Output Hazard-Free Transition Delay Fault Test Generation

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

Sreekumar Menon

Certificate of Approval:

Vishwani D. Agrawal
James J. Danaher Professor
Electrical and Computer Engineering

Adit D. Singh, Chair
James B. Davis Professor
Electrical and Computer Engineering

Victor P. Nelson
Professor
Electrical and Computer Engineering

George T. Flowers
Dean
Graduate School

Output Hazard-Free Transition Delay Fault Test Generation

Sreekumar Menon

A Thesis

Submitted to

the Graduate Faculty of

Auburn University

in Partial Fulfillment of the

Requirements for the

Degree of

Master of Science

Auburn, Alabama
May 09, 2009

Output Hazard-Free Transition Delay Fault Test Generation

Sreekumar Menon

_____

Signature of Author

_____

Date of Graduation

Vita

Sreekumar Menon, son of Mr. Narayanan Kutty and Mrs. Girija Menon, was born in Kondugallur, Kerala, India. He did his schooling in K.V. Pendharkar College, Mumbai, India. He graduated distinction with honors from Mumbai University, India with a Bachelors degree in Electronics in 2006. He joined the graduate programme in Electrical & Computer Engineering at Auburn University in August 2006.

Thesis Abstract

Output Hazard-Free Transition Delay Fault Test Generation

Sreekumar Menon

Master of Science, May 09, 2009
(B.E., University of Mumbai, 2006)

74 Typed Pages

Directed by Adit D. Singh

Architectural restrictions of scan 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. 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. Scan based timing comparison tests offer a potential solution to the problem of small delay detection in aggressive nanometer technologies. These tests require that circuit delays be unambiguously captured in the scan chains using multiple fast clocks. To ensure this, only those signals that are known to be hazard-free at captured are analyzed for timing information from the scan-out data. In this work we present the first systematic ATPG driven approach for generating high coverage Output Hazard-Free (OHF) TDF tests for scan delay testing. We have analyzed the effect of variations in process on test coverage using our approach. Results indicate that acceptable coverage can be achieved, no worse than about 10% below the unconstrained TDF coverage for both LOS and LOC tests, even

in the presence of significant process variations. The ATPG effort needed for the test set generation is modest as compared to using SPICE or other similar full circuit simulator for obtaining the OHF TDF vectors.

Acknowledgments

I would like to gratefully acknowledge the assistance, encouragement, support, patience and dedication provided to me by my adviser, Dr. Adit D. Singh, during my stay at Auburn University. I thank Dr. Vishwani Agrawal and Dr. Victor Nelson for being on my thesis committee and for their valuable suggestions. I am really grateful to Hillary Grimes for his extensive co-operation and for the use of his simulator which played a key role in my research. I would also like to thank my colleagues Nitin Yogi and Jins Alexander, for all the helpful discussions throughout this research. Most importantly, my heartfelt gratitude and thanks to my parents, whose encouragement and love has always been my strength in achieving my goals. Support from the Thomas Walter Center at Auburn University was very helpful in completing my studies at Auburn University.

Style manual or journal used Journal of Approximation Theory (together with the style known as "aums"). Bibliography follows van Leunen's *A Handbook for Scholars*.

Computer software used The document preparation package TEX (specifically LATEX) together with the departmental style-file `aums.sty`.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Recent technology generations display a noticeable increase in delay defects that impact circuit timing. Such defects are commonly caused by gate oxide failures and resistive opens in the vias and interconnect. Research interest has specially focused on small (fine) delay defects, which can often remain hidden within circuit timing slacks and timing margins during testing. While it is sometimes argued that such defects, not detectable at the rated clock speed during test, are functionally benign and can be ignored, there is an emerging consensus that small delay defects can result in functional and/or reliability failures in the field. Such defects must be detected to ensure acceptable product quality and reliability in high end ICs. There are two reasons that make it important to target small delay defects during test [1]. Switching delays in CMOS are highly input dependent. Meaningful delay testing requires that worst case delays be tested in the circuit paths. However, such worst case tests with even moderately high coverage are virtually impossible to generate and apply in practice. And even those effective tests, such as robust path delay tests, that can be generated often cannot be applied because of structural restrictions which allow only Launch-On-Capture (LOC) and Launch-On-Shift (LOS) tests in a scan environment. It is easily possible for a small delay fault to remain undetected during the test, but cause functional failure in the field when worse case input conditions are encountered [2]. Many small delay defects are known to degrade in operation and cause early life failure. For example, a resistive open caused by a minimally connecting via can become a complete open in operation due to metal migration. Traditional burn-in stress testing used to eliminate

such "latent" manufacturing flaws is becoming extremely expensive for delicate nanometer technologies; it also appears to be losing effectiveness in accelerating certain types of early life failures. As a result, industry is looking for alternate low cost methods for deleting latent defects. To quote from the 2006 "Research Challenges in Test and Testability" published by the Semiconductor Research Corporation: *"In order to achieve low DPPM levels without additional acceleration such as burn-in, fine delay defect screening appears to offer a solution."*

Unfortunately, the detection of small (fine) delay defects that fall within circuit timing margins is proving extremely challenging. Effective delay testing often requires faster than rated clock tests to discover defects within the timing margins. Even more aggressive detection of small delay defects hidden within the circuit timing slacks in short paths, can require the use of multiple fast clocks. Here each fast clock is used to test only those paths that are shorter than the corresponding clock period; the captured responses from longer paths must be masked out and ignored during test evaluation. However, this requires knowledge of the switching time for each signal for each applied test, information which has been traditionally obtained through timing simulation in test systems such as Cadence Encounter (True Time) [3]. However, given the high levels of performance variability from normal process variations observed in current technologies (path switching delays can easily span a 2X range across production lots), meaningful delay simulation to support such a small delay test methodology appears no longer viable. The problem is further aggravated by the "out-of-normal-mode" nature of single cycle scan delay tests which can display additional timing variations because of power supply noise [4, 5] temperature differences in the test model [6], and "clock stretching" [7].

An alternative to obtaining the expected switching delays from simulation in faster-than-rated-clock tests is to directly measure and compare such delays in matched ICs to detect delay faults. The Fmax, DDSI [8] and Self Timed [9] are all faster-than-rated clock scan based delay test approaches that have been proposed in recent years. The presence of multiple identical cores in state-of-the-art microprocessors makes a comparison delay test methodology even more attractive [7]. Since the performance of identical circuits located physically close to one another on silicon is generally well matched, except for random with-in die variations, the impact of process.variations is minimized in such comparison testing. (It is clearly impossible to distinguish delay defects that fall within random timing variations). An important additional advantage of such comparison tests is that all "out-of-normal-mode" effects associated with the scan delay tests equally affect timing in all the matched circuits that are tested and are thereby factored out in the comparison. A timing mismatch beyond the expected normal statistical variations between the matched circuits is an indication of a small delay defect.

While comparing timing between matched cores or die appears to be an attractive option for scan based faster-than-rated-clock testing, it imposes additional restrictions on the delay tests that can be applied: the applied tests must be hazard-free at all observed outputs [11, 8]. This is because the goal of the comparison tests is to compare the observed switching delays in final stable signals on circuit outputs; a hazard can cause the test to record an incorrect and arbitrary switching delay for a signal, leading to false error indications or test escapes.

1.1 Problem Statement

The problem addressed in this thesis is: *A systematic method to generate Output Hazard-Free Tests for transition delay faults and achieve a test coverage as close as possible to coverage attained by unrestricted transition delay fault testing.*

1.2 Contribution of Thesis

We have demonstrated how Output Hazard-Free transition delay fault (TDF) test generation can be implemented by using a systematic, yet low resource-intensive strategy. This is a first such attempt to develop a systematic ATPG based approach for generating Output Hazard-Free (OHF) transition delay fault (TDF) test for scan based delay testing. Such a methodology requires that the outputs be correctly identified if they can potentially display a hazard during the application of individual transition delay vectors. ATPG effort is modest as compared to using SPICE or other similar full circuit simulator for obtaining the OHF TDF vectors. Simulation results presented here indicate that such output hazard-free tests can be obtained with an average coverage of about 10% below the unrestricted transition delay fault coverage for both Launch-On-Shift (LOS) and Launch-On-Capture (LOC) modes.

1.3 Organization of Thesis

The thesis is organized as follows. In Chapter 2, we discuss a general background of scan based transition delay fault tests, and the motivation for generating Hazard-Free tests. In Chapter 3, previous works related to Output Hazard-Free test generation techniques are discussed. A general flow and a brief introduction to the general terms and methodologies used in our Output Hazard-Free testing is described in Chapter 4. In Chapter 5, Output Hazard-Free test generation techniques for combinational circuits are enumerated. Chapter 6 concludes with a detailed methodology dealing with OHF test generation for sequential circuits. Conclusions and future work are discussed in Chapter 7.

Chapter 2

Background

Tremendous changes have occurred within the semiconductor industry in recent years. This has been driven by industry requirements and consumer expectations of smaller, faster, more reliable, and less expensive integrated circuits. These integrated circuits form a key component in digital systems. Malfunction of these circuits is a common occurrence and will affect the behavior of digital systems. Incorrect behavior of digital systems might lead to serious accidents, as the digital systems are used as key components in many critical tasks. Therefore in order to realize dependable digital systems, VLSI circuits should be highly reliable. VLSI testing plays an important role in satisfying this requirement. Testing of the chips is used to check for faults existing in a circuit, and it consists of two main phases: test generation and test application. In test generation, stimuli, which but input sequences that are used to detect faults, are generated. In test application the generated test sequence is applied to the circuit. In the early days of digital systems the main concern was the logical correctness of the circuit. With improvements in semiconductor technology the speed of modern circuits has drastically increased. For such high speed circuits, delay testing is an important test feature that checks whether delay faults exist in a circuit. This has become an important technique to guarantee the timing correctness of the circuit because conventional testing for stuck-at faults is not sufficient to guarantee it. There also exists a possibility that switching delays can be affected by phenomena like residual partial charges on circuit-node capacitances that are the remnants from previous cycles. This is especially true for high-speed circuits with short-clock periods.

The strategy usually employed to detect such defects is to set up the worst case signal propagation conditions along each path for signals. Even this strategy does not guarantee detection of all delay defects. Delay defects whose effects are completely absorbed in the slack along a short path will remain undetected. Delay test generation for sequential circuits is a very challenging problem. It is very unlike the situation of generating tests for sequential circuits under simple fault models, such as the single stuck-at fault model.

Design for testability (DFT) is an important approach to reduce the test generation complexity for such sequential circuits. A scan based method has been proposed as a straight forward DFT technique for testing delay faults. Scan technology enables high levels of defect detection using automated tools. Each flip-flop in the circuit under test is implemented as an equivalent sequential element called a scan cell that includes extra test features. During scan mode, scan cells are concatenated into several long shift registers called scan chains. As a result, Automatic Test-Program Generation (ATPG) tools can create test patterns easily, since the complex sequential nature of the circuit is separated into the combinational logic (which is comparatively easy for test generation) between scan cells. Stuck-testing, also known as static testing, involves the generation of just one vector for a targeted fault site. This stuck vector will set the node to the logical 1/0 for targeting a stuck-at zero/one fault at that node. The other inputs will sensitize a path to the output lines to observe the logic value at that node. Static testing can also run at a slower clock speed. However testing for delays involves generation of a two-vector set for every delay fault site. The generation of the two vectors is limited by the design of the scan architecture used in sequential circuits. One of the vectors v2 from the vector pair [v1, v2] required for sequential delay testing has to be a stuck test for the fault site. The choice of vector v2 in

a scan environment, however, is dependent on the preceding vector v1. v2 can only be a shifted version of v1, or v2 can be the circuit response to vector v1 that is captured back in the scan chain. This constraint on the generation of vector v2 hinders the application of effective testing methodology, like path delay testing etc.

