Small Delay Testing Using On-chip Delay Measurement

January 2015

Wenpo Zhang

Graduate School of

Advanced Integration Science

CHIBA UNIVERSITY

（千葉大学審査学位論文）

オンチップ遅延測定を用いた微小遅延テスト

2015 年 1 月

千葉大学大学院融合科学研究科

情報科学専攻知能情報コース

張　文坡

Contents

1. INTRODUCTION 1
1.1. Background 2
1.2. Related work 6
1.3. Overview of this thesis 8
2. DELAY FAULT TESTING 11
2.1. Two-pattern test 12
2.2. LOS and LOC 13
2.3. Delay fault model 15
2.4. Gross delay and small delay 16
2.5. On-Chip Delay Measurement Method 18
3. SMALL DELAY FAULT COVERAGE IMPROVEMENT 21
3.1. SUMMARY 22
3.2. Constraint of Single-Path Sensitization 23
3.2.1. Single-Path Sensitization 23
3.2.2. Necessity of Single-Path Sensitization 24
3.2.3. Small-Delay Fault Coverage 26
3.3. Techniques for Improving Fault Coverage 28
3.3.1. Segmented Scan 28
3.3.2. Test Point Insertion 29
3.4. Evaluation 36
3.4.1. Effect of Segmented Scan 37
3.4.2. Effect of Test Point Insertion 40
3.4.3. Effect of Segmented Scan and Test Point Insertion 45
3.4.4. Effective fault coverage by the same overall hardware overhead 45
3.5. Conclusion 48
4. Scan Shift Time Reduction Using Test Compaction for On-Chip Delay Measurement 49
4.1. SUMMARY 50
4.2. Preliminaries 51
4.2.1. Terminology Related to Scan-Based Delay Testing 51
4.2.2. Operation of on-Chip Delay Measurement Method 52
4.3. The Proposed Method to Reduce Test Application Time and Test Data Volume for On-Chip Delay Measurement 55
4.3.1. LOS Operation of the On-Chip Delay Measurement 55
4.3.2. Scan-Based Test Pattern Merging 57
4.3.3. Procedure for Test Application Time and Test Data Volume Reduction 58
4.3.4. Test Application Time and Test Data Volume 59

4.4. Experimental Result 61
4.5. Conclusion 69

5. Fault Coverage Improvement and Test Compaction under LOS+LOC Test 71

5.1. SUMMARY 72
5.2. LOS Test vs. LOC Test 74
5.3. Proposed Coverage Improvement Method 76
5.4. Proposed Test Compact Procedure 79
5.5. EVALUATION 83
5.5.1. LOS Test vs. LOC Test 83
5.5.2. The Defect Coverage Improvement Effect 85
5.5.3. The Test Compaction Effect 86
5.6. CONCLUSION 90

6. CONSLUSION 91

ACKNOWLEDGEMENT 93

Reference 95

1. INTRODUCTION

1.1. Background

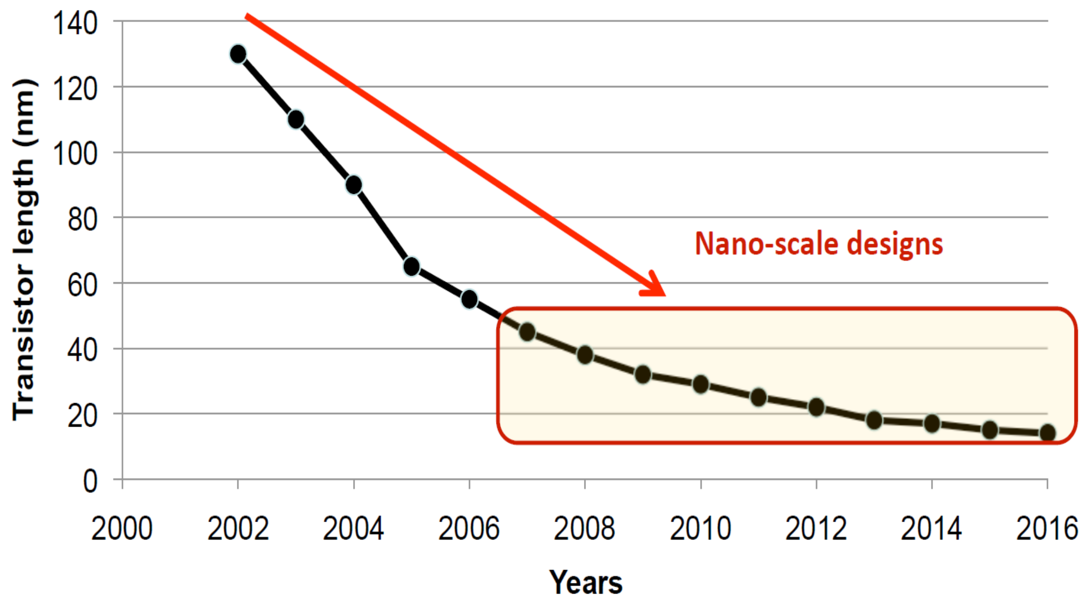


Figure 1.1 VLSI technology trend

Modern integrated circuits testing, which began with the commercial using of integrated circuits (ICs) in the early 1960s, has a history of more than 50 years [1-2]. Today, IC chips lie at the heart of ongoing advances across the electronics industry. With the development of IC chips, a small chip can provide a complex logic. The advent of high performance portable electronic systems such as the personal computer, mobile phone and wearable device are testament to this [3-5].

The scale of Integrated circuits has doubled every 18 months. Recently, as shown in Figure 1.1, *Very Large Scale Integrated circuits* (VLSI) technology scales to 45nm and below, semiconductor device scaling has significantly improved performance and circuit integration density. With the increasing system clock speed of IC chips, violations of the performance specifications are becoming a major factor affecting the product-quality level [6]. The small feature size has increased the probability that a manufacturing defect in the IC will result in a faulty chip. A very small defect can easily generate a transistor with fault or interconnect wire when the feature size is less than 45 nm. Delay defects that degrade performance and cause timing related failures are emerging as a

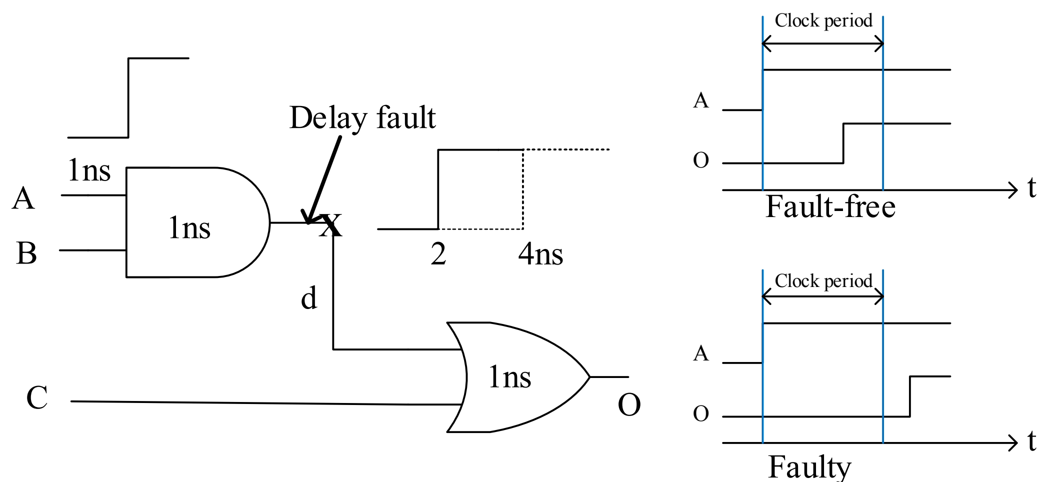


Figure 1.2 Example of delay fault

major problem in nanometer technologies. In IC chips, all changes of input signals are synchronized with the system clock signal and all outputs are expected to reach their final steady state values within one clock period after the input signals change. Thus, for a correct operation in one circuit the delay of one path should not exceed one clock period. As shown in Figure 1.2, if the transition transfer time is shorter than a clock period, we think it is fault-free and the chip operation is correct; if the transition transfer time is longer than a clock period, we think it is faulty and the chip operation is incorrect. It can lead to a logic fault on the chip if there has a delay fault. Delay testing is necessary, because many factors may delay a signal propagating along one path.

Several delay fault models and delay test methods have been proposed. Two commonly used model are transition fault and path delay fault [7]. With the growing complexity of designs, scan-based techniques of testing are becoming very popular. However, there has a precision of test clock frequency that the provided by the external automatic test equipment (ATE). The clock frequency can be affected by factors such as parasitic capacitance, resistance of probe and tester skew [8].

Small-delay defects are known to degrade in operation and cause early life failure. A small-delay defect has a defect size that is not large enough to cause a failure operation under the system clock cycle. Small-delay defects represent a significant reliability problem when resistive defects are present in a technology. For example, a minimally

connecting via caused resistive open can become a complete open fault in operation due to metal migration. Recently, increasing random process variations contribute to significant timing variability, which is indistinguishable from the effects of small-delay defects. Such variations can be beyond the performance variations caused by resistive small-delay defects. Therefore, these process variations need to be detected to improve the reliability of chips [9]. Since they might escape detection during traditional Pass-Fail delay fault testing with functional clock, small-delay defects have become a significant problem and it is essential to detect such defects during manufacturing tests [10]. Such manufacturing flaws have traditionally been eliminated through burn-in stress testing. Burn-in fallout can be as high as 0.5-1% (5,000-10,000 defect parts per million (DPPM) for large complex circuit, making stress testing necessary for high performance device such as CPUs. Unfortunately, the cost of traditional burn-in is becoming very expensive for nanometer technologies; it also appears to be losing effectiveness in accelerating certain types of early life failures [11]. As a result, industry is looking for a high quality, low-cost method to detect small-delay defect [12], [13].

To screen small-delay fault, small-delay defect screening with criteria based on statistical analysis is proposed [13]. In this technique, small-delay defects are detected as outliers. Delay distributions for each path can be obtained by measuring many chips of the same design. If a path delay time is beyond a specified time such as the three-sigma limit (users can set the specified time freely by taking into consideration the trade-off between yield and dependability), even if it is not beyond the system clock cycle, we regard it as a faulty path. Some previous works presented on-chip path delay time measurements based on this strategy [14]–[19]. By measuring delay time of the path under measurement (PUM), not only the gross delay fault but also small-delay faults can be detected. It also can obtain the amount of timing violation in the failing paths under certain environment conditions [20], [21].

However, there have some issues of testing using on-chip path delay measurement: its small-delay fault coverage is very low; and the test application time is long. In the measurement, PUMs are sensitized by delay fault test patterns. However, robust test patterns are not suitable for on-chip delay measurement. Specifically, they require test

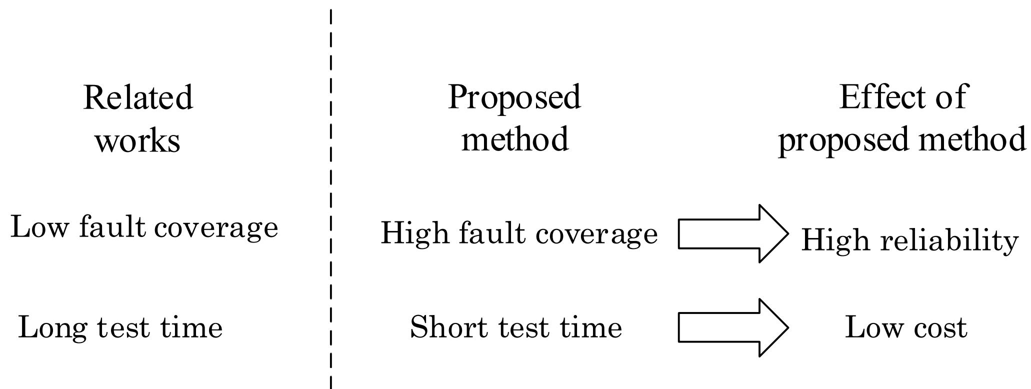


Figure 1.3 Objectives of this research

generation under the single-path sensitization condition, which causes its small-delay fault coverage to be very low. On-chip delay measurement incurs high test cost because it uses scan design, which brings about long test application time due to scan shift operation. Thus, a method reducing test application time is strongly required.

This thesis propose solutions to these problems as shown in Figure 1.3: we propose a method using segmented scan and *test point insertion* (TPI) to improve the small-delay fault coverage; we also propose a method to reduce scan shift time and test data volume using test pattern merging.

1.2. Related work

Table 1.1 Characteristics of methods of small-delay testing

Method	Resolution	Fault coverage	Test cost
Multiple clock frequencies [23-24]	○	○	×
Ring oscillator [25-27]	×	△	△
Time-to-voltage converter[28-30]	○	×	△
On-chip delay measurement[14-19]	○	×	△
Proposed method	○	○	○

○: Good   △: Not good   ×: Bad

Recently, as shown in Table 1.1, various methods for small-delay defect detection have been proposed. Methods using faster-than-at-speed have been proposed to detect delay faults [23], [24]. These methods use multiple clock frequencies that are higher than the system clock. These methods have a drawback that the test time is very long and it causes the test cost high. To detect small-delay faults, methods with delay fault testing using a ring oscillator have been proposed [25]–[27]. In these, the PUM is made a part of ring oscillator, delay of the target path can be translated into the oscillation period. However, the timing resolution is not very good. Some *time-to-voltage converter* (TVC) based schemes have been proposed [28]–[30]. The delay of the PUM is converted to a certain voltage, and by comparing the converted voltage with the reference voltage, the delay of the PUM can be obtained. These techniques give good timing resolution. However, the calibration is difficult.

Some on-chip path delay time measurement methods using embedded delay measurement were proposed [14]–[19]. In these, delay times of paths are measured. A modified *vernier delay line* (VDL) method for path delay measurement also has been proposed [14]. This delay measurement method can achieve a high-precision capability. The paper [15] presented modified boundary scan cells in which a *time-to-digital converter* (TDC) is embedded. An extension of the modified VDL technique, which has a *built-in delay measurement* (BIDM) circuit consisting of coarse and fine blocks, also has been proposed. A built-in-self delay testing methodology based on BIDM and self-calibration methods can be developed [16]. A modified VDL with small area overhead has also proposed [17]. The feature of this

method is delay range of each stage. The delay of each step increases exponentially, it reduces the hardware overhead. Thus, without decreasing delay measurement resolution, this method expands the range of delay measurement easily with small area overhead. The authors' group proposed a measurement system, which is different from the VDL method, to improve the accuracy of the measured value [18]. This method measures delay times of two paths: a path which includes the PUM, and the extra path whose length is almost equal to the redundant line of the path, is measured previously. The difference between the delay times gives the delay of the PUM. This method is able to give a precise measurement. In addition, a method with smaller execution time and circuit area has been proposed [19].

