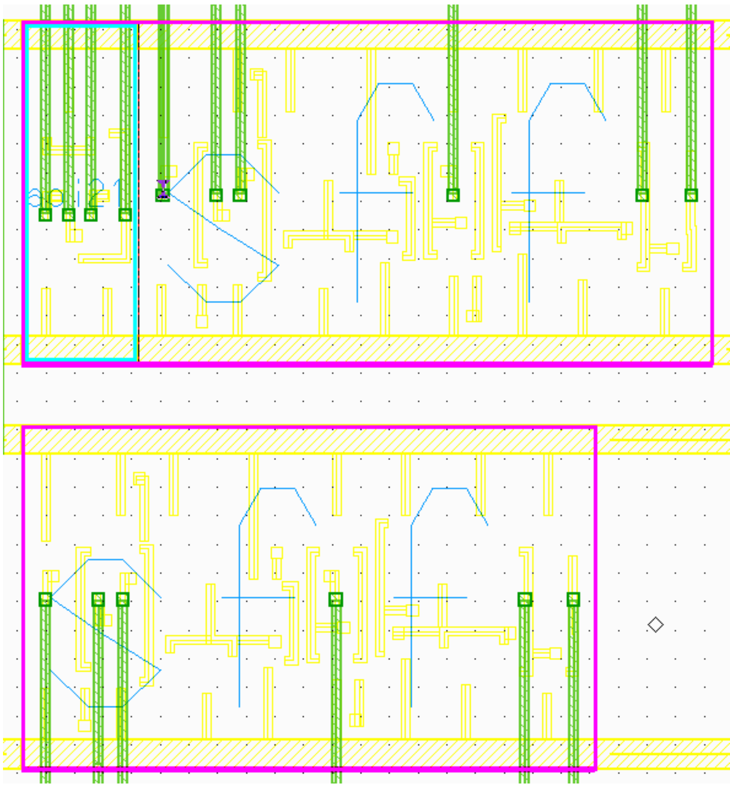

Figure 3-12 : The layout of DTSFF (AOI21 plus SFF) versus the layout of standard SFF

3.3 Internal Timing Analysis of DTSFF

Because the clock alignment logic is an asynchronous latch structure, it is important to analyze it for race conditions. Fortunately this is a very simple circuit. It is obvious that the tester must synchronize the input to this asynchronous cell with respect to the system timing. When Scan Enable signal drops from high to low after launching V1 and before launching V2, as illustrated by T1 in Figure 3-14, the tester must allow enough time for the slow Scan Enable signal to reach all the flip-flops across the chip before activating the next (launch) clock edge. This timing constraint is easy to realize on the tester since this is not the extra requirement for DTSFF delay tests, but is also the basic requirement for stuck-at scan based tests. When Scan enable signal is raised from low to high during T2, the timed MUX control follows after the OR gate delay Δt_2.

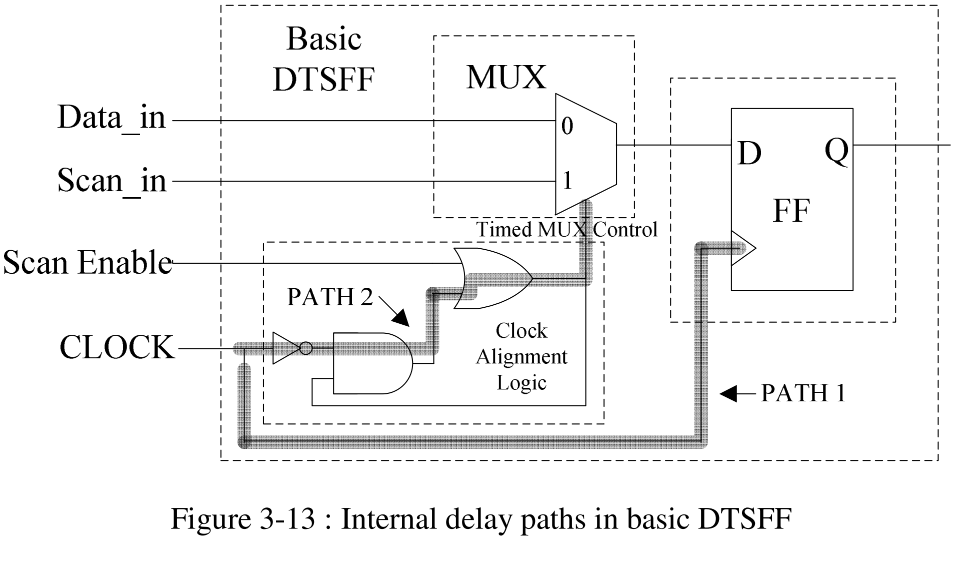

Figure 3-13 : Internal delay paths in basic DTSFF

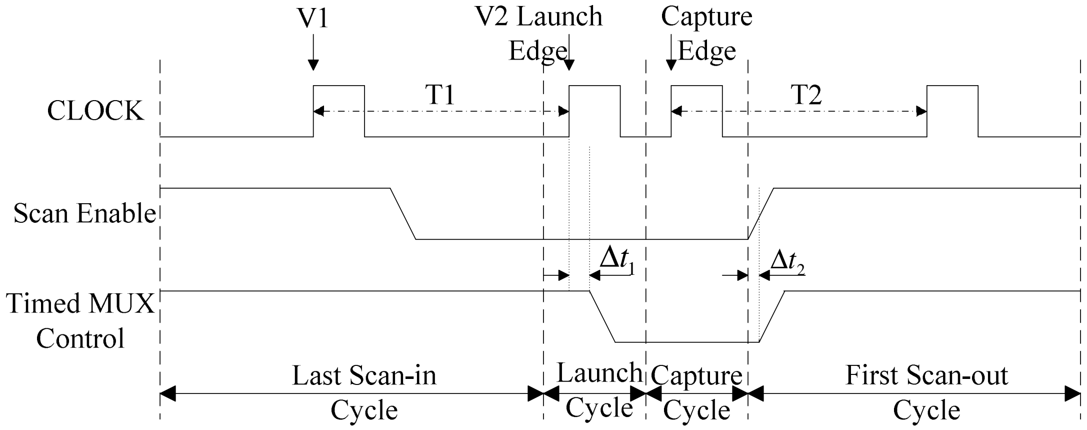

Figure 3-14 : Internal timing in basic DTSFF

The other timing concern is the possibility of the Timed MUX Control signal switching from high (scan) to low (functional) mode before the shifted pattern is latched into the flip-flop. This can happen if the change in the Timed MUX Control occurs before the clock edge arrives at the flip-flop, i.e. if the delay labeled Δt_1 in Figure 3-14 is negative. Observe however that in the Figure 3-13, the path delay on the Path 1, which is defined from the input signal CLOCK to the internal clock signal of the D flip-flop, only consists of the wire interconnection delay on this path. While the path delay on the Path 2, which is defined from the input signal CLOCK to the internal signal Timed MUX Control includes not only the wire interconnection delays but also three gate delays along this path. Since the DTSFF is designed as a standard logic unit, all components and wire interconnections will be placed and routed in a small continuous region. In this small region, the wire interconnection delays can be expected to be quite small since the interconnections have been restricted to a small area. Therefore, it is reasonable to expect that the Timed MUX Control signal will arrive with a delay of Δt_1 , which is approximately equal to three gate delays, later than the rising clock edge (V2 Launch

Edge) on which last shift occurs (Figure 3-14). Of course, layouts can vary with different implementation possibilities. It is the responsibility of the cell designer to ensure and verify timing so that the last shift is correctly achieved by the cell before the Timed MUX Control signal switches, as what cell designer does for a standard flip-flop which is also an asynchronous latch structure. (Note that since both Δt_1 and Δt_2 are small internal delays, for simplicity we do not display them in the other timing diagrams, except in Figure 3-14 used to explicitly discuss internal timing issues in this subsection.)

Thus implementing LOS tests using DTSFFs does not impose any new timing requirements on the scan enable signal whatsoever. All internal timing related issues can be robustly addressed while designing the DTS flip-flop cell. Externally, it appears identical to a standard scan cell to the design tools. The additional logic of the clock alignment block can be optimized as already discussed, and made quite small relative to the complexity of the standard scan cell, which typically contains the D flip-flop, implemented as two master-slave latches, and several additional inverters, buffers and drivers for clock and signal lines, not explicitly shown in Figure 1-1. Thus the area overhead of the proposed design is modest, particularly compared to flip-flop duplication based enhanced scan schemes, which additionally also require a second global control signal.

CHAPTER 4 FAULT COVERAGE IMPROVEMENTS TO DELAY TEST SCAN FLIP-FLOP

4.1 Employing Scan Chain Partition with Two or More Control Signals

For single LOS and LOC delay testing, all the flip-flops in the scan chain operate in one mode: either LOC or LOS. Intuition suggests that perhaps even higher coverage may be obtained if the two modes could be mixed, i.e. for a given pattern, some flip-flops made to operate in LOS and the rest in the LOC mode.

One solution is to partition the flip-flops into a small number of groups, each controlled by a separate Scan Enable signal common to that group. While too many such partitions can become prohibitive in the cost of additional global Scan Enable signals, the simplest possibility is to have just two partitions, controlled by two different global Scan Enable lines, SE1 and SE2. Note that each flip-flop is connected to only one of the two Scan Enable lines.

Table 4-1 shows the four possible combinations for the two signals and the characteristics of the corresponding delay test. If the Scan Enable lines for both partitions are switched early (before the last scan shift), a LOC test is realized; if they are both late (after the last scan shift), a LOS test is realized. For the other two combinations of SE1 and SE2, half of the flip-flops operate in the LOC mode and the other half in LOS, potentially yielding additional TDF coverage. Increasing the number of partitions can improve TDF coverage even further, at the expense of a larger number of global Scan Enable signals.

Table 4-1: Operation modes for partition size 2

SE1	SE2	Operation Mode
Early	Early	LOC
Early	Late	Partition 1 LOC + Partition 2 LOS
Late	Early	Partition 1 LOS + Partition 2 LOC
Late	Late	LOS

Another solution for mixed LOS and LOC test is to use modified DTSFF. Figure 4-1 shows a design where each DTS flip-flop has a Scan Test Select (STS) input to allow either LOC or LOS mode to be individually selected for that flip-flop. When Scan Test Select is held high ('1'), the output from the clock alignment logic is gated to the MUX input, and the DTS flip-flop operates in the LOS mode. A low ('0') on the Scan Test Select allows the high-to-low transition on the Scan Enable signal to bypass the clock alignment logic and arrive earlier at the MUX control input, thus resulting in LOC operation in the flip-flop. The timing diagrams of mixed LOS and LOC delay test using this kind of modified DTSFF is illustrated in Figure 4-2.

In practice, individual control of the Scan Test Select inputs at each flip-flop can be prohibitively expensive in interconnect costs. However, it may be more practical to partition the flip-flops into a small number of groups, each controlled by a common Scan Test Select signal.

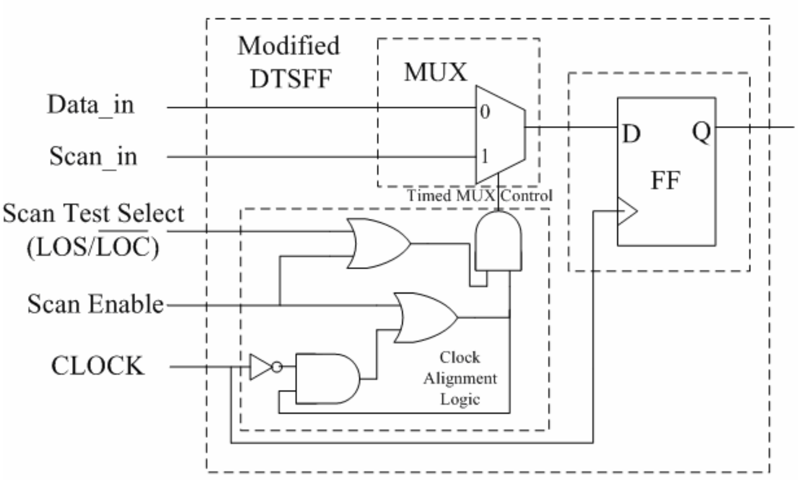

Figure 4-1: Modified DTSFF for mixed LOS and LOC delay test

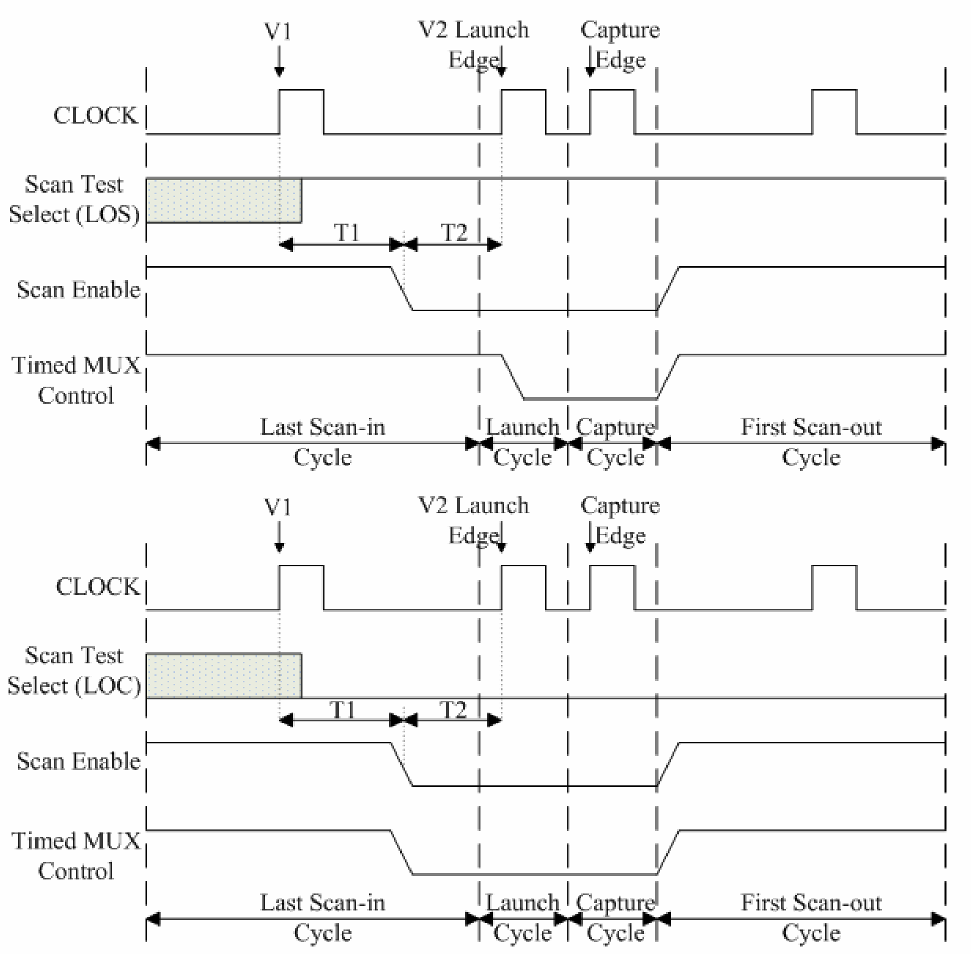

Figure 4-2: Timing diagrams of mixed LOS (top) and LOC (bottom) delay test using modified DTSFF

The simplest possibility is to have two partitions, controlled by two different global Scan Test Select lines, STS1 and STS2. Table 4-2 shows the four possible combinations for the two STS signals and the characteristics of the corresponding delay test. If the Scan Test Select lines for both partitions are low, a LOC test is realized; if they are both high, a LOS test is realized. For the other two combinations of STS1 and STS2 (01 and 10), half the flip- flops operate in the LOC mode and the other half in LOS, yielding additional TDF coverage. Increasing the number of partitions can potentially improve TDF coverage even further, at the expense of a larger number of global Scan Test Select signals.

Table 4-2: Operation modes for partition size 2

STS1	STS2	Operation Mode
0	0	LOC
0	1	Partition 1 LOC + Partition 2 LOS
1	0	Partition 1 LOS + Partition 2 LOC
1	1	LOS

4.2 Employing Scan Chain Partition without Extra Control Signals

An alternative to global Scan Test Select signals in the above partitioned approach is to make the Scan Test Select signals local to small regions of the circuit, and distribute them from local Control Cells that can be made part of the scan chain. This idea is similar in concept to the pipelined scan presented in [13], with the important difference that the

Scan Test Select signals in our design are slow signals, and do not need to meet the timing constraints of the fast scan enable signals distributed in [13].

Figure 4-3 shows the Scan Test Select connections between the local Control Cells and the DTS flip-flops for the case where the flip-flops are divided into two partitions. Notice that each DTS flip-flop in the scan chain is randomly connected to either Scan Test Select signal STS1 or STS2. The Control Cells are composed of three normal non-scan flip-flops connected in series and inserted as a 3-bit shift register into the scan chain.

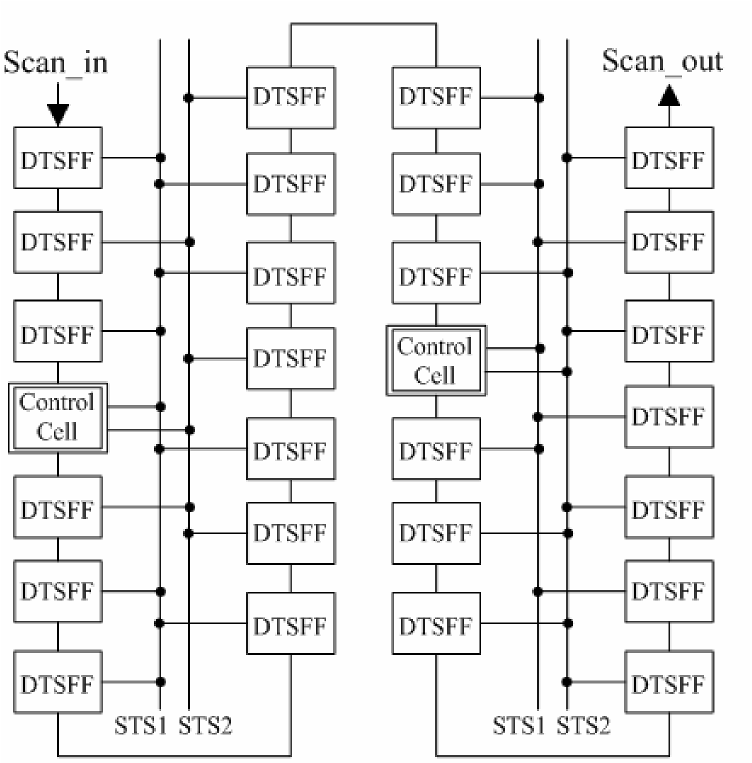

Figure 4-3: System view of the implementation mixed LOS and LOC delay test

As shown in Figure 4-4, the output of DFF1 provides the STS1 signal, while the output of DFF2 provides the STS2 signal. These signals can be set as desired by scanning in appropriate values into the Control Cells, to achieve the four operation modes shown in Table 4-2, while loading V1 into scan chain. Notice that the Control Cell contains an extra dummy flip-flop DFF3. This is to remove any dependency between the test mode controlled by DFF2 and the LOS V2 pattern at the input connected to the scan output of the Control Cell.

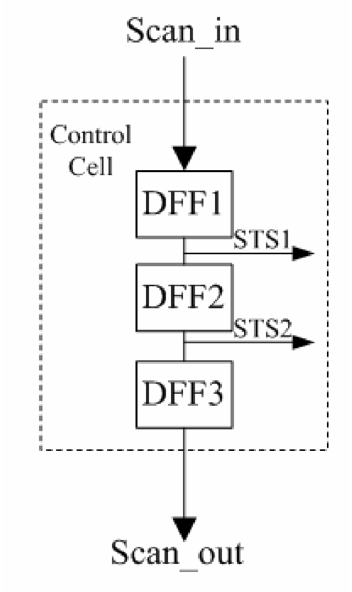

Figure 4-4: The structure of Control Cell

Each DTS flip-flop in Figure 4-2 is able to behave in either LOS or LOC mode based on the value of the STS signal. However, all STS signals throughout the chip must be stable before the Scan Enable signal goes low to switch the DTS flip-flops from the scan to capture mode. The required timing sequence is illustrated in Figure 4-2 for the two cases when Scan Test Select signal is required to be logic ‘1’ (LOS) and logic ‘0’ (LOC) during the capture cycle. (The following assumes that the STS signals are locally distributed by a Control Cell, although the same considerations hold if the signals are externally driven global signals.)

The positive clock edge of the last scan cycle that latches V1 at the scan outputs, also loads the appropriate STS signal values in the Control Cells. These signals may take some time to stabilize at all the driven flip-flop inputs because of the high fan out loading on the STS lines. However, sufficient slack (T1 in Figure 4-2) can be easily introduced in the test timing to allow all internal STS signals to be stable before the (slow) Scan Enable signal is switched low. Similarly, slack T2 is introduced to allow the slow Scan Enable to propagate to all the flip-flops before the launch clock edge. It is important to note that because T1 and T2 can be made arbitrarily long, the STS and Scan Enable signals do not need to meet any timing constraints, and can be implemented as inexpensive slow signals.

4.3 Reporting Fault Coverage and Discussion

Following the methodology in [7], we developed a simulation based ATPG program to generate transition delay fault (TDF) test patterns to evaluate the effectiveness of the applying different tests on a DTSFF based design. We generated 100,000 pseudo-random V1 vectors to create transition test pattern pairs and applied them to the ISCAS89 benchmarks in both the LOC and LOS test modes. The TDF coverage (no fault collapsing) obtained is presented in the second and third columns of Table 4-3. Except for a couple of the largest circuits tested in LOC mode, the TDF coverage saturated well before all 100,000 random patterns were applied. The TDF coverage for a combined LOC and LOS test employing 200,000 pseudo-random patterns (100,000 for LOS and 100,000 for LOC) is listed in column 4. The simulation results in column 4 of Table 4-3 (LOS+LOC) indicate that, on average, a combined 92.67% TDF coverage can be achieved for the ISCAS89 benchmark circuits. Note that comparable simulation results corresponding to the first three columns in Table 4-3 have been available in the literature for over a decade[7] (and were also presented in [30]), although the precise coverage values can show significant variations because of their strong dependence on the flip-flop ordering in the scan chains, and also on the random pattern generator used in the simulation. Also notice that the results in Table 4-3 and the results in [7] are from two different experiments in which different scan-chain orderings and pseudo-random vectors were adopted. Therefore, the fault coverage for the same circuit might be different in these two experiments.