2.1 Scan based transition delay fault tests

Scan based delay fault testing is commonly based on the transition delay fault (TDF) model. The two possible faults detected at each node are slow-to-rise and slow-to-fall. For detecting a slow-to-rise fault on the line, a test for a stuck-at-0 is generated for that fault. This will set the node to logical 1 in the fault free circuit. The other inputs sensitize a path to the output lines in such a manner that any change in the logic value at that node is visible at the output. This vector v2 is preceded by any vector v1 that will set that node to 0. Now this vector pair [v1, v2] is a test for the slow-to-rise transition fault on the line. In a faulty circuit where the line is slow to rise, the effect observed at the output will be a 0 instead of the expected value 1. Similarly for detecting a slow-to-fall fault on the line, vector v1 sets the line to 1 and the vector v2 is a stuck-at-1 test for that line.

This basic structure makes it easier to apply the test vectors at the rated clock-speed which is also known as at-speed testing. At-speed testing allows the circuit to be tested under its normal operating condition. Fig. 2.1 shows a conceptual overview of the scan-based delay testing using a two-vector test pattern. Scan based structural delay testing involves applications of two test vectors [v1,v2] via the scan chain. The first vector v1, also known as the initialization vector is first scanned into the scan chains using a slow scan clock. This vector v1 initializes the internal logic values of the CUT (Circuit-Under-Test).

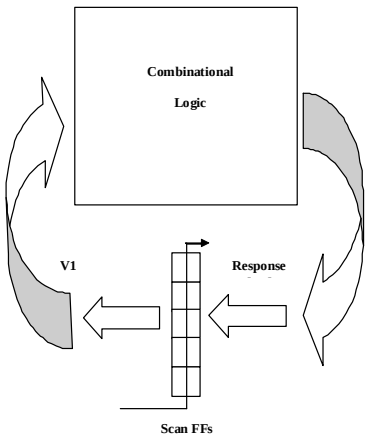

Figure 2.1: Illustration of a general Scan design circuit

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. The launch to capture period is dependent on the operational frequency of the Circuit-Under-Test. If the circuit is free of delay defects, then the transition that had propagated (within the specified clock period) to the end of the circuit paths will be the correct value. Otherwise, if a delay causes a slow propagation, the transition from launch to capture cell will be slow and an erroneous value will be captured leading to the detection of the defect. If explained in terms of the vector pair [v1,v2], the first vector v1 is scanned in with a slow scan clock and is then replaced in the scan register to launch v2 by applying a one-bit shift to the scan register or by propagating v1 to the combinational logic in the normal mode and capturing the outputs in the scan register. This divides the transition delay fault tests into two categories: Skewed-load delay (Launch-On-Shift) test or Broad-side (Launch-On-Capture) delay test.

2.1.1 Launch-On-Shift tests

In Launch-On-Shift (LOS) test mode, the test patterns are at first loaded to the scan chains. The launch occurs during the last shift while loading the scan chain. Scan in of vector v1 is followed by one extra cycle of slow clock while the circuit is still in the scan mode. The test is so designed that the next vector v2 is obtained by a 1 bit shift of the previous vector v1. The circuit is placed into functional/capture mode very quickly so that an at-speed functional clock can be pulsed. ATPG is generally considered easier with LOS as compared to the Launch-On-Capture pattern generation. It is a simple ATPG activity to load the starting value for a transition directly to a scan cell one shift before the last shift of the data and then load the transition value in the last shift. LOS patterns usually report higher coverage than Launch-On-Capture patterns. However the industry is reluctant to use LOS patterns for commercial testing. There are two primary reasons for this reluctance. The first is the difficulty to make the circuit change from shift mode to functional/capture mode between the last shift and the functional clock pulse. If a standard scan enable architecture is used then the scan enable must be routed as a clock. Furthermore since scan enable goes to all sequential elements it acts as a global clock and must settle at the system clock frequency. LOS patterns can shift in a transition that is impossible during normal circuit operation. The timing diagram for a Launch-On-Shift delay test is shown in Fig. 2.2.

2.1.2 Launch-On-Capture tests

The procedure for the Launch-On-Capture (LOC) test is described as follows. At first the scan enable (SE) signal is asserted and the first pattern v1, also called the initialization

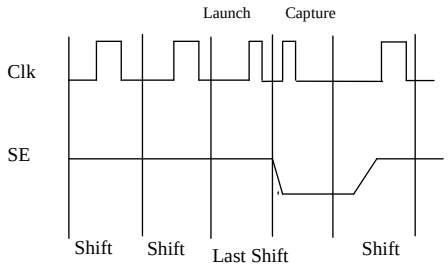

Figure 2.2: Timing Diagram for Launch-On-Shift delay tests

pattern is scanned into the scan cells. For a slow-to-rise (slow-to-fall) TDF, v1 sets the fault site to 0(1). In LOC the second pattern is generated through the combinational logic of the Circuit-Under-Test (CUT) andcaptured in the scan cells by applying a clock pulse with SE deasserted. This is referred to as the launch cycle. Application of v2 activates the fault by launching a transition at the fault site and also propagates the fault effect to an observed output (primary output or a scan cell). For a slow-to-rise (slow-to-fall) TDF, v2 is a test for a stuck-at-0 (stuck-at-1) fault at the fault site. Following the application of v2, another clock pulse is applied with SE still de-asserted to capture the CUT response to the test. The advantage of this approach is that it does not require the scan enable signal to operate at full speed and the sensitizable paths under the Launch-On-Capture constraints are also sensitizable in functional modes unless the first vector represents an illegal state. Though the Launch-On-Capture approach is more promising and practical for industrial use, the Launch-On-Shift approach is also included in commercial testing because it might detect some faults that are missed by Launch-On-Capture tests.

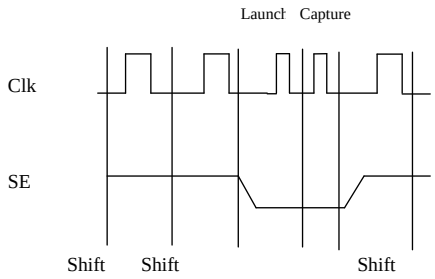

Figure 2.3: Timing Diagram for Launch-On-Capture delay tests

2.2 Motivation for Hazard-Free tests

Speed of a high-end ICs is one of the prime concerns in today's highly competitive market. It is becoming increasingly difficult to design a reliable and an easily testable chip. Transition Delay Fault (TDF) testing is one of the most effcient and reliable ways employed by the industry to keep the frequency of operation of the chips to be reliable to specifications. However it is not a fool-proof method of testing chips for delay defects.

Small (fine) delay defects, which can often remain hidden within circuit timing slacks and timing margins during testing are increasingly becoming a major force that can cause reliability issues in high-end ICs. This has been made worse by the fact that design complexity has increased exponentially over the last few years. The number of metal layers has been predicted to incrementally increase from the current ten to reach fifteen in the near future. Due to the continuous scaling of processes the interconnects have become narrower and taller, creating increased interconnect delays. Transition scaling also causes other problems like tunneling, hot electron effect, etc. Use of a lower voltage in chips for power

considerations has become routine. Such circuits are more susceptible towards noise. Such noise has been known to cause extra delays [3]. It is indeed questionable whether a static timing analysis technique prevalent today can alone predict all the variations with any level of certainty. Differences in timing could be a result of process variations over a local region in a die or over different lots and wafers. The amount of timing variation seen will depend on the spatial variations of all the effects discussed above and the design itself. Certain designs could be immune to changes in certain parameters and other designs might be more prone towards certain parameters. Resource intensive techniques like statistical methods have been proposed by Duvall [21] and Agrawal et al. [22] to predict the effects of spatial variations on delay sizes. The size and distribution of delay defects affects the quality of delay tests [30, 55, 56]. The size of delays complicates things further, as the test vector set designed to catch a delay fault of one size might not be able to catch the same delay fault if its size varies. Pramanick and Reddy [14] have also explored similar phenomena for designing delay tests with the considerations for delay defect and slack sizes. A delay test for a fault of a certain size might not be a test for a similar fault of a smaller or larger size [55]. In addition to these issues it should be kept in mind that potentially effective tests like robust path delay tests often cannot be applied in practice because of structural restrictions which allow only Launch-On-Capture (LOC) and Launch-On-Shift (LOS) tests in a scan environment. The problems in traditional scan-based testing are compounded by the "out-of-normal-mode" nature of single cycle scan delay tests.

Various alternative solutions have been proposed by many researchers to detect small delay defects. One such solution proposes to look for delays in the short paths within the

slack interval. (Theslackassociated with each connection is the difference between the required time and the arrival time. A positive slacksat a node implies that the arrival time at that node may be increased byswithout affecting the overall delay of the circuit. Conversely,negative slackimplies that a path is too slow, and the path must be sped up (or the reference signal delayed) if the whole circuit is to work at the desired speed.) This technique calls for using a clock that is faster than the nominal rated clock to capture the scan based test response. There are however a variety of issues plaguing this approach. All the issues that were discussed earlier like process variations, heat issues, power issues and noise play their part in causing effects like hazards that can confuse timing tests. All these make it very difficult to test for small delay defects. Simulations can incorporate a tolerance to accommodate all these effects but it reduces its accuracy in catching the small delay defects. It should be kept in mind that a traditional scan environment is very different than a normal functional environment. Non-functional tests have a probability to activate certain paths that are not functionally used and might cause certain unwanted issues like local-heating, etc. This could lead to an incorrect pass/fail rate and ultimately yield loss. This issue is not limited to the traditional scan based testing, it can also be observed in BIST based architectures.

It is not feasible to model all the effects like process variations and test setup parameter variations in a timing simulator. An effective and more accurate method of obtaining timing information is from the silicon itself. Silicon-Calibrated delay testing proposed by Singh et al. [10] measures for each signal transition by sampling the output lines with faster-than-rated-clocks for every test stimulus. This is termed as the 'learned' response which is obtained from a golden die in the wafer. The key point is that the stimuli applied

to the golden die are the same as those applied to nearby die in the wafer. Assuming the test setup (electrical and temperature settings) doesn't change for testing the local regions of the wafer; Silicon-calibrated testing can potentially detect any variations in performance. The degree of accuracy in this test is dependent on the rate of the sampling of the output lines. An effective tradeoff can be reached to ensure minimal test costs and accurate timing results.