1.3. Overview of this thesis

This thesis contains 6 chapters. Chapter 2 provides introduction to delay fault testing of VLSI. Chapter 3 introduces techniques which use segmented scan and test point insertion. Chapter 4 proposes method reduces scan shift time and test data volume using test pattern merging. Chapter 5 presents a method using LOS+LOC based on a conventional method. Chapter 6 concludes this thesis.

Chapter 2 provides a brief introduction to delay fault testing of VLSI. It covers various topics such as delay fault models, delay test method, etc.

Chapter 3 introduces techniques which use segmented scan and test point insertion. Our pre-simulation results show that when using on-chip delay measurement method to detect small-delay defects, test generation under the single-path sensitization is required. This constraint makes the fault coverage very low. To improve fault coverage, this chapter introduces techniques which use segmented scan and test point insertion (TPI).

Chapter 4 proposes a method reduces scan shift time and test data volume using test pattern merging. On-chip delay measurement incurs high test cost because it uses scan design, which brings about long test application time due to scan shift operation. Our solution is a test application time reduction method for testing using the on-chip path delay measurement. The testing with on-chip path delay measurement does not require capture operations, unlike the conventional delay testing. Specifically, FFs keep the transition pattern of the test pattern pair sensitizing a *path under measurement* (PUM) (denoted as p) even after the measurement of p. The proposed method uses this characteristic. The proposed method reduces scan shift time and test data volume using test pattern merging.

To improve the small-delay defect coverage of on-chip delay measurement method with small hardware overhead, Chapter 5 presents a method using LOS+LOC based on a conventional method. We also propose a test compaction procedure under LOS+LOC that reduces scan shift time and test data volume using test pattern merging. The evaluation results show that, compare with the conventional LOS+LOC

method, the proposed method reduces the test application time by 47.87~54.02% and test data volume by 71.72~74.50%. Compare with the conventional LOS based methods in section 3 and 4, the proposed procedure can provide similar or higher defect coverage with very small hardware overhead. Specifically, the hardware overhead is 9.27~35.21% smaller than the conventional method. The proposed test compaction procedure reduces the test application time by 4.47~29.29% and test data volume by 4.46~29.96%.

Chapter 6 concludes the above chapters.

2. DELAY FAULT TESTING

2.1. Two-pattern test

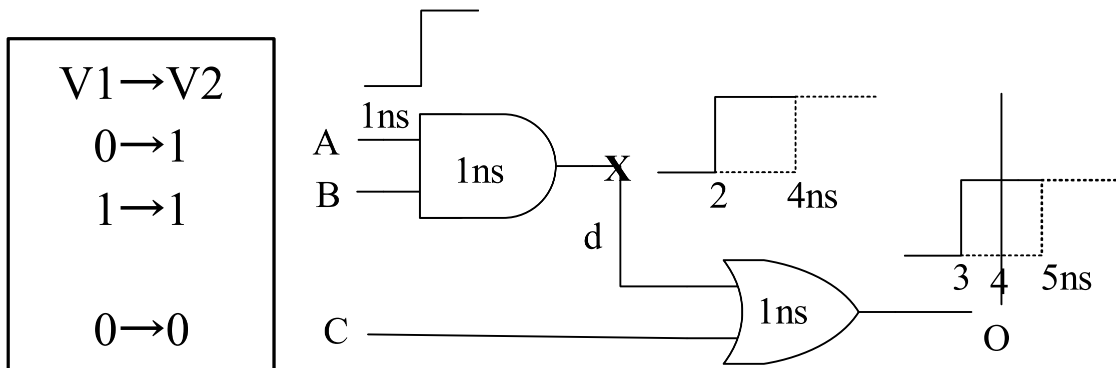


Figure 2.1 example of a part of circuit

The operation of IC chips is usually synchronized with the system clock. It is necessary that all logic elements reach steady state within a clock cycle.

Figure 2.1 shows an example of a part of circuit. All changes of input signals are synchronized with a the system clock and all outputs are expected to reach their final steady states in one clock cycle after the change of input signals. Thus, for a correct operation in one circuit the delay of one path should not exceed one system clock cycle. In order to examine the timing operation of a circuit, we must propagate a transition through the combinational path. To examine the circuit shown in Figure 2.1, we need two test patterns: 010→110. The pattern v1 applied in first is an initial pattern, the pattern v2 is a transition pattern. We call the two patterns as one pattern pair. [31]

Transitions of input signals occur at the same time with the operation of system clock. The right edge of the output transition region is determined by the last transition, or the delay of the critical path. In this Figure, the system clock is set to 4ns, and the 0→1 transition at the output O is expected before than the clock edge. Because there has a delay fault at line *d*, we cannot observe the 0→1 transition on the output O before the system clock edge. We say there have a delay fault in the circuit. A delay fault means that the delay of one or more path exceeds one system clock.

2.2. LOS and LOC

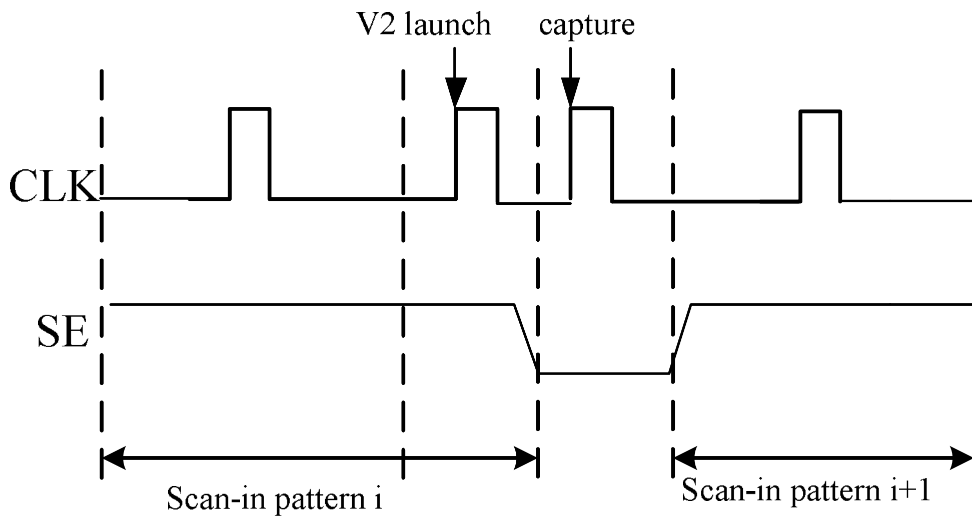


Figure 2.2 LOS

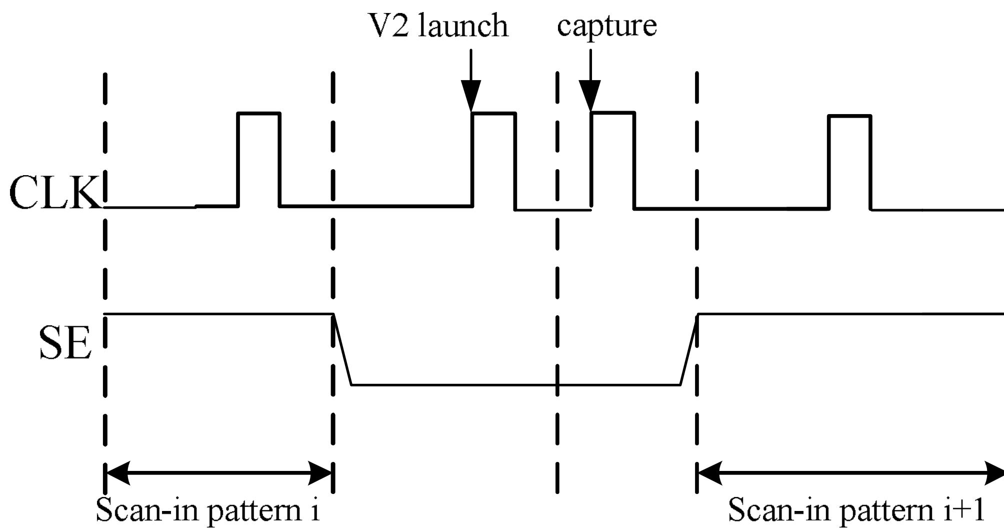


Figure 2.3 LOC

The operation of two-pattern test have 3 cycles: 1) Initialization, the circuit is initialized by apply v1; 2) Launch, the transition is launched by apply v2; and 3) Capture, we propagate the transition and capture it at an FF or output. We need use scan-based test method for two-pattern test. Depending the operation of the transition

launch and capture, two methods are widely used: *launch off shift* (LOS) (or referred as *skewed-load*), and *launch off capture* (LOC) (or referred as *broadside*). [32-33]

Figure 2.2 shows the operation of LOS. In this method, after the scan-in operation, pattern v1 is set and the circuit is set to an initial state. The second pattern is generated by 1 bit shifting the first pattern v1.

Figure 2.3 shows the operation of LOC. In this method, after the scan-in operation, pattern v1 is set and the circuit is set to an initial state. The second pattern v2 is the function response of v1.

2.3. Delay fault model

Two commonly used model are transition fault and path delay fault. The transition fault (slow-to-rise and slow-to-fall) model targets each gate output in one circuit. Path delay fault model targets the delay time of a path (accumulating delay of all the gates and lines on the path).

In one circuit, if the delay of any of its paths exceeds a specified limit (system clock), we think there have a path delay fault in the circuit. To detect a path delay fault on one path, we should to propagate a transition through the path. Therefore, to detect a path delay fault, we need to specify the path and transition type. Typically, there are three classes of path delay faults according to the sensitization criteria: single-path sensitization, robust sensitization and non-robust sensitization. [34-35]

Transition fault model assumes that there have delay fault on one gate line in the circuit. As the same with path delay fault, there have two types of transition faults: slow-to-rise and slow-to-fall. We think the delay of each gate line in one circuit should equal to its design delay. If there have delay faults on one gate line, the delay of that gate line will increase. The addition delay caused by the transition fault is assumed to be large enough that we cannot observe the transition. In other word, it made the transition time longer than the system clock. [36-38]

In one circuit, the number of transition fault is equal to twice the number of gate lines, and the number of path delay fault is exponential relationship with the number of gates. In modern circuit, transition fault model is more widely used than path delay fault. We also note that, with the increasing of the number of paths, the test generation will be very difficult [12]. This chapter uses the transition fault coverage as the small-delay fault coverage because the aim of this chapter is to detect increases of gate and line delays caused by resistive faults to reduce early life-failure. If there has a transition on one gate line, and it causes the delay time of one path that includes this fault exceed the system clock. We think we can detect this transition fault.

2.4. Gross delay and small delay

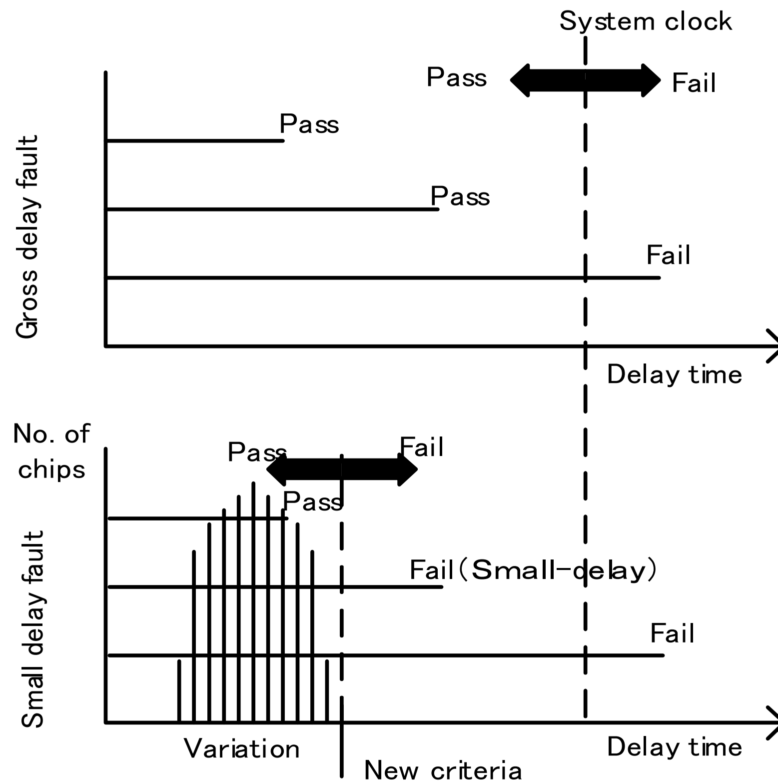


Figure 2.4 Delay fault

Figure 2.4 shows the two kinds of delay fault: gross delay and small delay. Gross delay means the delay size is beyond the system clock. We can detect gross delay using two-pattern test. If the transition can be observed before the clock edge, we think it is fault-free. If the transition cannot be observed before the clock edge, we think there has a fault. A small-delay defect has a defect size that makes the path delay is smaller than the system clock. Since they might escape detection during traditional Pass-Fail delay fault testing with functional clock, small-delay defects have become a significant problem and it is essential to detect such defects during manufacturing tests [10]. To screen small-delay fault, small-delay defect screening with criteria based on statistical analysis is proposed [13]. In this technique, small-delay defects are detected as outliers. Delay distributions for each path can be obtained by measuring many chips of the same design. If a path delay time is beyond a specified time such as the three-sigma limit

(users can set the specified time freely by taking into consideration the trade-off between yield and dependability), even if it is not beyond the system clock cycle, we regard it as a faulty path.