Table 4-3: Simulation results on ISCAS89 benchmark circuits

Circuit	LOC (%) (100k)	LOS (%) (100k)	LOS+LOC (%) (200k)	MIX (%) (200k)
S208	57.45	88.94	92.79	94.71
S298	81.21	84.23	94.97	98.49
S344	93.75	94.04	97.67	100
S349	93.12	93.41	96.99	99.28
S382	76.83	90.71	93.06	96.73
S386	52.72	79.40	88.08	93.52
S400	75.63	89.50	91.87	95.38
S420	64.76	87.74	92.62	95.24
S444	75.11	86.60	92.23	94.93
S510	89.41	90.39	96.47	98.82
S526	64.35	87.45	93.35	98.19

Circuit	LOC (%) (100k)	LOS (%) (100k)	LOS+LOC (%) (200k)	MIX (%) (200k)
S526n	64.35	87.64	93.54	98.29
S641	91.60	96.70	97.17	98.12
S713	85.13	90.81	91.23	92.08
S820	51.83	78.17	84.63	88.05
S832	51.08	77.04	83.41	86.84
S953	91.55	91.03	96.22	97.22
S1196	81.65	85.54	85.83	85.87
S1238	79.08	81.99	82.31	82.35
S1423	87.10	95.99	98.24	98.7
S1488	87.40	79.67	96.2	98.82
S1494	86.98	79.08	95.68	98.23
S5378	89.61	93.05	96.78	97.46
S9234	74.71	88.28	89.83	89.71
S13207	82.38	94.04	96.2	96.77
S15850	78.82	90.66	92.05	92.56
Average	77.22	87.77	92.67	94.86

We also studied this mixed LOS and LOC test mode by randomly assigning each flip-flop into one of two partitions, as illustrated in column 5 of Table 4-3 (MIX). For each of the four combinations of SE1 and SE2, we applied 50,000 pseudorandom patterns for a total of 200,000 patterns. Since the TDF coverage achieved by this mixed LOS and LOC test can vary, depending on which flip-flops get randomly assigned to which partition, we ran 10 simulations for each circuit using different random partitions and recorded the best coverage achieved in the last column of Table 4-3. These results are clearly sub-optimal since there are a very large number of ways of partitioning the flip-flops, of which only 10 were explored for each circuit. Nevertheless, the results in the last column of Table 4-3 show an average TDF coverage of almost 95%. Observe that while on average the coverage improvement is just over 2% compared to LOS+LOC, some circuits show up to a 5% improvement. It is our expectation that better partitioning of the circuit flip-flops using algorithms that exploit controllability and observability heuristics will allow somewhat higher TDF coverage to be achieved by a two partition mixed mode test. Recall that such a DFT capability requires a separate Scan Enable signal for each partition, i.e. one extra global control signal when compared to traditional scan design. While in theory even higher coverage is possible by creating a larger number of partitions, given the potentially near-complete coverage obtained from two partitions, the benefits of further partitioning may not be sufficient to justify the costs of additional global control signals.

Our simulation results confirm the well known result that LOS tests provide better coverage than LOC. However, due to the nature of the pseudo-random simulation, the real fault coverage of the LOC test might be underestimated in our experiments. The discrepancy of the fault coverage between the LOS and LOC tests might not be substantially high, as is shown in [11] where an academic ATPG has been implemented. Commercial ATPG results on ISCAS benchmark circuit are also reported in Table 4-4, where DFT Compiler tool and TetraMax ATPG tool from Synopsys Corporation have been used to synthesize the circuits and generate test patterns separately. Fault Coverage (after collapsing) of LOS, LOC, LOS+LOC and MIX mode delay tests are listed in column2-5. Test pattern numbers (after dynamic and static compression) of LOS, LOC and LOS+LOC delay tests are listed in columns 6-8. Test pattern numbers (without compression due to ATPG limitations) of the MIX mode delay test are listed in column 9.

Table 4-4: ATPG results on ISCAS89 benchmark circuits

Circuit	Fault Coverage (%)				Test patterns			
	LOC	LOS	LOS+LOC	MIX	LOC	LOS	LOS+LOC	MIX
s298	85.36	85.04	96.34	98.12	37	32	48	54
s344	95.58	92.60	99.04	99.04	35	29	38	38
s349	94.71	91.39	98.34	98.34	34	29	40	40
s382	81.61	85.15	94.82	97.68	44	34	45	52
s386	84.11	58.43	92.44	98.06	59	50	96	120
s400	79.91	84.16	93.32	95.88	43	37	49	55
s420	90.65	82.25	97.00	97.72	106	75	122	128
s444	80.85	85.06	93.14	95.73	52	33	53	61
s510	92.23	52.49	95.93	98.80	75	57	120	146
s526	68.36	84.60	95.81	97.71	72	54	89	99
s641	99.14	97.27	99.61	99.69	45	34	46	47
s713	87.07	85.17	87.45	87.52	57	34	47	48
s820	86.88	55.99	91.65	98.15	129	88	177	216
s832	85.70	54.46	90.48	97.42	130	88	180	225
s838	89.05	86.13	97.35	97.70	217	180	246	252
s953	95.31	95.99	99.22	99.55	110	103	132	137

Circuit	Fault Coverage (%)				Test patterns			
	LOC	LOS	LOS+LOC	MIX	LOC	LOS	LOS+LOC	MIX
s1196	100.00	96.24	100.00	100.00	194	157	199	199
s1238	97.04	92.45	97.04	97.04	204	159	203	203
s1423	94.83	96.25	98.88	98.93	85	62	73	75
s1488	93.49	61.83	95.52	99.08	146	108	213	268
s5378	89.16	93.83	98.13	98.46	178	176	248	270
s9234	90.89	92.70	95.38	95.74	342	232	272	287
s13207	91.61	91.03	98.54	98.90	457	281	455	497
s15850	87.65	96.58	96.79	96.93	247	185	230	242
s35932	87.55	87.77	88.06	88.06	40	42	55	55
s38417	98.35	97.77	99.71	99.73	216	203	245	253
s38584	93.15	96.02	96.10	96.12	412	279	274	279
Ave.	89.64	84.39	95.78	97.26	139	105	148	161

Notice that in this commercial ATPG experiment, benchmark circuits have different synthesizing and scan chain ordering compared with same benchmark circuits under simulation based ATPG experiment environment, which is discussed in the earlier sections of this chapter.

Also notice that, although the fault coverage of the LOS test is higher than the LOC test, the risk of over-testing for the LOS test is also higher because the LOC test only uses logic states as V2 while the LOS test does not. Therefore, higher fault coverage of the LOS test might induce an extra yield loss. Over-testing issues of scan designs are due to scan insertion, which makes redundant faults in the original circuit detectable after scan insertion. There are already two techniques available to eliminate over-testing issues. One technique [32, 33] masks a small portion of positions in the output response of a circuit in order to avoid the detection of those redundant faults that become detectable after scan insertion. The other technique [34-37] attempts to ensure that faults are detected only under functional operation conditions.

CHAPTER 5 USING PARTIAL DTSFF DESIGN AND PARTIAL ENHANCED DTSFF DESIGN FOR LOW COST DELAY TESTING

5.1 Partial Delay Test Scan Flip-flop Delay Testing Scheme

While the area increase of the proposed DTSFF over the traditional scan flip-flop is quite modest, the design can become even more attractive if this overhead can be further significantly reduced. One possible solution is to develop a partial DTSFF delay test methodology, which typically requires partial scan flip-flops in the design to be replaced by DTSFFs, without significant degradation in the TDF coverage. In this partial DTSFF scheme, only some of the flip-flops in the circuit are implemented using DTSFFs, while the rest remain normal scan flip-flops, significantly reducing the area penalty for using DTSFFs.

A full DTSFF design is illustrated in Figure 5-1 (a) and a partial DTSFF design is illustrated in Figure 5-1 (b). The difference between them is that the former uses the DTSFFs in place of all regular scan Flip-Flops (SFFs), while the latter uses a mix of the two flip-flop types. Observe that the partial DTSFF design employs two independent (slow) scan enable signals to control each flip-flop type. This is necessary to allow the design to support traditional stuck-at tests.

In a full DTSFF design, all memory elements are implemented with the Delay Test Scan Flip-Flops. These DTSFFs are under the control of a single slow Scan Enable signal so that they are able to operate in either LOS mode (if the Scan Enable signal is switched after the last scan shift clock edge) or LOC mode (if the Scan Enable signal is switched before the last scan shift clock edge).

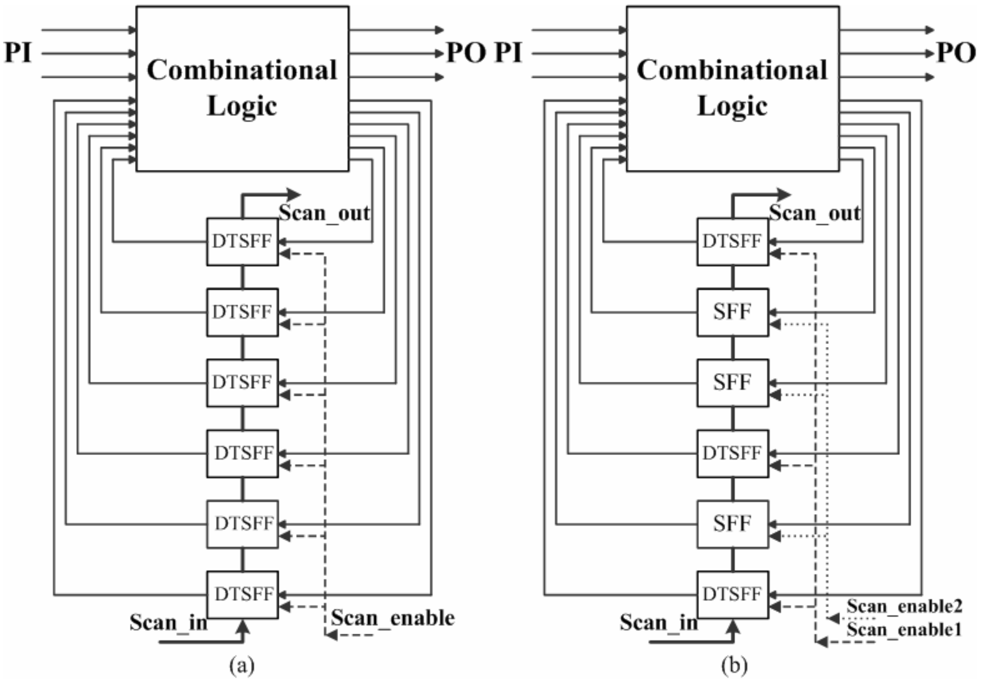

Figure 5-1: (a) Full DTSFF design; (b) Partial DTSFF design (SFF are regular scan Flip-flops)

In a partial DTSFF design, the two scan enable signals offer four possible test modes as shown in Table 5-1. If both the Scan Enable signals are switched after the last scan shift clock edge (as shown in Figure 3-2), then the memory elements implemented with DTSFFs operate in the LOS mode, while others implemented with regular SFF operate in the LOC mode. Notice from Figure 3-2 that while the DTSFF input multiplexers will get the timed MUX control signal after the next clock edge for a LOS test, the regular SFFs will use the unmodified scan enable signal to control their input multiplexers, resulting in LOC operation. Since these two test modes now coexist in the scan chain, we say that the partial DTSFF scan chain operates in the LOS-LOC mode. (Note that, our LOS-LOC mode is different from the test mode in [15, 16] where the scan flip-flops cannot capture any responses when in the shift mode). Similarly, if both the Scan Enable signals switch before the last scan shift clock edge, as seen in Figure 3-3, the DTSFFs operate in the LOC mode, while the SFFs launch the transition test on the clock edge following a second capture cycle; named LOCC (Launch-on-capture-capture). Thus we say that, in this case, the scan chain operates in the LOC-LOCC mode. Note that the tests generated by this LOC-LOCC mode are not the same as those from the conventional LOC (or LOC-LOC) mode.

Table 5-1: Operation modes for partial DTSFF delay test

Mode	Scan_enable1	DTSFF	Scan_enable2	SFF
LOS-LOC	Switch after last bit shift	LOS	Switch after last bit shift	LOC
LOC-LOCC	Switch before last bit shift	LOC	Switch before last bit shift	LOCC
LOC-LOC	Switch before last bit shift	LOC	Switch after last bit shift	LOC
LOS-LOCC	Switch after last bit shift	LOS	Switch before last bit shift	LOCC

The LOC-LOC mode is implemented in the partial DTSFF design by switching the scan enable for the DTSFFs in the scan cycle before the last V1 bit is scanned in (to achieve LOC operation in the DTSFFs as shown in Figure 3-3), while switching the scan enable for the regular flip-flops in the cycle just after the last shift as in traditional LOC tests. Finally, an additional test mode is possible by switching the scan enable for the DTSFFs after the last shift, and the scan enable for the regular flip-flops before the last shift; this is a LOS-LOCC mode. However, TDF test generation for this mode is difficult because the DTSFFs shift after the first capture cycle in the regular flip-flops, requiring a mix of sequential analysis and pattern shifting to predict the new scan chain contents (V2 test vector). This is currently not supported by commercial test tools, and consequently only the first three test modes were used in test generation for the experiments reported in this dissertation. While this LOS-LOCC test mode can be expected to detect a few unique faults, the loss of reported coverage is likely to be quite modest since most of the TDF coverage in partial DTSFF designs is provided by the LOS-LOC and LOC-LOC modes. Of course, with improved tools, this mode can also be exploited for better tests.

Our interest in the partial DTSFF designs is motivated by the intuition that the TDF coverage of traditional LOC tests may be significantly impacted by the limited controllability at the inputs of relatively few flip-flops in the circuit, which bias some bits in the V2 vector generated by the logic block to be mostly ‘0’ or ‘1’ . If these flip-flops can be identified and replaced by DTSFFs that can also support LOS tests, the constraints on the V2 vector will be reduced. Consequently, most of the benefits, in terms of TDF coverage, of a fully DTSFF design may be achieved with lower overhead costs.

For the partial DTSFF approach to be cost effective, the relationship between the percentage of DTSFFs in the design and TDF coverage should ideally be as shown in Figure 5-2. Notice that the TDF coverage of a traditional design with a slow scan enable input is just the coverage attainable by LOC tests. A full (100%) DTSFF design can attain the combined LOC+LOS coverage, since both tests are fully supported. Coverage for partial DTSFF designs typically falls somewhere in between. If the flip-flops selected to be changed to DTSFFs from regular SFFs are randomly picked, on average coverage can be expected to generally increase with the percentage of DTSFFs; some DTSFFs introduced into the design may bring very little benefit, while others may have a more significant impact on coverage. On the other hand, if a good selection procedure can be developed to incrementally identify the flip-flops that yield the greatest coverage improvement, relatively few DTSFFs may bring about most of the coverage gains. In such a case the coverage versus percentage DTSFF plot may ideally look as shown in Figure 5-2, offering an attractive trade-off between TDF coverage and DTSFF overhead. Using only a small fraction of DTSFFs at low cost may still allow high coverage.

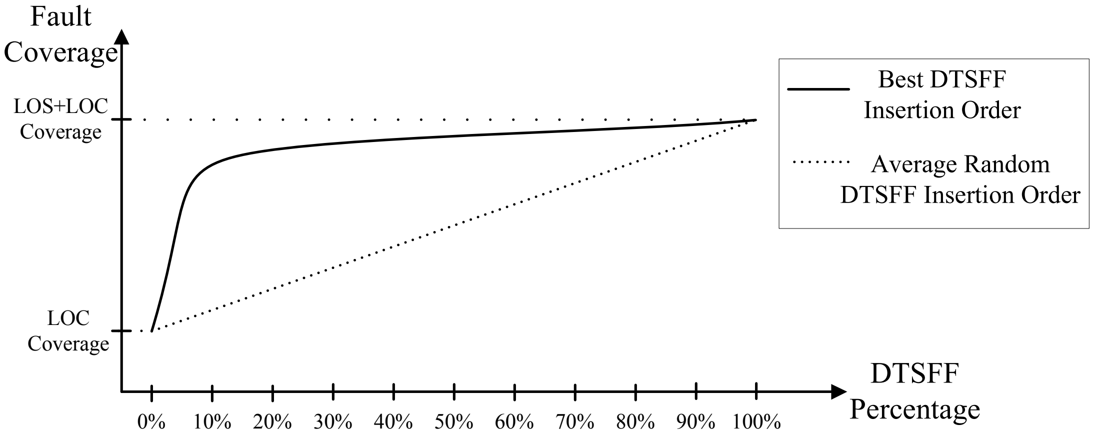

Figure 5-2: Desired relationship between DTSFF percentage and Fault Coverage

5.2 Partial Enhanced Scan design

Although enhanced scan techniques have been around for a long time, 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 scan based tests, beyond what is possible from traditional LOC tests, to detect small delay defects and perhaps also avoid the need for at-speed functional tests, has revived interest in such schemes [16]. In this section we investigate a strategy for realizing most of the TDF coverage gains achievable from enhanced scan at a fraction of the cost by implementing partial enhanced scan designs.

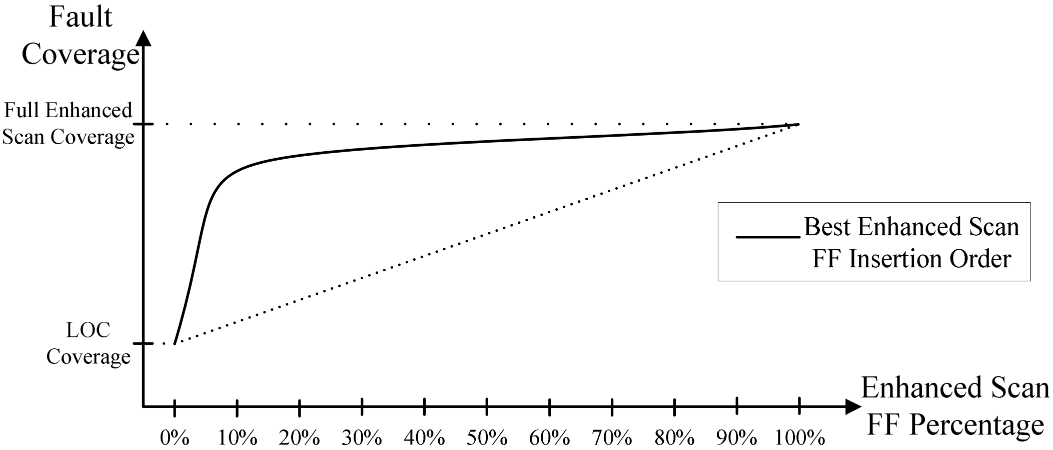

Figure 5-3: Ideally desired relationship between partial enhanced scan flip-flop percentage and TDF coverage

While the idea of partial enhanced scan chains had been presented before, within the context of increasing path delay coverage [38]. One of the specific problems addressed in the earlier work was: starting from a (full) enhanced scan chain, what flip-flops can be replaced with regular flip-flops without reducing the achievable path delay coverage. Our objective here is very different. We want to identify a relatively small number (10-30%) of flip-flops in the scan chain such that when these are replaced by two-stage enhanced scan flip-flops, the majority (60-90%) of the additional coverage achievable by going to an all-enhanced scan flip-flop design is already realized. This is possible if we can develop a flip-flop selection scheme that gives a coverage versus fraction of enhanced scan flip-flops trade-off as shown in Figure 5-3. Such a methodology can then offer attractive low cost options for partial enhanced scan designs as shown in Figure 5-4.

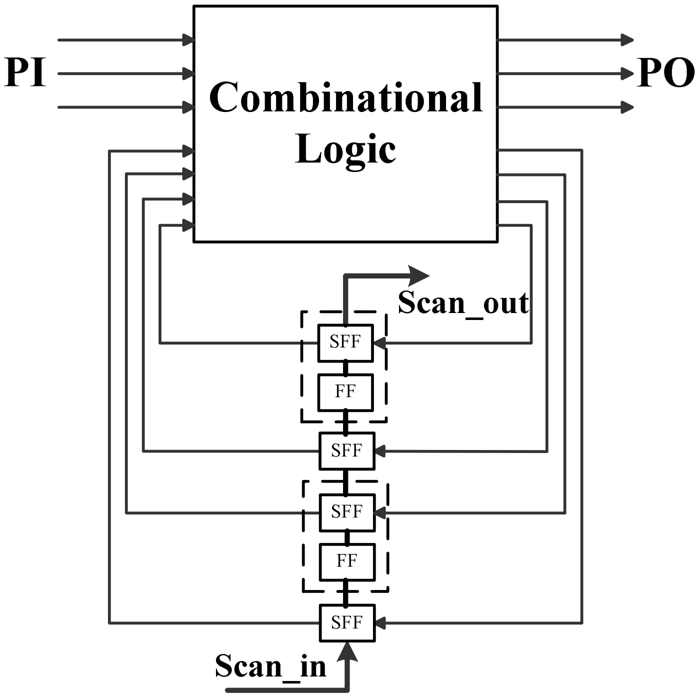

Figure 5-4: Partial enhanced scan (enhanced scan flip-flop pairs are enclosed in the dashed boxes)

Unfortunately, in all of the enhanced scan designs discussed in Chapter 2, control signals capable of switching at operational clock speeds are needed to ensure proper test timing. For example, it is well understood that the scan enable signal must be capable of at-speed switching to support the LOS tests needed by the design in Figure 2-3 and Figure 5-4. Similarly the individual hold control signals at each latch in Figure 2-4 must also switch is a timed and synchronized manner throughout the IC to launch the V1 to V2 transition at the same instant at all the inputs. This is not practical using a slow global broadcast signal which can be subject to substantial timing skews at different locations on the chip. Implementing high speed control signals is very expensive, loosely comparable in cost to an extra clock signal. Such signals must be avoided in any low cost design which attempts cost savings from a partial enhanced scan methodology.

The need for a high speed scan enable is alleviated in the dual flip-flop enhanced scan design in [16], where the enhanced scan flip-flop comprises two cascaded standard scan flip-flops as shown in Figure 2-5. As discussed in Chapter 2, the shifting operation and the capture operation of this design can be achieved using slow speed scan control signals, without any need for high speed switching within the launch to capture window.