However, it is very critical to get a learned response which is guaranteed to be a true circuit response. Hazards are the prime concern in techniques involving learning timing from silicon. Hazards might cause an incorrect failure or an incorrect pass when the learned timing is being recorded. Hazards need to be avoided to ensure that the circuits passing the timing tests are indeed good circuits and that they do not have any delay faults in them or that the circuits rejected on the basis of the timing tests do indeed contain small delay defects. The concept of "learning" timing information from a test circuit thus requires the observed signals in the circuit to be free of hazards. But hazards are a common occurrence in CMOS technology. There always exists a possibility that some test vector pair in the test set designed might catch a hazard and record false timing for a signal path. A hazard can easily fail a timing test and can cause an incorrect reject. This is especially true in the learning phase; if a hazard causes an incorrect golden timing value to be recorded the consequences would be that good circuits will be declared bad leading to yield loss. One safe way of preventing such a possibility is to ensure that the outputs remain hazard-free for the test inputs which are applied for the purpose of learning timing information from silicon. The hazard-free timing tests are the conventional TDF tests with an additional constraint. The constraint is nothing but a condition that states that, for every TDF stimulus applied

to the circuit, it will not generate any hazards at any of the output lines that are observed. It also should be kept in mind that for such a set of TDF stimuli, not all outputs can be expected to remain hazard-free, only a certain set of outputs will remain hazard-free. This small set of hazard-free outputs is to be used for learning circuit timing that can later detect faults in other dies. Thus hazard-free test generation coupled with Silicon-calibrated delay tests can prove to be a viable alternative for detecting small delay defects.

Chapter 3

Previous Work

The first section in this chapter describes a Philips experiment [11] aimed at studying the potential of detecting "fine" delay defects with conventional gross delay defect vectors. The experiment included a process flow that created hazard-free TDF tests from the conventional gross delay vectors. The second section describes a technique that employed logical masking to obtain Output Hazard-Free TDF tests from random TDF vectors.

3.1 Detection of fine "delay" defects with gross delay fault vectors

Gross delay faults are those delay faults for which the combined delay of the defect and the tested path is longer than the delay of the critical path. To ensure manufacturability for most designs (based on a slow process corner) gross delay fault testing usually attempts to guarantee the chips running at the specified minimum frequency. As explained earlier, conventional TDF delay patterns are designed to detect gross-delay faults. As these patterns traverse unequal paths, small delays may be suppressed during the delay tests. However these small delays might cause a failure during a functional mode if they activate a long path. These small delay defects problems are usually the result of variations in the process and the test setup environment. These process variations result in an increased resistance of a via or a wire. Depending on the location and the test conditions the delay observed due to the increased resistance may vary. It has been estimated that the number of delays in the 1-7 ns range caused by this increase in resistance are nearly the same as the number of delays seen in the 7-49 ns range [63]. The data clearly indicates that using gross delay

testing alone to pass or fail chips might lead to test escapes. Also the number of delay faults in the small delay category is nearly the same as the gross delay fault numbers. This makes a very compelling case for creating a test methodology for detecting small delay defects.

ATPGs usually have a tendency to generate patterns over shorter paths. There have been approaches intending to modify the ATPG function to generate tests for shorter (nearly equal) paths for certain desired fault coverage and then use the same patterns for targeting longer paths. This approach might solve the problem of detecting small delay defects. But currently existing complex designs hardly make it feasible. Highly complex designs make it a very resource intensive task for the ATPG to generate patterns for equal lengths that will have satisfactory test coverage. An adaptive pass/fail technique [64] has also been suggested to improve the delay coverage. This adaptive pass/fail technique relies on a comparative timing analysis of the blocks or uses the on chip PLL frequency. The use of a reference in the adaptive fail/pass methodology leads to a detection rate greater than the gross delay detection. But, it also allows a considerable margin of error to creep in to the comparison. Studies [11] showed that delay faults that can lead to a 30% increase in timing will remain undetected even with the use of the adaptive pass/fail feature. In addition the glitch behavior of a circuit is dependent on the process parameters and the input combinations to the circuit. Logic simulation cannot be used alone to determine the hazard-free response to test stimuli. A statistical analysis of the rise and fall time of each gate, using a Monte-Carlo analysis, could be performed to generate a hazard-free test set; but in designs that contain nearly a billion transistors, it would be totally unfeasible.

Bram Kruseman et al. [11] introduce a strategy for creating Output Hazard-Free tests from a group of conventional gross delay test patterns. This strategy called for dividing the

ATPG generated gross delay patterns into sets of delay vectors that test for nearly equal path lengths. These sets of grouped delay vectors test nearly equal path lengths at higher speeds to detect the small delay defects. A concept of tolerance was also introduced in the simulation run to account for the various changes in the process parameters that might cause hazards. They used a region 'X' for every transition during simulation to account for the delays induced by process variations. The 'X' region was also designed to incorporate clock-skew. The 'X' region was taken as a first approximation of the absolute difference between the fastest and the slowest simulation time observed. They introduced this 'X' region in every gate and lumped the delays for logic like flip-flops so that a faster simulation time can be achieved. The 'X' region introduced at the input propagates through various nodes at the outputs. Figure 3.1 is an illustration of how they determined an output to be gltichy or gltich-free. The first line indicates the ambiguity introduced at the inputs of the circuit. Xa represents the width of the region. The second line represents the same ambiguity region visible at a node n2. The third line is a representative figure of these ambiguity regions coming together at an output line. An output is termed glitchy if there are two 'X' regions observed at the primary output as shown in the Figure 3.1. The degree of accuracy in catching the glitch-free outputs depends on the width of the 'X' region. Making the width too high will result in a pessimistic result and theoretically can catch all the glitches. All the outputs, which show the uncertainty region greater than the delay region 'X' that was introduced at the primary inputs, are termed as glitchy and discarded. They ran this 'X' based simulation with the use of the grouped gross-delay patterns generated by the ATPG to give a hazard-free coverage. In the simulation run to get the OHF coverage, they dropped

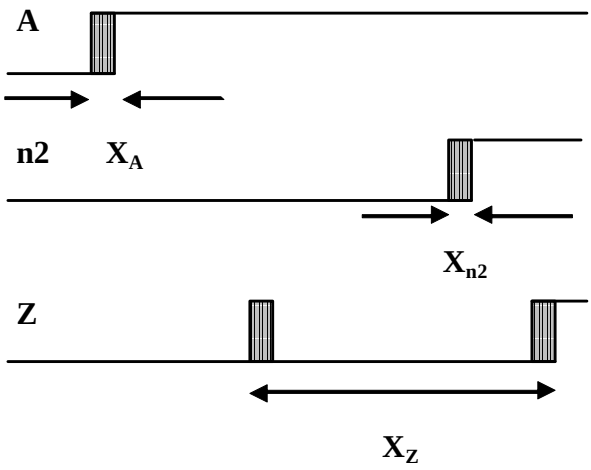

Figure 3.1: Selection of unstable regions ([Ananta K. Majhi et al.][11])

Characteristics	**Design1**	**Design2**
Technology	0.18 um	0.13 um
Size in Gates	50k	250k
gross delay Fault Coverage	97.50%	95%
Hazard-free Fault Coverage	40%	56%

Table 3.1: OHF coverage on two industrial circuits

the outputs that did not meet the timing goals and the hazard affected outputs (which were treated as don't-care) and ran the set of grouped gross delay test vectors.

They employed this technique on two industrial circuits, the results of which are shown in Table 3.1.

In conclusion this work resulted in approximately 40% and 60% hazard-free test coverage for the two circuits considered. They however did not proceed further with hueristics that could have improved their hazard-free test coverage.

3.2 OHF Test generation using logical masking

A hazard-free testing methodology [10] was proposed by using the difference in the depth of the logic level in cones leading to the output. Logical hazard masking was used to obtain Output Hazard-Free TDF tests from a conventional TDF test set that was generated from random pattern. Hazards usually occur when the delay at the input of a gate is greater than the gate's inertial delay. This occurs due to the propagation delay occurring in the logic cone behind the input lines. In other words, the delay observed in a signal is a function of the logic depth on that signal path. If a signal on the longer path in the logic cone behind the observed output switches much before than the signal at the shorter path, a hazard-free condition can be observed at the output line. Singh et al. [10] took this conclusion further, by assuming that a hazard can only occur if the signal line from the top input is delayed by one or more inertial gate delay more than the lower input signal. This translates to the observation that the possibility of the generation of a hazard becomes less if the signal paths differ by two or more logic levels. This logic can also be applied on signal paths of the same length, as it would take at least one gate inertial delay to create a hazard.

This might however be an optimistic view, as in CMOS logic gates can exhibit different gate delays which might vary due to subtle changes resulting from process variations and other factors. Hazards generated by such factors are overlooked by this proposed technique. It could also work in the favor of hazard-free situations, as such variations might cancel out the difference in gate delays and cause no hazards to be observed at the output lines. This work makes an assumption that for a huge number of timing tests, the effect of timing variability causing hazards and suppressing hazards will eventually cancel each other out. While somewhat better coverage was reported, the results were still conservative when

compared with the coverage that can be obtained if timing information is also taken into consideration. Many hazards that are logically possible are filtered out in circuits because of the actual timing relationship between signals.

Chapter 4

Introduction to Output Hazard-Free (OHF) Test Generation Methodology

4.1 Introduction

Hazard-free testing involves application of TDF tests that do not generate hazards on the outputs lines of the circuit-under-test. Test stimuli when applied would generate hazards on one or more output lines of the circuit-under-test. It is very rare to observe a situation when the test stimuli will generate hazards on all the output lines of the circuit. The definition of a hazard in hazard-free simulation terms is very important. A very loose definition of a hazard might render the test ineffective. On the other hand a very strict definition of the hazard will lead to a very pessimistic test solution. Here, in this introduction we make our assumptions for a hazard-free condition in the simulation environment clear. The experimental setup to detect glitches has been explained in detail in the following section.

4.2 Min-Max delay simulator used for hazard detection

We have used a min-max delay simulator to predict the generation of hazards for an applied set of stimuli. Figure 4.1 illustrates a min-max simulator [59] which was used to determine the glitch behavior of circuits. This min-max delay simulator developed by fellow researchers, uses a bounded delay model [59], where each gate is assigned lower and upper bounds for delay. This assignment refers to the min-max delay specifications. These delays depend on the best and worst case switching delays (gate inertia) for the gate [60]. To define and represent signal timing in the bounded delay model, we use the term ambiguity region.

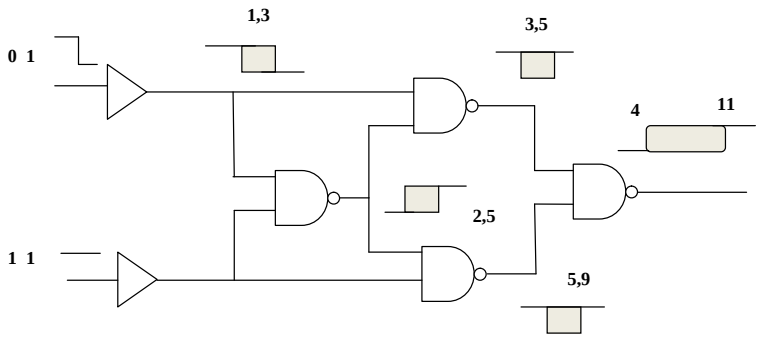

Figure 4.1: Min-max delay Simulator

This timing region is shown in Figure 4.1 as a shaded region at the outputs of each gate. This ambiguity region represents an area of signal uncertainty. This area of signal uncertainty translates into a probable region for creating a hazard, i.e. the signal may transit in that region, possibly more than once, creating a hazard, but the exact timing of the transition is uncertain. This simulator incorporates a 30% random variation on the inertial delay of each gate. This introduction of a tolerance in the simulation is to accommodate the various process variations and setup issues that affect a test procedure. Figure 4.1 illustrates the formation of a glitch at the output due to specific [v1, v2] inputs. **The simulation makes the conservative assumption that a hazard always occurs at the output of a gate if the ambiguity region is large enough to accommodate the multiple transitions needed to create the hazard, based on the (most favorable) minimum switching delays for the gate.** A circuit output can be termed hazard free for the TDF test vector applied if there is no possibility of a hazard on that specific output line. It should be noted here that our definition of a hazard-free condition is dependent the status of one or more output lines of the circuit with respect to the test vector application. Those output lines

that exhibit glitches during simulations are masked out as don't-care and ignored for fault observation.