2.5. On-Chip Delay Measurement Method

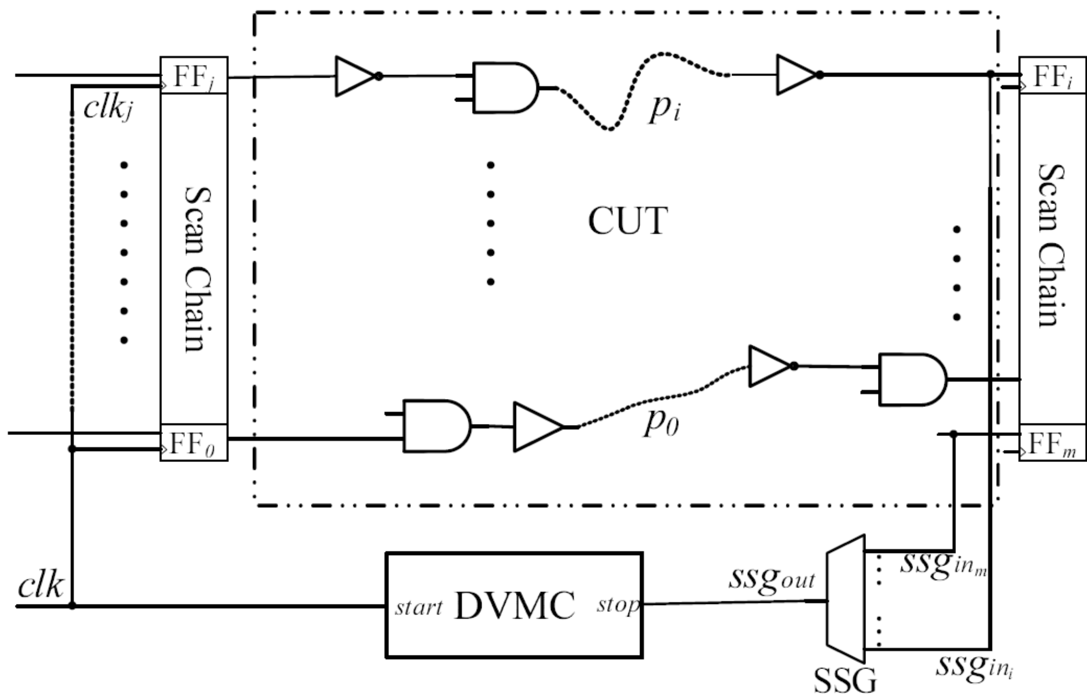


Figure 2.5 Architecture of on-chip path delay measurement system.

Some previous works presented on-chip path delay time measurements based on the technique introduced in section 2.4 [14]–[19]. By measuring delay time of the path under measurement (PUM), not only the gross delay fault but also small-delay faults can be detected. It also can obtain the amount of timing violation in the failing paths under certain environment conditions [20], [21].

Figure 2.5 shows the architecture of the on-chip path delay measurement method of [19]. The on-chip delay measurement method measures the delay time of each path including a PUM. the input (output) of the PUM is the output (input) of a flip-flop (FF) in the left (right) scan chain. The delay measurement system consists of *delay value measurement circuit* (DVMC), *stop signal generator* (SSG, which is an N-to-1 multiplexer), and *circuit under test* (CUT). The embedded delay measurement circuit DVMC has two input lines, *start* line and *stop* line. The DVMC, which is a class of TDC, consists of a delay chain and an n-bit up counter. DVMC measures the time

difference of two transitions sent to *start* line and *stop* line. The clock line of the CUT *clk* is directly connected to *start* of the DVMC; the transition of *start* triggers the measurement. The DVMC starts the measurement when a positive transition of *clk* is sent to *start*. The input line of the FF_i in the right scan chain, ssg_{ini}, is connected to the SSG; the SSG detects the transition at ssg_{ini} and sends the transition to *stop* of the DVMC, by setting the corresponding control data of SSG. The input line *clk* is the clock signal of CUT. The line clk_i is the clock line of FF_i. The input of FF_i is connected to ssg_{out} through ssg_{ini} and the SSG. The system measures a path including one clock line clk_j, a PUM p_i, and some redundant lines ssg_{ini} and ssg_{out}. For example, after the measurement of the path $p' = clk_j$-p_i-ssg_{ini}-ssg_{out}, by comparing the measured delay time with the expected delay time, small-delay defects on clk_j and p_i can be detected [19]. In this chapter, we insert one DVMC circuit in one CUT. Thus, only one path is selected to be measured for each test.

Before the measurement, a test pattern (which sensitizes p_i) is assigned to the primary inputs and FFs of CUT. SSG is controlled to send the transition propagating to the output of p_i, to ssg_{out}. Then, we start the measurement by launching a positive transition to *start*. The transition propagates to the *start* of DVMC after the rising edge of *clk*, and DVMC starts the measurement. At the moment the clock rising edge reaches FF_j, a transition is launched to p_i. The transition reaches *stop* of DVMC through p_i, ssg_{ini} and ssg_{out}. Then, DVMC stops the measurement. The measured delay time of p^0 contains the delay of redundant lines ssg_{ini} and ssg_{out}. By using the criterion of [13], we can detect small-delay defects occurring on p_i and the segments of clock trees. This method has a good measurement resolution enough to detect defects even if the path delay is short [18]. However, there is an important point for small-delay fault test. When the intended transition reaches the *stop* of DVMC through SSG, DVMC stops the measurement. If the transition on the off-input of a PUM affects measured delay time, an incorrect measurement result will be recorded. The incorrect measurement result leads to false error indications or test escapes. We can satisfy this by using the single-path sensitization condition.

3. SMALL DELAY FAULT COVERAGE IMPROVEMENT

3.1. SUMMARY

With IC design entering the nanometer scale integration, the reliability of VLSI has declined due to small-delay defects, which are hard to detect by traditional delay fault testing. To detect small-delay defects, on-chip delay measurement, which measures the delay time of paths in the *circuit under test* (CUT), was proposed.

This chapter points out a considerable issue of testing using on-chip path delay measurement: in the measurement, PUMs are sensitized by delay fault test patterns. However, this chapter reveals that the robust test patterns are not suitable for on-chip delay measurement. Specifically, they require test generation under the single-path sensitization condition, which causes its small-delay fault coverage to be very low. Thus, a method improving fault coverage is strongly required. This chapter uses the transition fault coverage as the small-delay fault coverage because the aim of this chapter is to detect increases of gate and line delays caused by resistive faults and other reasons (for example process variation) to reduce early life-failure.

This chapter gives evidence that, for improving small-delay fault coverage of on-chip delay measurement, the use of segmented scan and *test point insertion* (TPI) is efficient. Evaluation results indicate that we can get an acceptable fault coverage, by combining these techniques for *launch off shift* (LOS) testing under the single-path sensitization condition. Specifically, fault coverage is improved 27.02~47.74% with 6.33~12.35% of hardware overhead.

The rest of the chapter is organized as follows. Section 3.2 analyzes the constraint of single-path sensitization and the reasons for low small-delay fault coverage. Section 3.3 explains methods for improving fault coverage. Section 3.4 evaluates the introduced methods. Finally, section 3.5 concludes the chapter.

3.2. Constraint of Single-Path Sensitization

In this section, we introduce the necessity of single-path sensitization in on-chip delay measurement using an example. We give some simulation results to prove that the robust test is not appropriate for on-chip delay measurement. We also analyze the reasons for low small-delay fault coverage.

3.2.1. Single-Path Sensitization

If there has a transition on one gate line, and it causes the delay time of one path that includes this fault exceed the system clock. We think we can detect this transition fault. From this reason, a path through the fault should be sensitized for testing the transition delay fault. In path delay fault testing, a signal is an on-input of path p if it is on path p. If a gate g is on path p and an input line of the gate g is not on p, the line is called an off-input of p [6], [22]. A logic value is the controlling value to a gate if it determines the output value of the gate regardless of the values on the other inputs to the gate. Otherwise, we say that is a non-controlling value. Typically, there are three classes of path delay faults according to the sensitization criteria: single-path sensitization, robust sensitization and non-robust sensitization. Consider an AND gate with two inputs, this gate is a part of a target path p with one input as the on-input and the other input as the off-input for p. Table 3.1 shows the sensitization versus pairs of initial and final values in on-input

Table 3.1 sensitization versus pairs of initial and final values in on and off-input of an AND gate.

Sensitization	On-input	Off-input
Single path sensitization	T1	S1
	T0	S1
Robust	T1	S1, T1 or H1
	T0	S1
Non-robust	T1 or H1	S1, T1 or H1
	T0 or H0	S1, T1 or H1

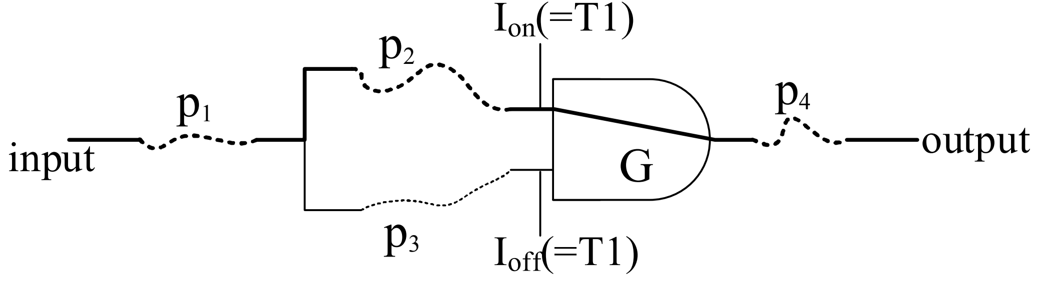


Figure 3.1 Example of PUM not satisfying single-path sensitization condition.

and off-input of the AND gate. Symbol S0 (S1) represents a stable 0 (1) value on some signal under the initial value and final value. Symbol T0 (T1) represents a 1 to 0 (0 to 1) transition. H0 (H1) represents a 0 (1) value that might have hazard.

3.2.2. Necessity of Single-Path Sensitization

Unlike the traditional delay fault test, for the on-chip delay measurement method we must consider the single-path sensitization condition. As already known, in traditional delay fault test, we have to test sensitizing PUM robustly and not non-robustly to detect delay faults on PUM regardless of the delay time to off-inputs. The same is true for the delay measurement method. However, this is not sufficient, which is explained below using the example of Figure 3.1 and the simulation data in Figures 3.2 and 3.3.

In Figure 3.1, PUM is the path p_1-p_2-G-p_4, denoted with a bold line. Assume that all sub-paths p_1, p_2 and p_4 are sensitized, and the values of on-input I_{ON} and the off-input I_{OFF} of the AND gate G are T1. From Table 3.1, PUM is robustly tested, but is not single-path sensitized. Here, we set $T_{ON}= t_1+t_2+t_{ON}+t_4$ is the delay time of the PUM, and $T_{OFF}= t_1+t_3+t_{OFF}+t_4$ is the delay time of the path p_1-p_3-G-p_4, where t_i is the delay times of p_i and, t_{ON} and t_{OFF} are the gate delay of G from I_{ON} and I_{OFF}. By measuring the delay time from the input to the output, we obtain the following time:

$T = max(T_{ON},T_{OFF})$.

If the expected delay time of t_3 is longer than t_2, we need to set the expected path delay to T_{OFF}+3σ. In other words, we should compare T with T_{OFF}+3σ. We cannot

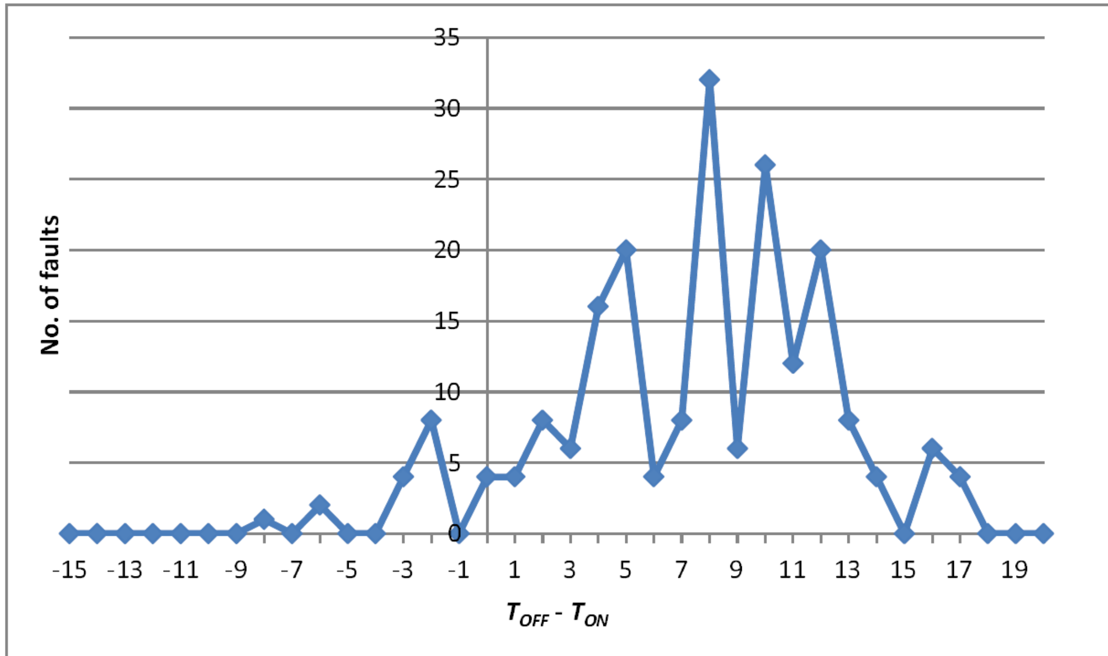


Figure 3.2 Relation of number of faults and TOFF−TON for s5378.