If the enhanced scan flip-flop in Figure 2-5 is used in a partial enhanced scan design, along with a slow scan enable, the enhanced scan flip-flops can launch arbitrary two bit patterns at their outputs during the V1 to V2 transition, while the regular scan flip-flops must operate in the LOC mode (LOS is not supported by a slow scan enable). This implies that the lower bound TDF coverage of such a partial enhanced scan design is just the LOC coverage (with 0% enhanced scan flip-flops in the scan chain).

Our goal is to select an increasing number of flip-flops in the scan chain to convert to enhanced scan flip-flops in such a manner that for a given number of enhanced flip-flops in the partial scan chain, the TDF coverage is the maximum, as shown in Figure 5-3. We present our flip-flop selection procedure in the next section.

5.3 Using Controllability Analysis for Scan Element Selection

In a scan delay test, the first vector V1 can be selected without any restrictions since V1 is scanned into the CUT from the tester. We therefore say that V1 is un-biased by the structural limitations of scan design. The second vector V2, in both LOS and LOC applications, depends on V1 and cannot be arbitrarily selected. For a LOC test, each bit in V2 does not have 50% probability to be ‘0’ or ‘1’ since V2 now is the response of V1, and is “conditioned” by the combinational logic. Some bits in V2 are more frequently found to be ‘0’ or ‘1’. We call these the biased bits of the vector.

For example, in Figure 5-5, we assume each bit of V1 has an unbiased 0.5 (50%) probability to be logic ‘0’. The bits of V2, where V2 is the response of V1 for the LOC test, have different probability values (estimated to be 0.44, 0.78, 0.56, 0.91 and 0.23 in this example). If we (quite arbitrarily) pick 0.75 to be the upper threshold value and 0.25 to be the lower threshold value for defining an unbiased bit, then the 0.91 and 0.23 probabilities in Figure 5-5 indicate bits that are biased (towards ‘0’ and ‘1’, respectively), while a 0.56 probability of ‘0’ indicates that the bit is mostly unbiased.

In a LOC delay test, circuit bias often makes some flip-flops acquire a biased value (‘0’ or ‘1’) in the V2 vector during the test; this restriction degrades the fault coverage of the LOC test. If these biased bits can be changed to unbiased values, the fault coverage achievable for the LOC test can be improved. Therefore, we proposed to replace the flip-flops receiving “biased” logic values from the combinational logic in the LOC mode with DTSFFs. This allows the corresponding bits of V2 to be generated by shifting. Meanwhile unbiased signals can continue to be captured in regular SFFs to reduce DFT overhead when compared with the full DTSFF designs.

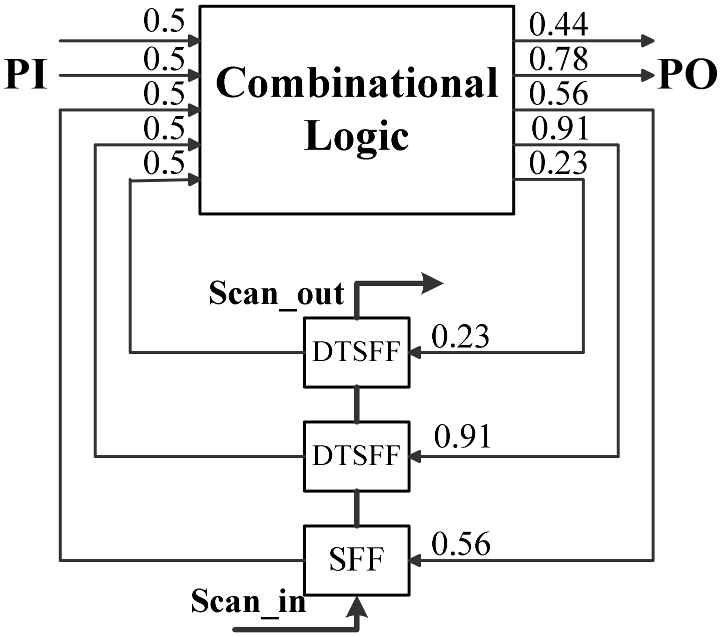

Figure 5-5: Example circuit with each node labeled with the probability to be logic ‘0’ (Partial DTSFF Design)

The example in Figure 5-5 illustrates this strategy. Using 0.25 and 0.75 signal probabilities as bias thresholds, the bottom “unbiased” memory element (0.56) is implemented with a SFF and the other biased ones (0.23, 0.91) are implemented with DTSFFs.

Signal probabilities at the outputs of a combinational block can be readily estimated for unbiased random inputs either by running Monte Carlo simulations or by probabilistic analysis. Table 5-2 shows how the probability of a ‘0’ can be calculated at the outputs of different gates, based on the input probabilities. A similar table can be generated for computing the ‘1’ probabilities. All signal probabilities in a circuit can be iteratively calculated using these tables. The starting probabilities for the inputs of the combinational block are taken to be the unbiased value, 0.5.

Table 5-2: Calculating the probability to be ‘0’

Logic	Inputs $P_I(0)$	Output $P_O(0)$
AND	$P_{I1}(0), P_{I2}(0) \ldots P_{IN}(0)$	$1 - \prod_{i=1}^{N} (1 - P_{I_i}(0))$
NAND	$P_{I1}(0), P_{I2}(0) \ldots P_{IN}(0)$	$\prod_{i=1}^{N} (1 - P_{I_i}(0))$
OR	$P_{I1}(0), P_{I2}(0) \ldots P_{IN}(0)$	$\prod_{i=1}^{N} P_{I_i}(0)$
NOR	$P_{I1}(0), P_{I2}(0) \ldots P_{IN}(0)$	$1 - \prod_{i=1}^{N} P_{I_i}(0)$
XOR	$P_{I1}(0), P_{I2}(0)$	$P_{I1}(0)P_{I2}(0) + (1 - P_{I1}(0))(1 - P_{I2}(0))$
XNOR	$P_{I1}(0), P_{I2}(0)$	$P_{I1}(0)(1 - P_{I2}(0)) + (1 - P_{I1}(0))P_{I2}(0)$
BUF	$P_{I1}(0)$	$P_{I1}(0)$
NOT	$P_{I1}(0)$	$1 - P_{I1}(0)$

It is important to note that this simple signal probability estimation method does not account for signal correlations at reconvergent gates, and can therefore be subject to some error, which can be significant in unusual cases. While Monte Carlo simulations do not suffer from this inaccuracy, statistically obtaining the signal probabilities through such random simulation can be time consuming for large circuits, for which the input space explodes. We will employ both these approaches in the experiments in Section 5.5.

Recall that signal probabilities at the flip-flop inputs help us to order the flip-flops in terms of signal bias, and allow us to incrementally find the next flip-flop to convert to DTSFF for the best TDF coverage improvement as we introduce an increasing number of DTSFFs in a design.

The above DSTFF selection methods using controllability analysis and Monte Carlo simulation can also be used for the scan element selection in partial enhanced scan design. For example, in Figure 5-6, we assume each bit of V1 has an unbiased 0.5 (50%) probability to be logic '0'. The bits of V2, where V2 is the response of V1 for the LOC test, have different probability values (estimated to be 0.44, 0.78, 0.56, 0.91 and 0.23 in this example). If we (quite arbitrarily) pick 0.75 to be the upper threshold value and 0.25 to be the lower threshold value for defining an unbiased bit, then the 0.91 and 0.23 probabilities in Figure 5-6 indicate bits are biased (towards '0' and '1', respectively), while a 0.56 probability of '0' indicates that the bit is mostly unbiased.

In a LOC delay test, circuit bias makes some flip-flops often acquire a biased value ('0' or '1') in the V2 vector during the test; this restriction degrades the fault coverage of the LOC test. If these biased bits can be changed to unbiased values, the fault coverage achievable for the LOC test can be improved. Therefore, we propose to replace the flip-flops receiving “biased” logic values from the combinational logic in the LOC mode with enhanced scan flip-flops. This allows the corresponding bits of V2 to now be arbitrarily selected and scanned in. Meanwhile unbiased signals can continue to be captured in regular SFFs to reduce DFT overhead when compared with the full enhanced scan designs.

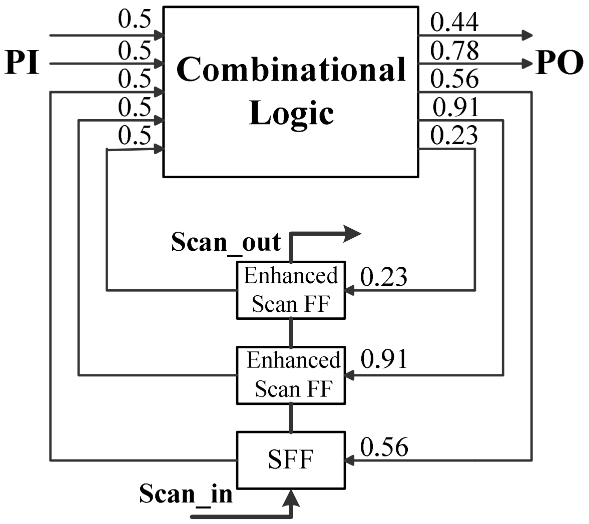

Figure 5-6: Example circuit with each node labeled with the probability to be logic ‘0’ (Partial Enhanced Scan Design)

The example in Figure 5-6 illustrates this strategy. Using 0.25 and 0.75 signal probabilities as bias thresholds, the bottom “unbiased” memory element (0.56) is implemented with a SFF and the other biased ones (0.23, 0.91) are implemented with enhanced scan flip-flops.

Signal probabilities at the output of a combinational block can be readily estimated for unbiased random inputs either by running Monte Carlo simulations or by analytical controllability analysis methods. For the experiments described in Section 5.7, we have used a Monte Carlo approach.

5.4 The Interchange Procedure

Recall that signal probabilities at the flip-flop inputs help us to order the flip-flops in terms of signal bias, and allow us to incrementally find the next flip-flop to convert to a DTSFF for the best TDF coverage improvement as we introduce an increasing number of DTSFFs in the partial DTSFF design. Unfortunately, simple controllability estimates do not capture all the complex interactions between the inputs required to activate and propagate fault effects through the logic, and are only somewhat loosely related to TDF coverage. We therefore implement an additional interchange procedure, using a selected level of resolution, to see if the results can be further improved. The idea here is to divide the ordered list of flip-flops into subsets, and check if the shape of the coverage versus % DTSFF plot can be improved by iteratively interchanging adjacent subsets of flip-flops. The interchange procedure is described below. Notice that the Interchange Procedure discussed here can also be applied to partial enhanced scan design.

Procedure_Interchange

1) Assumptions and Definitions:

Assume that the resolution of the Interchange Procedure is 5%. Then according to the controllability ordering of each flip-flop, all flip-flops in the circuit are assigned into 20 groups (G_1, G_2… G_{20}). G_0 is also defined as a group which contains zero flip-flop.

FC_N (N=0, 1, 2 …20) is defined as the fault coverage when flip-flops in groups indexed $\leq$ N (i.e. G_0, G_1…G_N) are implemented with DTSFFs.

Slope$_N$ (N=1, 2… 20) is defined as FC$_N$ - FC$_{N-1}$. (The goal of ideal FF selection is to ensure that Slope$_N$ decreases monotonically when N increases, as in Figure 5-2.)

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.)

2) Interchange Procedure:

```
Calculate $FC_N$ (N=0, 1, 2…20);
Interchange_times=0;
Delta_Slope_allowable_value= -0.03;
Interchange_allowable_times= 30;
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;
}
```

In the Interchange Procedure for the experiments reported later, we arbitrarily choose the Interchange_allowable_times to be 30. 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. Interchange between very small differences in slopes will not yield a meaningful difference in the fault coverage versus DTSFF percentage trade-off. Note that each time we Update FC_{M-1}, we must re-synthesize a new partial DTSFF circuit and re-run ATPG using Synopsys TetraMax.

5.5 Experimental Results on Partial DTSFF Design

In order to study the effectiveness of the partial DTSFF methodology, we studied six of the larger ISCAS89 benchmark circuits, containing 74 to 1426 flip-flops; the smaller circuits have too few flip-flops to provide meaningful results. We also dropped those benchmark circuits from this study which showed only limited fault coverage improvement from adding the LOS test to LOC, since TDF coverage in such circuits would not significantly benefit from partial TDSFF insertion.

For each design, we obtained signal probabilities at the outputs of the combinational block, assuming unbiased (0.5) probabilities at the inputs, using both the analytic method and Monte Carlo simulation (up to 200,000 vectors). These signal lines, which are the inputs to the flip-flops, were then ordered based on the computed probabilities; the flip-flops receiving the most biased inputs, with probabilities furthest from 0.5, were the first to be replaced by DTSFFs as we inserted an increasing number of DTSFFs to configure the partial DTSFF scan chains. In this manner, we obtained a number of partial DTSFF designs for each benchmark with 0, 5, 10, 15, 20… 95, and 100% DTSFFs in the scan chains. For example, the 10% DTSFF design had 10% of the most biased flip-flops replaced by DTSFFs.

For test generation, a netlist for the partial DTSFF circuits was generated by replacing SFFs with DTSFFs in the netlist of the corresponding full SFF scan circuit generated by the Synopsys DFT Compiler tool. Scan based TDF tests were next generated for each of the partial DTSFF designs for the combined LOS-LOC and LOC-LOCC test modes shown in Table 5-1 using Synopsys TetraMax tools. To generate tests in the LOS-LOC mode for our partial DTSFF design, the ATPG was run for LOC test generation with the capture cycle set to 2. Recall that the DTSFFs introduce one more shift when the scan enable signal is timed for an LOC test; this results in the LOS-LOC test mode. Similarly, performing LOC test generation with the capture cycle set to 3 generates tests for the LOC-LOCC mode. Tests for the combinational part of the circuit in the pure LOC (LOC-LOC) mode were separately generated using a non (0%) DTSFF version of the circuit, and the additional faults detected were taken credit for in the reported fault coverage. As discussed before, the LOS-LOCC mode is not easily supported by the tools, omitting it has a small, pessimistic impact on the fault coverage reported.

Note that in modeling the circuit for test generation, the clock alignment logic in the DTSFF design is replaced by a functionally equivalent D-latch cell from the Synopsys library shown in Figure 5-7; this allows TetraMax to generate tests without flagging a combinational feedback violation.

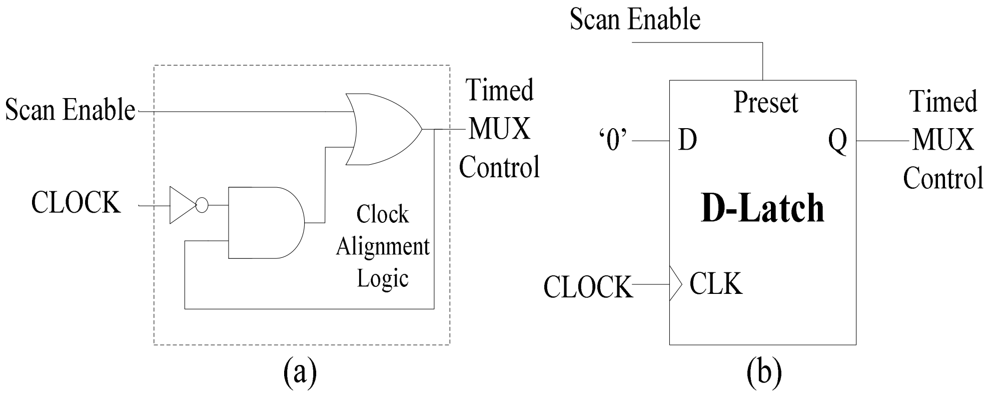

Figure 5-7: (a) The Clock Alignment Logic (b) Functionally equivalent D-latch cell with preset line

TDF coverage versus % DTSFFs results for the six circuits are presented in Figure 5-8 (a)-(f) for analytically estimated signal probabilities and Figure 5-9 (a)-(f)for Monte Carlo based signal probability estimation. The plots show fault coverage when the DTSFFs are introduced in the "Normal Order" as discussed before, and also when this DTSFF insertion order is reversed. We shall discuss the "Reverse Order" results later.

Note that the figures show TDF fault coverage, which as reported by TetraMax [39] includes undetectable and ATPG undetectable faults in the fault list. TDF test coverage numbers, which ignore the undetectable faults, could unfortunately only be computed for full (100%) DTSFF designs supporting complete LOS and LOC testing. These are reported in the captions for Figure 5-8, and are above 99% for all six circuits. This indicates that virtually all of the missing fault coverage in the plots is due to undetectable faults. These results again highlight the capability of full DTSFF designs to achieve near perfect TDF delay test coverage with a minimal overhead of 6 transistors per flip-flop.

The results for partial DTSFF designs varied with the circuit, with the Monte Carlo based flip-flop selection performing somewhat better than the analytical method.

Generally, most of the TDF coverage benefits of introducing DTSFFs were reached with 50% or fewer DTSFFs in the scan chain in the plots in Figure 5-9. We shall shortly discuss how these results can be further improved as shown in Figure 5-10(a)-(f).

Observe that s13207 in Figure 5-8(d) behaves closest to the ideal case shown in Figure 5-2, with TDF coverage monotonically increasing with the percentage of DTSFFs. However, in some cases, such as s1423, the plot is much more erratic. In fact, going from 55% to 60% DTSFFs in s1423, TDF coverage actually goes down with an increase in the percentage of DTSFFs. This is because, in addition to errors in computing the signal probabilities, our incremental greedy flip-flop selection approach is not globally optimal. TDF coverage is the result of a complex interaction between the logic values and test modes of the flip-flops, and cannot be fully manipulated by managing the "signal bias" in flip-flops alone.

To see if our flip-flop selection procedure to achieve high TDF coverage with the fewest number of DTSFFs is in fact better than a purely random selection, we conducted experiments that reversed this order of adding flip-flops to the scan chain. Now the least biased flip-flops were introduced in the partial DTSFF scan chain first, and the most biased last. The results are captioned "Reverse Order" in the figures. Notice that in most cases such an unfavorable selection resulted is TDF coverage improving much more slowly with an increase in the DTSFF percentage. This was most pronounced for s5378, where initially the coverage does not improve at all, even past 50% DTSFFs.

These reverse order results indicate the effectiveness of the proposed approach in identifying a useful insertion ordering of the flip-flops. They also suggest that Monte Carlo based signal probability estimation performs better in general, given the better separation between the normal and reverse order results in Figure 5-9. However, this is not always the case; for s13207 and s38584, the fault coverage reported in Figure 5-8 for the analytical approach is higher in some instances.

It is also important to understand why a 95% DTSFF design sometimes shows higher TDF fault coverage than the 100% DTSFF design in the plots; for example for s9234 in Figure 5-8(c). The reason is that we have allowed the partial DTSFF designs two independent scan enable signals (necessary to support traditional stuck-at scan tests) which allow the four delay test modes shown in Table 5-1. On the other hand, for the results reported for full (100%) DTSFF designs, we only assume a single scan enable signal, which can only support two tests modes: LOS and LOC. (A single scan enable line is sufficient to support traditional tests in full DTSFF designs.) Allowing two scan enable signals in full DTSFF designs will further improve TDF coverage by allowing mixed LOS and LOC tests [30], but raises the question of how the flop-flops are assigned to the two scan partitions. Since such a study is outside the scope of partial DTSFF design, we have reported results for 100% DTSFF with only a single scan enable.

Also notice that the available V2 patterns for different percentage partial DTSFF designs are different. For example, a 75% partial DTSFF design may generate a unique V2 pattern which can not be generated by an 80% partial DTSFF design. If this unique V2 pattern can detect extra defaults which are not detected by the 80% partial DTSFF design, then the 75% DTSFF design may obtain even higher fault coverage than the 80% partial DTSFF design. “Reverse Order” results in Figure 5-8(b) demonstrate such a case.

Recall that, ideally, we seek a flip-flop insertion ordering that allows fault coverage to increase monotonically with the percentage of DTSFFs as shown in Figure 5-2, and approximated by the plots in Figure 5-9(d) and Figure 5-9(f). Furthermore, the slope of the TDF coverage curve should be constantly decreasing with as the percentage of DTSFFs increase. This is not always the case in practice; for example in Figure 5-9(b) the slope increases sharply as we go from 20% to 30% DTSFFs.

This suggests that interchanging the flip-flops in the 20-25% and 25-30% groups with those in the 5-10% and 10-15% groups may yield higher TDF coverage with fewer DTSFFs, although given the complex interrelation between the flip-flops, this cannot be guaranteed. The results in Figure 5-10(b) show that for s5378, such an interchange can indeed yield an improved trade-off between TDF coverage and the percentage of DTSFFs; in this case nearly 98% fault coverage can be reached with only 20% DTSFFs.

Similarly, iteratively interchanging the order of selection for adjacent sets of 5% of the flip-flops any time the slope of the coverage plots increases instead of decreasing (with increasing DTSFFs) yields the improved results for the six circuits shown in Figure 5-10. This brings these coverage plots closer to the smoothly increasing coverage desired in Figure 5-2. We now have an attractive DTSFF selection ordering to trade off TDF coverage versus DTSFF overhead for all six circuits. While the above interchange procedure is somewhat computationally intensive, it does show the potential for obtaining even further gains in TDF coverage with a lower number of DTSFFs in the scan chains than suggested by the results in Figure 5-8 and Figure 5-9. Efficient methods for exploiting this potential are the subjects of ongoing work. For example, the methodologies used for partial scan [40-45] designs may be borrowed to refine our current methodology.