4.3 Glitch profiling

Glitch Profiling is an important technique employed to improve the effectiveness of the ATPG tool. The hazard response to test vectors varies for different circuits. Hazards are a function of circuit design as well as of the variations that occur in the manufacturing process. Some designs can actually be more immune to glitches and some very sensitive to glitches.We decided to use this characteristic of hazards to rely on design variations and incorporate that in the ATPG tool. In a normal test case scenario the ATPG tool generates delay test vectors for the circuit, considering any of the primary output lines as a path to propagate this fault. We decided to refine this approach to make a more efficient use of the ATPG. We subjected every design under hazard-free testing to 1000 random TDF vectors. Using the min-max delay simulator described above, we made a database recording the response of the circuit (every output line) to the 1000 random TDF vectors. This database stored the glitch behavior shown by the primary outputs for every design subjected to the random 1000 TDF vectors. In every design it was observed that certain output lines tend to be very glitchy and certain outputs lines remain relatively immune to glitches. Those output lines that showed a very high tendency towards glitches were masked out and the circuit was subjected to TDF vector generation. As we had already masked out the most glitchy output lines, the ATPG-generated TDF vectors would contain a richer set of TDF vectors that will have a lower glitch profile, thus increasing the effectiveness of the test process. By using this technique, ATPG was forced to generate TDF vectors for some of

the faults that would avoid the glitchy lines for propagation. The following is an example of the database from an experiment to study the glitch profile on ISCAS-85 circuit c880 which proved to be very important in our strategy to generate an effective test methodology for Output Hazard-Free testing. c880 is an 8 bit ALU with 60 inputs and 26 outputs. The aim of the experiment was to understand the glitch behavior of the circuit on application of test vectors. We used the above described min-max simulator to identify glitches. The experimental setup included an application of one thousand random TDF vectors to the circuit. The resulting glitch information on each output line for each applied test vector input was recorded. This experimental setup was repeated five times on the same circuit for the same set of one thousand random vector inputs. These five setups on the c880 circuit included a gradual increase in tolerance on the inertial gate delay of each gate. These five experimental setups were basically five versions of c880. The first version had a tolerance limit on the gate inertial delay of 10%, the next version had a limit of 20% and this was continued till the last version of the circuit had a tolerance of 50%.

Table 4.1 illustrates the results of the above described experiment. Column 1 of the table lists the 26 outputs of the circuit. Column 2 to column 5 represent the behavior of the output lines on the application of a specific set of one thousand random TDF vectors. The numbers in these columns represent the glitch behavior of the output lines in the circuit i.e. a number 61 means that the output line corresponding to the number was found be glitch-free for 61% of the applied thousand random inputs. The number 100 for a specific output line would indicate it to be glitch-free for all the TDF test vectors applied. On the other hand the number 0 for a specific output line would indicate it to be glitchy for all the TDF test vectors applied.

O/P no	**10%**	**20%**	**30%**	**40%**	**50%**
1	61	61	61	61	61
2	58.2	58.2	58.2	58.2	58.2
3	60.3	60.3	60.3	60.3	60.3
4	50.6	50.6	50.6	50.6	50.6
5	68.4	68.4	68.4	68.4	68.4
6	68.7	68.7	68.7	68.7	68.7
7	59.1	59.1	59.1	59.1	59.1
8	59.1	59.1	59.1	59.1	59.1
9	61.5	61.5	61.5	61.5	61.5
10	46	46	46	46	41.6
11	93.3	86	86	86	86
12	60.2	60.2	60.2	60.2	60.2
13	90.5	90.5	90.5	83	83
14	94.4	94.4	94.4	86.4	86.4
15	46	46	46	46	41.9
16	0.04	0.04	0.04	0.04	0.04
17	0	0	0	0	0
18	28.6	27.7	26.4	24.8	23
19	20	19.4	19.1	18.6	17.3
20	27	26.9	26.6	26.2	24
21	29.9	29.6	29.2	28.9	25.9
22	36.3	34.4	32.3	29	26.2
23	22.4	21.3	21	20	18.4
24	23	22.5	20.8	19.4	18.5
25	23.9	22.5	22	21.5	20.5
26	23.6	22.9	21.6	20.9	19.4

Table 4.1: Illustration of Glitch-Profiling (c880 Results)

4.3.1 Observations from glitch profiling Technique

Interesting and important conclusions can be derived from Table 4.1. The first obvious conclusion to be derived is that the glitch frequency observed increases with the increase in tolerance on gate inertial delay. However this increasing glitch behavior is output dependent. It can be observed that some output lines are more prone to glitches. Certain output lines deteriorate more with increasing tolerance. Some output lines remain steady throughout the entire setup variations. The most interesting observation is the output line 17 which is glitchy for all the random TDF test vectors applied. Useful information can be gleaned from the above set of observations. As the TDF vectors that generate hazards on output lines need to be dropped, the strategy of generating output hazard-free tests involves generating multiple TDF vectors for a fault. We can use this prior information regarding the glitch behaviour of a circuit to generate hazard-free TDF tests from the ATPG more effectively. c880 has an output line number 17 which is hazard prone; we can direct the ATPG to avoid this output line for fault propagation. This will help to reduce ATPG effort in generating transition delay vectors that would have been wasted by propagating the faults to this glitchy output line. This technique of glitch profiling is used extensively in our hazard-free test generation methodology.

4.4 TDF vector generation from stuck-at vectors

Recall that a transition delay fault (TDF) test is a two pattern test [v1, v2] which causes a rising (falling) transition at the target node, while v2 is simultaneously also a stuck-at-0 (stuck-at-1) test for the node. The test pattern set obtained from conventional TDF ATPG is further augmented by a unique methodology which provides additional TDF vectors for

the TDF faults derived from stuck-at vectors obtained from N-detect stuck-at ATPG. The stuck-at vectors generated from the conventional ATPG provide the v2 vector in the two pattern TDF delay test, [v1, v2], for the target fault. The v1 vector to be associated with each v2 is constrained by the scan-chain structure. The techniques for generation of vector v1 depend upon the capture procedures applied in the scan environment and the type of logic.

Unlike the sequential circuit scan environment, in a combinational environment, there is no restriction on the vector [v1,v2] generation. The TDF pattern generated from this mode constitutes the seed vector (stuck-at seed) as the second vector v2 and the vector v1 is a one bit shift of the seed vector. We get multiple TDF patterns by generating multiple v1s per v2 by continuing to generate v1 by shifting v2 (the number of shifts per v1 depend on the number of primary inputs).

In a sequential environment, the traditional scan environment poses limitation on the generation of the TDF vectors. For a Launch-On-Shift (LOS) test, vector v1 for the two pattern delay test is simply obtained by applying a one-bit shift to v2 in the direction opposite to that of the normal scan shift. This v1 however does not guarantee the desired transition at the target node when applied as an LOS test; several v2 vectors from the N-detect stuck-at set for v2 are therefore tried until the desired TDF test is obtained. For Launch-on-Capture (LOC) tests, finding a v1 vector to go with the v2 vector generated by stuck-at ATPG is more challenging; the v1 vector is obtained as a stimulus for the combinational block that leads to the v2 vector as the next state.

4.5 N-detect Incremental Test Generation

Conventional TDF vectors are applied to the circuit and the hazards generated by them on every output are measured by using the min-max delay simulator. The output lines that show hazards for the applied stimuli are then masked off and the TDF vectors are applied to the circuit again. This will in effect reduce the fault coverage of the TDF vectors. In order to get a good Output Hazard-Free test coverage it is necessary to generate as many TDF vectors as possible. N-detect is the obvious answer to generate multiple TDF vectors. We have added a unique step to the N-detect strategy to increase the effectiveness of the vector set.

The incremental N-detect flow starts with the generation of N=1 detect TDF vectors from a conventional ATPG. These vectors are then filtered for hazards and then applied to the circuit. The detected fault list from this vector run is dropped from the total fault list. Using the undetected fault list from this run, the circuit is then again subjected to the ATPG for N-detect testing (Now N=2). This procedure of filtering the vectors for hazards and creating a new fault list for an incremental N-detect is continued till the fault coverage is saturated. This flow helps us to take advantage of the ATPG simulator to create a rich set of different patterns for every run. The incremental flow of N-detect vectors gives us an effective OHF test vector set by using a simple fault dropping strategy.

Chapter 5

Output Hazard-Free (OHF) test generation for combinational logic

5.1 Ad-hoc OHF Testing

Ad-hoc testing methodology was adopted for Output Hazard-Free Transition Delay Test generation for ISCAS85 circuits. The basic objective was to generate TDF vectors that detect transition delay faults without generating a hazard on the output lines. Hazards may occur on some of the outputs for the applied stimuli, but those outputs are masked out and ignored. The basic test generation proceeds in two steps: TDF test patterns are first generated using a conventional ATPG tool, then a timing simulator is used to filter out those tests that can potentially display hazards at the output. The important part is to generate enough TDF vectors for the two step process to be repeated until sufficient coverage is obtained.

The ATPG flow is illustrated in Figure 5.1. This ATPG flow illustrates an ad-hoc technique of generating TDF patterns for OHF testing. The first step involves generating conventional TDF vectors from a standard commercially available ATPG tool. These vectors are then subjected to filtering using the min-max delay simulator explained earlier. The filtering involves masking out the output lines for which the vector produces a glitch. These filtered vectors are then applied to the test circuit. The detected faults from this round of test vector generation are dropped and the undetected faults are now taken as the new fault list for the second step in the ATPG flow. The next round of TDF fault generation involves the use of stuck-at vectors from the undetect fault list passed from step 1. These stuck vectors are then subjected to the same filtering procedure explained earlier.

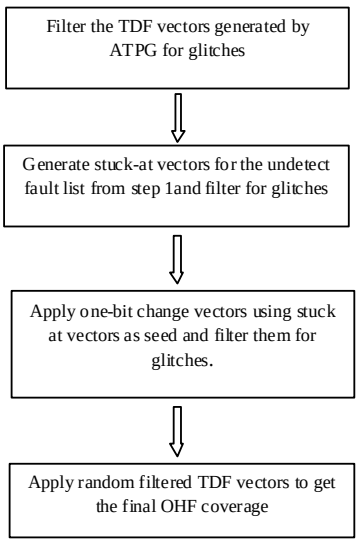

Figure 5.1: The Ad-hoc ATPG Flow for OHF test generation for combinational logic

Recall that a transition delay fault (TDF) test is a two pattern test [v1, v2] which causes a rising (falling) transition at the target node, while v2 is simultaneously also a stuck-at-0 (stuck-at-1) test for the node. The TDF vector pairs are generated from stuck-at vectors. Vector v1 is generated from the one-shift of the vector v2 (which is the stuck-at vector). The filtered stuck vectors are paired with this vector v1 to complete the TDF vector pair. These are then applied to the circuit and the detected faults are removed from the total fault list.