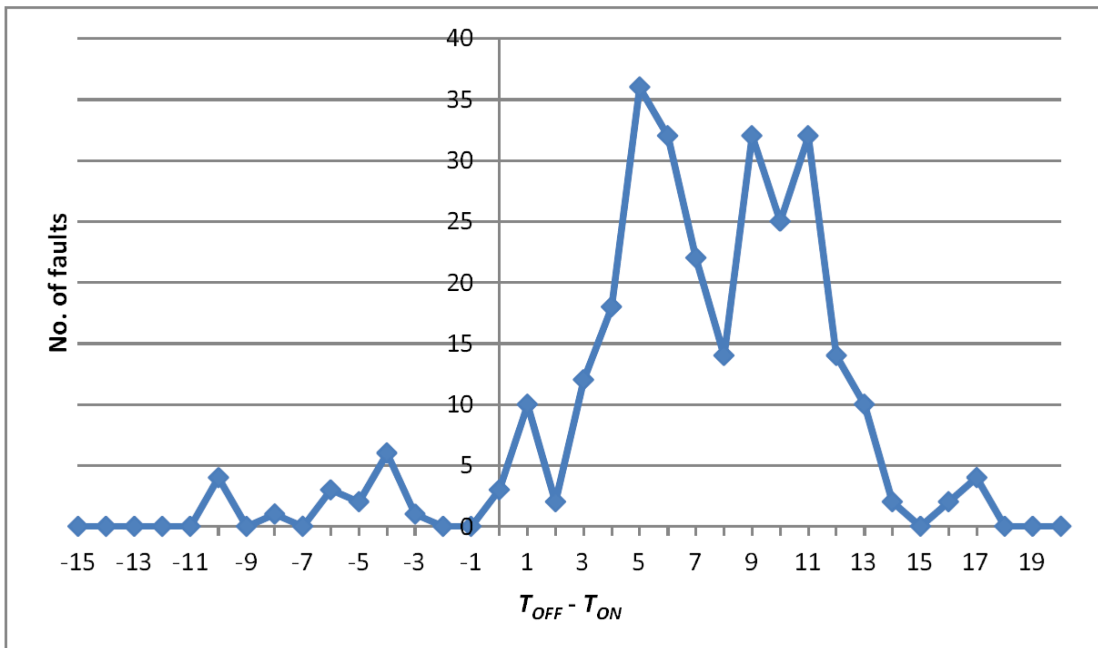


Figure 3.3 Relation of number of faults and TOFF−TON for s9234.

detect a resistive fault on p_2 with statistical analysis of T, unless the PUM has a sufficiently delay fault such that $T_{ON} \geq T_{OFF} + 3\sigma$. It is likely to bring fault escapes in

Table 3.2 Small-delay fault coverage of ISCAS89 Benchmark of LOS test.

Circuit	F_{all}	F_{det}	C_{ov}
s5378	8880	3886	43.76%
s9234	16442	7718	46.94%
s13207	23910	12297	51.43%
s35932	60634	36732	60.58%
s38584	68972	21340	30.94%

manufacturing testing, as a PUM typically has plenty of off-inputs. Else if the expected delay time of t_2 is longer than t_3, we need to set the expected path delay to T_{ON}+3σ. We can detect small-delay defects on the PUM, unless the off-input has a delay fault that causes $T_{OFF} \geq T_{ON}+3\sigma$. It is noted that in both of the two cases, we cannot detect small-delay defects on the PUM when $T_{OFF}>T_{ON}$. Even there has just one off-input of the PUM makes t_3 is longer than t_2.

To prove that the robust test is not appropriate for on-chip delay measurement, we show the relationship of T_{OFF} and T_{ON} using some experimental data. We use the test set (robust but not single-path sensitization) of s5378 and s9234; Figures 3.2 and 3.3 show results of s5378 and s9234, respectively. In these Figures the x axis shows the delay time difference of T_{OFF} and T_{ON} using number of inverters, and the y axis shows the numbers of faults. The results show that in most cases (over than 90%) $T_{OFF}>T_{ON}$. It causes incorrect measurement result, and defects can be masked. These results demonstrate that, in the actual testing, it is difficult to ensure that the transition of on-input earlier than the transitions of all off-inputs. In other words, it is difficult to detect small-delay defects on the PUM under robust but not single-path sensitization. To guarantee that faults on the PUM are detected, we must make the measured time include t_2 regardless of t_3. We can satisfy this by using the single-path sensitization condition, setting S1 to I_{OFF}.

3.2.3. Small-Delay Fault Coverage

Because of the constraint of single-path sensitization, small-delay fault coverage of the test method using on-chip delay measurement is very low. Table 3.2 shows the

small-delay fault coverage of ISCAS89 benchmark circuits of LOS test by using on-chip delay measurement. For example fault coverage of s38584 is less than 31%. From this, a method for improving small-delay fault coverage using on-chip delay measurement is strongly required.

3.3. Techniques for Improving Fault Coverage

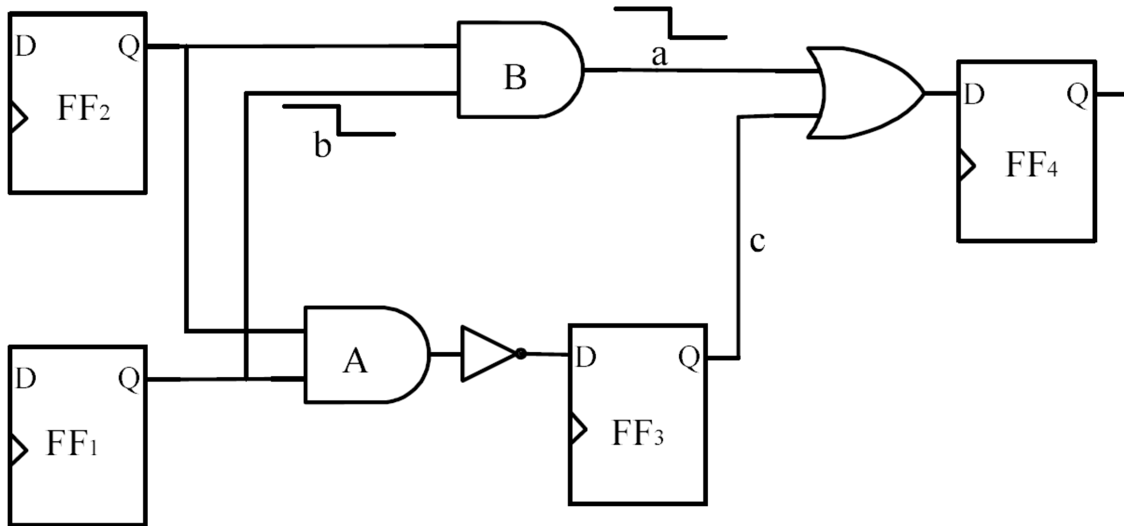


Figure 3.4 Example circuit.

This section introduces some techniques for improving fault coverage. These techniques are available for improving small-delay fault coverage of on-chip delay measurement. For higher fault coverage with an acceptable area overhead, we propose a procedure for using these techniques at the same time. The proposed method is a class of DFTs (designs for test), which facilitate testing. Although DFT increase area resulting in increase in the probability of defects, detecting small-delay faults with the DFT accomplishes shipment of dependable chips, which are free from manufacturing defects bringing about early-die.

3.3.1. Segmented Scan

Segmented scan is one of the techniques for improving the fault coverage [39]. Consider the part of a sequential circuit shown in Figure 3.4, assume that the FFs are connected in the order of their numerical indices. Assume that a slow-to-fall small-delay fault occurs on line *a*. Under the single-path sensitization condition, the path (which includes the line *a*) cannot be sensitized since the initialization condition $FF_1 = FF_2 = 1$ implies $FF_3 = 1$ by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line *c*) is set to controlling value, and the fault is blocked and it cannot propagated to FF_4 in the next cycle. Consider again, assume that

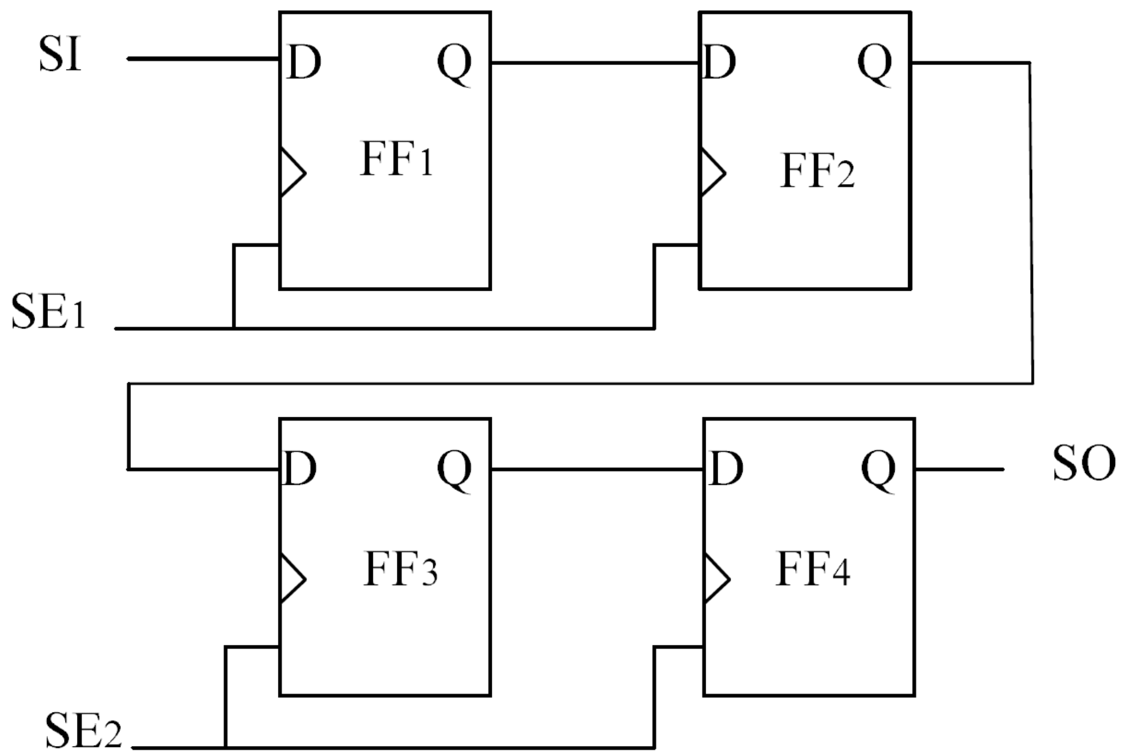
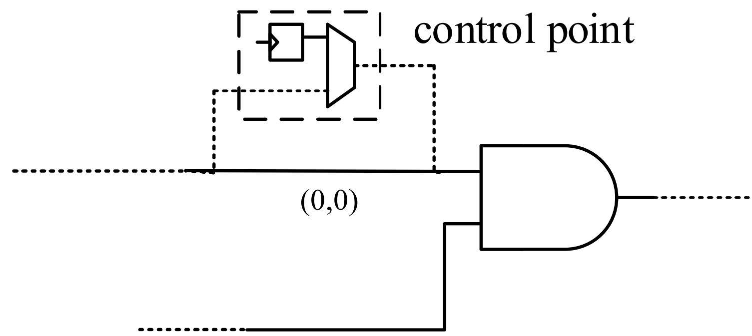


Figure 3.5 Segmented scan.

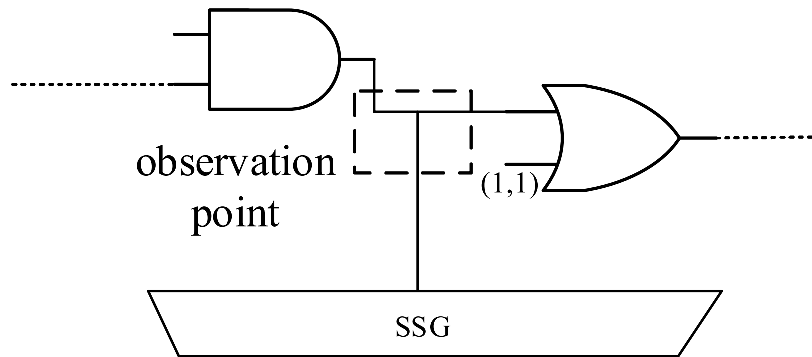
the scan chain is partitioned into two segments and each segment is controlled by two scan enable signals SE_1 and SE_2 as shown in Figure 3.5. The initial test pattern (FF_1, FF_2, FF_3, FF_4) = (1, 1, 0, X) (X is do not care value) is scanned in when both the scan enables SE_1 and SE_2 is set to 1. In the next cycle, we set SE_1 = 1, SE_2 = 0, then FF_3 contains the value of “0” instead of set to “1”. Therefore, the fault effect is propagated to FF_4 and single-path sensitization is achieved. The example demonstrates that the path cannot be sensitized under the single-path sensitization condition. However, it can be achieved by using two scan segments. Thus, segmented scan technique can be utilized to improve small-delay fault coverage of on-chip delay measurement method.

3.3.2. Test Point Insertion

Test point insertion (TPI) for improving fault coverage is popular on design. There are two types of TPI methods, namely observation point insertion and control point insertion [40]. As shown in Figure 3.6, observation point insertion involves making a



(a) control point



(b) observation point

Figure 3.6 Example of the Test Point Insertion.

node can be observed connecting it to the SSG. Control point insertion involves a selector with an enable signal where the enable signal is driven by a dedicated FF and can be set to a non-controlling value during the scan operation. Control point is enabled during the test operation and disabled during normal operation. As shown in Figure 3.6 (a), when the SE signal is "1", the node is driven by the dedicated FF and set to a non-controlling value. The FFs are inserted into scan chain; the non-controlling values are provided by scan operation. The dedicated FF is driven by the system clock signal just like other scan FFs. In this section, we investigate a strategy for maximizing the fault coverage from a small number of observation points and control points, respectively.

(1) Observation Point Insertion

Consider the sequential circuit shown in Figure 3.4, assume that a slow-to-fall small-delay fault occurs on line *b*. Under the single-path sensitization condition, the path (which includes this fault) cannot be sensitized since the initialization condition $FF_1 = FF_2 = 1$ implies $FF_3 = 1$ by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line *c*) is set to controlling value, the fault is blocked and it cannot propagated to FF4 in the next cycle. Here we insert an observation point after gate B (on line *a*) to observe the fault. We just need to ensure the values of FF_1 and FF_2 are (1, 1) and (0, 1) in the initialization pattern and launch pattern, respectively. By inserting the observation point, the transition on line *b* can reach stop of the DVMC, the sub-path (which includes the fault) is sensitized under the single-path sensitization condition. Thus, by comparing the measured delay time with the expected delay time, the transition fault on line *b* can be detected with criteria based on statistical analysis.

We insert observation points in order to detect small-delay faults that cannot be detected by LOS test under the single-path sensitization condition. For keeping the discussion general, we assume a set of all faults in one circuit denoted F, and the set of faults, which are detected by LOS under the single-path sensitization condition, denoted F_{SINGLE}. We denote the set of target faults as $U = F - F_{SINGLE}$.