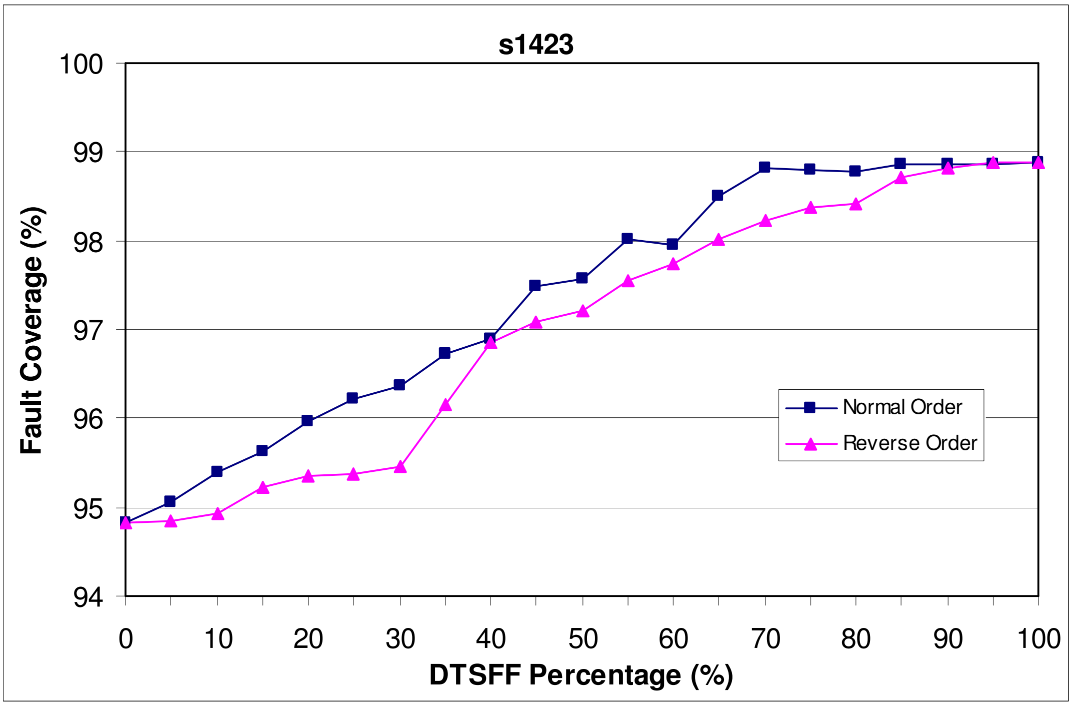

Figure 5-8(a): Benchmark S1423 (LOS+LOC Test Coverage 99.89%)

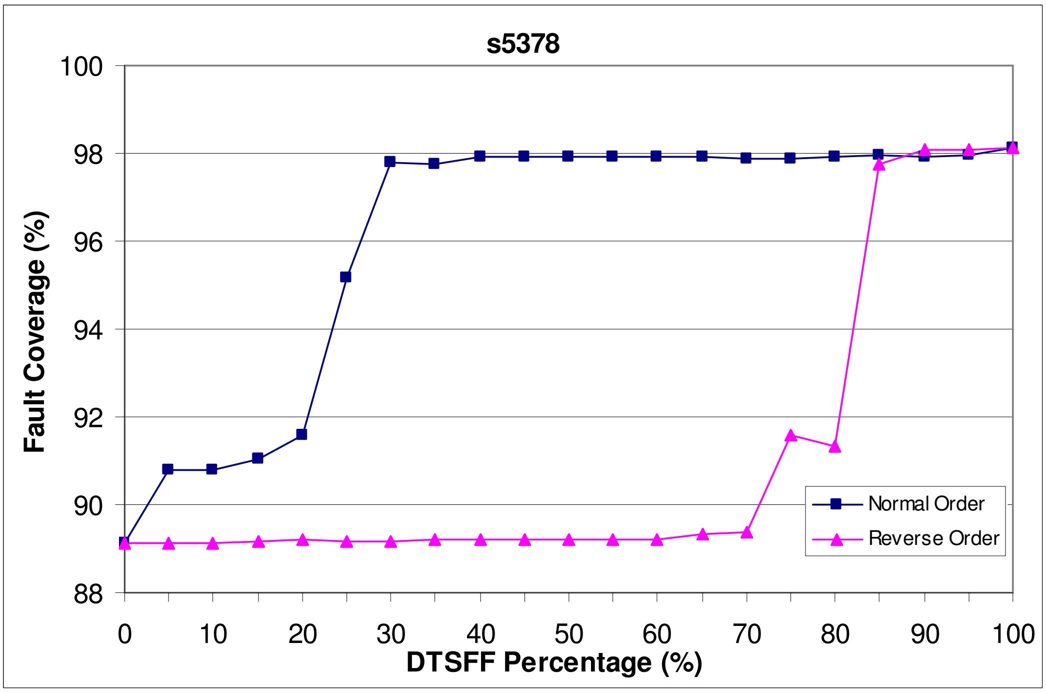

Figure 5-8(b): Benchmark S5378 (LOS+LOC Test Coverage 99.39%)

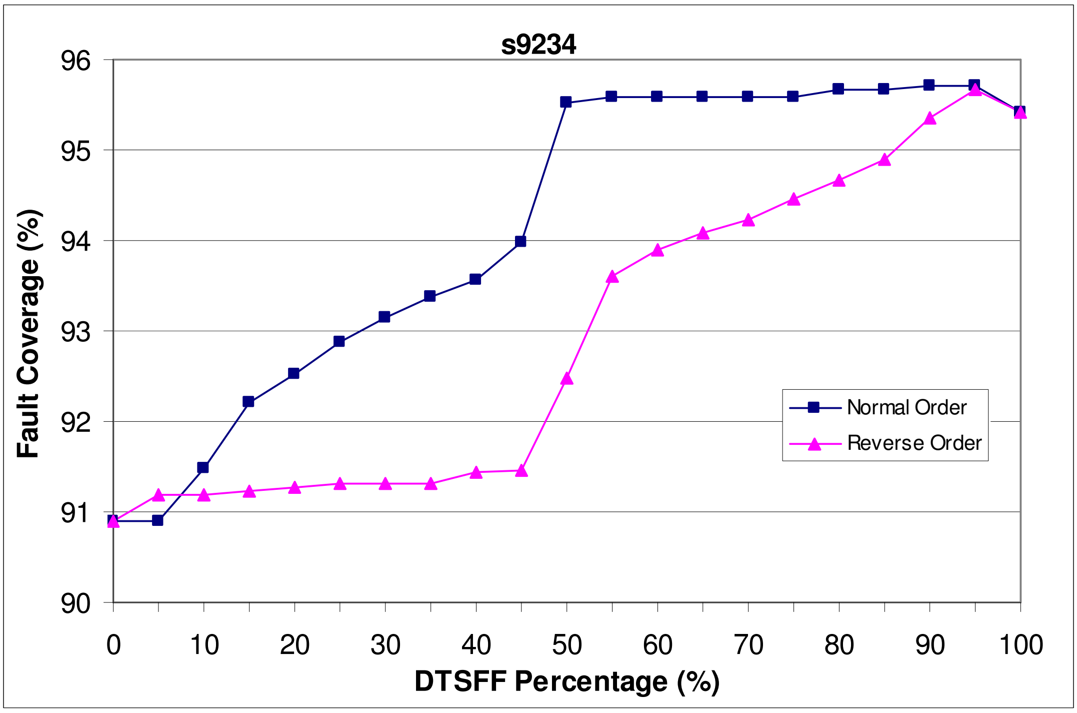

Figure 5-8(c): Benchmark S9234 (LOS+LOC Test Coverage 99.65%)

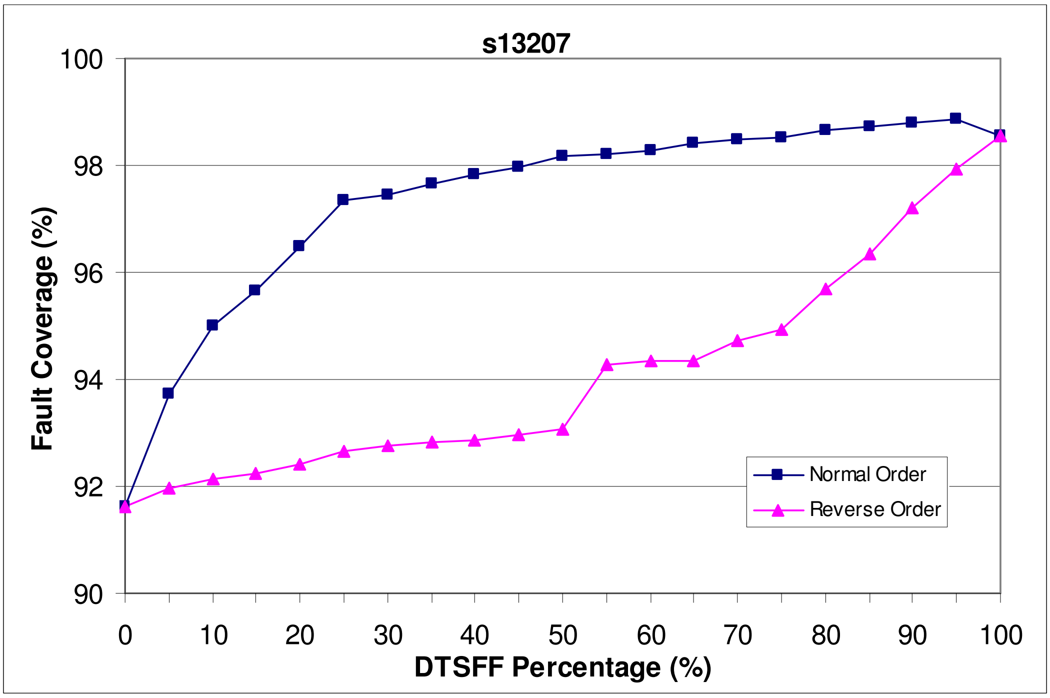

Figure 5-8(d): Benchmark S13207 (LOS+LOC Test Coverage 99.61%)

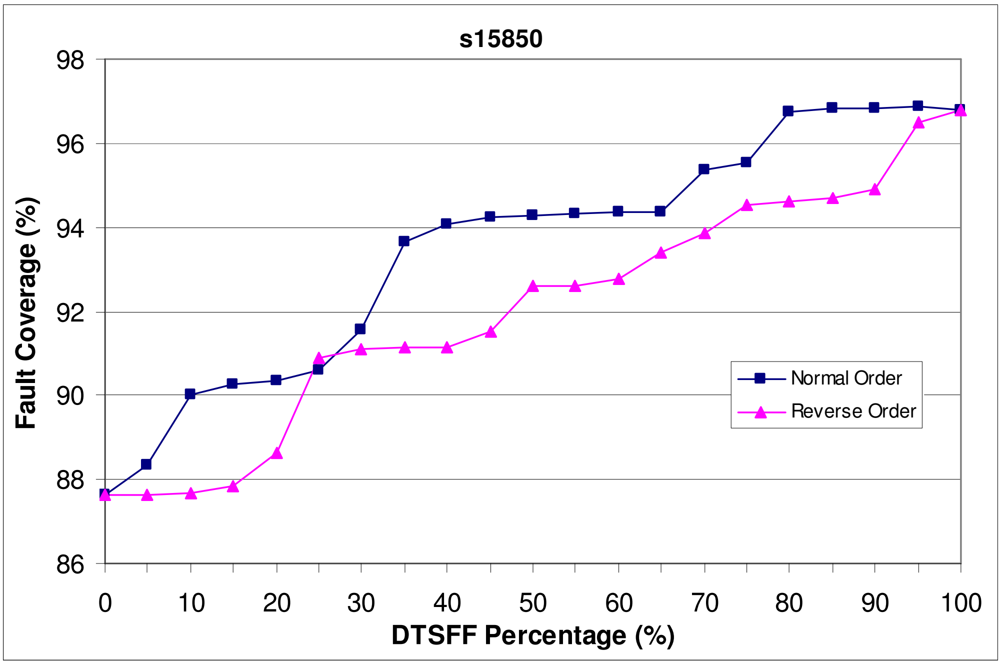

Figure 5-8(e): Benchmark S15850 (LOS+LOC Test Coverage 99.82%)

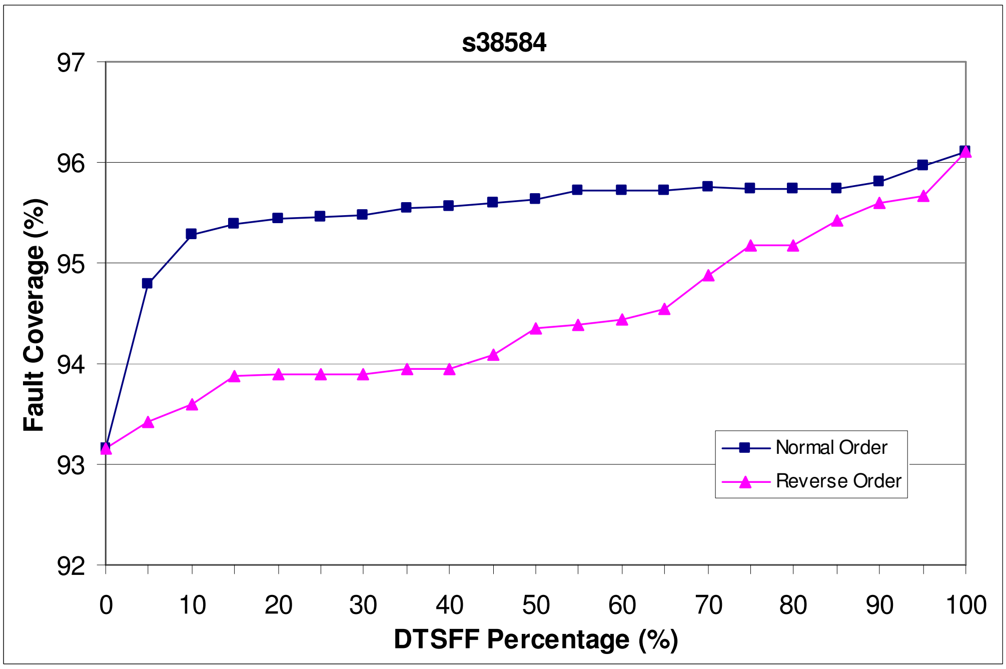

Figure 5-8(f): Benchmark S38584 (LOS+LOC Test Coverage 99.97%)