The next TDF generation technique involves a unique ATPG vector generation flavor. We used one-bit change vectors for our third round of TDF vector generation. These one-bit change vectors use the stuck at vectors as seeds, and the number of primary inputs of the circuit under test determines the number of vector pairs generated for each stuck-at vector. The TDF pattern generated from this mode constitutes the seed vector (stuck-at seed) as the second vector v2 and the vector v1 is a one bit shift of the seed vector. We

Circuit	Unrestricted ATPG (%)	OHF (%)	Filtered ATPG (%)	Stuck-at Vectors(%)	One Bi Change (
C17	100	**100**	44.11	21.92	4.56
C432	99.17	**98.05**	38.03	28.22	7.87
C880	99.94	**98.61**	43.98	30.71	6.23
C1355	98.17	**96.34**	27.89	33.89	10.11
C1908	99.3	**97.88**	34.19	24.15	13.76
C2670	99.25	**97.54**	35.99	18.75	8.48
C3540	99.57	**98.14**	46.99	15.36	9.12
C5315	99.76	**96.93**	39.99	22.48	11.12
C6288	99.98	**96.78**	33.99	30.29	3.85

Table 5.1: Ad-hoc Output Hazard-Free Test Generation Coverage on ISCAS85 circuits

get multiple TDF patterns by generating multiple v1s per v2 by continuing to generate v1 by shifting v2 (the number of shifted v1s depend on the number of primary inputs). This set of TDF vectors generated from stuck-at vectors is then applied to the circuit and subjected to hazard detection. Again the detected faults from this round are dropped and the undetected faults are taken as the new fault list to be detected. The fourth round of ATPG generation includes a simple random TDF generation of 1000 random vectors which are filtered for hazards and applied to the circuit. The total detected faults from all the rounds were concatenated and the final OHF coverage is calculated.

Table 5.1 illustrates the effectiveness of the above described ad-hoc methodology. Column 1 of the table lists the ISCAS85 circuits which were chosen for OHF test generation and coverage. Column 2 shows the unrestricted TDF coverage obtained. Column 3 demonstrates the effective total OHF coverage obtained by applying the above described flow. Column 4 shows the effectiveness of the first step of our flow i.e. filtered ATPG TDF vectors. Column 5 gives us the coverage of filtered TDF vectors using stuck vectors as seed. Column 6 is the coverage obtained from TDF vectors formed by using one-bit change vectors. Column

7 is the effectiveness of the filtered 1000 random TDF vectors. The OHF coverage obtained was as close as to the unrestricted ATPG TDF coverage. The major contribution to the OHF coverage was given by the ATPG vectors themselves which amounted to around 35% on an average. However the contribution of random vectors amount to an average of about 25% to the OHF coverage. Random vectors need to be avoided in a systematic OHF ATPG methodology. We have introduced a OHF strategy which avoids the use of random vectors in the next section.

5.2 N-detect OHF testing of combinational logic

The ad-hoc test generation methodology for OHF testing was a semi-random process. It involved the use of random vectors which are generally not preferred in a systematic test generation procedure of this nature. The novel approach used to counteract this semi-random process and help generate a systematic and efficient way to generate OHF tests is explained in the flow diagram shown in Figure 5.2. Our OHF TDF generation strategy relies heavily on the ability to generate multiple diverse TDF vectors for the same targeted TDF fault so as to maximize the probability of detecting the fault. This helps if many of the tests are invalidated because of hazards at the output lines of the circuit-under-test. The two step strategy remains the same. The first step is to create a rich set of TDF patterns to act as Output Hazard-Free tests. The second step involves filtering these TDF tests for hazards and applying them to the circuit-under-test to get the desired TDF Output Hazard-Free coverage. In the first step, our strategy uses incremental N-detect transition delay vectors generated from conventional TDF ATPG to generate Output Hazard-Free tests. The success of this incremental N-detect methodology lies in adapting the ATPG

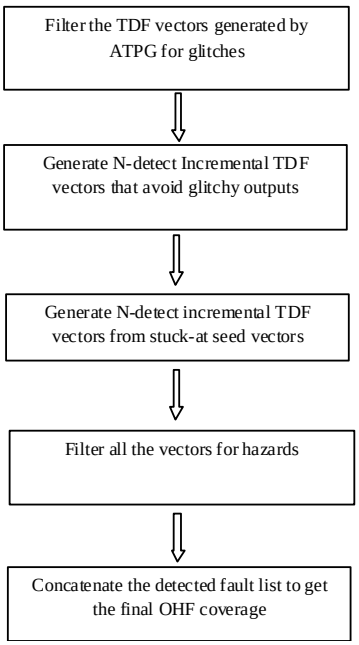

Figure 5.2: N-detect Flow for OHF test generation for combinational logic

tool to the constraint posed by the timing screen used to detect hazards. This adaptation of the ATPG tool requires profiling the circuit to identify the most glitchy output lines. This profiling is done so as to avoid ATPG effort to generate transition delay vectors that would propagate the faults through these glitchy output lines. Every circuit undergoes this profiling for hazards using a stimulus of 1000 random vectors and the above described min-max delay simulator. Two versions of the circuit are then used for TDF test generation. One version contains the most glitchy outputs to be masked out and the other version is the unedited version of the same circuit. The two versions help the ATPG to favor outputs that are statistically observed to have fewer hazards for propagating faults. This creates a rich set of TDF patterns both from the unedited version of the circuit and the masked out glitchy outputs. We then apply the incremental N-detect transition delay vector generation technique on these two versions of the circuit. The first run is with N=1 detect TDF vectors

generated from the conventional ATPG. The faults that are detected by this run (N=1) are dropped and the pattern set from this run is preserved. The undetected fault list from this run serves as a new fault list for the next run (N=2). This sequence of steps for the runs (from N=1 to N=2 and so on) involving dropping the detected faults, and forwarding the undetected fault list to the next run and preserving the vectors, continuing until the fault coverage is saturated. The combined test pattern set from these runs on both versions of the circuit are appended to each other to form a comprehensive test set. This test pattern set obtained from TDF ATPG is further augmented by a unique methodology for obtaining additional TDF vectors for the same fault based on stuck-at vectors obtained from N-detect stuck-at ATPG. We use the same incremental N-detect flow to generate multiple stuck-at fault vectors. The TDF vectors created from the stuck-at vectors follow the same principle as used in the ad-hoc technique. The vector v2 from a TDF vector pair [v1,v2] is the stuck-at vector derived from a conventional ATPG and the vector v1 is the one-bit shifted vector of vector v2. Our novel ATPG based N-detect TDF test generation methodology creates a richer set of TDF patterns, which are more likely to contain the desired OHF TDF test patterns. Recall that the stuck-at vectors for the corresponding undetected TDF faults are generated as a stuck-at-0 for a slow to rise transition fault and stuck-at-1 for a slow to fall transition fault.

The results of this test generation methodology are presented in the Table 5.2. Column 1 of the table lists the ISCAS85 circuits which were chosen for OHF test generation and coverage. Column 2 tells us the unrestricted TDF coverage obtained. Column 3 demonstrates the effective total OHF coverage obtained by applying the above described flow. Column 4 shows the effectiveness of the first step of our flow, i.e. filtered N-detect incremental ATPG

Circuit	Unrestricted ATPG (%)	OHF (%)	Filtered ATPG Vectors(%)	N-detect (%)
C17	100	**94.11**	44.11	50
C432	99.17	**95.29**	38.03	57.26
C880	99.94	**92.83**	43.98	48.85
C1355	98.17	**92.02**	27.89	64.13
C1908	99.3	**95.51**	34.19	61.32
C2670	99.25	**96.04**	35.99	60.05
C3540	99.57	**94.36**	46.99	47.37
C5315	99.76	**96.53**	39.99	56.54
C6288	99.98	**95.44**	33.99	61.45

Table 5.2: OHF Test coverage on ISCAS85 circuits using N-detect methodology

TDF vectors and N-detect incremental TDF vectors using stuck vectors as seed. The table clearly shows the effectiveness of the Incremental Methodology as opposed to using the random vectors in the ad-hoc methodology. The incremental N-detect methodology provides a TDF OHF coverage of more than 50% and the total OHF coverage obtained is nearly the same as that of the unrestricted TDF ATPG coverage.

5.3 Observations

Devising a strategy for developing Output Hazard-Free tests for combinational logic is a question of generating multiple flavors of TDF vectors which will get through the filtering procedure. It is easier to generate multiple variants of TDF vectors for combinational circuits than for sequential circuits, where the scan environment enforce restrictions on the generation of TDF vectors. However, it would be useful to observe certain strategies and their effectiveness in OHF test generation to devise a similar OHF strategy for sequential circuits. It can be clearly seen that the conventional ATPG generated TDF vectors contribute significantly towards OHF coverage. The filtered TDF vectors contributed a maximum of

44% to a minimum of 28% in OHF coverage. The next best effective coverage was obtained from N-detect vectors. However, to boost the OHF coverage close to the unrestricted TDF coverage, it seems that the incremental TDF vectors prove to be a good canditate.

Chapter 6

Output Hazard-Free (OHF) Test Generation Methodology for sequential circuits

6.1 Introduction

This chapter presents our OHF TDF test generation approach for Launch-On-Shift (LOS) scan delay tests, including coverage results for the ISCAS89 benchmark circuits. We also present the OHF results for Launch-On-Capture (LOC) and coverage results for combined LOS and LOC tests. We have also included a study on the impact of process variation on OHF TDF coverage.

Recall that a Transition Delay Fault (TDF) test is a two pattern test [v1, v2] which causes a rising (falling) transition at the target node, while vector v2 is simultaneously also a stuck-at-0 (stuck-at-1) test for the node. The basic objective is to generate TDF vectors that detect transition delay faults without generating a hazard on the observed output lines. The output lines on which the hazards are generated for a specific test pattern [v1, v2] are masked out and ignored. The test generation proceeds in two steps: conventional TDF test patterns are first generated and then a timing simulator is used to filter out those tests that can potentially display hazards at the output. Additional TDF tests are then generated for the dropped faults, and the process repeated until sufficient coverage is obtained. Our timing simulation also accounts for the significant process variations observed in advanced nanometer technologies. The results indicate that acceptable coverage can be achieved no more than 10% below the unconstrained TDF coverage for both LOS and LOC tests, even in presence of significant process variation.