We determine the placement of observation points for U as follows. For illustration, we show an example of part of a CUT in Figure 3.7. Assume that there are a fault on line G_1 and a fault on line G_2. Target to the fault G_1, because the off-input is controlling value, the fault is blocked at $Gate_4$ and $Gate_5$. Thus, the path including the fault G_1 cannot be sensitized under the single-path sensitization condition. The fault effects to lines G_2, G_3, G_4 and G_5. We insert observation point on any one of these lines allows the fault to be detected under the single-path sensitization condition. However, observation points on some of these lines may allow other faults to be detected. For example, the fault on line G_2 can be detected by inserting an observation point on line G_4. Thus, we should find the line which allows the most faults to be detected.

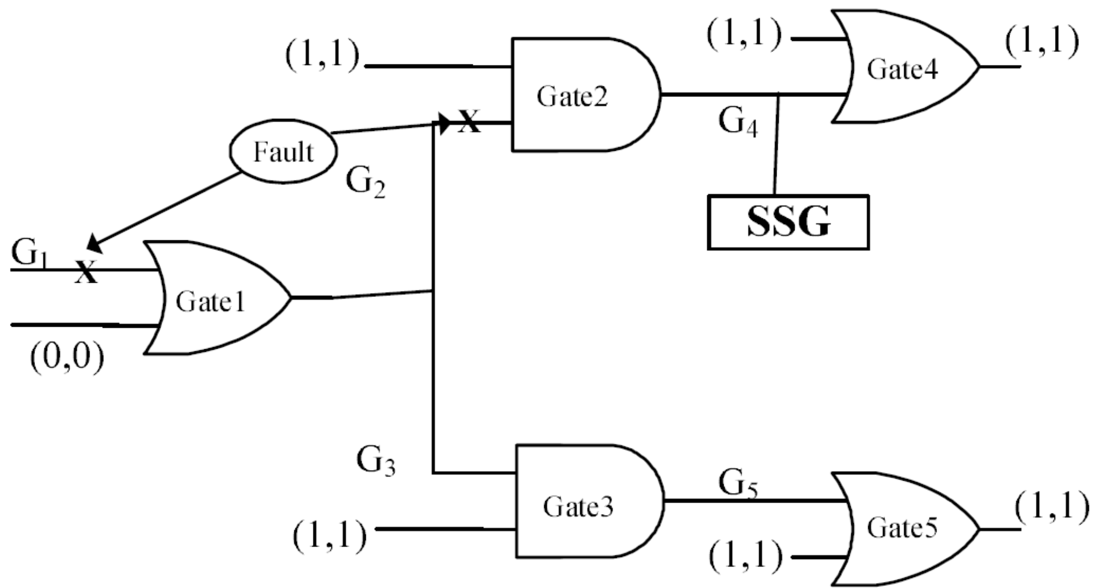


Figure 3.7 Example of observation point insertion.

If we insert a test point on a line *l* that on a critical path, it may increase the delay time of critical path. Then this insertion will degrade the performance of the chip. In order to prevent performance degradation due to this, we do not insert any test point into signal lines on a critical path. Before the test point insertion procedure, we identify all signal lines that lie on a critical path. We delete these lines from the test point set.

We select a minimal subset of observation points applied for U by using a greedy covering procedure. We show the detail procedure for inserting observation point in Procedure 1. To achieve higher fault coverage with an acceptable hardware overhead, the number of inserted observation point (N_O) is decided by results of hundreds of pre-simulation tests.

Procedure 1: Observation Point Insertion

1. Let U be the set of target faults. Let OB be an empty set, it is the set of lines used to insert observation points.

2. For every line g_i, let OBS(g_i) be an empty fault set. Find the set of faults OBS(g_i) such that can be detected with an observation point inserted on the line g_i.

3. Select a line g_j such that OBS(g_j) has the largest number of faults $tf_j \in$ U.

4. Add the line g_j to OB. Remove faults $tf_j \in$ OBS(g_j) from U.

5. If U = ∅, or the number of observation points = pre-set value N_O, stop; else go to Step 3.

(2) Control Point Insertion

Consider the circuit shown in Figure 3.4, assume that a slow-to-fall small-delay fault occurs on line *a*. Under the single-path sensitization condition, the path (which includes this fault) cannot be sensitized since the initialization condition

FF_1 = FF_2 = 1 implies FF_3 = 1 by 1 bit shift during the launch cycle. Thus, the off-input of the OR gate (line *c*) is set to controlling value, and the fault is blocked from being propagated to FF_4 during the capture cycle. Here we insert a control point after gate A (on line *c*). Under the single-path sensitization condition, to detect the slow-to-fall delay fault of line *a*, we just need to set the value of the off-input of OR gate (line *c*) as non-controlling value (S0) in both the initialization pattern and launch pattern.

We insert control points to detect faults that cannot be detected by LOS test under the single-path sensitization condition. We select a minimal subset of control points by using a greedy covering procedure. As the same with observation point insertion, to achieve higher fault coverage with minimal control point, we should consider the number of faults that can be detected by one control point insertion. In other words, control point which detects the largest number of faults will be first inserted. In order to prevent any possible performance degradation, we do not insert any test point into signal lines that are on a critical path. The procedure for control point insertion is given next as Procedure 2. To achieve higher fault coverage with an acceptable hardware overhead, the number of inserted observation point (N_C) is decided by results of hundreds of pre-simulation tests.

Procedure 2: Control Point Insertion:

1. Let U be the set of target faults. Let CO be an empty set, it is the set of lines used to insert control points.

2. For every line g_i, let COS(g_i) be an empty fault set. Find the set of faults COS(g_i) such that can be detected with a control point inserted on the line g_i.

3. Select a line g_j such that COS(g_j) has the largest number of faults $tf_j \in U$.

4. Add the line g_j to CO. Remove faults $tf_j \in$ COS(g_j) from U.

5. If $U = \varnothing$, or the number of control points = pre-set value N_C, stop; else go to Step 3.

(3) Procedure for Segmented Scan and Test Point Insertion

Based on Procedure 1 and Procedure 2, test points are inserted according to the number of faults that can be detected. In other words, test point which detects the largest number of faults will be first inserted. Thus, after a number of test points inserted, the effect for the coverage improvement will be not very notable. Hence, by using only one of the introduced techniques, we may not be able to get an ideal fault coverage under the single-path sensitization condition. For higher fault coverage, we use the segmented scan and test point insertion at the same time. The procedure for using all the introduced techniques is given next as Procedure 3. The values of N_S, N_C and N_O are decided by results of hundreds of pre-simulation tests. We can use these values to get a higher coverage with an acceptable hardware overhead.

Procedure 3: Segmented scan and test point insertion

1. Let L_0 be the length of CUT's scan chain, U be the set of undetectable faults under the single-path sensitization condition, and N_S be the number of scan segments. We divide the scan chain to N_S segments, the length L of these segments is calculated by $L = L_0 / N_S$. Each segment is controlled by a corresponding scan enable signal. After this step, let $U = U - F_S$, where F_S is the set of new detectable faults.

2. Let N_C be the number of control point will be inserted. Insert these control

points using procedure 2.

3. Let N_O be the number of observation points will be inserted. Insert these observation points using procedure 1.

This procedure inserts control points before observation point insertions. This is because that the effect of the control point is better than the observation point in term of area overhead; the results of the pre-simulation tests which confirm this fact will be shown in the next section (Figure 3.11).

3.4. Evaluation

In this section, we study the effects of these introduced techniques on the set of faults undetectable by LOS test under the single-path sensitization condition. The corresponding hardware overhead also will be evaluated. In this evaluation, we use ISCAS89 benchmark circuits. The results of cell area are obtained by synthesis with Synopsys design compiler using Rohm 180μm process [41]. We also reported the maximal core utilization after layout with Synopsys IC Compiler. The maximal core utilization means the maximal value of core utilization that the IC Compiler can make layout (placement and routing) without errors. The chip area depends on both the maximal core utilization and cell area. The smaller value of maximal core utilization means larger chip area resulting from more complex routing. Figures 3.8~3.10 show the relation between maximal core utilization (in s5378 and s9234) and the numbers of scan segments, inserted observation points and inserted control points, respectively. These Figures are explained later. The test patterns are generated with in-house ATPG based on what is used in [19]. First, we evaluate the effect of segmented scan. Next, we evaluate the effect of test point insertion (observation point insertion and control point insertion). We also compared the effects of control point insertion and observation point insertion for the pre-simulation tests to explain why the proposed procedure inserts control points before observation point insertions (as noted in the last paragraph of the previous section). For higher fault coverage, we evaluate the effect of segmented scan and test point insertion. Tables 3.3~3.6 show the evaluation results. In these Tables, the column *Circuit* shows the circuit name. The columns *[19]* and *Proposed* show the evaluation results of the method of [19] and the methods using introduced fault coverage improving techniques. The column N_T gives the number of the test pattern pairs (As only one path is selected to be measured for each test pattern pair, N_T also gives the number of measurements). Columns C_0(%) and C_1(%) report the fault coverage of [19] and methods using introduced fault coverage improving techniques, respectively. Columns $S_0(mm^2)$ and $S_1(mm^2)$ report the cell area of [19] and methods using introduced fault coverage improving techniques, respectively. The column C_{uti} reports the maximal core utilization after layout. Chip area can be calculated by $S_C = S_0/C_{uti}$ (or $S_C = S_1/C_{uti}$). Column N_S, N_C and N_O show

the numbers of scan segments, inserted control points and inserted observation points, respectively. The column V shows the test data volume in 10^5 bit. The column T shows the test application time in 10^5 clocks. The column C_{IMP}(%) reports the effect of fault coverage improvement by using fault coverage improving techniques. The column AO(%) reports the area overhead, which is calculated by $AO=(S_1-S_0)/S_0\times 100(\%)$.

In addition, to achieve a still higher fault coverage with the same overall hardware overhead, we implemented the proposed procedure several times with the same overall hardware overhead (by changing the area ratio of control point insertion and observation point insertion in the same overall hardware overhead). Figures 3.12 and 3.13 show the results of s5378 and s9234.

3.4.1. Effect of Segmented Scan

We improved the fault coverage 12.67~22.31% by only using 8 scan segments. However, with the increasing of scan segment's numbers, the effect for the coverage improvement is not very notable for some circuits. For example, as shown in Table 3.3 in the circuit s35932, we used 8 scan segments, the coverage improvement was 27.55%. However, when we used 64 scan segments, the fault coverage is just improved 0.64% compared to 8 scan segments. For higher fault coverage, we must consider to use other techniques. Figure 3.8 shows that the maximal core utilization became smaller with increasing the number of scan segments. It means that increasing the number of scan segments increases the number of redundant lines and makes routing more difficult. As a result, it causes the larger total area.

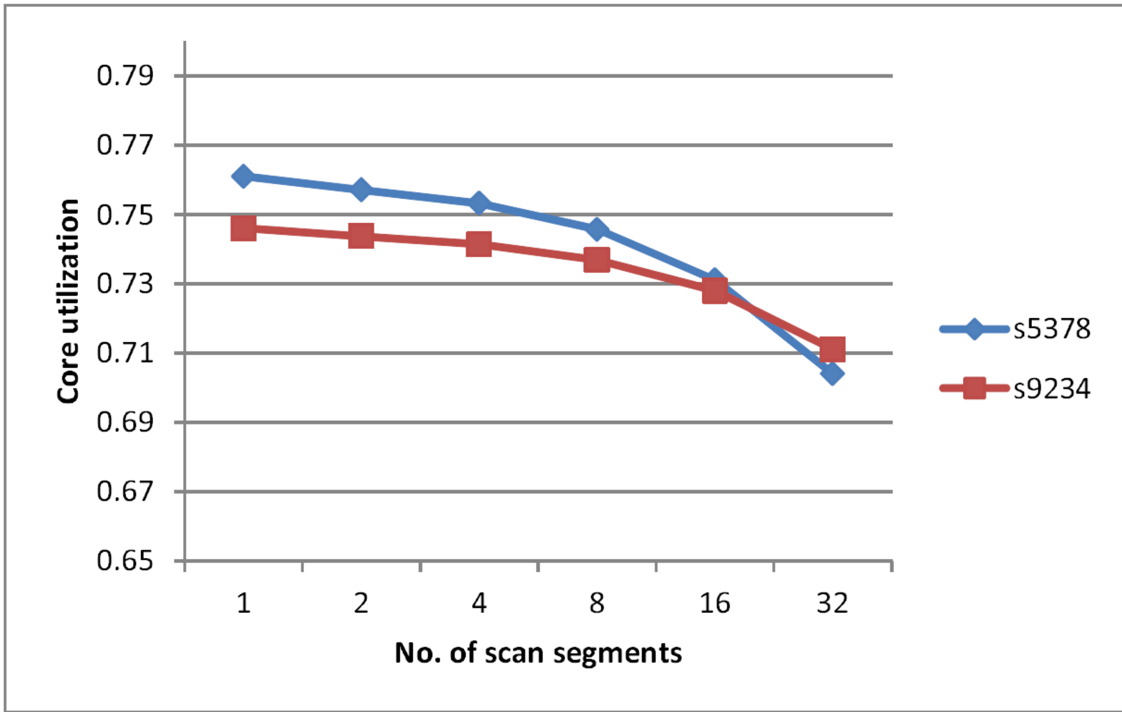


Figure 3.8 Maximal core utilization vs. scan segments.