Figure 5-8: DTSFF selection based on analytical probability calculation

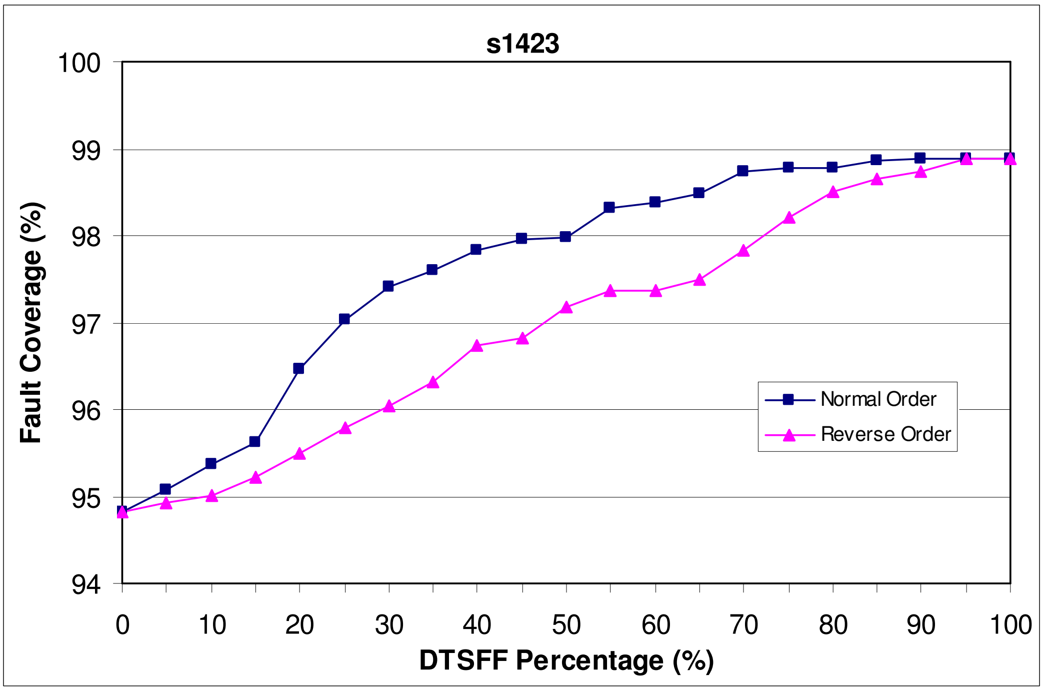

Figure 5-9(a): Benchmark S1423

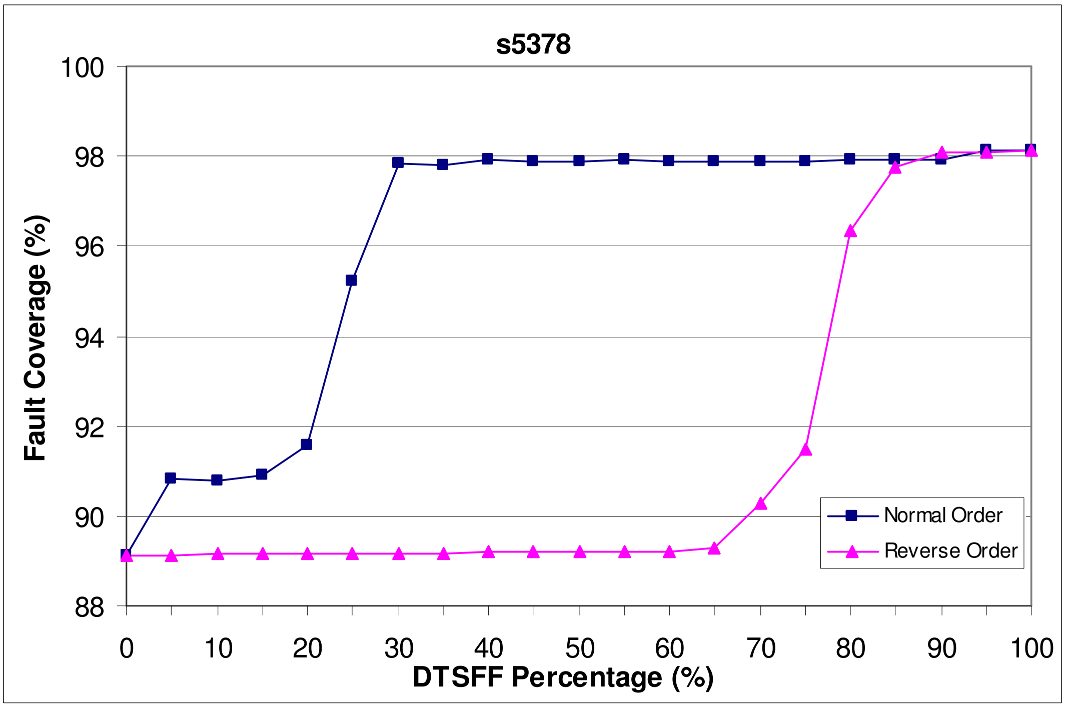

Figure 5-9(b): Benchmark S5378

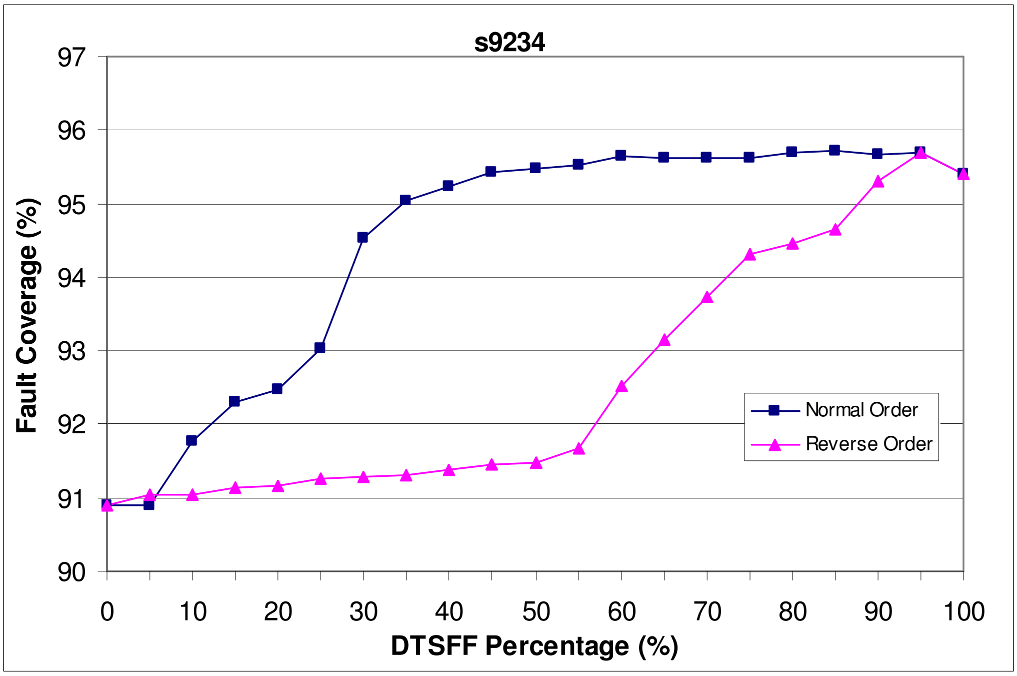

Figure 5-9(c): Benchmark S9234

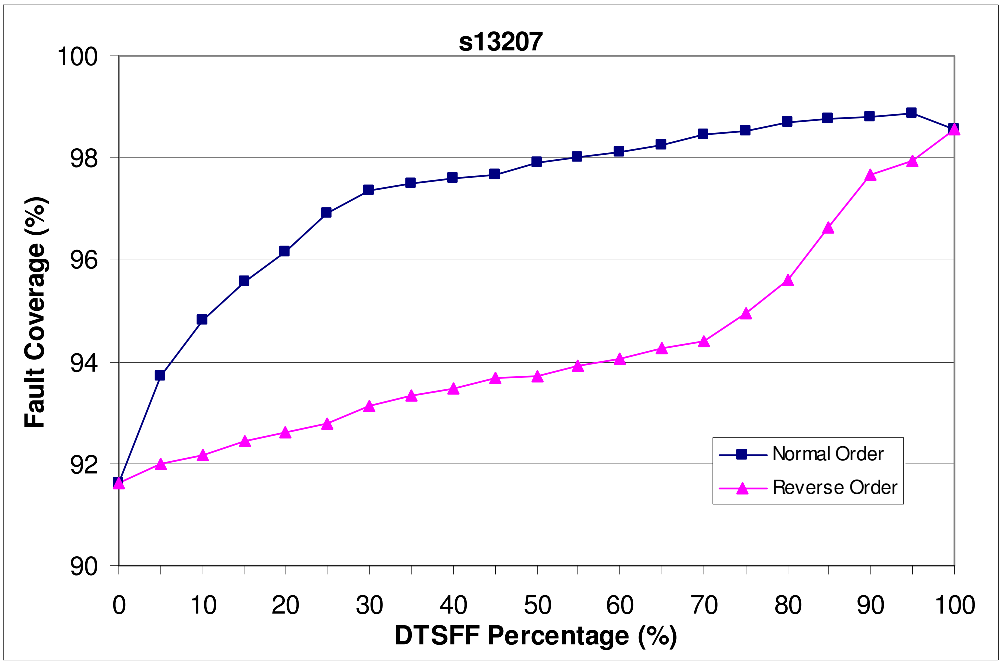

Figure 5-9(d): Benchmark S13207

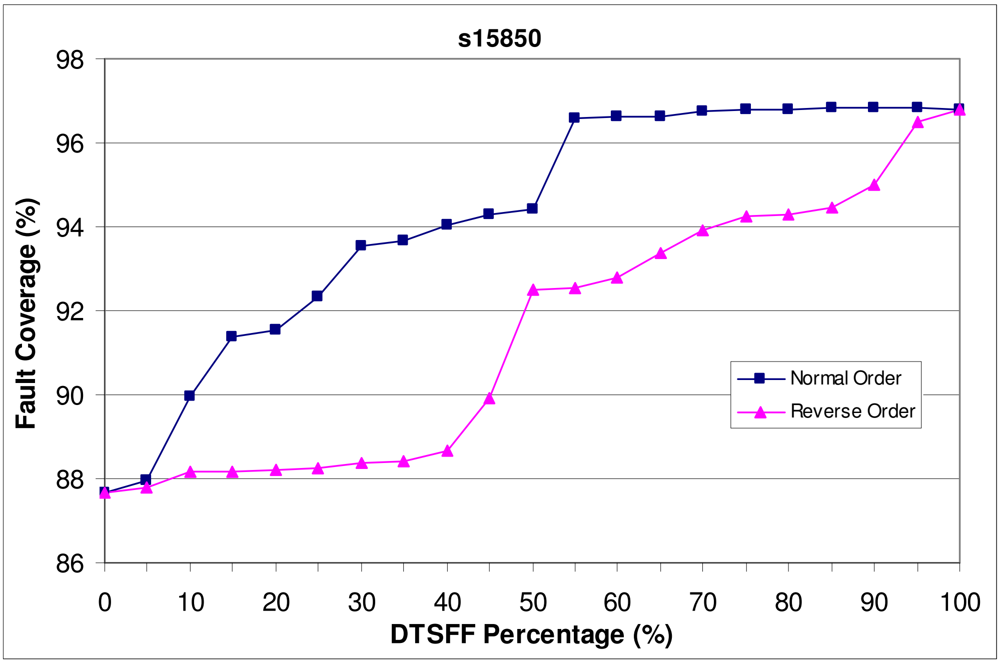

Figure 5-9(e): Benchmark S15850

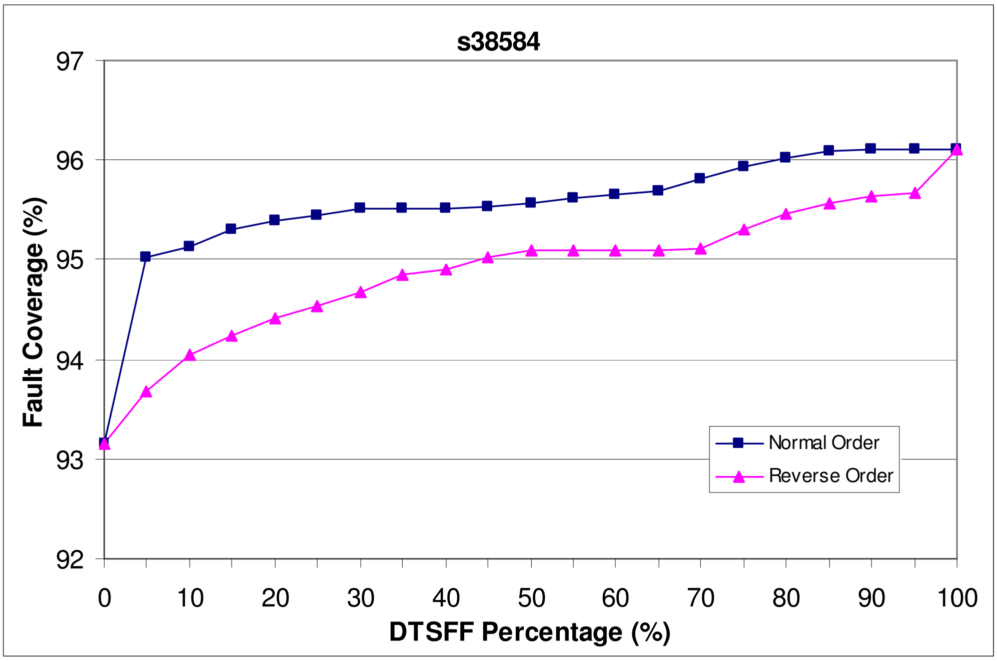

Figure 5-9(f): Benchmark S38584

Figure 5-9: DTSFF selection based on Monte Carlo simulation

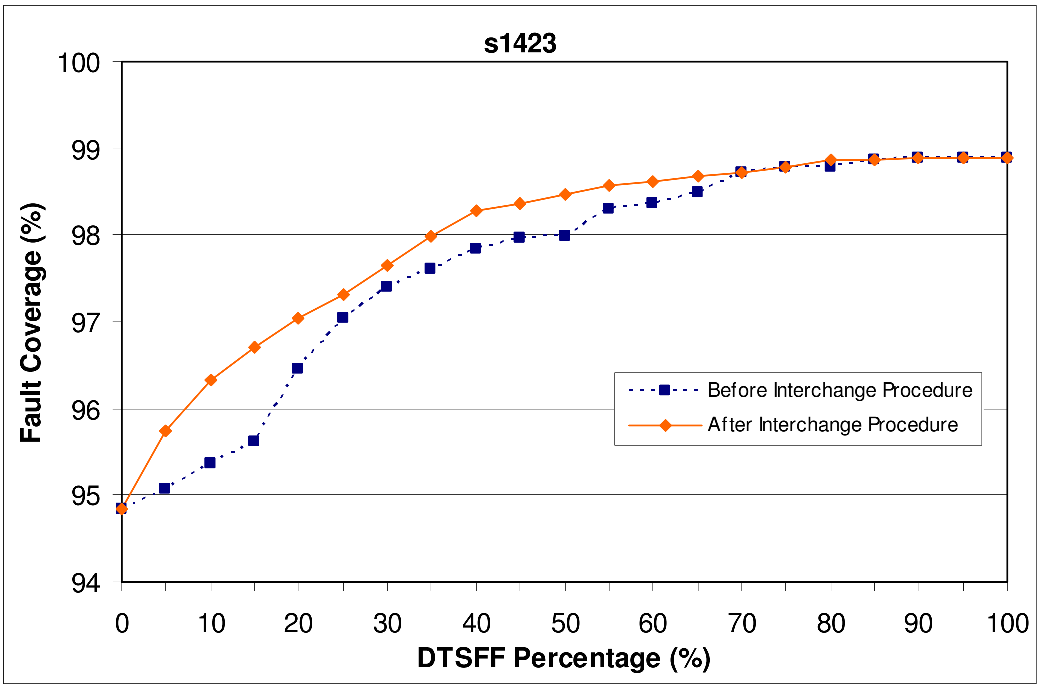

Figure 5-10(a): Benchmark S1423

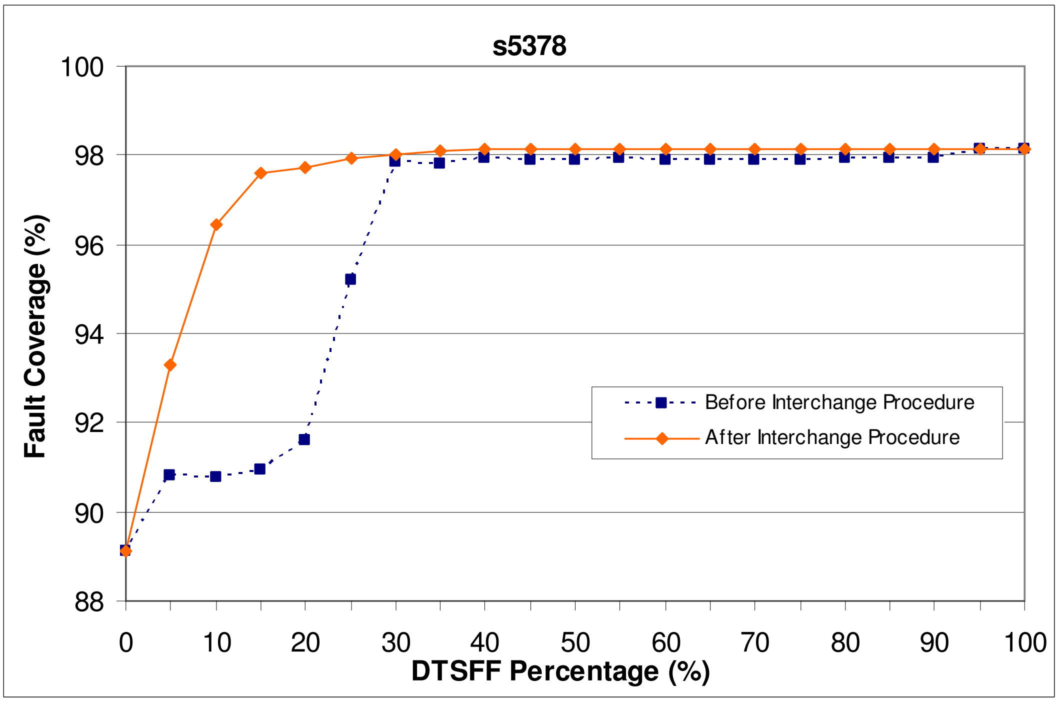

Figure 5-10(b): Benchmark S5378

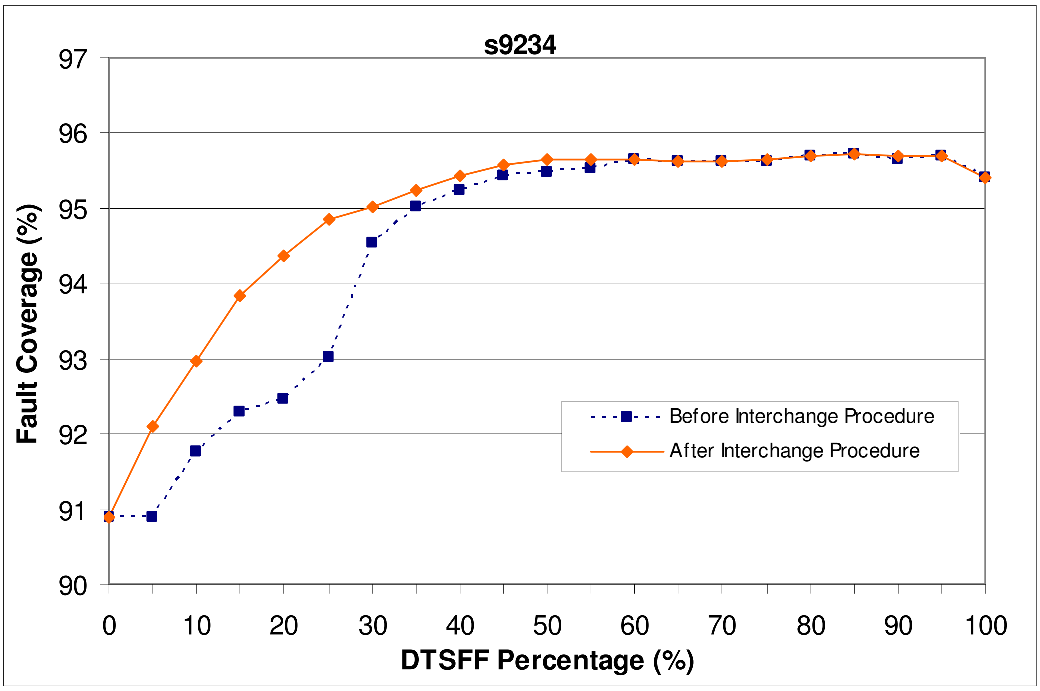

Figure 5-10(c): Benchmark S9234

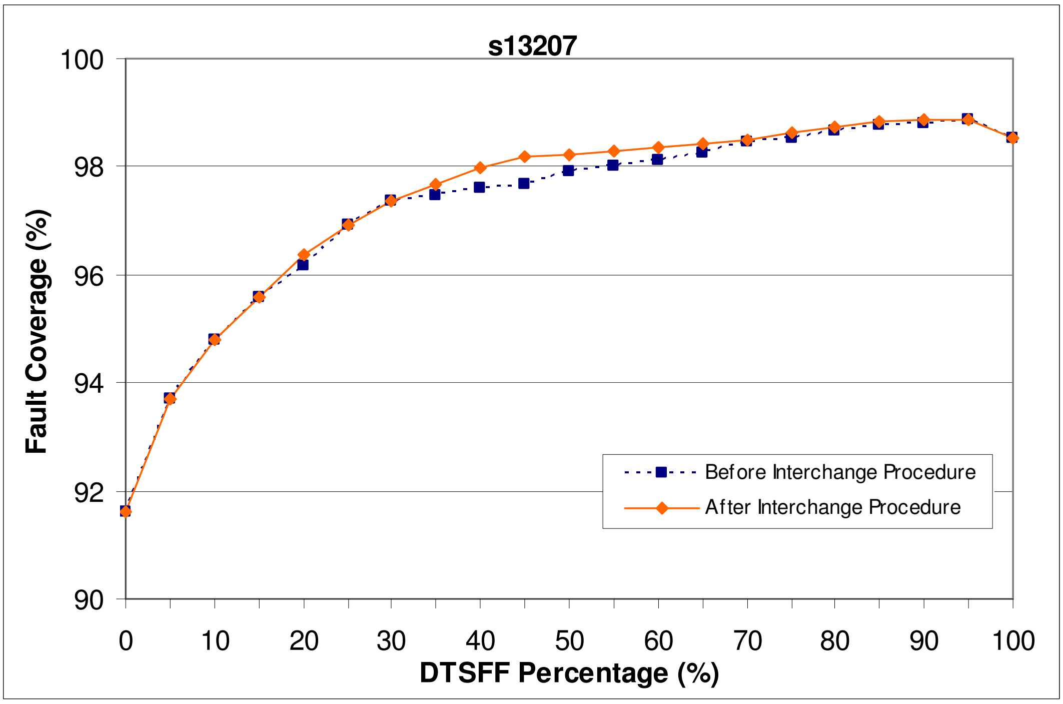

Figure 5-10(d): Benchmark S13207

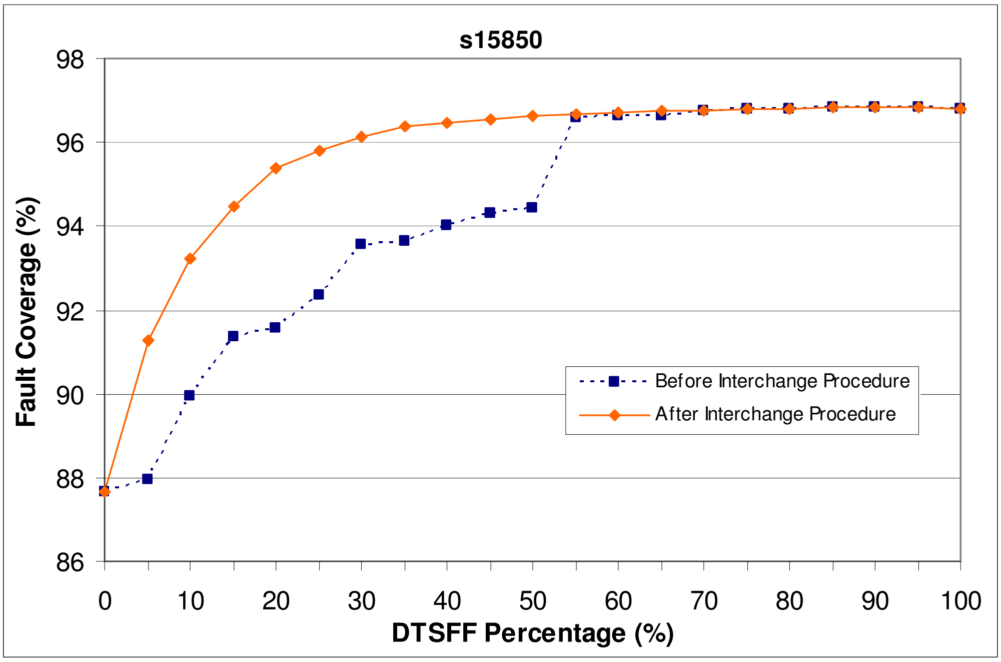

Figure 5-10(e): Benchmark S15850

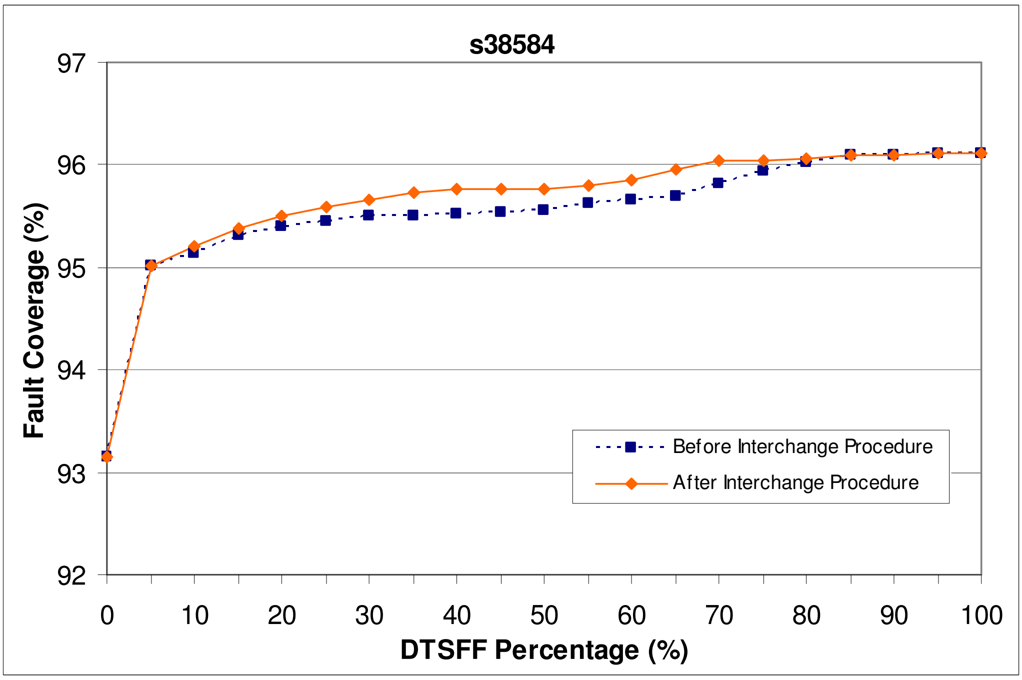

Figure 5-10(f): Benchmark S38584

Figure 5-10: DTSFF selection after Interchange Procedure

5.6 Enhanced Delay Test Scan Flip-flop

The motivation for a partial enhanced scan methodology is to achieve high TDF coverage at low cost. The goal is to employ only 10-30% enhanced scan flip-flops in the design. To keep down costs it is also critically important to avoid the need for a high speed scan enable control signal. Figure 2-5 presents a design from [16] of an enhanced scan flip-flop that can achieve this, albeit using two slow global scan control signals.

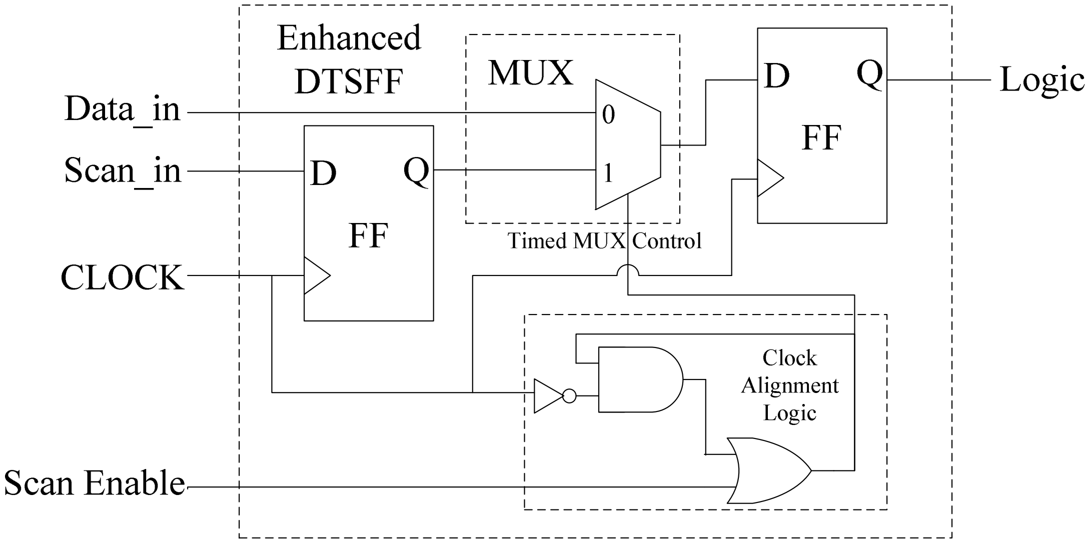

Figure 5-11: The structure of Enhanced DTSFF

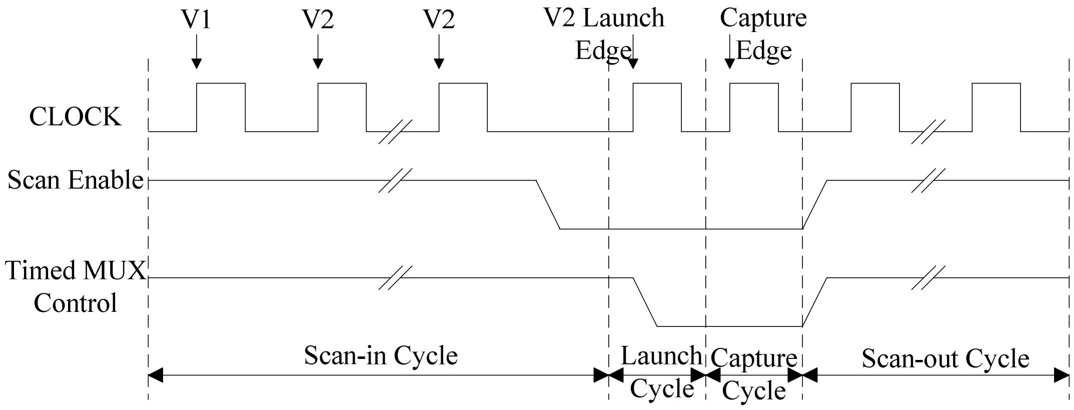

Figure 5-12: Timing waveform for Enhanced DTSFF based test

Figure 5-11 and Figure 5-12 presents an alternative design, based on our proposed [30, 31] Delay Test Scan Flip-Flop (DTSFF), that achieves the same capability. Observe in this design that the redundant flip-flop that holds the V2 bit to be shifted in and launched at a MUX input is actually located in the standard cell in front of the functional flip-flop. The advantage to the Enhanced DTSFF over the design in [16] is that it is comparable in hardware complexity to full enhanced scan design. The Enhanced DTSFF cell has identical pinout as compared to a standard multiplexer based scan flip-flop, and therefore it can be seamlessly integrated into industry standard design flows. Observe that the design cell in Figure 2-5 has an extra scan enable input.