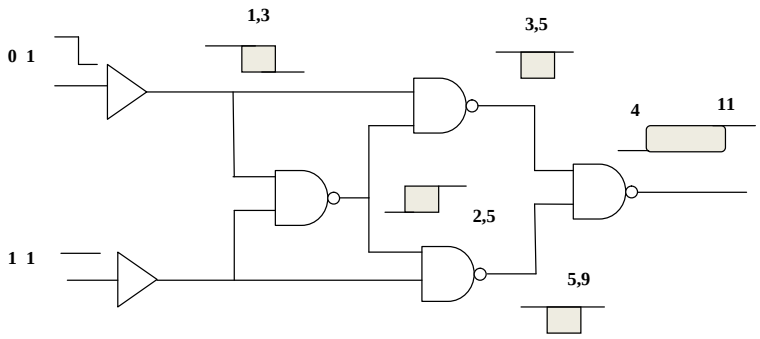

Figure 6.1: Min-max delay simulator

Our OHF TDF generation strategy relies heavily on the ability to generate multiple diverse TDF vectors for the same targeted TDF fault so as to maximize the probability of detecting the fault, even if many of the tests are invalidated because of output hazards. In the first step, our strategy uses N-detect transition delay vectors generated from conventional TDF ATPG that is configured to drive faults effects towards outputs that are statistically observed to have fewer hazards. This test pattern set obtained from TDF ATPG is further augmented by a unique methodology for obtaining additional TDF vectors for the same fault based on stuck-at vectors obtained from N-detect stuck-at ATPG. The stuck-at vectors generated from the conventional ATPG provide the v2 vector in the two pattern TDF delay test, [v1, v2], for the target fault. The v1 vector to be associated with each v2 is constrained by the scan-chain structure. The techniques for generation of vector v1 depend upon the capture procedures applied in the scan environment. For a Launch-On-Shift (LOS) test, vector v1 for the two pattern delay test is simply obtained by using a one shift in the direction opposite to that of the normal scan shift. This v1 however does not guarantee the desired transition at the target node when applied as an LOS test; several v2 vectors from the N-detect stuck-at set for v2 are therefore tried until

the desired TDF test is obtained. For Launch-On-Capture (LOC) tests, finding a v1 vector to go with the v2 vector generated by stuck-at ATPG is more challenging; the v1 vector is obtained as a stimulus for the combinational block that leads to the v2 vector as the next state. Our novel ATPG based N-detect TDF test generation methodology creates a richer set of TDF patterns, which are more likely to contain the desired OHF TDF test patterns. OHF TDF test patterns are next selected from a candidate TDF test using efficient timing simulation to achieve high coverage tests as described below. The second step of our test generation methodology involves filtering of the above generated diverse set of transition delay patterns to eliminate those tests that generate hazards at the observed outputs. This filtration is achieved by using a min-max delay simulator developed by other fellow researchers [59]. The min-max simulator uses a bounded delay model, wherein each gate is assigned the lower and upper bounds for delay, also called the min-max delay specification [60]. These delays depend on the best and worst case switching delays (gate inertia) for the gate. To define and represent signal timing in the bounded delay model, we use the term "ambiguity" region. This timing region is shown in Figure 6.1 as the shaded regions at the outputs of the gates. This ambiguity region represents an area of signal uncertainty, i.e. the signal may transit in that region, possibly more than once, creating a hazard, but the exact timing of the transitions is uncertain. The simulation makes the conservative assumption that a hazard always occurs at the output of a gate if the ambiguity region is large enough to accommodate the multiple transitions needed to create the hazard, based on the most favorable (minimum) switching delays for the gate. A circuit output can be termed hazard-free for the TDF test vector applied if there is no possibility of a hazard. On the other hand if the test stimulus applied to the circuit generates hazards on certain output

lines, these output lines are masked and not observed for fault detection. The min-max delay simulator can also be used to deal with performance variability occurring due to manufacturing process by including a tolerance with the use of bounded delay model [59, 60]. For the simulations results presented in this paper, this variance is conservatively taken to be 30%, i.e. any switching delay can vary 30% from its nominal value. This min-max modeling approach dramatically decreases the computational complexity that is required to filter transition delay vectors as compared to full circuit SPICE simulation, which is not a feasible option for large circuits. Since many of the transition delay vectors get invalidated due to output hazards detected by the min-max delay filtering procedure, we need N-detect TDF vectors to cover the dropped faults. As outlined above, these are obtained by using our novel techniques to generate multiple TDF test vectors for a given fault.

6.1.1 Profiling and Incremental N-detect methodology

While standard TDF ATPG can generate N-detect TDF vectors, it is usually unable to provide a diverse enough N-detect vector set to give a satisfactory coverage after the vectors are screened for the hazards. Based on observations over many experiments, we have developed an effective method for incrementally creating additional TDF vectors for a given fault. Figure 6.2 illustrates the step by step procedure for the incremental TDF test generation methodology. The success of this incremental N-detect methodology lies in adapting the ATPG tool to the constraints posed by the timing screening that is used to detect hazards. This adaptation of the ATPG tool requires a profiling of a circuit to identify the most glitchy output lines. This profiling is done so as to avoid ATPG effort to generate transition delay vectors that would propagate the faults through these glitchy

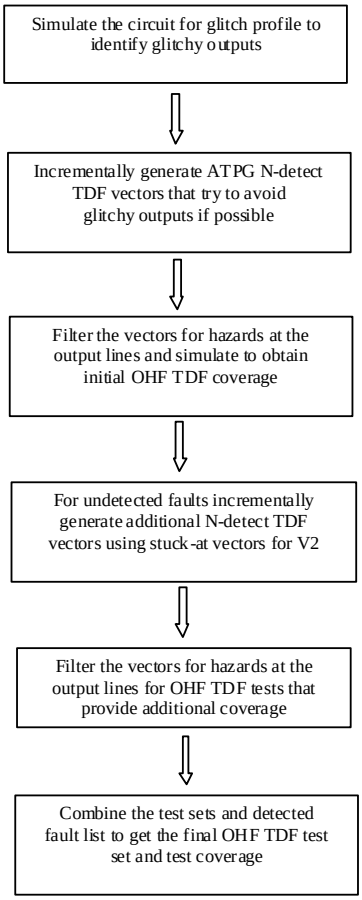

Figure 6.2: The ATPG Flow for OHF test generation for sequential logic

output lines. Every circuit undergoes this profiling using a stimulus of 1000 random vectors and the above described min-max delay simulator. We create two versions of every circuit during test generation. One version of the circuit has those output lines that have a high likelihood of a hazard (above some threshold) constrained to a 'X'. This constraining of the most glitchy output lines to an unobserved value forces the ATPG to avoid these lines for fault propagation. The second version of the circuit is the normal netlist without any constraint. These two versions of the circuit are provided to the ATPG for transition delay vector generation. We then apply the incremental N-detect transition delay vector generation technique on these two versions of the circuit. This technique involves multiple run of N-detect TDF vector generation. The first run is with N=1 detect TDF vectors generated from the conventional ATPG. The faults that are detected by this run (N=1) are dropped and the pattern set from this run is preserved. The undetected fault list from this run serves as a new fault list for the next run (N=2). This sequence of steps involving dropping of the detected faults, forwarding the undetected fault list to the next run and preservation of the vectors is continued till the fault coverage is saturated. This cyclic process is repeated for both versions of the circuit. The combined test pattern sets from these runs on both versions of the circuit are appended to each other to form a comprehensive test set. This unique approach of using constraint outputs and incremental TDF vectors forces the ATPG to generate multiple TDF vectors. This in turn enhances the probability of obtaining a filtered TDF test set which is guaranteed to prevent a hazard on the output line. The cumulative undetected fault set generated by the above run is then used for generating stuck-at vectors from conventional ATPG. Here too we use the same two versions of the circuit obtained by glitch profiling. The stuck-at vectors for the corresponding undetected

TDF faults are generated as a stuck-at-0 testsfor a slow to rise transition fault and stuck-at-1 for a slow to fall transition fault. We use the same incremental flow to generate N-detect stuck-at faults. These stuck-at fault vectors form the vector v2 in the basic two pattern [v1, v2] delay test. The corresponding v1 for this v2 is generated using mimic techniques for Launch-On-Shift and Launch-On-Capture procedures.

6.2 Launch-On-Shift (LOS)Test Generation

Launch-On-Shift (LOS) tests involve the use of the two pattern [v1, v2] tests for transition delay faults, where v2 is the scan shift vector for v1. Recall that a hazard can potentially appear at the output for the applied two pattern tests and be propagated (sensitized) at the output. Therefore it is essential to generate multiple two pattern TDF vectors to propagate faults through outputs that do not have a hazard. We use our innovative technique of incremental vector generation to achieve the above objective. The first phase involves implementing the glitch profile technique using the min-max delay simulator to create the two versions of every test circuit as explained earlier. The circuits are first targeted using conventional N-detect TDF vectors generated from the conventional ATPG. Multiple incremental runs are executed from N=1 to N=x where x is the stage where the fault coverage has saturated.

These incremental runs start by applying N=1 detect TDF vectors on the two versions of the circuit. These vectors are then filtered for hazards using the min-max delay simulator and then reapplied to the circuit. The pattern set from the run N=1 is preserved. The undetected fault set from this run (N=1) is forwarded as a new fault set for the two versions of the circuit. Then an incremental N-detect run (N=2) of TDF vectors are generated from

the conventional ATPG. The sequence of filtering the vectors for hazards and forwarding the undetect fault list to the next run of N-detect test is continued. This systematic approach of generating N-detect patterns and filtering for hazards is continued until the undetected fault set that is to be forwarded remains constant, i.e. the maximum attainable Output Hazard-Free Transition Delay Fault coverage has been obtained from these conventional ATPG delay test vectors. A cumulative pattern set is obtained by combining these filtered TDF patterns (N=1 to x). The undetect fault set from the above conventional TDF incremental run is next used to generate stuck-at test for those faults using stuck-at ATPG. The stuck-at 0 test is targeted to obtain v2 for a slow to rise transition and the stuck-at-1 test is targeted for a slow to fall transition. These stuck-at vectors are then used as vector v2 in the conventional TDF test pattern pair [v1, v2]. The corresponding vector v1 is obtained by using one shift in the direction opposite to that of the normal scan shift. These vector pairs generated are then filtered for hazards. We use the same approach of forwarding the undetect fault set (from filtered vectors) to a second run, where N=2 detect stuck-at tests are generated. This novel technique for generating TDF test vectors using stuck-at vectors is applied with the same incremental N-detect strategy, as explained earlier to generate multiple stuck vectors (v2) for every undetected TDF fault. This results into a diverse pattern set which increases the possibility of obtaining a hazard-free delay test.

Table 6.1 presents the simulation results for the OHF coverage obtained using Output Hazard-Free TDF Tests on ISCAS 89 circuits using LOS capture procedure. Column 2 shows the saturated coverage from incremental conventional TDF vectors. Column 3 shows the residual coverage obtained from TDF vectors generated by using stuck-at vectors as seed. Column 4 gives the total OHF Delay coverage. Recall that the timing simulations

Circuit	N-detect transition	N-detect stuck at	Total	ATPG
S208	57.34	19.67	77.01	89.7
S298	60.11	16.44	76.55	83.25
S344	67.58	15.4	82.98	94.21
S349	66.65	20.37	87.02	93.54
S382	50.58	24.25	74.83	89.91
S386	61.22	13.09	74.31	81.21
S400	62.31	16.78	79.09	89.68
S420	47.14	27.63	74.77	85.53
S444	56.78	16.39	73.17	85.65
S510	60.38	23.44	83.82	90.48
S526	54.87	17.22	72.09	86.92
S526n	55.23	14.67	69.9	86.55
S641	68.86	21.51	90.37	95.39
S713	62.11	18.44	80.55	89.76
S820	42.39	22.13	64.52	80.11
S832	46.77	22.63	69.4	79.4
S953	64.51	24.54	89.05	94.99
S1196	54.39	23.82	78.21	88.21
S1238	53.86	11.47	65.33	78.35
S1423	68.76	20.71	89.47	96.13
S1488	42.67	24.65	67.32	77.58
S1494	46.75	17.21	63.96	80.14
S5378	58.76	20.01	78.77	93.12
S9234	61.26	18.92	80.18	86.28
S13207	68.54	19.21	87.75	90.77
S15850	61.22	17.69	78.91	88.88
Ave.	**57.7**	**19.6**	**77.3**	**87.5**

Table 6.1: Output Hazard-Free test coverage on ISCAS89 benchmark circuits in LOS mode

used to eliminate the possibility of hazards on circuits outputs, conservatively allow for a 30% random variability in the delays associated with individual transitions from their nominal (min-max) values. It can be clearly seen that the average OHF coverage obtained was 77.3% which is just 10% below the unrestricted TDF coverage. The unique technique of creating TDF vectors from stuck-at vectors provide about 20% of the OHF coverage thus proving their effectiveness.