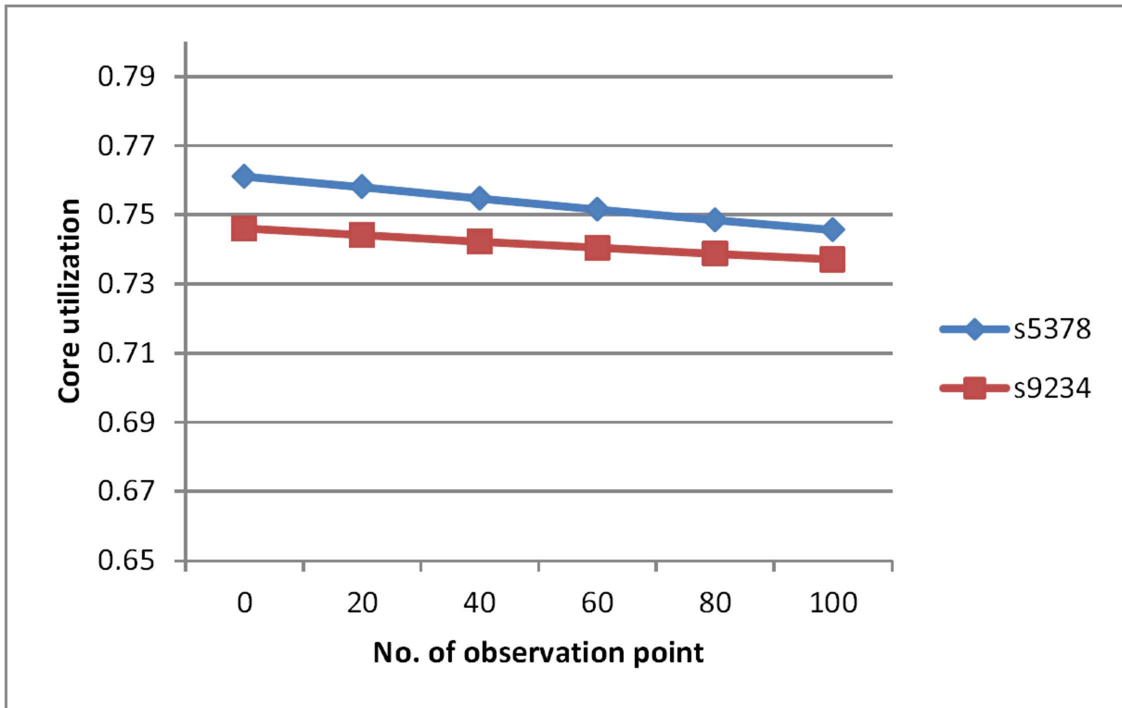


Figure 3.9 Maximal core utilization vs. inserted observation points.

Table 3.3 Effect of segmented scan.

Circuit		[19]			*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C_{uti}	N_S	N_T	C_1(%)	S_1(mm^2)	C_{uti}	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	8	345	66.07	0.118	0.75	22.31	0.56
s5378	160	43.76	0.118	0.76	16	638	77.64	0.119	0.73	33.88	1.11
s9234	199	46.94	0.191	0.75	8	576	64.86	0.191	0.74	17.92	0.35
s9234	199	46.94	0.191	0.75	16	726	68.35	0.192	0.73	21.41	0.68
s13207	330	51.43	0.357	0.73	8	507	66.14	0.358	0.73	14.71	0.18
s13207	330	51.43	0.357	0.73	32	1048	72.45	0.360	0.71	21.02	0.73
s38584	3778	60.58	0.889	0.74	8	5734	73.25	0.890	0.73	12.67	0.07
s38584	3778	60.58	0.889	0.74	64	7208	79.48	0.895	0.72	18.90	0.58
s35932	3213	30.94	0.963	0.75	8	6425	58.49	0.963	0.75	27.55	0.07
s35932	3213	30.94	0.963	0.75	64	6498	59.13	0.968	0.74	28.19	0.54

3.4.2. Effect of Test Point Insertion

We improved the fault coverage with 4.83~22.06% by inserting 100~400 observation points. The fault coverage can be improved with 8.21~38.29% by inserting 40~100 control points. However, after a number of test points inserted, the effect for the coverage improvement became not very notable. For the aim of higher fault coverage, we need to insert more test points or use other techniques. From the results in Figures 3.9 and 3.10, the maximal core utilization became smaller with increasing the number of inserted test points. It means that increasing the number of inserted test points leads to more difficult routing, and it causes the total area becomes larger. As shown in Table 3.4, the decrease of maximal core utilization is 0.03 (from 0.76 to 0.73) when we insert 200 observation points in s5378. However, the decrease of maximal core utilization is only 0.01 (from 0.74 to 0.73) even we insert 400 observation points in s34584. We can get similar results in control point insertion in Table 3.5. We found that the decrease is smaller in large circuits when we insert the same (or more in large circuit) number of test points. This means that the increases of the redundant lines and difficulty in routing caused by our DFT design are small in large circuits. In other words, the effect on layout of our proposed method is less serious in large circuits.

To decide whether observation point insertion follows control point insertion, we also compared the effects of control point insertion and observation point insertion with the same hardware overhead. Figure 3.11 presents the results of s13207. In Figure 3.11, the y axis and the x axis show the fault coverage (%) and the hardware overhead (%). Here, we set the hardware overhead of test point insertion from 1% to 10%. From the experiment result, we found that the effect of the control point insertion is better than the observation point. For example, the fault coverage is improved more than 36% by using control point insertion with only 3% hardware overhead, while the improvement of the fault coverage is less than 6% by using observation point insertion with the same hardware overhead. Therefore, control points are inserted before observation points in Procedure 3.

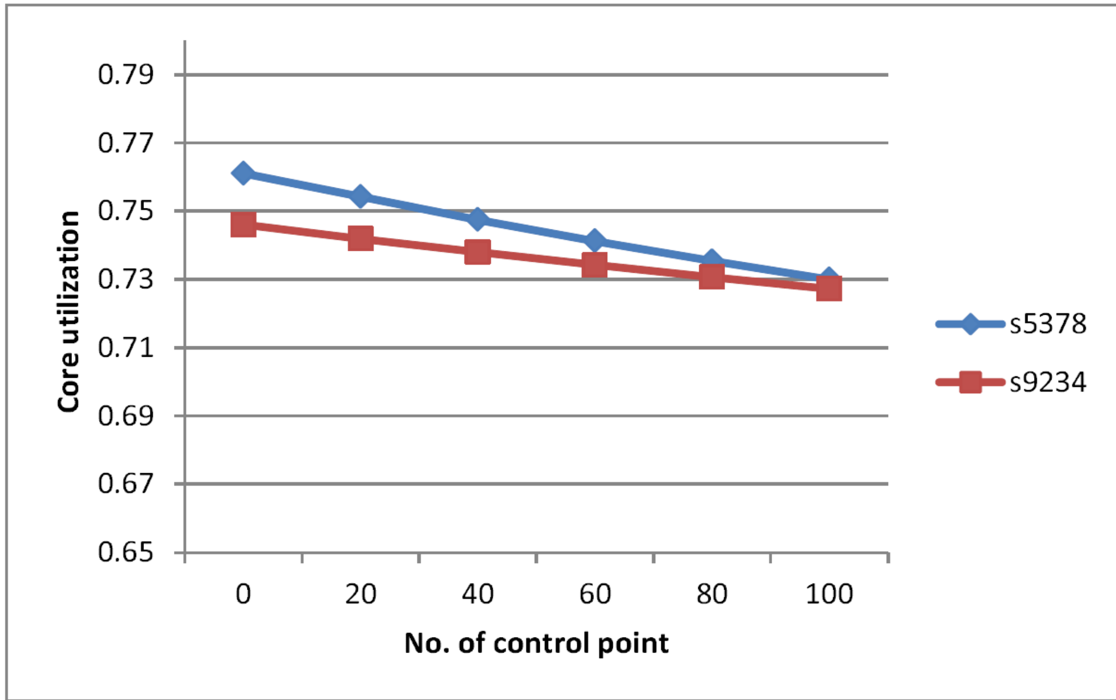


Figure 3.10 Maximal core utilization vs. inserted control points.

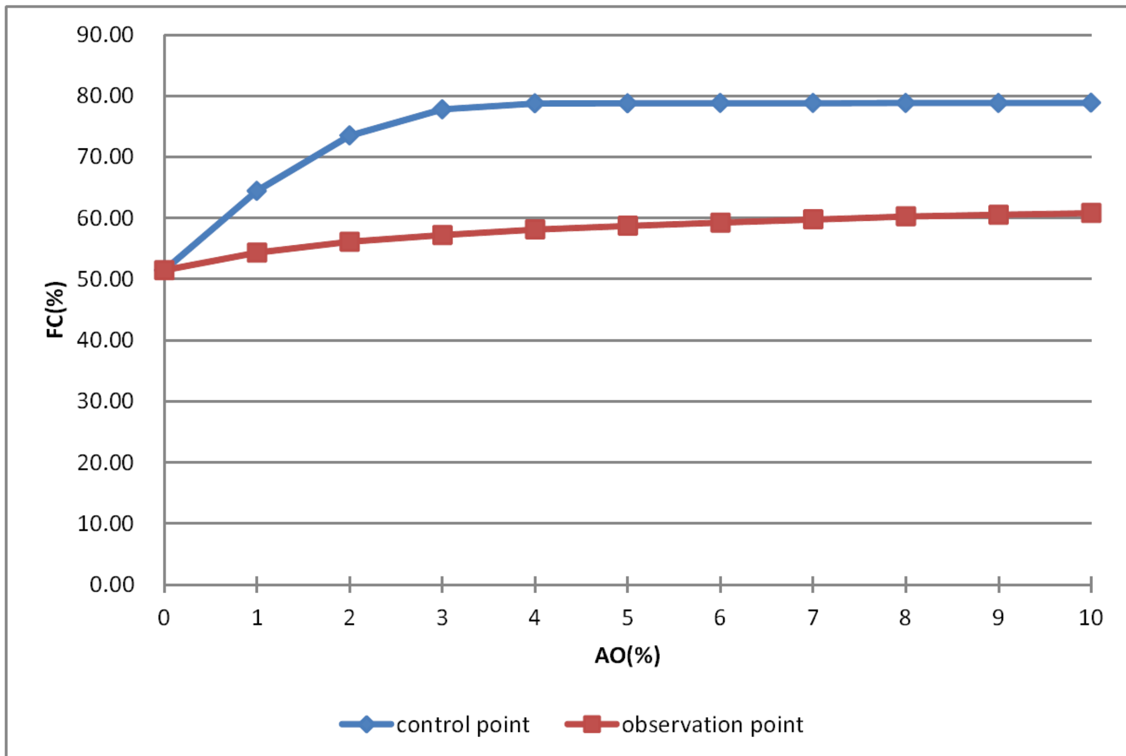


Figure 3.11 Comparison of the effects of control point insertion and observation point insertion.

Table 3.4 Effect of observation point insertion.

Circuit		[19]			*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C*uti*	N*O*	N_T	C_1(%)	S_1(mm^2)	C*uti*	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	100	315	57.11	0.131	0.75	13.35	11.33
s5378	160	43.76	0.118	0.76	200	336	62.43	0.144	0.73	18.67	22.65
s9234	199	46.94	0.191	0.75	100	538	64.24	0.204	0.74	17.30	6.99
s9234	199	46.94	0.191	0.75	200	722	69.00	0.217	0.73	22.06	13.98
s13207	330	51.43	0.357	0.73	100	476	56.26	0.370	0.73	4.83	3.73
s13207	330	51.43	0.357	0.73	300	502	61.34	0.397	0.72	9.91	11.20
s38584	3778	60.58	0.889	0.74	400	4213	66.39	0.943	0.73	5.81	6.00
s35932	3213	30.94	0.963	0.75	400	5947	51.74	1.016	0.75	20.80	5.54

Table 3.5 Effect of control point insertion.

Circuit		[19]			*Proposed*						
	N_T	C_0(%)	S_0(mm^2)	C*uti*	N_C	N_T	C_1(%)	S_1(mm^2)	C*uti*	C_{IMP}(%)	AO(%)
s5378	160	43.76	0.118	0.76	40	622	58.90	0.125	0.75	15.14	6.34
s5378	160	43.76	0.118	0.76	100	717	72.61	0.136	0.73	28.85	15.86
s9234	199	46.94	0.191	0.75	60	527	57.04	0.202	0.73	10.10	5.87
s9234	199	46.94	0.191	0.75	100	564	62.09	0.209	0.73	15.15	9.78
s13207	330	51.43	0.357	0.73	60	1944	77.76	0.368	0.73	26.33	3.14
s38584	3778	60.58	0.889	0.74	100	4568	68.79	0.897	0.73	8.21	0.84
s35932	3213	30.94	0.963	0.75	40	10414	69.23	0.981	0.75	38.29	1.94

Table 3.6 Effect of segmented scan and test point insertion.

Circuit	[19]						*Proposed*								
	N_T	C_0	S_0	C_{uti}	V	T	$(N_S/N_C/N_O)$	N_T	C_1	S_1	C_{uti}	V	T	C*IMP*	AO(%)
s5378	160	43.76	0.118	0.76	0.69	0.31	32/40/20	856	91.5	0.13	0.69	3.8	2.5	47.74	10.81
s9234	199	46.94	0.191	0.75	0.98	0.49	64/70/40	1162	84.56	0.214	0.67	5.93	3.11	37.62	12.35
s13207	330	51.43	0.357	0.73	4.62	2.26	64/100/50	1956	87.66	0.388	0.69	27.78	13.33	36.23	8.54
s38584	3778	60.58	0.889	0.74	110.62	55.44	128/200/100	11425	87.6	0.95	0.7	337.04	165.23	27.02	6.85
s35932	3213	30.94	0.963	0.75	113.29	56.02	128/200/100	10684	72.77	1.024	0.71	379.07	186.02	41.83	6.33

3.4.3. Effect of Segmented Scan and Test Point Insertion

To achieve higher fault coverage with an acceptable hardware overhead, we use segmented scan, observation point insertion and control point insertion at the same time by using procedure 3. The evaluation results are shown in Table 3.6. As shown in the results, we got an acceptable fault coverage by using these techniques at the same time. Fault coverage can be improved 27.02~47.74% with 6.33~12.35% of hardware overhead.

3.4.4. Effective fault coverage by the same overall hardware overhead

To achieve a still higher fault coverage with the same overall hardware overhead, we implemented the proposed procedure several times with the same overall hardware overhead (by changing the area ratio of control point insertion and observation point insertion in the same overall hardware overhead). Figures 3.12 and 3.13 show the results of s5378 and s9234.

The y axis in Figures 3.12 and 3.13 shows the fault coverage, and the x axis shows the area ratio of the overhead for the observation point insertion to the whole overhead.

For example, in the case of the overall hardware overhead is 10% and the area ratio for the observation point insertion is 40%, we insert control points and observation points with area ratio of 6:4. Here, we set the overall hardware overhead as 5%, 10%, 15% and 20%. Figure 3.12 shows the result of s5378 using 32 scan segments, and Figure 3.13 shows the result of s9234 using 64 scan segments.