5.7 Experimental Results for Partial Enhanced Scan Design

In order to study the effectiveness of the partial enhanced scan methodology presented in Section 5.2, we studied five of the larger ISCAS89 benchmark circuits, containing 74 to 638 flip-flops; the smaller circuits have too few flip-flops to provide meaningful results. For each design, we obtained signal probabilities at the outputs of the combinational block, assuming unbiased (0.5) probabilities at the inputs, using Monte Carlo simulation (up to 10,000 random vectors). These signal lines, which are the inputs to the flip-flops, were then ordered based on the computed statistical probabilities; the flip-flops receiving the most biased inputs, with probabilities furthest from 0.5, were the first to be replaced by enhanced scan flip-flops as we inserted an increasing number of enhanced scan flip-flops to configure the partial enhanced scan chains.

In our experiments, for 5% resolution of partial enhanced scan designs, all flip-flops will be assigned into 20 groups (G_1, G_2… G_{20}), i.e. first 5% biased flip-flop is assigned to G_1 and the next 5% biased flip-flop is assigned to G_2 and so on. We also define an empty group G_0 which contains zero flip-flops. In this manner, we obtain a number of partial enhanced scan designs for each benchmark circuit with 0, 5, 10, 15, 20 … 100% enhanced scan flip-flops in the scan chains. For example, the 15% enhanced scan design has the flip-flops in groups (G_0, G_1, G_2, G_3) replaced by enhanced scan flip-flops.

Scan based TDF tests were next generated for each of these partial enhanced scan designs using Synopsys DFT Compiler and TetraMax tools. For these five circuits, the TDF coverage FC_N (N=0, 1, 2 …20), which is defined as the fault coverage when flip-flops in groups (G_0, G_1…G_N) are implemented with enhanced scan flip-flops, are presented in Figure 5-13(a)-(e) for Monte Carlo based signal probability estimation (before the interchange procedure is implemented). Note that the fault coverage reported here does not count in undetectable faults. Also notice that partial enhanced scan circuits with the same flip-flop selection but with different scan chain orderings may yield different fault coverage, because our flip-flop selection method does not need any information from scan chain ordering.

Recall that, ideally, we seek a flip-flop ordering that allows fault coverage to increase monotonically with the percentage of enhanced scan flip-flops as shown in Figure 5-3. This appears best approximated by S13207 in the plots in Figure 5-13(d). Furthermore, the slope of the TDF coverage curve should be constantly decreasing as the percentage of enhanced scan flip-flops increase, as in Figure 5-3. This is not always the case for the controllability based results in Figure 5-13; for example in Figure 5-13(e) the slope increases sharply as we go from 5% to 10% enhanced scan flip-flops.

The interchange procedure attempts to achieve higher TDF coverage with fewer enhanced scan flip-flops by interchanging adjacent sets of (in the above example 5%) flip-flops until decreasing slopes are observed in the TDF coverage versus percentage enhanced flip-flops plots. The results after the Interchange Procedure have been plotted in Figure 5-14(a)-(c). (S9234 and S13207 are not subjected to the Interchange Procedure since the results for these two circuits are already satisfactory). Observe that the plots are now monotonically decreasing in slope and achieve the highest TDF coverage with the least number of enhanced scan flip-flops. Furthermore, they suggest that 60-90% of the coverage gain achievable by full enhanced scan designs over LOC can be obtained from 10-30% carefully selected enhanced scan flip-flops.

Note that we have quite arbitrarily picked 5% set size for the flip-flops to interchange. Using a larger, a granularity set size can reduce the computational effort on the part of the interchange procedure, but leads to somewhat lower TDF coverage for the same fraction of enhanced flip-flops, as shown for a 10% set size in Figure 5-14. This is because the “best” flip-flops cannot be individually selected from inside these 10% sets. Using a smaller set size improves the results, but can become computationally prohibitive.

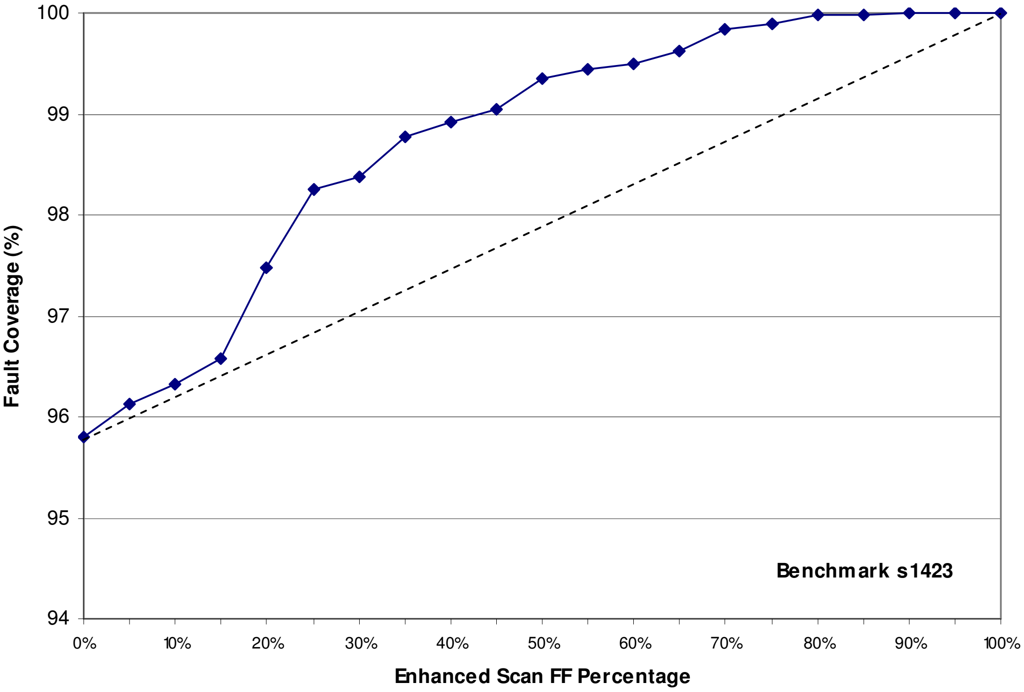

Figure 5-13(a): Benchmark s1423

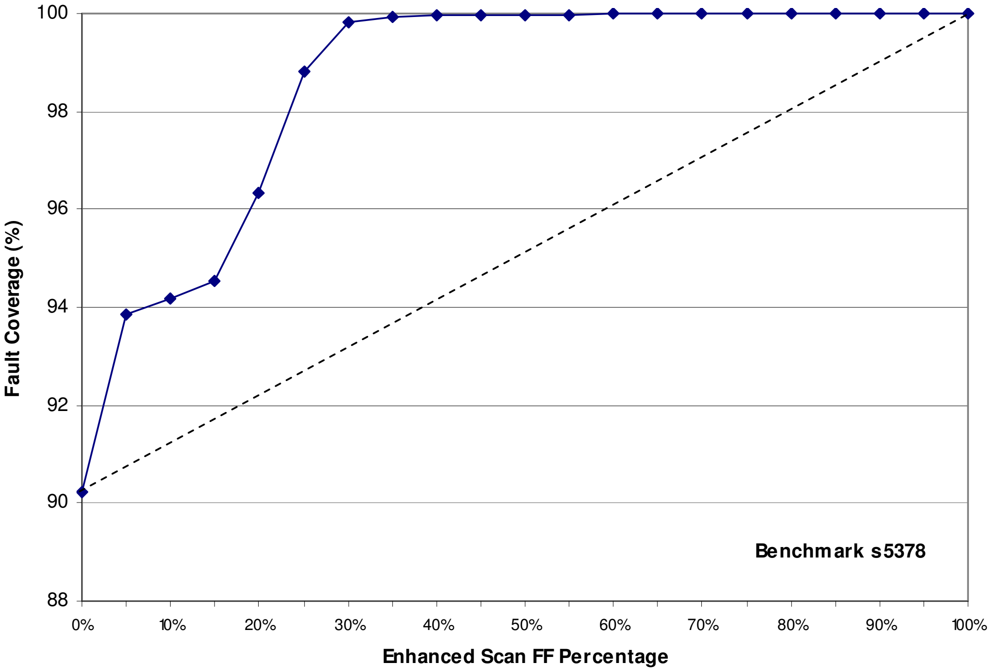

Figure 5-13(b): Benchmark s5378

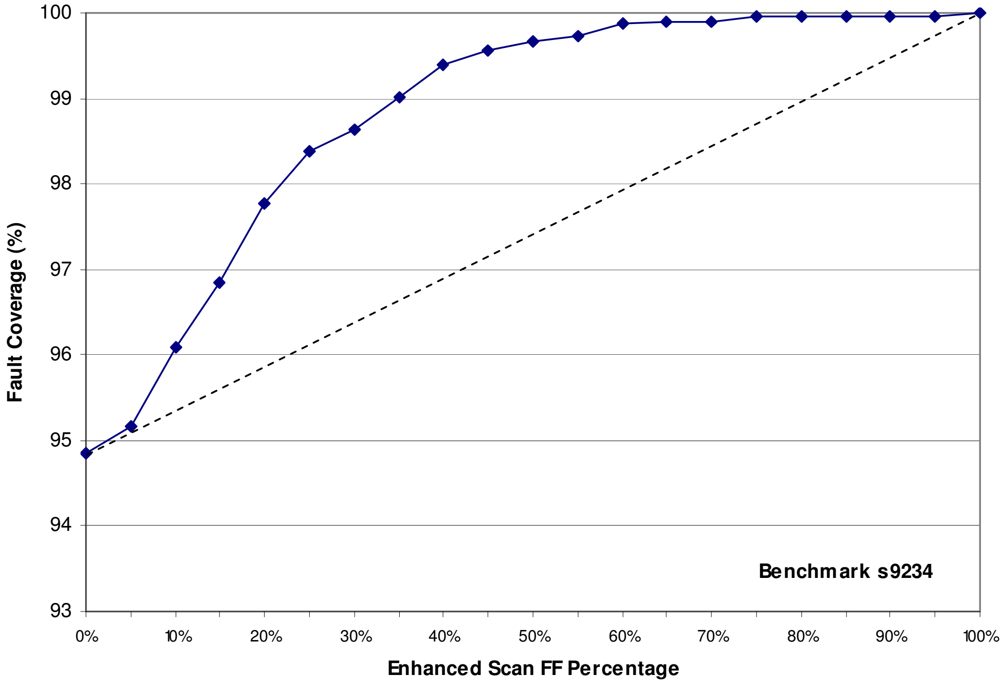

Figure 5-13(c): Benchmark s9234

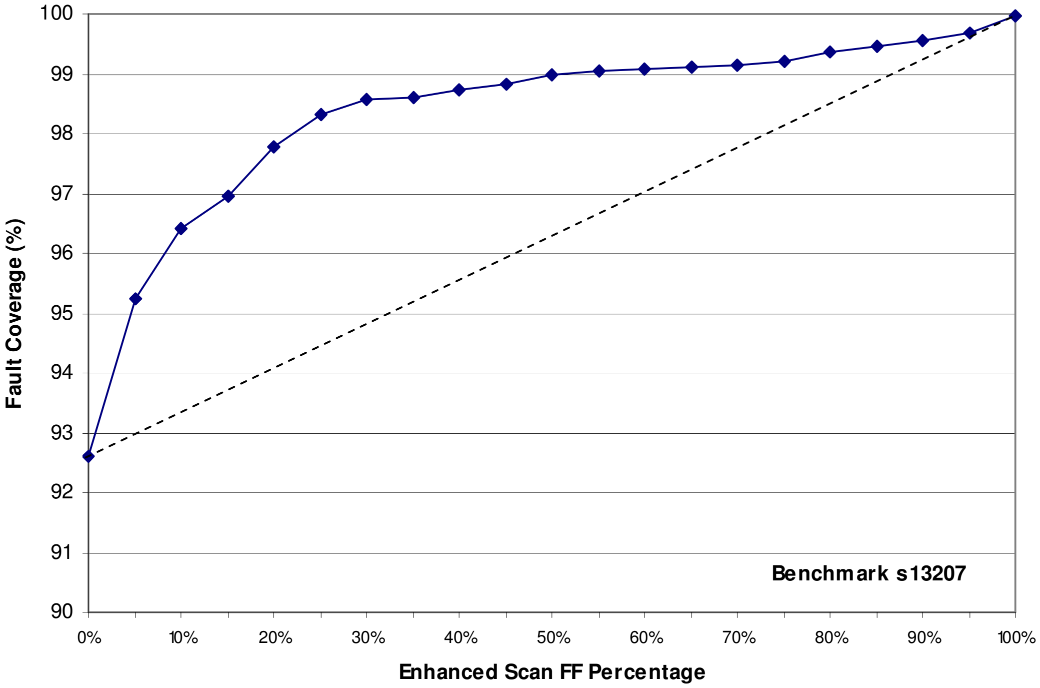

Figure 5-13(d): Benchmark s13207

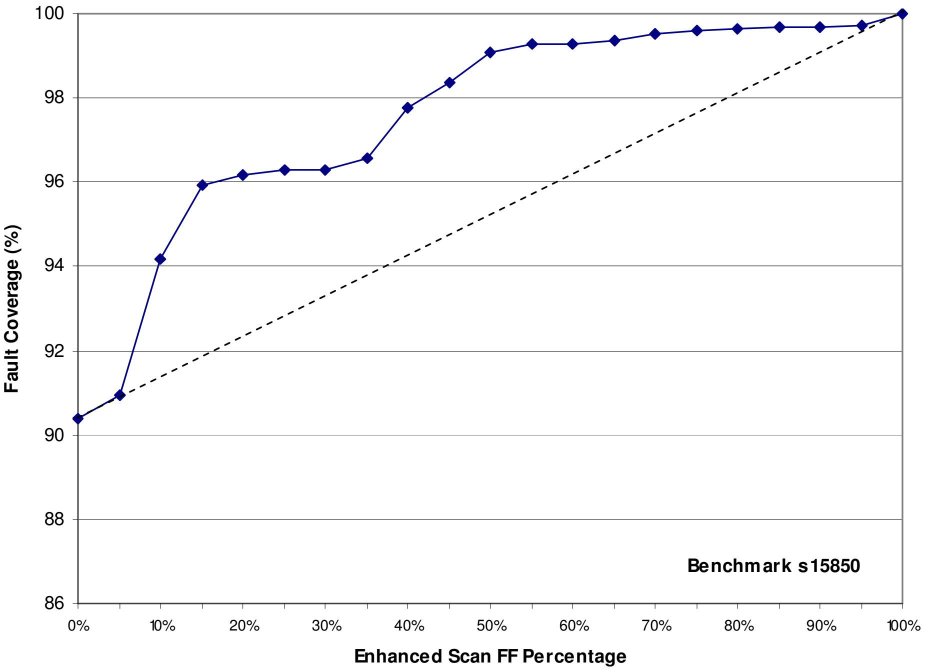

Figure 5-13(e): Benchmark s15850

Figure 5-13: Enhanced Scan Flip-flop selection based only on Monte Carlo simulation.

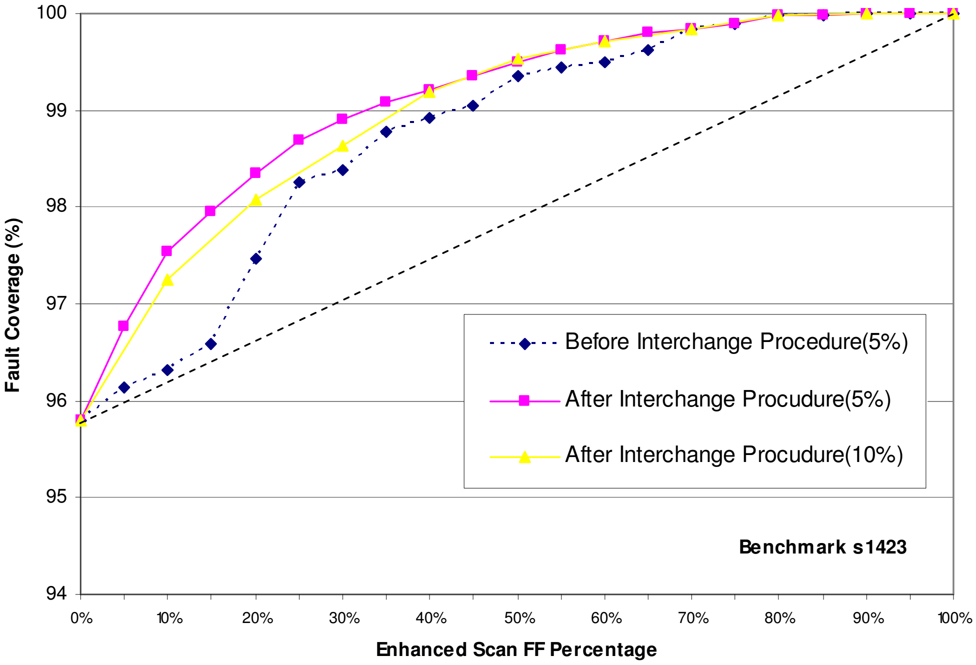

Figure 5-14(a): Benchmark s1423

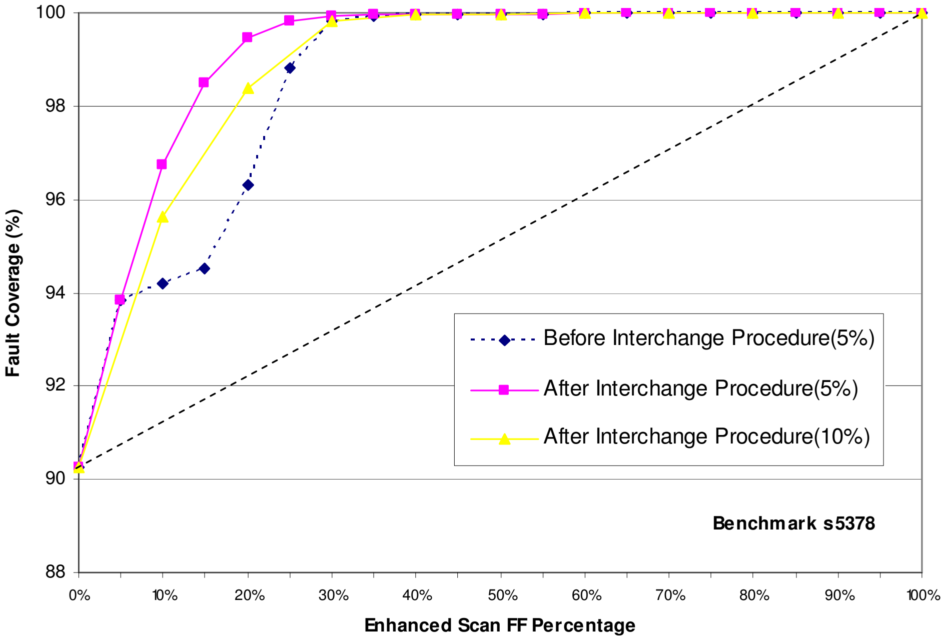

Figure 5-14(b): Benchmark s5378

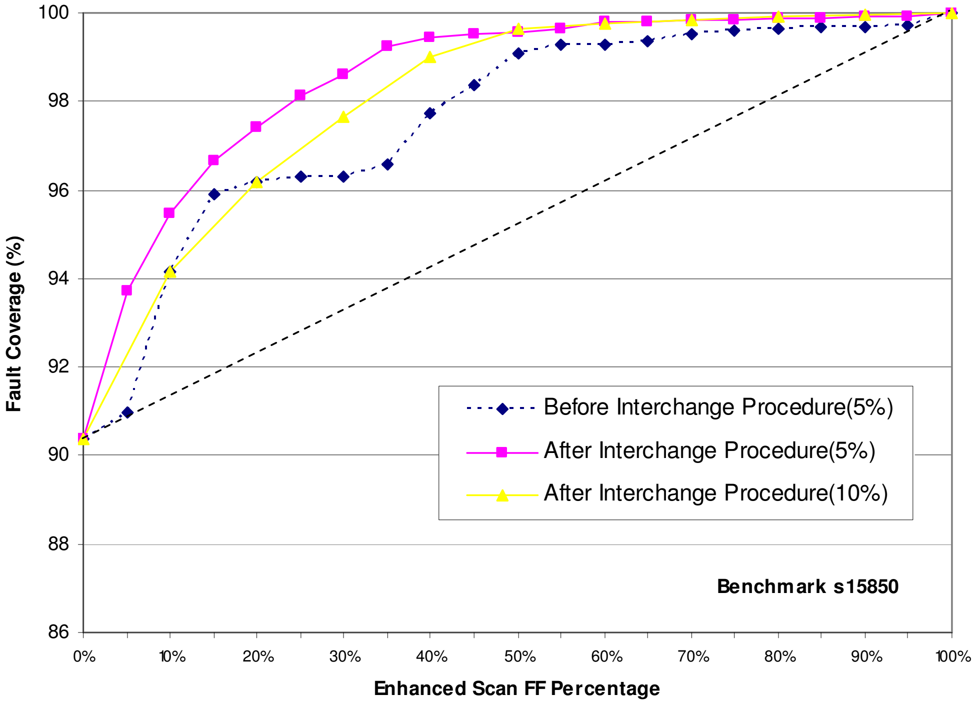

Figure 5-14(c): Benchmark s15850

Figure 5-14: Enhanced Scan Flip-flop selection after Interchange Procedure

CHAPTER 6 SILICON CALIBRATED TESTING

6.1 Background of Silicon Calibrated Delay Tests

Delay defects that degrade performance and cause timing related failures are emerging as a major problem in nanometer technologies. The physical causes of these defects include via voids, resistive opens in the interconnect, and gate oxide failures. Testing for such defects presents a serious challenge because signal delays for gates along a path in CMOS logic can vary greatly depending on the logic state and switching waveforms at both the on- and off-path inputs. For example, the rise time at the output of a 3-input NAND gate can vary by as much as 300% depending on how many of the inputs switch low to cause the low-to-high transition at the output. In high-speed circuits, with short clock periods, switching delays can also be affected by residual partial charges on circuit node capacitances from previous cycles. These factors gives rise to varying (input dependent) timing slacks at each gate input as the input signals arrive at different times. Delay effects of size smaller than this slack do not propagate to the gate output and remain undetected. Maximizing delay fault detection requires setting up worst case signal propagation conditions along each path. However, even these worst case timing tests still cannot detect defects whose effects are completely absorbed in the slack at some gate input.