6.3 Launch-On-Capture (LOC) Test Generation

Conventional Launch-On-Capture transition delay test involves a stimulus-response mechanism to create a two pattern [v1, v2] delay test. We generate multiple delay patterns using a similar strategy as employed in generating LOS test set.

The initial phases, like glitch profiling to generate the two versions of the circuit and using an incremental methodology on conventional N-detect TDF LOC vectors generated from the traditional ATPG, remain the same. The cumulative filtered delay test coverage after this phase is noted and the filtered test patterns are preserved. The undetected fault set from this cumulative run is used as a fault set to generate stuck-at fault vectors (s-a-0 for slow to rise and s-a-1 for slow to fall). These stuck-at fault vectors form the v2 vector in the two pattern [v1, v2] test. LOC test generation demands that the vector v1 has to be the stimulus for the circuit in such a way that v2 would be the expected response. Hence to generate the corresponding v1, we use v2 as a constraint on our circuits and force the ATPG to generate the corresponding stimulus using a stuck-at test approach. The stimulus obtained is nothing but the vector v1 which completes our required [v1, v2] delay test set. These test sets are then filtered for hazards and the same approach of forwarding the

Circuit	N-detect transition	N-detect stuck at	Total	ATPG
S208	48.92	10.31	59.23	74.11
S298	60.45	20.89	81.34	82.16
S344	66.78	22.45	89.23	93.91
S349	63.56	27.56	91.12	93.24
S382	55.42	20.67	76.09	82.45
S386	47.87	13.44	61.31	83.92
S400	57.28	18.51	75.79	85.78
S420	50.13	17.36	67.49	83.12
S444	46.14	18.58	64.72	78.67
S510	57.88	21.28	79.16	89.49
S526	41.35	15.78	57.13	83.78
S526n	40.01	18.76	58.77	74.94
S641	60.61	24.35	84.96	90.35
S713	59.78	18.83	78.61	84.14
S820	44.76	11.34	56.1	77.12
S832	36.36	16.45	52.81	84.64
S953	63.93	23.52	87.45	95.53
S1196	49.43	23.9	73.33	84.19
S1238	49.54	25.57	75.11	85.56
S1423	55.43	19.97	75.4	89.26
S1488	57.44	17.83	75.27	88.76
S1494	56.79	17.24	74.03	91.77
S5378	61.11	18.49	79.6	91.55
S9234	48.87	16.4	65.27	83.08
S13207	58.34	21.27	79.61	84.74
S15850	53.16	16.82	69.98	70.03
Ave.	**53.5**	**19.1**	**72.7**	**84.9**

Table 6.2: Output Hazard-Free test coverage on ISCAS89 benchmark circuits in LOC mode

undetected faults to create N=2 detect stuck-at vectors is used. Again, the two pattern set is completed for this run by generating vector v1 as a stimulus-response to these N=2 stuck vectors. This process of creating incremental N-detect stuck vectors, generating v1 from this incremental run and filtering the combined [v1, v2] for hazards is continued until the fault coverage gets saturated.

Table 6.2 presents the simulation results for the coverage obtained using Output Hazard-Free TDF Tests on ISCAS 89 circuits using LOC capture procedure. Column 2 demonstrates the effectiveness of the incremental conventional TDF vectors obtained from the ATPG. Column 3 shows the effectiveness of the use of the innovative technique of generating TDF vectors from the stimulus-response behavior (using N-detect stuck-at vectors). Column 4 shows the effective hazard-free LOC coverage obtained. As compared to column 5 which is the unrestricted LOC coverage, we have again achieved reasonable fault coverage. Again we were able to achieve an average OHF coverage of about 73% which is nearly 10% below the unrestricted TDF coverage. Using the ATPG to generate our v1 from the stuck-at seed created a rich set of TDF vectors which gave us a significant 20% coverage boost.

6.4 LOC and LOS combined OHF test generation

LOC and LOS tests can be combined to achieve even higher overall test coverage. The LOC + LOS coverage in Table 6.3 was obtained by combining the detected fault list from the individual hazard-free LOC and LOS tests. The last column in Table 6.3 shows the total effective coverage obtained from LOC and LOS combined tests. Results indicate that acceptable coverage can be achieved, no worse than about 10% below the unconstrained TDF coverage for both LOS and LOC tests.

Circuit	LOS	LOC	LOS+LOC
S208	77.01	59.23	80.95
S298	76.55	81.34	86.67
S344	82.98	89.23	91.73
S349	87.02	91.12	93.86
S382	74.83	76.09	78.42
S386	74.31	61.31	76.86
S400	79.09	75.79	81.31
S420	74.77	67.49	77.89
S444	73.17	64.72	76.03
S510	83.82	79.16	85.3
S526	72.09	57.13	74.68
S526n	69.9	58.77	73.14
S641	90.37	84.96	93.57
S713	80.55	78.61	85.48
S820	64.52	56.1	70.11
S832	69.4	52.81	74.56
S953	89.05	87.45	92.15
S1196	78.21	73.33	83.22
S1238	65.33	75.11	81.18
S1423	89.47	75.4	93.71
S1488	67.32	75.27	80.03
S1494	63.96	74.03	79.53
S5378	78.77	79.6	84.46
S9234	80.18	65.27	87.68
S13207	87.75	79.61	89.75
S15850	78.91	69.98	83.34
Ave.	**77.28**	**72.65**	**82.91**

Table 6.3: OHF Test Coverage for ISCAS89 benchmark circuit in LOC+LOS mode

6.5 Effect of variations on OHF coverage

The detection of small (fine) delay defects that fall within circuit timing margins in the presence of process variations is proving extremely challenging. Given the high levels of performance variability from normal process variations observed in current technologies (path switching delays can easily span a 2X range across production lots), it is essential to study the delay simulation in the presence of significant process variations.

Our timing simulation in the above runs allowed for 30% variations in the inertial delay of each gate. This was done to mimic the presence of realistic process variations. However we have set the above runs with process variations reaching from 10% to 50%. Table 6.4 shows the result of this setup.

Table 6.4 presents the simulation results for the OHF coverage obtained by changing the tolerance from 10% to 50%. The OHF coverage noticeably decreases with increase in tolerance. On an average there is a 10% decrease in coverage between a tolerance of 10% and 50%. Certain output lines in circuits are more prone to variability in tolerance. These output lines generally behave as good candidates for hazard generation. The use of glitch profiling helped to avoid such glitchy output lines. This unique use of the profiling technique helped to reduce the impact of such a huge variation difference in the OHF coverage to 10%.

6.6 Discussions

Table 6.5 and Table 6.6 provide a comparison analysis on the OHF coverage between our work and the previous work [10] which proposed logical masking as a technique for Hazard-Free test generation. Results from this previous study suggested that the average lower bound that can be obtained in OHF testing in LOS mode is 63% and the average higher

Circuit	10%	20%	30%	40%	50%
S208	84.34	81.48	80.95	76.45	74.83
S298	89.32	87.24	86.67	84.57	78.46
S344	92.05	91.73	91.73	88.47	83.64
S349	94.13	93.86	93.86	91.12	88.67
S382	79.56	79.56	78.42	76.85	74.66
S386	80.64	78.83	76.86	74.18	73.43
S400	82.78	82.05	81.31	79.64	78.58
S420	84.78	81.61	77.89	75.18	72.53
S444	78.57	77.15	76.03	71.22	69.45
S510	87.53	85.16	85.3	81.42	78.67
S526	82.57	79.32	74.68	71.18	71.18
S526n	76.23	74.74	73.14	71.37	69.45
S641	94.12	93.57	93.57	89.23	88.57
S713	89.23	86.28	85.48	83.51	81.74
S820	72.18	71.46	70.11	67.33	64.95
S832	77.18	75.68	74.56	70.14	68.3
S953	92.78	92.64	92.15	89.31	87.45
S1196	85.31	84.5	83.22	81.65	74.37
S1238	83.01	81.99	81.18	79.08	73.61
S1423	95.45	94.13	93.71	87.1	82.78
S1488	83.52	81.49	80.03	79.67	77.31
S1494	81.76	79.08	79.53	77.45	75.41
S5378	86.65	85.1	84.46	82.35	82.35
S9234	88.28	87.91	87.68	85.72	84.01
S13207	89.12	89.75	89.75	80.66	80.05
S15850	85.35	84.23	83.34	78.64	76.43
Ave.	**85.25**	**83.87**	**82.91**	**79.75**	**77.34**

Table 6.4: Effect of variations on OHF coverage in LOS+LOC mode

Circuit No	LB from Logical Masking %	OHF %
S208	60.1	77.01
S298	60.74	76.55
S344	82.7	82.98
S349	81.66	87.02
S382	70.94	74.83
S386	51.17	74.31
S400	68.62	79.09
S420	60	74.77
S444	59.57	73.17
S510	57.84	83.82
S526	65.78	72.09
S526n	65.87	69.9
S641	87.52	90.37
S713	53.09	80.55
S820	36.83	64.52
S832	36.06	69.4
S953	71.09	89.05
S1196	57.82	78.21
S1238	57.31	65.33
S1423	76.88	89.47
S1488	45.43	67.32
S1494	44.48	63.96
S5378	65.45	78.77
S9234	64.71	80.18
S13207	78.24	87.75
S15850	75.66	78.91
Ave.	**62.91**	**77.3**

Table 6.5: A comparative illustration of OHF results in LOS mode with previous work

Circuit No	LB from Logical Masking %	OHF %
S208	36.78	59.23
S298	54.19	81.34
S344	72.97	89.23
S349	71.92	91.12
S382	52.49	76.09
S386	34.97	61.31
S400	50.88	75.79
S420	28.21	67.49
S444	42.12	64.72
S510	55	79.16
S526	37.26	57.13
S526n	37.26	58.77
S641	80.46	84.96
S713	46.91	78.61
S820	32.87	56.1
S832	32.27	52.81
S953	70.46	87.45
S1196	52.55	73.33
S1238	50.85	75.11
S1423	56.01	75.4
S1488	42.07	75.27
S1494	41.67	74.03
S5378	73.64	79.6
S9234	37.33	65.27
S13207	53.34	79.61
S15850	51.91	69.98
Ave.	**49.86**	**72.7**

Table 6.6: A comparitive illustration of OHF results in LOC mode with previous work

bound is 78%. Using our test generation methodology we have achieved OHF coverage of about 77%, which is nearly the same as their predicted higher bound OHF coverage. It is also within 10% of unrestricted TDF coverage. We achieved significant improvement in the estimated higher bound OHF coverage in LOC mode as predicted by the previous study. As compared to their prediction of achievable higher bound average OHF coverage in LOC mode of 66%, we achieved coverage of 73% for the same which is again within 10% of the unrestricted OHF coverage. There is a significant improvement in OHF coverage in the LOC mode. The vector generation technique used by Singh et al. was random TDF vectors. Random TDF vectors do not give a sufficient OHF coverage in the LOC mode. In the Launch-On-Capture mode, the circuit response is designed to be the second vector v2, which acts as the stuck fault vector. The probability that a random TDF vector would generate as effective a LOS test vector as an ATPG is considerably less. We have introduced a systematic technique for generating OHF transistion delay fault vectors that can provide a test coverage within 10% of the unconstrained TDF coverage.