From the experiment result, we can achieve a still higher fault coverage with the same hardware overhead. For example, for s5378 using 32 scan segments, we can achieve the most effective fault coverage of 91.57% when the observation point insertion occupies 50% of the overall hardware overhead (when the overall hardware overhead is 10%). For s9234 using 64 scan segments, we can achieve the most effective fault coverage of 91.22% when the observation point insertion occupies 30% of the overall hardware overhead (when the overall hardware overhead is 20%).

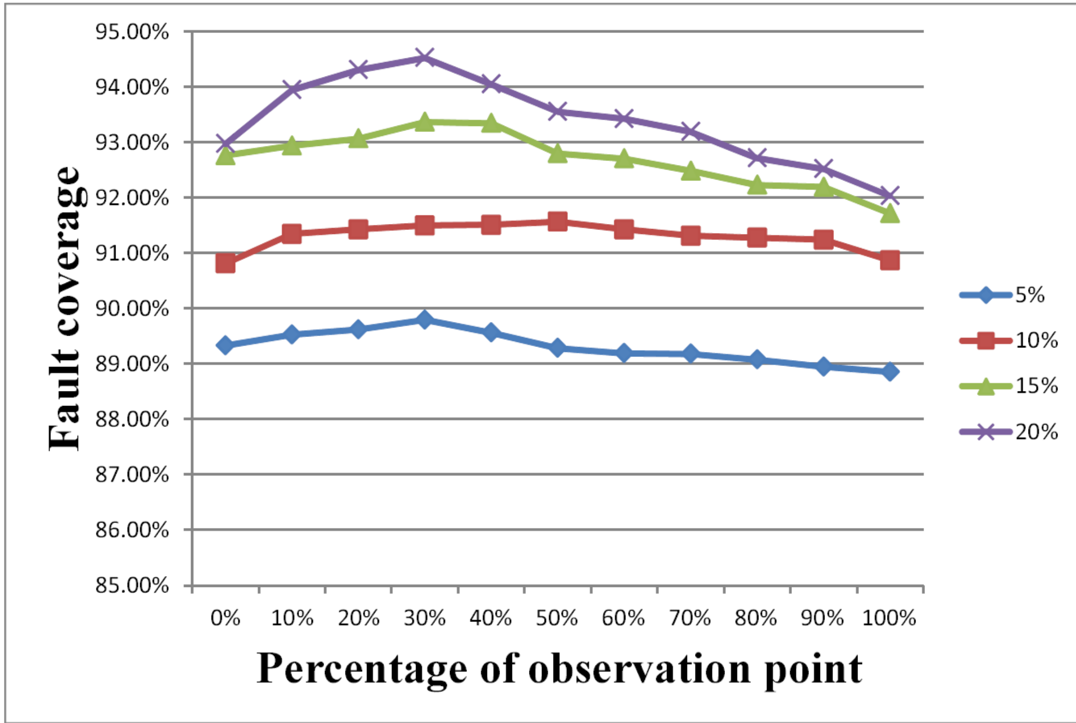


Figure 3.12 Result of s5378 using 32 scan.

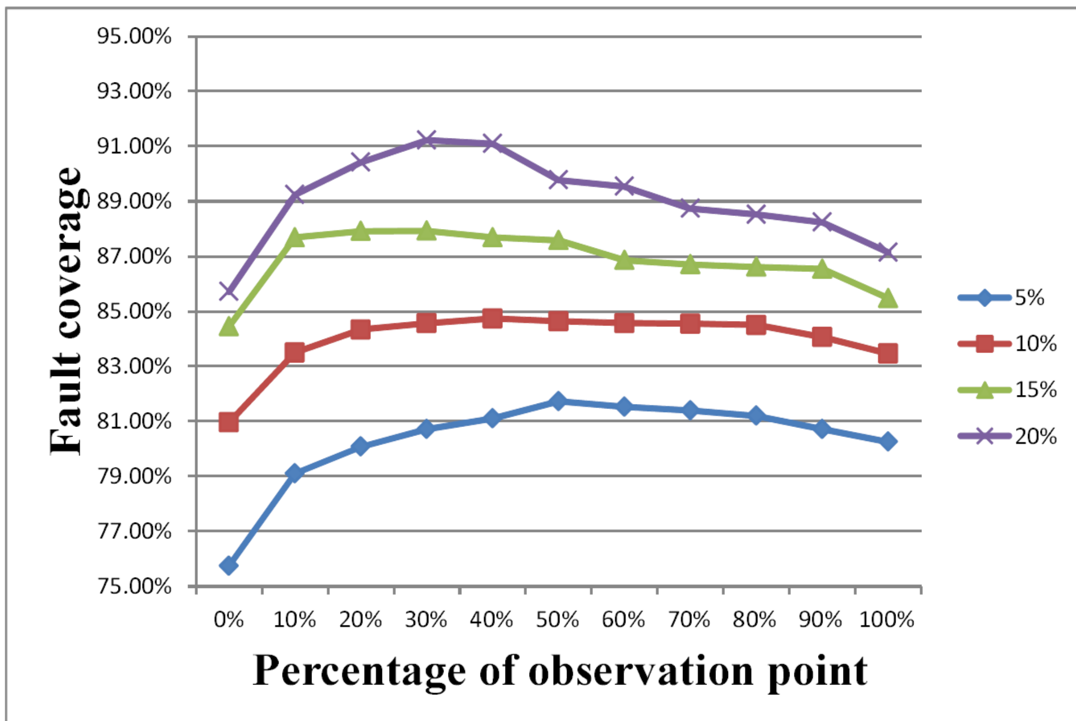


Figure 3.13 Result of s9234 using 64 scan.

From the experiment result of Figures 3.12 and 3.13, to achieve a still higher fault coverage with the same hardware overhead, we should set the area of observation point insertion occupies 30~50% of the overall hardware overhead.

3.5. Conclusion

Our pre-simulation results show that single-path sensitization is required when using on-chip delay measurement method. This constraint makes the fault coverage very low. To improve small-delay fault coverage under the single-path sensitization condition, this chapter introduced techniques using segmented scan and test point insertion. For higher fault coverage, we propose a procedure for using these techniques at the same time. As the evaluation results, the proposed procedure improved the fault coverage 27.02~47.74% with 6.33~12.35% of hardware overhead. To achieve a still higher fault coverage with the same hardware overhead, we should set the area of observation point to occupy 30~50% of the overall hardware overhead.

In this chapter, we focus only on the LOS test. However, the proposed procedure can be extended to improve fault coverage for other test designs. As our future work, for higher fault coverage and smaller hardware overhead, we will apply the proposed procedure on LOC test and LOS+LOC. Our future work also includes test data and test application time reduction.

4. Scan Shift Time Reduction Using Test Compaction for On-Chip Delay Measurement

4.1. SUMMARY

In recent VLSIs, small-delay defects, which are hard to detect by traditional delay fault testing, can bring about serious issues such as short lifetime. Small-delay defects have become a significant problem and it is necessary to detect such defects during manufacturing tests [42]. Firstly, the increase of small delay on paths might trigger a time-related failure [43]. The second reason is that small-delay defects can become a reliability fault because the defect can be worsened during subsequent aging in the field [44]. To detect small-delay defects, on-chip delay measurement which measures the delay time of paths in the circuit under test (CUT) was proposed. However, on-chip delay measurement incurs high test cost because it uses scan design, which brings about long test application time due to scan shift operation. Thus, a method reducing test application time is strongly required. In on-chip path delay measurement, the capture operation is unnecessary unlike the conventional delay testing. Thus, FFs keep the transition pattern (denoted as $v_{m,1}$) of the test pattern pair sensitizing a PUM *p* even after the measurement of *p*. If $v_{m,1}$ can be used as the initial pattern (denoted as $v_{n,0}$) of another test pattern pair (v_n which sensitizes another path *p'*), we can sensitize *p'* by just shifting 1 bit of the transition pattern (under LOS test). The proposed method uses this characteristic. This thesis presents a method reduces scan shift time and test data volume by using scan-based test pattern merging. We can also reduce the switching activity induced by the launch pulse. As a result, this also reduces excessive IR-drop in scan testing avoiding test-induced yield loss.

The proposed method reduces scan shift time and test data volume using test pattern merging. Evaluation results on ISCAS89 benchmark circuits indicate that the proposed method reduces the test application time by 6.89~62.67% and test data volume by 46.39~74.86%.

The rest of the chapter is organized as follows. Section 4.2 describes some terminologies related to scan-based delay testing and the on-chip delay measurement. Section 4.3 explains methods to reduce test application time and test data volume. Section 4.4 evaluates the introduced method. Finally, section 4.5 concludes this chapter.

4.2. Preliminaries

This section introduces some terminologies related to scan-based delay testing and the on-chip delay measurement.

4.2.1. Terminology Related to Scan-Based Delay Testing

(1)Test pattern pair

In scan-based delay testing, a pair of test patterns $v_m = (v_{m,0}, v_{m,1})$ are applied to the PUT in two consecutive clock cycles. The pattern $v_{m,0}$ applied in first is an initial pattern, the pattern $v_{m,1}$ is a transition pattern. A pair of initial and transition patterns is called a test pattern pair. If the number of pattern pairs in one test pattern set V is n, we call n is the length of V.

(2) Transition fault and path delay fault

In one circuit, the number of transition fault is equal to twice the number of gate lines, and the number of path delay fault is exponential relationship with the number of gates. In modern circuit, transition fault model is more widely used than path delay fault. We also note that, with the increasing of the number of paths, the test generation will be very difficult [12]. This chapter uses the transition fault coverage as the small-delay fault coverage because the aim of this chapter is to detect increases of gate and line delays caused by resistive faults to reduce early life-failure. If there has a transition on one gate line, and it causes the delay time of one path that includes this fault exceed the system clock. We think we can detect this transition fault.

In this chapter, we try to detect increases of gate and line delays caused by resistive faults to reduce early-life failure, the transition fault model is adopted. The small-delay fault coverage is equal to the transition fault coverage. Note that in this chapter we focus only on the LOS test.

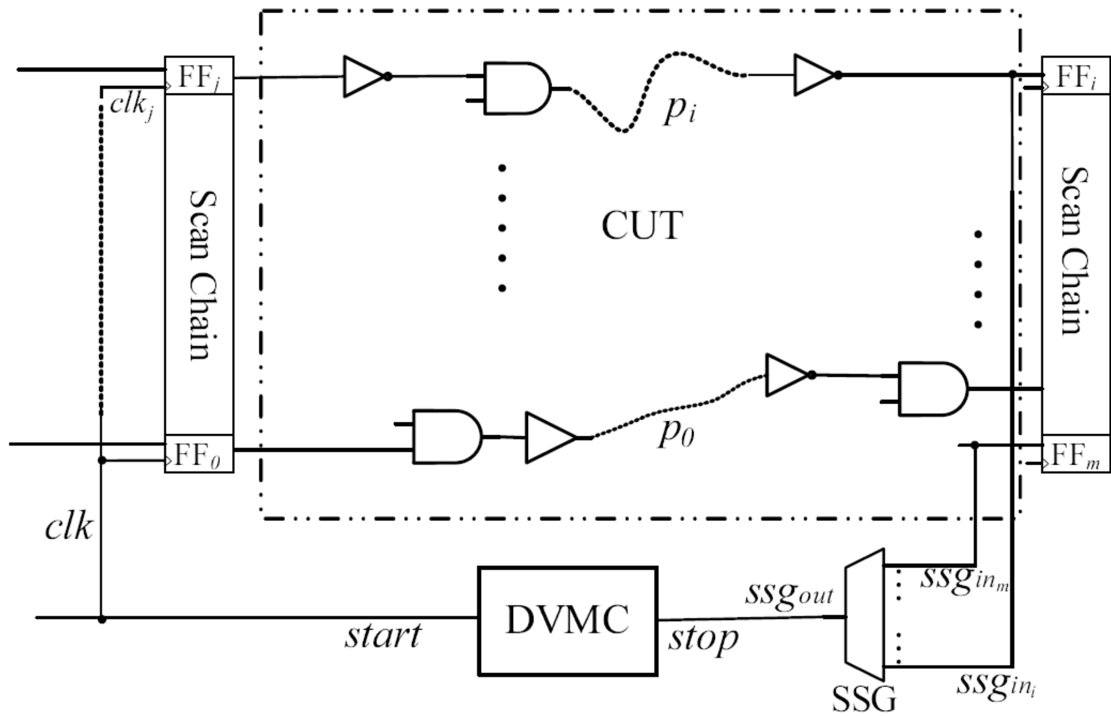


Figure 4.1 Architecture of on-chip path delay measurement system.

4.2.2. Operation of on-Chip Delay Measurement Method

Figure 4.1 shows the architecture of the on-chip path delay measurement. The on-chip delay measurement system measures the delay of each path including a PUM, whose input and output are start and stop, respectively. The delay measurement system consists of *delay value measurement circuit* (DVMC), *stop signal generator* (SSG, which is an N-to-1 multiplexer), and *circuit under test* (CUT).

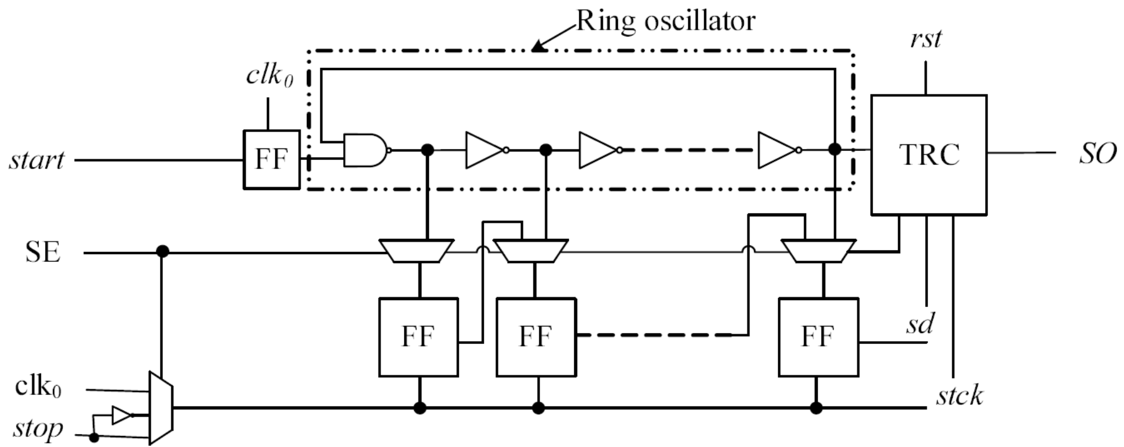


Fig4.2 Architecture of DVMC.