Circuit timing is also greatly influenced by other factors such as layout and circuit electrical parameters actually realized from fabrication. Given this complex relationship between the applied input vectors and circuit timing, it is extremely difficult to develop test vectors that ensure worst case propagation of delay fault effects to the circuit outputs to ensure detection. (Theoretically, it is even possible that very different test patterns may be needed for a worst case test of the same path in different copies of the same die, given the up to 2X variation in performance observed today from parameter variations in cutting edge manufacturing processes.) The cost and complexity of such comprehensive delay test generation is prohibitive. The robust path delay fault model does generate logic level tests for switching delays along circuit paths that capture many, although not all [46] of these timing effects. Unfortunately, for many circuits, such robust tests do not exist for a large fraction of the paths. Furthermore, the structural limitations of scan designs make it impossible to apply many effective timing tests even for the limited cases where such vectors can be generated by ATPG programs. Recall that when a two pattern delay test, <V1, V2>, is applied in the scan environment, because of the serial structure of scan chains, vector V2 must either be a one bit shift of V1 (launch-on-shift), or the circuit's response to V1 (launch-on-capture).

Given the difficulties in generating good tests for scan based delay testing, industry is currently largely relying on tests generated using the basic transition delay fault (TDF) model. Here rising (and falling) transitions at each node are tested by ensuring that the two-pattern test <V1,V2> causes a rising (falling) transition at that node, and that V2 is a stuck-at 0 (stuck-at 1) test for the node. This TDF fault model assumes "gross" lumped delay faults at the node; detection is guaranteed only if the delay fault size exceeds a clock period. Smaller defects may be detected depending on the slacks on the paths sensitized to the output. Because TDF tests are closely related to stuck-at tests, test coverage (for gross delay faults) approaching the stuck-at test coverage for the circuit can be achieved if no restrictions are placed on choosing the V1 and V2 vectors. Under the structural limitations of scan, studies have shown [7] that TDF coverage between 75% and 95% can be achieved, with launch-on-shift performing somewhat better than launch-on-capture. However, application of launch-on-shift tests requires that the scan enable signal switch at-speed; this capability is not supported in most designs that employ multiplexer based scan flip-flops, except DTSFF design introduced in this dissertation.

Table 6-1: Delay defect size versus detection coverage (LOS tests) [23]

ISCAS89 Full Scan	Size of Fault List	Delay Fault Size (% of Tcritical)	Traditional Delay Test Coverage		Transition Test Coverage
			fmax	10% Timing margin	
S13207.1	26414	15%	3.08%	1.03%	89.9%
		25%	15.40%	3.61%	
S15850.1	31700	15%	6.94%	0.52%	94.8%
		25%	21.87%	6.77%	
S35932	71864	15%	1.07%	1.05%	86.7%
		25%	12.63%	1.14%	
S38584.1	77168	15%	1.36%	1.08%	88.8%
		25%	9.30%	1.44%	
S38417	76834	15%	1.24%	0.13%	93.6%
		25%	5.98%	1.40%	

Table 6-1 shows some results from a simulation study [23] on the effectiveness of scan based TDF tests in detecting delay faults of size smaller than a full clock period (gross delay). In this study, the effect a delay fault of a given size at a node was simulated by adding an additional delay at that node during timing simulation. This injected delay was set equal to the desired delay fault size. The entire TDF delay test set was simulated using a timing simulator, one test vector pair <V1, V2> at a time, for each injected fault. The fault was considered detected if it caused a signal to be delayed at a primary output beyond the rated clock time for any vector pair in the delay test set. The table shows results for the most optimistic case where the clock period is made equal to the delay of the circuit critical path (fmax). Also shown is a more realistic situation where the clock period includes a 10% timing margin beyond the critical path delay. The injected fault size is measured as a percentage of the critical path delay for that circuit.

Observe from Table 6-1 that even though the TDF coverage of the test sets is relatively high, actual coverage for delay defects of size 25% of the critical path is quite limited, even at the most aggressive clock speed (fmax). Such faults are virtually undetectable for the more realistic case where 10-20% timing margins are used to avoid failing good circuits due to performance variation from varying process parameters. It has been sometimes argued that small delay defects may not cause functional failures and are therefore benign and need not be detected; in fact, screening out such defects will lead to unnecessary yield loss. Note however that an average gate contributes 12.5% to the total delay of an eight level critical path. A delay fault that adds 25% extra path delay at some gate output increases the gate delay by 200%. Such an increase can rarely be brought about be a benign minor manufacturing flaw. For example, the resistance in a via must increase 1000-10,000X from the typical 0.1 ohm value to cause such a delay. The greatly increased current density in the minimal contact formed by such a defective via will very likely lead to reliability failure in the field. In many cases, such defects may also be actual test escapes that fail for untested inputs immediately when deployed in the system, since TDF delay tests make no attempt to exercise worst case signal propagation conditions. There is a growing consensus among researchers [23, 25, 47, 48] that such “small” delay defects must be detected and screened out to achieve satisfactory defect levels (DPM) and failure in time (FIT) rates in high end integrated circuits.

A solution to this small delay defect detection problem is to look for delays in the short paths within the slack interval by capturing the test response during scan based testing using a clock faster than the nominal rated clock frequency. Figure 6-1 shows the switching signals for some test vector pair at four outputs of a combinational logic block feeding scan flip-flops. The small delay defect shown clearly cannot be observed at the nominal clock time. But if the outputs are also captured at a faster sample clock T1, the defect can potentially be detected. However, this requires accurate timing information for each output signal so that a decision can be made regarding its expected logic state at each (faster than nominal) sample clock applied during the test. Repeating the test multiple times can also significantly increase test application time.

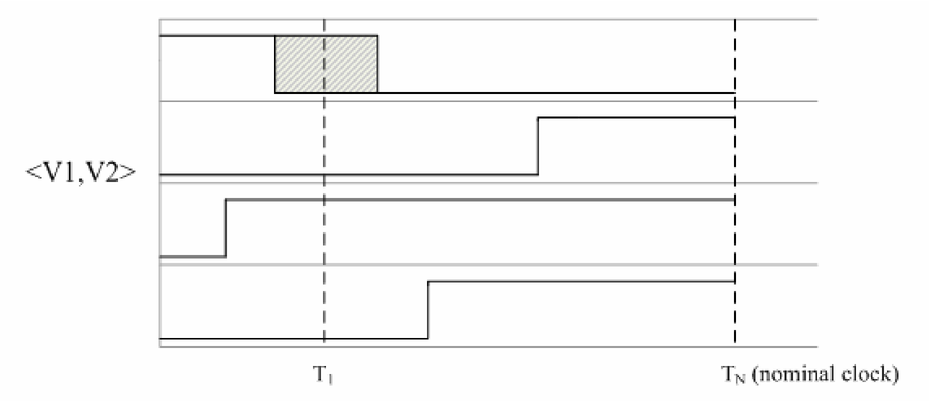

Figure 6-1: Delay defect detection in the slack

While the concept of sampling faster than rated clock for detecting timing defects has been in the literature for decades [49], simulation tools to support such testing are now becoming available [48, 50]. For example, Cadence's Encounter Test System supports "True-Time" delay test for short paths. The ATPG here is based on the transition fault model, and generates delay tests for each node that are sensitized along the shortest path to a primary output. The timing simulator then computes the path delays for each test. These delays are binned and used to select the sample clock rates. Tests are applied first using the shortest test clock; outputs whose delay exceeds the clock are masked off and not observed during that test. The detected faults are dropped, and then tests are applied with a longer clock, again masking off slower outputs. The process is repeated until the longest paths are tested with no masking. This approach allows different tests to be used for each sample clock depending on the faults to be tested at that clock rate. The result is that the total number of tests applied is only ~2X times the basic TDF test set, which is considerably less overhead than if the complete test set was repeated at each sample clock. Published results [47] report better than a 2X improvement in DPM levels in production even with relatively modest testing of the short paths.

In addition to the concerns about defect coverage discussed above, there are other real practical issues associated with the application of scan based delay testing that limit the accuracy of such tests. Many of these have to do with the fact that a scan based delay test is an "out of normal mode" test. Scan based delay tests can cause switching activity significantly in excess of normal functional operation in local areas of the chip, leading to power supply droop and ground bounce. Such noise on the power grid can slow circuitry during test, resulting in a false indication of timing faults [51, 52]. Abnormal switching activity can also aggravate cross talk. Finally, different chip temperature profiles during test can result in significantly different circuit timing at test when compared to normal operation; a 40C temperature change can cause as much as a 20% performance change variation.

These test conditions are virtually impossible to simulate accurately enough for tight timing testing of the short paths as described in the previous section. In practice they must be guardbanded with an appropriate timing margin during test, which can further compromise the effectiveness of scan based delay tests.

6.2 Silicon Calibrated Delay Tests

Architectural restrictions of scan greatly limit the effectiveness of traditional scan based delay testing. It has been recently shown that enhancing scan tests by also testing for delays on short paths using fast clocks can significantly lower DPM. However, obtaining accurate timing for such tests from simulations that accurately account for the effects of process parameter variations, as well as power supply noise and crosstalk from the excessive switching activity of scan tests, is extremely difficult. We show that learning signal timing information on silicon to "calibrate" such tests is much more accurate and effective. However, such an approach requires that the outputs of the applied tests be hazard-free to avoid learning incorrect timing due to glitches at the output. Simulation results presented later indicate that such output hazard-free tests can be obtained with an average coverage only about 10 % below the transition delay fault coverage for both launch-on-shift (LOS) and launch-on-capture (LOC) modes.

The difficulty in accurately predicting circuit timing information for use in tight delay testing of individual circuit paths arises from the observed variations in process related circuit parameters among the individual circuits that are manufactured. In nanometer technologies, extremely small changes in the calibration, positioning and alignment of the manufacturing equipment can sufficiently impact circuit features so as to cause a substantial variation in device performance. Unintended timing variations of 50% or more are currently observed from the same process. Furthermore, these manufacturing variations can impact different circuit components in different ways, giving rise to significant variations in the relative timing between signals as well. This makes it virtually impossible to accurately predict the expected delays along the different paths in each individual circuit under test (CUT). Substantial timing margins must be added when using delay predictions from timing simulators to avoid failing good die. These margins limit the effectiveness of the tests in detecting small delay faults.

The key idea behind the proposed Silicon Calibrated Delay Tests is that the circuit switching timing information needed for delay testing be "learned" from "golden" circuits on silicon, rather than being obtained through less accurate and much more expensive timing simulation. This can be done on the tester as follows. For each delay test vector pair <V1, V2> in the test set, we propose measuring the switching time for each signal transition by repeated sampling of the outputs at multiple fast clocks. This is illustrated in Figure 6-2. There will obviously be some quantization error in the measurement, but this can be made as small as desired by reducing the sampling time. This "learning" on silicon can provide the same timing information obtained from a timing simulator to develop a delay test set for short paths, but with much greater accuracy.

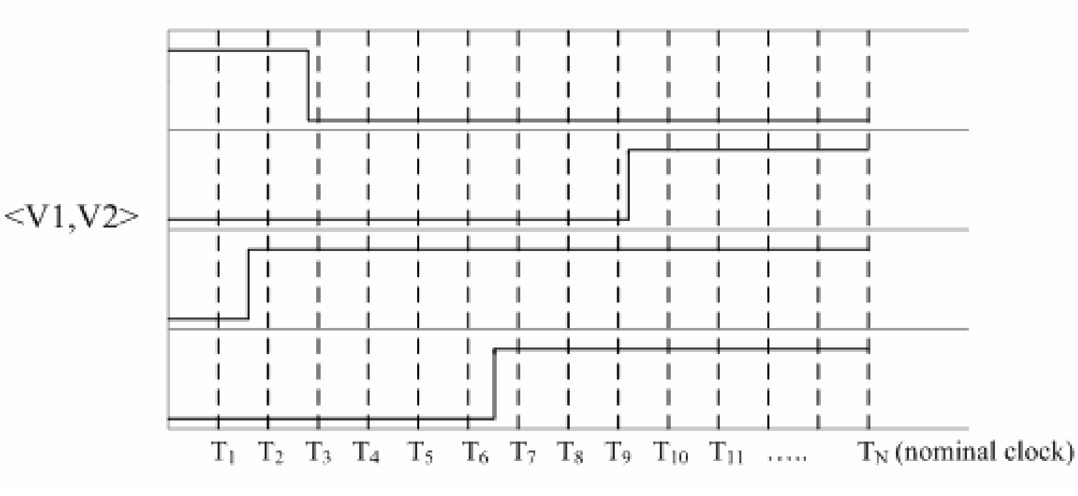

Figure 6-2: "Learning" signal transition timing by repeated sampling

Process related circuit parameters are generally observed to track most closely for matched devices within a die, and within local regions of a wafer. This is because identical circuits that are located in physical proximity on the same silicon wafer undergo very similar, if not identical, manufacturing conditions. Variations increase across wafers and the different wafers in a lot, and is most pronounced over multiple lots from different processing runs. This correlation of device parameters in local regions of a wafer allows timing information "learned" for one representative golden die in the region to be used for all the other die in that region. Furthermore, the number of die in the "local region" represented by the golden die can be varied depending on the required timing accuracy of the delay test. For example, "learning" timing from a golden die for each local region comprising 24 die will allow timing tests to run with smaller timing margins than, say, if a single golden die was used to obtain timing for the entire wafer. While tests with smaller timing margins can detect smaller delay defects, the increased delay test sensitivity must be traded off against the increased costs of learning timing from more die on the wafer.

Silicon Calibrated Delay Tests offer another vitally important advantage over timing simulation. The two-vector delay test patterns used to "learn" circuit timing from a representative golden die are exactly the same test vectors that are applied to nearby die in the local region of the wafer to test for delay faults. Thus timing effects of ground bounce, power supply droop, cross talk, etc. associated with the application of that test vector pair will all be captured in the "learning" process during the test calibration phase. The expected response to the test, will therefore accurately reflect the timing impact of these effects; details virtually impossible to accurately model in simulation. If care is taken to ensure that identical electrical and thermal conditions are maintained at test as during the "learning" phase, the delay test will be able to accurately detect any significant deviation in performance of the CUT from the golden die. Note that it is immaterial whether this difference is caused by a delay defect along some signal path, or by some other defect in the CUT, such as a defect in the power rail; in both cases the circuit is defective.

6.3 Output Hazard-free Transition Delay Tests

Learning the circuit timing response on the tester imposes an important condition on the test set: all circuit responses to be learned must be guaranteed to be hazard-free. This is necessary to avoid learning incorrect timing for an output change due to the existence of hazards, because incorrect timing information will make a good chip fail the test.

Figure 6-3 illustrates this point, showing just one example of how a hazard can confuse a timing test. Observe in the figure that the signal shown reaches its final low logic state at sample time T_f, but because of the timing of the sampling strobe signals, the final transition is missed and an earlier switching time for the signal, Ts is learned due to the hazard. Now when this output is tested for delay defects in other circuits, based on this timing learned on the golden circuit, a relatively small and quite acceptable variation in circuit timing in the CUT (due to parameter variations) can result in a different logic value being captured at time T_f during the test as shown. This suggests a significant delay defect in the CUT, although none exists.

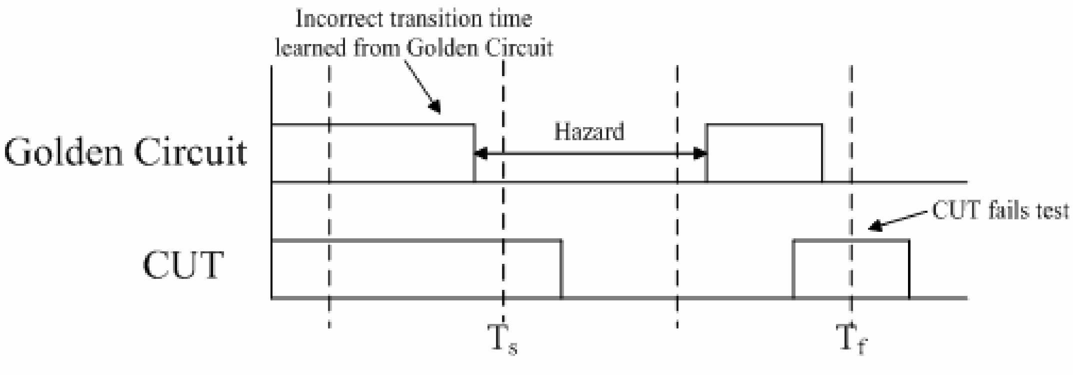

Figure 6-3: How a hazard can confuse a timing test

Hazards are quite common in CMOS circuits, and today's circuits contain thousands, even millions of scan flip-flops. The possibility that circuit timing is incorrectly sampled and learned due to a hazard in just one of these many flip-flops for some test vector pair in the test set is significant. As a result, hazards can lead to an unacceptably high number of false delay defect indications if they cause incorrect timing to be observed and recorded during the learning phase. We propose to overcome this problem by ensuring that only outputs that are guaranteed to be hazard-free are used for the purpose of learning circuit timing from silicon.

Note that we still work with TDF tests, but with the additional constraint that a delay fault is considered detected at an output only if that output switches without any possibility of a hazard. For each two-pattern test applied, typically only a subset of outputs can be guaranteed to be hazard-free. Only these outputs are useful for learning circuit timing, and then, based on this learned timing, for delay fault detection in other die.

6.4 Fault Coverage for Output Hazard-free Delay Tests

Output hazard-free delay tests have also been investigated in [48], not for learning circuit timing, but for the reliable detection of small delay defects in the slack (called “fine” delay defects in [48]). Here the hazard-free tests were obtained by screening a normal ATPG generated TDF test set for hazards at the output using a fast timing simulator. For the two circuits presented, 40% and 60% TDF fault coverage for the screened test sets was reported.

Since our silicon calibrated test methodology critically depends on output hazard-free tests, it is important to investigate a larger set of circuits to study what range of output hazard-free fault coverage can be obtained in general. An accurate study requires detailed timing simulation of the TDF test sets for the target benchmark circuits with electrical parameters extracted from actual layouts. However, here our goal is not to actually develop output hazard-free TDF test sets, but only to estimate the expected coverage for such tests, so as to see if the proposed method is viable. We therefore use logic simulation and simple gate level delay models to estimate this coverage, making some simple assumptions regarding the generation and propagation of hazards.

Table 6-2: Estimated coverage (%) of output hazard-free TDF tests (LOS)

Circuit	Hazard-Free TDF (Lower Bound)	Mode 3	Mode 2	Mode 1	Mode 0	Unrestricted TDF
S208	60.10	72.84	78.37	79.33	79.33	88.94
S298	60.74	62.92	62.92	64.60	76.51	84.23
S344	82.70	83.43	85.76	86.63	87.50	94.04
S349	81.66	82.38	84.67	85.53	86.39	93.41
S382	70.94	73.56	75.79	78.01	83.38	90.71
S386	51.17	54.79	57.51	58.16	68.26	79.40
S400	68.62	70.88	73.62	76.38	82.13	89.50
S420	60.00	75.83	77.98	78.57	78.81	87.74
S444	59.57	62.39	63.40	70.72	77.70	86.60
S510	57.84	63.14	67.06	74.41	81.37	90.39
S526	65.78	67.11	68.25	70.53	80.13	87.45
S526n	65.87	67.21	68.35	70.63	80.04	87.64
S641	87.52	91.13	92.23	92.54	93.80	96.70
S713	53.09	58.49	59.75	74.96	76.72	90.81
S820	36.83	38.41	40.30	44.09	57.99	78.17
S832	36.06	37.44	39.30	42.85	56.97	77.04
S953	71.09	81.11	83.16	84.42	87.51	91.03

Circuit	Hazard-Free TDF (Lower Bound)	Mode 3	Mode 2	Mode 1	Mode 0	Unrestricted TDF
S1196	57.82	67.02	69.06	71.28	74.37	85.54
S1238	57.31	65.06	66.60	68.86	71.61	81.99
S1423	76.88	82.64	82.78	84.50	91.22	95.99
S1488	45.43	51.31	53.53	56.79	63.31	79.67
S1494	44.48	50.50	52.64	55.76	62.45	79.08
S5378	65.45	68.75	70.98	75.27	89.59	93.05
S9234	64.71	70.01	71.26	76.60	80.04	88.28
S13207	78.24	80.66	81.20	84.31	87.49	94.04
S15850	75.66	78.45	78.79	81.17	82.15	90.66
Ave.	62.91	67.59	69.43	72.57	78.34	87.77

Table 6-2 and Table 6-3 present simulation results for the estimated coverage achievable by output hazard-free TDF tests. These were generated by simulating 100,000 random scan vectors for both launch-on-shift and launch-on-capture modes. Column 2 shows lower bound results evaluated under strict logical conditions that eliminate all possibility of hazards in the outputs observed. The last column, column 7 shows unrestricted TDF coverage results for the circuit. The other columns are based on some timing assumptions discussed below.