Chapter 7

Conclusions

While there has been some prior work, we are the first to present a systematic ATPG based approach for generating Output Hazard-Free (OHF) Transition Delay Fault (TDF) tests for scan-based delay testing. Our timing simulation also accounts for the significant process variations observed in advanced nanometer technologies. Also in this work we have demonstrated how Output Hazard-Free TDF test generation can be implemented by using a systematic, yet low resource intensive, strategy. Such a methodology requires that outputs be correctly identified if they can potentially display a hazard during the application of individual transition delay vectors. We have introduced new steps like the incremental N-detect, profiling for adapting the ATPG, etc., which greatly enrich the TDF set to provide a hazard-free test environment.

Simulation results presented here indicate that such output hazard free tests can be obtained with an average coverage of about 10% below the unrestricted transition delay fault coverage for both Launch-On-Shift (LOS) and Launch-On-Capture (LOC) modes. ATPG effort is modest as compared to using SPICE or similar full circuit simulator for obtaining the OHF TDF vectors.

Future work in this methodology should be focussed on finding a more optimistic way of modelling hazards in a test simulation. Various heuristics can also be developed to improve the effectiveness of the OHF methodology. A combination of these two would result into a more commercially viable OHF methodology that might push the test coverage to near that of unrestricted TDF coverage.

Bibliography

[1] A.Pierzynska, and S.Pilarsky,"Non-Robust versus Robust", Proceedings International Test Conference, 1995, pp. 123-131.

[2] R. Ahmadi and F. N. Najm, "Timing analysis in presence of power supply and ground voltage variations", Proc. International Conference on Computer Aided Design, 2003, pp. 176-183.

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

[4] A. Krstic, Y.-M. Jiang and K.-T. Cheng, "Delay testing considering power supply noise effects", Proceedings International Test Conference, 1999, pp. 181-190.

[5] 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", Proceedings International Test Conference, 2003, pp. 1098-1104.

[6] 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.

[7] A. D. Singh "Scan Based Testing of Dual/Multi Core Processors for Small Delay Defects", Proceedings 2008 International Test Conference, Santa Clara CA, October 2008

[8] H. Yan and A. D. Singh, "A New Delay Test Based on Delay Defect Detection Within Slack Intervals (DDSI)" IEEE Transactions on Very Large Scale Integration Systems, vol. 14, 2006, pp. 1216-1226, 2004.

[9] A. D. Singh, "A self-timed structural test methodology for timing anomalies due to defects and process variations", Proceedings International Test Conference, 2005.

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

[11] B. Kruseman, A. K. Majhi, G. Gronthoud and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing", Proceedings International Test Conference, 2004, pp. 213-222.

[12] J. Savir and S. Patil, "Broad-side Delay Test", IEEE Transactions on Computer-Aided Design, 13(8):1057- 1064, August 1994.

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

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

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

[16] H. Yan, A. D. Singh, "Evaluating the Effectiveness of Detecting Delay Defects in the Slack Interval: A Simulation Study", Proceedings of International Test Conference, 2004.

[17] Cadence Encounter Test, Product Datasheet, Aug. 2005

[18] Sreekumar Menon, Adit D. Singh, Vishwani Agrawal,"Output Hazard-Free Transition Delay Fault Test Generation",VLSI Test Symposium, 2009

[19] J.J. Liou et al., "Analysis of Delay Test Effectiveness with a Multiple-Clock Scheme," Proc. IEEE Int'l Test Conf. (ITC 02), IEEE CS Press, 2002, pp. 407-416.

[20] R. Kumar, "Interconnect and Noise Immunity Design for the Pentium 4 Processor," Intel Technology J., Q1 2001, 12

[21] S.G. Duvall, "A Practical Methodology for the Statistical Design of Complex Logic Products for Performance," IEEE Trans. VLSI, vol. 3, no. 1, Mar. 1995, pp. 112-123.

[22] A. Agrawal et al., "Statistical Delay Computation Considering Spatial Correlation," Proc. Asian South Pacific

[23] N. Hedenstiema and K. 0. Jeppson. CMOS circuit speed and buffer optimization. IEEE Trons. on Computer-Aided Design, 6(2):270-281, March 1987

[24] E. B. Eichelberger and T. W. Williams, "A logic design structure for LSI testing," in Proc. 14th Des. Autom. Conf., 1977, pp. 462-468.

[25] B. Dervisoglu and G. Strong, "Design for testability: Using scanpath techniques for path delay test and measurement," in Proc. IEEE Int. Test Conf., 1991, pp. 365-374.

[26] J. L. Avra and E. J. McCluskey, "Synthesizing for scan dependence in built-in self-testable design," in Proc. Int. Test Conf., 1993, pp. 734-743.

[27] P. Varma, "On path-delay testing in a standard scan environment," in Proc. Int. Test Conf., 1994, pp. 164-173.

[28] J. Savir, "Scan latch design for delay test," in Proc. Int. Test Conf., 1997, pp. 446-453.

[29] R. Tekumalla and P. Menon, "Delay testing with clock control: An alternative to enhanced scan," presented at the Int. Test Conf., Atlantic City, NJ, 1997.

[30] V. S. Iyengar, B. K. Rosen, and I. Spillinger, "Delay test generation: I. Concepts and coverage metrics," in Proc. Int. Test Conf., 1988, pp. 857-866.

[31] B. Kruseman, A. K. Majhi, G. Gronthoud, and S. Eichenberger, "On hazard-free patterns for fine-delay fault testing," in Proc. Int. Test Conf., 2004, pp. 213-222.

[32] K. T. Cheng, "Transition Fault Testing for Sequential Circuits," IEEE Trans. on Computer Aided Design of Integrated Circuits and Systems, vol. 12, no. 12, pp. 1971-1983, Dec. 1993.

[33] H. Konuk, "On Invalidation Mechanisms for Non-Robust Delay Tests," in Proc. International Test Conference, Oct. 2000, pp. 393-399.

[34] A. Krstic and K. T. Cheng, Delay Fault Testing for VLSI Circuits. Boston: Springer, 1998.

[35] C. J. Lin and S. M. Reddy, "On Delay Fault Testing in Logic Circuits," IEEE Trans. Computer Aided Design, vol. 6, no. 5, pp. 694-703, Sept. 1987.

[36] J. A. Waicukauski, E. Lindbloom, B. Rosen, and V. Iyengar, "Transition Fault Simulation," IEEE Design & Test of Computers, vol. 4, no. 2, pp. 32-38, Apr. 1987.

[37] S. Chandra, K. Lahiri, A. Raghunathan, and S. Dey,"Considering Process Variations During System-level Power Analysis," in Proc. International Symposium on Low Power Electronics and Design, 2006.

[38] Hawkins, A, Keshavaru, and Segura, "Para. metric Timing Failures and Defect-based Testing in Nanotechnology CMOS Digital ICs", Pmc. of NASA Symp. 2003.

[39] R. Kumar, "Interconnect and Noise Immunity Design for the Pentium 4 Processor," Intel Technology J., Q1 2001, 12 Feb. 2001; http://www.intel.com/technology/itj/q12001.htm.

[40] M. Gumm, VLSI Design Course Notes: VHDL-Modeling and Synthesis of The DLXS RISC Processor, Univ. Stuttgart, Germany, 1995.

[41] P. C. Maxwell, R. C. Aitken, K. R. Kollitz and A. C. Brown, "IDDQ and AC scan: The war against unmodelled defects," in Proc. 1996 IEEE Int. Test Conf., Oct. 1996, pp. 250-258.

[42] "So what is an optimal test mix? A discussion of the SEMATECH methods experiment," in Proc. 1997 IEEE Test Conf., Nov.1997

[43] K. Tumin, C. Vargas, R. Patterson and C. Nappi, "Scan vs. functional testing - A comparative effectiveness study on Motorola's MMC2107TM," in Proc. 2001 IEEE Int. Test Conf., Oct.-Nov. 2001, pp. 443-450.

[44] M. A. Breuer and S. K. Gupta, "Process aggravated noise (PAN): New validation and test problems," in Proc. 1996 IEEE Int. Test Conf.,Oct 1996.

[45] K. L. Shepard and V. Narayanan,"Noise in Deep Submicron Digital Design," IEEE International Conference on Computer Aided Design, San Jose, CA, Nov. 1996, pp. 524-531.

[46] W. Daasch, K. Cota, J. McNames, and R. Madge, "Neighbor selection for variance reduction in IDDQ and other parametric data," in Proc. Int. Test Conf., 2001, pp. 92-100.

[47] Cadence Inc., San Jose, CA, "Cadence encounter test," Aug. 2005, Product Datasheet.

[48] A. L. Crouch, "Exploring the Basic of AC Scan", Evaluation Engineering, July, 2004.

[49] 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.

[50] E.B. Eichelberger and E. Lindbloom, "Random-Pattern Coverage Enhancement and Diagnosis for LSSD Logic Self-Test," IBM J. Research and Development, May 1983, pp. 265-272.

[51] Z. Barzilai and B.K. Rosen, "Comparison of AC Self-Testing Procedures," Proc. IEEE Int'l Test Conf., Oct. 1983, pp. 89-94.

[52] T.M. Storey and J.W. Barry, "Delay Test Simulation," Proc. Design Automation Conf., June 1977, pp. 492-494.

[53] G. Smith, "Model for Delay Faults," Proc. IEEE Int'l Test Conf., Nov. 1985, pp. 342-349.

[54] J.A. Waicukauski et al., "Fault Simulation for Structured VLSI," VLSI Systems Design, Dec. 1985, pp. 20-32.

[55] V. S. Iyengar, B. K. Rosen, and J. A.Waicukauski, "On computing the sizes of detected delay faults," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 9, no. 3, pp. 299-312, Mar. 1990.

[56] B. Koenemann et al., "Delay test: The next frontier for LSSD test systems," in Proc. IEEE Int. Test Conf., 1992, pp. 578-587.

[57] "On the fault coverage of delay fault detecting tests," in Proc. Eur. Des. Autom. Conf. (EDAC), 1990, pp. 334-338.

[58] "On the fault coverage of gate delay fault detecting tests," IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst., vol. 16, no. 1, pp. 78-94, Jan. 1997.

[59] Saumitra Bose, Hillary Grimes and Vishwani D. Agrawal, "Delay Fault Simulation with Bounded Gate Delay Model," ITC, 2007, paper 26.3.

[60] C. J. Seger, "A Bounded Delay Race Model," in Proc. of the IEEE International Conf. Computer Aided Design, Nov. 1989, pp. 130- 133.

[61] . Agarwal, D. Blaauw, and V. Zolotov, "Statistical Timing Analysis for Intra-Die Process Variations with Spatial Correlations," in Proc. International Conference on Computer Aided Design, Nov. 2003, pp. 900-907.

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

[63] B. Kruseman, "Comparison of Defect Detection Capabilities of Current-based and Voltage-based Test Methods", European Test Workshop, 2000, pp. 175.180.

[64] D. Belete, A. Razdan, W. Schwm, R. Raina, C.Hawkins. a "Use of Dft Techniques in Speed Grading a 1 GHz+ Microprocessor". Proc. Int. Test Conf., Oct 2002,pp. 1111-1119.