Table 6-3: Estimated coverage (%) of output hazard-free TDF tests (LOC)

Circuit	Hazard-Free TDF (Lower Bound)	Mode 3	Mode 2	Mode 1	Mode 0	Unrestricted TDF
S208	36.78	48.08	50.24	50.72	50.72	57.45
S298	54.19	55.03	57.55	59.90	70.30	81.21
S344	72.97	77.47	82.27	84.59	86.19	93.75
S349	71.92	76.36	81.09	83.38	84.96	93.12
S382	52.49	53.27	54.45	56.81	68.85	76.83
S386	34.97	36.92	37.82	39.90	41.71	52.72
S400	50.88	51.63	52.88	55.25	68.37	75.63
S420	28.21	53.81	54.88	55.48	55.48	64.76
S444	42.12	43.47	44.03	53.49	58.11	75.11
S510	55.00	61.18	63.92	70.10	75.49	89.41
S526	37.26	37.74	40.02	43.54	53.23	64.35
S526n	37.26	37.74	40.02	43.54	53.33	64.35
S641	80.46	85.48	86.34	86.50	88.07	91.60
S713	46.91	54.63	56.73	68.09	70.48	85.13
S820	32.87	33.84	34.94	36.95	43.48	51.83
S832	32.27	33.23	34.38	36.36	42.91	51.08
S953	70.46	78.70	80.90	82.06	85.94	91.55
S1196	52.55	61.04	64.67	66.72	69.73	81.65
S1238	50.85	60.10	63.29	65.79	68.30	79.08
S1423	56.01	61.35	61.74	66.02	76.88	87.10

Circuit	Hazard-Free TDF (Lower Bound)	Mode 3	Mode 2	Mode 1	Mode 0	Unrestricted TDF
S1488	42.07	48.59	51.34	55.95	62.33	87.40
S1494	41.67	47.86	50.77	55.09	61.55	86.98
S5378	73.64	76.61	86.65	82.35	86.65	89.61
S9234	37.33	44.03	45.49	52.41	59.00	74.71
S13207	53.34	56.42	57.43	63.82	70.01	82.38
S15850	51.91	58.60	59.17	62.12	64.23	78.82
Ave.	49.86	55.12	57.42	60.65	66.01	77.22

Recall that a hazard appears at an output if it can be potentially generated at some gate for the applied two-pattern test, and propagated (sensitized) to the output. In general, a hazard is generated at a gate output if appropriate inputs arrive at the gate with a skew greater than the inertial gate delay. It is reasonable to assume that the delay in a signal generally depends on the number of logic gates in the path. Therefore, two signals at different levels in the circuit cannot cause a hazard that requires the signal on a longer path to switch before that on a shorter path. Figure 6-4 illustrates this with an example. To see a hazard at the output, the top input at the output gate must arrive at least an inertial gate delay before the lower input, which is generally unlikely.

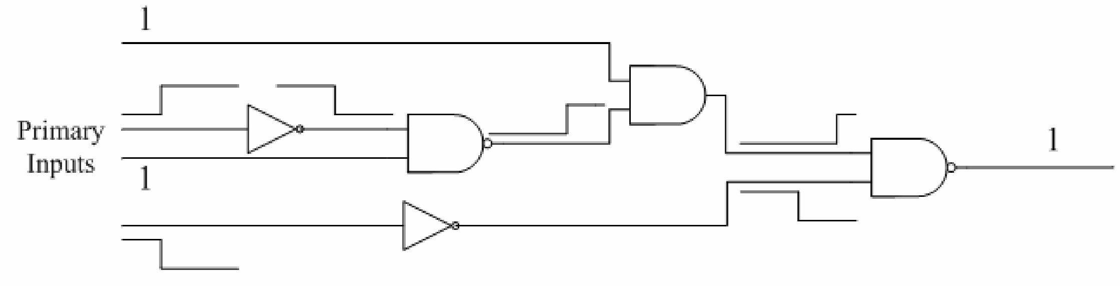

Figure 6-4: Hazard masking due to paths of differing lengths

Based on the above, it is relatively safe to assume that signal paths that differ by two or more logic levels will not cause hazards that require the longer path to switch first. Column 3, titled Mode 3 in Table 6-2 and Table 6-3, shows the output hazard-free coverage attainable if it can be assumed that signals that differ in three or more levels of logic cannot cause a hazard because of the longer path switching earlier. The Mode 2 column similarly assumes that paths differing in two or more logic levels cannot cause a hazard. In Table 6-2 and Table 6-3 however, we also show coverage results where no hazards are assumed when inputs have paths that differ by only one level (Mode 1), and even when paths are of equal length (Mode 0). These have been included because we are trying to estimate the possible coverage achievable by output hazard-free tests, and not develop such tests. Even if signals arrive at a gate at about the same time along paths of equal length, because it takes a timing skew of at least the gate inertial delay to cause a hazard at the output, in most cases no hazards will be generated. Note that it is indeed possible that because not all gates in CMOS exhibit equal gate delays, hazards may be generated in some cases, making us overestimate the achievable output hazard-free coverage. However, such timing disparities between actual signal delays and logic levels can also result in valid hazard-free tests in other cases where we predict hazards under our equal date delay assumption. Statistically, over a large number of tests, the effects of such timing disparities over the equal gate delay assumption should balance out. Thus if output hazard-free TDF tests are generated using careful timing simulation, for example using wireload model [53], it should be possible to achieve a defect coverage close to the most optimistic estimates (Mode 0) in the tables. This suggests that in general, output hazard-free TDF coverage only about 10% below the unconstrained TDF coverage can be achieved, both for launch-on-shift and launch-on-capture test modes.

6.5 Practical Implementation of Silicon Calibrated Delay Tests

Our proposed methodology assumes that some default information on the expected switching delays for each output (including outputs with hazards) is available from timing simulation to begin with. (Alternatively, it is also possible to initially specify a worst case switching delay for all signals based on the rated clock speed.) Depending on the projected accuracy of the simulations, sufficient timing margins can be added to these estimates to safely test the timing of circuit paths, binned in groups of approximately equal length. When a group of similar length paths is tested for delays using an appropriate fast clock, the response bits for longer paths are masked out in evaluating the response. The introduction of timing margins is to allow for simulation inaccuracies, parameter variations, circuit noise etc. and ensure that these factors do not cause good circuits to fail the timing test. However, these margins also allow defects to escape detection, as illustrated earlier in Table 6-1. The purpose of learning timing from silicon is to tighten up these margins significantly, wherever possible, so that small delay defects are detected by the test. Hence the name: Silicon Calibrated Delay Testing. Where timing cannot be learned, for example in the case of outputs with potential hazards, simulation based worst case timing values can still be used to detect delay faults, albeit at a considerably reduced detection sensitivity.

The fault coverage studies of the previous section suggest that output hazard-free TDF tests can potentially be generated with coverage only about 10% lower than that for unrestricted scan based delay tests. This indicates that most delay defects can in fact be tested using tight timing learned from silicon.

The one key question that we have not addressed so far is how known good "golden" die are identified on the wafer to allow the learning of timing information. This is not difficult if one recognizes that a high quality stuck-at test screens out the vast majority of the defective die on the wafer. Defect rates for die that fail only timing tests following high quality stuck-at fault screening are rarely more than a few thousand DPM (less than 1%). While this number is unacceptably high compared to the < 500 DPM targets for commercial parts, (making delay testing essential,) the probability of a delay-only fault in a typical die is in fact small. We propose to learn timing information with respect to the output response to the hazard-free TDF test set from two matched adjacent die from a representative location in the wafer region of interest. Figure 6-5 shows this conceptually. These die are first extensively tested to ensure that they pass all stuck-at tests, and any (less sensitive) timing tests that may be available at this stage. The likelihood that any one of the two dies contains a delay defect after this initial screening is small (safely less than 1%). We next compare the "learned" timing (for each transition) from the two die to ensure that they are consistent, i.e. within the acceptable inter-die timing variation for matched die typically observed for the process. Even in the rare case that a delay defect exists and the "learned" timing from the two matched die is at variance, such a defect will typically impact timing on only a few circuit paths, for a few test inputs. Most other paths will still match in timing. For such cases we drop the mismatched timing information learned for the test and instead use the more conservative worst case default value available from simulation. In practice, this will generally impact only a few output bits in a test response comprising millions of response bits. Recall that less than 1% of golden die pairs are likely to experience such defects in the first place; the impact on overall defect coverage will be negligible. (Including a third golden die in the group to resolve mismatches with a vote is another obvious solution, but the extra cost of learning timing on 50% additional golden die is not justified by the minimal improvement in defect coverage.)

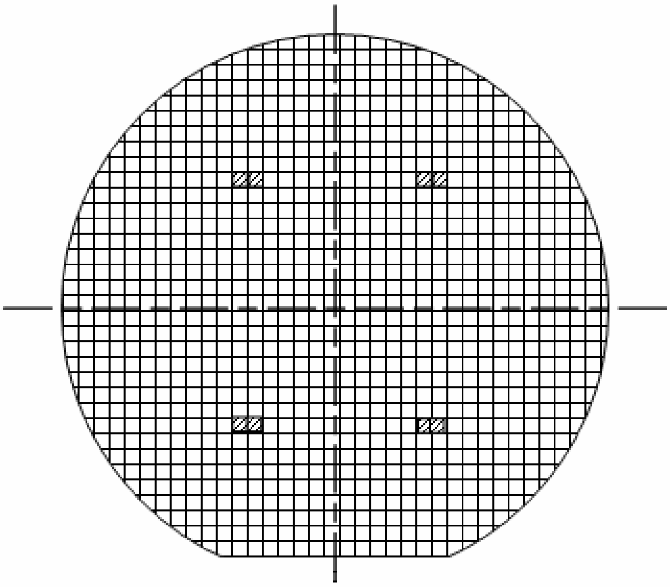

Figure 6-5: Golden die in local regions of the wafer

Identifying matched regions on wafers whose timing can be best captured by representative golden die is a problem that can be analyzed using timing Shmoo plots for wafers obtained during characterization tests for the IC. The size of these regions, i.e. the number of die "represented" by a pair of golden die, offers a trade off between timing accuracy and the cost of learning timing from additional golden die. Smaller regions with less timing variation will require a small timing margin to be added to the timing learned from the golden die, allowing for tighter timing tests. In the extreme, the two golden die by themselves can form a region on the wafer; in that case the test methodology reduces to the highly effective, but expensive, DDSI approach presented in [23, 25].

Developing scan based delay tests based on the captured timing responses from sampling the golden die on the wafer requires some computation on a considerable volume of data. Computing the expected timing based test responses online and having them ready for application to individual die during wafer sort will be challenging. Golden die will need to be tested and sampled first. Test responses based on results from the first pair of golden die can be computed while the other golden dies are being evaluated.

.

CHAPTER 7 CONCLUSIONS

The difficulty of applying good functional timing tests to SOCs has greatly increased the interest in scan-based delay testing. Most designs implement the scan enable as a slow speed global control signal, and can therefore only implement launch-on-capture (LOC) delay tests. These display relatively modest delay fault coverage. Launch-on-shift (LOS) tests are generally more effective, achieving better fault coverage with significantly fewer test vectors, but require a fast scan enable.

A Delay Test Scan Flip-flop (DTSFF) unit [30, 31, 54] is developed for implementing LOS tests by adding a minimal amount of logic (six extra transistors) in a standard scan flip-flop to align the slow scan enable signal to the clock edge. This new design is much more efficient and effective when compared to other recent proposals. It can support full LOS and LOC testing, achieving a combined extremely high TDF coverage for the ISCAS89 benchmark circuits. In some designs, it may also be possible to employ a limited amount of scan chain reordering to further modestly increase coverage. If needed, TDF coverage can be further improved by applying two-partition mixed LOS/LOC tests. Intelligent partitioning of the flip-flops, rather than the random partitioning is even more effective in further raising the delay test coverage for mixed LOS/LOC tests. Also recall that the DTSFFs are pin compatible with standard scan flip-flops and can be controlled by a common slow scan enable signal, so they can be transparently integrated into conventional design flows.

Research shows that TDF coverage of traditional scan delay tests applied in the LOC mode is significantly impacted by the limited controllability at the inputs of some of the flip-flops. If these flip-flops can be identified and replaced by the recently proposed Delay Test Scan Flip-Flops (supporting LOS tests) or enhanced scan flip-flops (supporting arbitrary two-pattern tests), very high TDF coverage can be achieved by the resulting partial DTSFF scan design [55, 56] or partial enhanced scan design [57] with low overhead costs. Besides the basic flip-flop selection procedure which identifies the scan units according to their bias/non-bias status, an Interchange Produce is further developed to refine the scan unit selection. Recall that the DTSFF cell can be implemented with an overhead of as little as 6 additional transistors. Therefore, a partial DTSFF scan design with 20-40% DTSFFs will incur an average overhead of only about 2 transistors per flip-flop, while delivering high 90% TDF coverage. Similarly, a partial enhanced scan test methodology has been shown to provide 75-90% of the TDF coverage benefits achievable by full enhanced scan designs at 10-20% of the cost.

Because of the unpredictability of process parameter variations, as well as the unmodeled effects of power supply noise and crosstalk from excessive switching activity, the simulation based timing estimates used to evaluate delay tests must often be guardbanded to the point where many significant defects escape detection. Here we have shown how that many these problems can be addressed by calibrating (‘tightening”) the expected timing response of the applied delay tests by learning accurate signal switching timing from silicon [58]. Such a methodology requires that the outputs of the applied tests be hazard-free so as to avoid learning incorrect timing due to a glitch at the output. Simulation results presented indicate that such output hazard-free tests can be obtained with an average coverage only about 10 % below the transition delay fault coverage for both launch-on-shift and launch-on-capture modes.

In summary, the DTSFF based design-for-test methodology offers a promising and cost effective solution to achieving high TDF coverage in a scan based delay testing environment. Also it is viable to use low DFT cost partial DTSFF design or partial enhanced scan along with the slow scan enable signals to implement a high fault coverage delay testing. In practice, silicon calibrated delay test “learned” accurate timing from “golden” circuits on silicon so that it could provide high quality delay testing for both LOS and LOC tests.

BIBLIOGRAPHY

[1] A. Krstic, J.-J. Liou, K.-T. Cheng and L. C. Wang, "On structural vs. functional testing for delay faults", *in Proc. Quality Electronic Design Symposium*, 2003, pp. 438-441.

[2] M. L. Bushnell and V. D. Agrawal, "Essentials of electronic testing for digital, memory and mixed-signal VLSI circuits", Springer, 2000.

[3] E. B. Eichelberger, E. Lindbloom, J. A. Waicukauski and T. W. Williams, "Structured logic testing", Prentice-Hall, 1991.

[4] J. Savir, "Skewed-load transition test: part I, calculus", *in Proc. International Test Conference*, 1992, p. 705.

[5] S. Patil and J. Savir, "Skewed-load transition test: part II, coverage", *in Proc. International Test Conference*, 1992, p. 714.

[6] A. Krstic and K. T. T. Cheng, "Delay fault testing for VLSI circuits", Springer, 1998.

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

[8] 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.

[9] J. A. Waicukauski, E. Lindbloom, B. Rosen and V. Iyengar, "Transition fault simulation", *IEEE Design & Test of Computers,* vol. 4, 1987, pp. 32-38.

[10] S. Wang, X. Liu and S. T. Chakradhar, "Hybrid delay scan: a low hardware overhead scan-based delay test technique for high fault coverage and compact test sets", *in Proc. Design, Automation and Test in Europe*, 2004, pp. 1296-1301.

[11] Z. Zhang, S. M. Reddy, I. Pomeranz, X. Lin and J. Rajski, "Scan tests with multiple fault activation cycles for delay faults", *in Proc. VLSI Test Symposium*, 2006, pp. 343-348.

[12] J. Abraham, U. Goel and A. Kumar, "Multi-cycle sensitizable transition delay faults", *in Proc. VLSI Test Symposium*, 2006, pp. 306-311.

[13] N. Ahmed, C. P. Ravikumar, M. Tehranipoor and J. Plusquellic, "At-speed transition fault testing with low speed scan enable", *in Proc. VLSI Test Symposium*, 2005, pp. 42-47.

[14] Synopsys Application Note, "Tutorial on pipelining scan enables".

[15] N. Ahmed, M. Tehranipoor and C. P. Ravikumar, "Enhanced launch-off-capture transition fault testing", *in Proc. International Test Conference*, 2005, pp. 246-255.

[16] 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.

[17] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy and I. Pomeranz, "A novel method of improving transition delay fault coverage using multiple scan enable signals", *in Proc. International Conference on Computer Design*, 2005, pp. 471-474.

[18] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy and I. Pomeranz, "Improved delay fault coverage using subsets of flip-flops to launch transitions", *in Proc. Asian Test Symposium*, 2005, pp. 202-207.

[19] H. Yan, G. Xu and A. D. Singh, "Low voltage test in place of fast clock in DDSI delay test", *in Proc. International Symposium on Quality of Electronic Design*, 2005, p. 316.

[20] H. Yan, A. D. Singh and G. Xu, "Delay defect characterization using low voltage test", *in Proc. Asian Test Symposium*, 2005, pp. 8-13.

[21] H. Yan and A. D. Singh, "A delay test to differentiate resistive interconnect faults from weak transistor defects", *in Proc. International Conference on VLSI Design*, 2005, pp. 47-52.

[22] H. Yan and A. D. Singh, "Reduce yield loss in delay defect detection in slack interval", *in Proc. Asian Test Symposium*, 2004, p. 372.

[23] H. Yan and A. D. Singh, "Evaluating the effectiveness of detecting delay defects in the slack interval: a simulation study", *in Proc. International Test Conference*, 2004, p. 242.

[24] H. Yan and A. D. Singh, "On the effectiveness of detecting small delay defects in the slack interval", *in Proc. IEEE International Workshop on Current and Defect Based Testing*, 2004, p. 49.

[25] H. Yan and A. D. Singh, "Experiments in detecting delay faults using multiple higher frequency clocks and results from neighboring die", *in Proc. International Test Conference*, 2003, p. 105.

[26] S. DasGupta, R. G. Walther, T. W. Williams and E. B. Eichelberger, "An enhancement to LSSD and some applications of LSSD in reliability, availability, and serviceabilit", *in Proc. International Symposium on Fault-Tolerant Computing*, 1995, p. 289.

[27] 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.

[28] 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.

[29] 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.

[30] G. Xu and A. D. Singh, "Low cost launch-on-shift delay test with slow scan enable", *in Proc. European Test Symposium*, 2006, pp. 9-14.

[31] G. Xu and A. D. Singh, "Delay test scan flip-flop: DFT for high coverage delay testing", *in Proc. International Conference on VLSI Design*, 2007, pp. 763-768.

[32] I. Pomeranz and S. M. Reddy, "On application of output masking to undetectable faults in synchronous sequential circuits with Design-For-Testability logic", *in Proc. International Conference on Computer Aided Design*, 2003, pp. 867-872.

[33] I. Pomeranz and S. M. Reddy, "On generating tests that avoid the detection of redundant faults in synchronous sequential circuits with full scan", *Transactions on Computers,* vol. 55, 2006, pp. 491-495.

[34] J. Rearick, "Too much delay fault coverage is a bad thing", *in Proc. International Test Conference*, 2001, pp. 624-633.

[35] X. Liu and M. S. Hsiao, "Constrained ATPG for broadside transition testing", *in Proc. International Symposium on Defect and Fault Tolerance in VLSI Systems*, 2003, pp. 175-182.

[36] I. Pomeranz, "On the generation of scan-based test sets with reachable states for testing under functional operation conditions", *in Proc. Design Automation Conference*, 2004, pp. 928-933.

[37] L. Hangkyu, I. Pomeranz and S. M. Reddy, "A test generation procedure for avoiding the detection of functionally redundant transition faults", *in Proc. VLSI Test Symposium*, 2006, pp. 294 - 299.

[38] K. T. Cheng, S. Devadas and K. Keutzer, "A partial enhanced-scan approach to robust delay-fault test generation for sequential circuits", *in Proc. International Test Conference*, 1991, p. 403.

[39] Synopsys User Manual, "TetraMAX ATPG user guide", Version X-2005.09, September 2005, pp. 249-264.

[40] O. I. Khan, M. L. Bushnell, S. K. Devanathan and V. D. Agrawal, "SPARTAN: a spectral and information theoretic approach to partial-scan", *in Proc. International Test Conference*, 2007.

[41] D. Kagaris and S. Tragoudas, "Retiming-based partial scan", *Transactions on Computers,* vol. 45, 1996, pp. 74-87.

[42] T. Takasaki, T. Inoue and H. Fujiwara, "Partial scan design methods based on internally balanced structure", *in Proc. Asia and South Pacific Design Automation Conference*, 1998, pp. 211-216.

[43] D. Xiang and J. H. Patel, "Partial scan design based on circuit state information and functional analysis", *Transactions on Computers,* vol. 53, 2004, pp. 276-287.

[44] M. L. Flottes, R. Pires, B. Rouzeyre and L. Volpe, "Scanning datapaths: a fast and effective partial scan selection technique", *in Proc. Design, Automation and Test in Europe*, 1998, pp. 921-922.

[45] J. Rearick, "The case for partial scan", *in Proc. International Test Conference*, 1997, p. 1032.

[46] A. Pierzynska and S. Pilarski, "Non-robust versus robust", *in Proc. InternationalTest Conference*, 1995, pp. 123-131.

[47] C. Barnhart, "Delay testing for nanometer chips ", in *Chip Design*, August/September 2004, pp. 8-14.

[48] B. Kruseman, A. K. Majhi, G. Gronthoud and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing", *in Proc. International Test Conference*, 2004, p. 213.

[49] A. K. Pramanick and S. M. Reddy, "On the computation of the ranges of detected delay fault sizes", *in Proc. IEEE International Conference on Computer-Aided Design*, 1989, pp. 126-129.

[50] Cadence Encounter Test, "Product datasheet", Aug. 2005.

[51] J. Saxena, K. M. Butler, V. B. Jayaram, S. Kundu, N. V. Arvind, P. Sreeprakash and M. Hachinger, "A case study of ir-drop in structured at-speed testing", *in Proc. International Test Conference*, 2003, pp. 1098-1104.

[52] T. M. Mak, A. Krstic, K. T. Cheng and L. C. Wang, "New challenges in delay testing of nanometer, multigigahertz designs", *IEEE Design & Test of Computers,* vol. 21, 2004, pp. 241-248.

[53] C. K. Cheng, A. B. Kahng, B. Liu and D. Stroobandt, "Toward better wireload models in the presence of obstacles", *IEEE Very Large Scale Integration (VLSI) Systems,* vol. 10, 2002, pp. 177-189.

[54] G. Xu and A. D. Singh, "Scan cell design for launch-on-shift delay tests with slow scan enable", *IET Computers & Digital Techniques,* vol. 1, 2007, pp. 213-219.

[55] G. Xu and A. D. Singh, "High Coverage Delay Test with Partial DTSFF Scan Chains", *in Proc. North Atlantic Test Workshop*, 2007, pp. 26-32.

[56] G. Xu and A. D. Singh, "Achieving High Transition Delay Fault Coverage with Partial DTSFF Scan Chains", in *International Test Conference*, 2007.

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

[58] A. D. Singh and G. Xu, "Output hazard-free transition tests for silicon calibrated scan based delay testing", *in Proc. VLSI Test Symposium*, 2006, pp. 349-355.