The clock line is directly connected to *start* of the DVMC; the DVMC starts the measurement when a positive transition is sent to *start*. The SSG detects the transition on the input of a designated flip-flop (FF) and sends the transition to *stop* of the DVMC, by setting the corresponding control data of SSG. The input line *clk* is the clock signal of the CUT. The line clk_i is the clock line of FF_i. The input of FF_i is connected to ssg_{out} through ssg_{ini} and the SSG. The system measures a PUM including one clock line clk_j, a path p_i, and some redundant lines ssg_{ini} and ssg_{out}. For example, after the measurement of the path $p^0 = clk_j\text{-}p_i\text{-}ssg_{ini}\text{-}ssg_{out}$, by comparing the measured delay time with the expected delay time, small-delay defects on clk_j and p_i can be detected. In this thesis, we insert one DVMC circuit in one CUT. Thus, only one path is selected to be measured for each test.

The architecture of the embedded delay measurement circuit DVMC is shown in Figure 4.2. The DVMC is a ring oscillator based TDC, it measures the time difference of two transitions that sent to *start* line and *stop* line. The transition of *start* triggers the measurement. The TRC (an *n*-bit up counter) counts the round cycles of the oscillation. Synchronizing the transition of *stop*, the FFs capture the states on the output of corresponding NOT gates and the TRC. From these values, the delay value is calculated.

In one circuit, the set of paths under measurement is denoted by *P* (includes paths p_0, p_1, ..., $p_{(m-1)}$). Let *D* (which includes d_0, d_1, ...,$d_{(m-1)}$) be the control data of SSG. The data d_i selects the path p_i as the PUM. Let *V* (which includes test pattern pairs

$v_0, v_1, ..., v_{(m-1)}$ for sensitizing paths p_0, p_1, ..., $p_{(m-1)}$) be the test data. The test flow of the on-chip delay measurement is as follows:

1. Select a path p_i from P for delay measurement by setting the corresponding control data of SSG to d_i.
2. Assign the test pattern which sensitizes p_i to the primary inputs and flip-flops of the CUT.
3. The transition reaches *stop* of the DVMC through p_i and the SSG, thus, the DVMC stops the measurement.
4. We get the measurement result through scan out of the DVMC. Consequently, the path delay of the PUM is calculated from the read out values.
5. Delete p_i from P. If $P = \varnothing$, stop the test; else go to Step1.

4.3. The Proposed Method to Reduce Test Application Time and Test Data Volume for On-Chip Delay Measurement

This section proposes a method which reduces test application time and test data volume of the on-chip delay measurement by using scan-based test pattern merging. The LOS operation of the on-chip delay measurement is introduced in section 4.3.1 Scan-based test pattern merging technique is explained in section 4.3.2. In section 4.3.3, we introduce our procedure for test application time and test data volume reduction. In 4.3.4 we analyze the test application time and test data volume.

4.3.1. LOS Operation of the On-Chip Delay Measurement

Figure 4.3 shows the LOS operation of the on-chip delay measurement. When using the on-chip delay measurement to detect small-delay defects on path *p* (from FF_i to FF_j), we set the SSG to detect the transition on the input of FF_j. At the moment the transition reaches the input *D* of FF_j, the transition is sent to *stop* of the DVMC through the SSG. Then, the DVMC stops the measurement. In this process, the capture operation of LOS test is unnecessary unlike the conventional LOS operation. Thus, FFs keep the transition pattern $v_{m,1}$ of the test pattern pair sensitizing *p* even after the measurement of *p*. If $v_{m,1}$ can be used as the initial pattern $v_{n,0}$ of another test pattern pair (v_n which sensitizes another path *p'*), we can sensitize *p'* by just shifting 1 bit of the transition pattern. Therefore, we can reduce the test application time. We can also reduce the switching operation caused by the capture operation. As a result, the proposed method also reduces excessive IR-drop in scan testing avoiding test-induced yield loss.

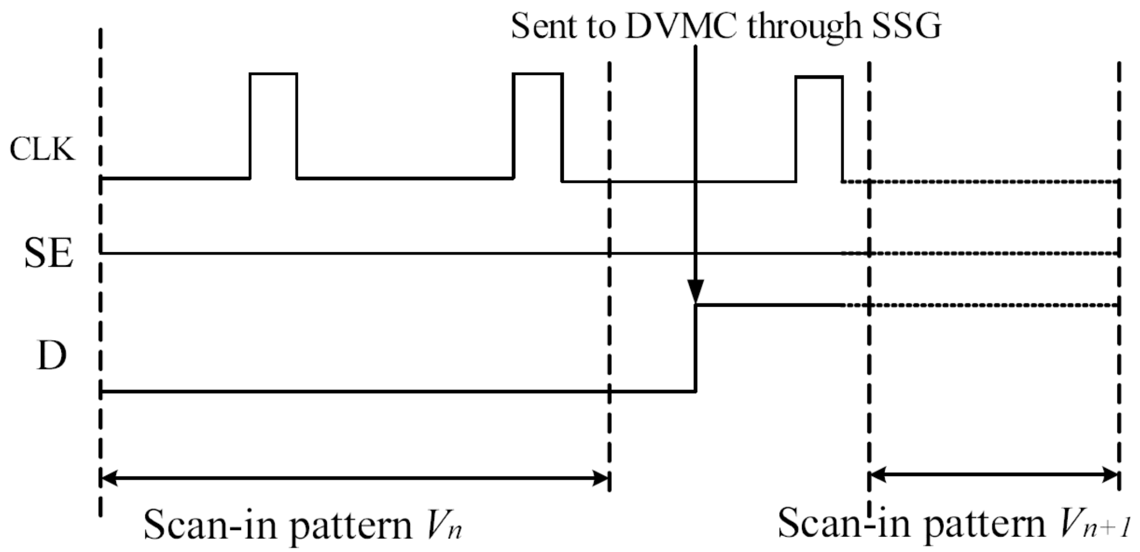
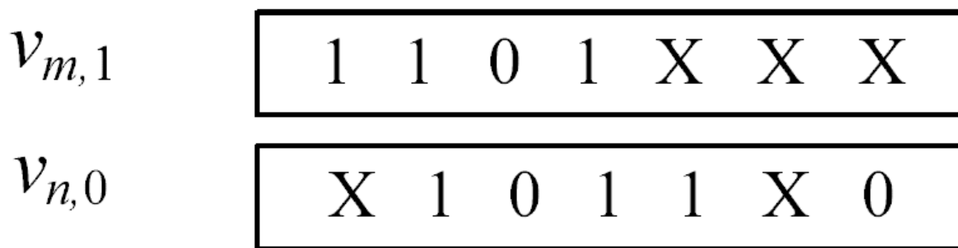
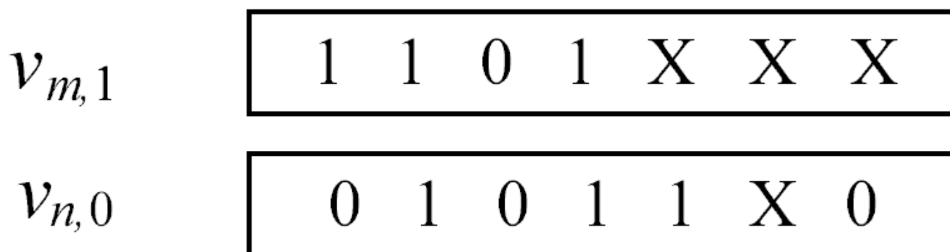


Figure 4.3 LOS operation of on-chip delay measurements.



(a) compatible



(b) Compatible With 2bit shift

Figure 4.4 Example of test pattern merging.

4.3.2. Scan-Based Test Pattern Merging

The scan-based test pattern merging technique is based on merging compatible patterns using scan shift operation. Two bits are compatible if they have the same value or any one of them is an X. Two patterns are considered compatible if every two corresponding bits in the two patterns are compatible.

Let $v_m = (v_{m,0}, v_{m,1})$, $v_n = (v_{n,0}, v_{n,1})$ be two test pattern pairs. As shown in Figure 4.4 (a), if $v_{m,1}$ and $v_{n,0}$ are compatible, we say that the two pattern pairs are compatible. As shown in Figure 4.4 (b), if we can make $v_{m,1}$ and $v_{n,0}$ compatible by shifting r bit of $v_{m,1}$, and $v_{m,1}$ and $v_{n,0}$ are not compatible without shifting, then we say that the two pattern pairs are compatible with r bit shift (in Figure 4.4(b) $r = 2$). If $v_{m,1}$ and $v_{n,0}$ are compatible, we do not need to scan-in all bits in $v_{n,0}$, and thus we need to scan-in only the last one bit in $v_{n,1}$ for v_n. If $v_{m,1}$ and $v_{n,0}$ are compatible with r bit shift, we need to scan-in the r bits in $v_{n,0}$ as well as the last one. In sum, we can reduce test data for v_n to 1 or (r+1) bits. Beside the 1 or (r+1) bits, we need to know the control data of shift times (denoted as S which includes s_0, s_1, ..., $s_{(m-1)}$ for controlling the shift time of test pattern pairs v_0,v_1, ...,$v_{(m-1)}$).

4.3.3. Procedure for Test Application Time and Test Data Volume Reduction

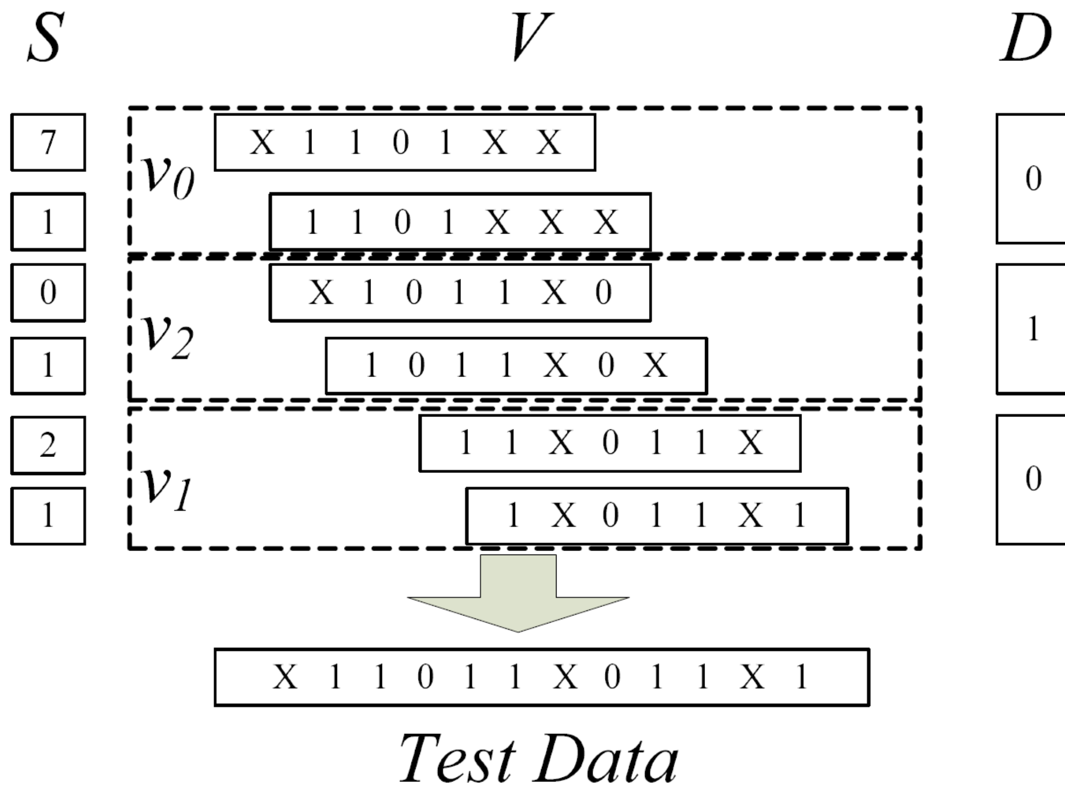


Figure 4.5 Example of test compaction.

In this subsection, we introduce the procedure for test application time and test data volume reduction. Specifically, we introduce the generation of the test data, the corresponding control data of SSG and the control data of shift times.

For illustration, an example of test compaction is shown in Figure 4

.5. We assume that the CUT contains three PUMs: p_0, p_1, p_2. Paths p_0, p_1 end in FF_i and p_2 ends in FF_j. Test pattern pairs for sensitizing paths p_0, p_1, p_2 are v_0 = (X1101XX, 1101XXX), v_1 = (11X011X, 1X011X1), v_2 = (X1011X0, 1011X0X). Here we use the LOS method to sensitize paths.

At first we need to decide the first path to be sensitized, here we choose p_0. Because p_0 ends in FF_i, the control data of SSG d_0 is set to 0 (00). The control data of shift time s_0 is 7 (111), which equals to the length of the scan chain. After the sensitization of p_0, the data stored in the FFs are $v_{0,1}$ = 1101XXX. Next, we try to

sensitize p_2. The reason why we do not select p_1 is to reduce the test application time (in greedy way). Here, v_0 and v_2 are compatible, and v_0 and v_1 are compatible with 3 bit shift. This means that sensitizing p_1 requires shifting of 4=3+1 bits while sensitizing p_0 requires only 1 bit shift. Here, the control data of SSG d_1 is set to 1 (01), and the control data of shift time s_1 is set to 0 (000). At last we sensitize p_1, when the control data of SSG d_2 is set to 0 (00), and the control data of shift time s_2 is set to 2 (010) (v_2 and v_1 are compatible with 2 bit shift). After all the steps, we get the compacted test data V = (X11011X011X1), the corresponding D (the control data of SSG) and S (the control data of shift times). The procedure for reducing test application time and test data volume is given as follows.

Procedure 1: test application time and test data volume reduction.

1. Let V' be a set of test pattern pairs without applying the proposed method. Let V be an empty set (The objective compacted data will be obtained as V). Let i be an integer, and set i=0. Select and delete one test pattern pair v_m' from V'. Add v_m' to V as v_i.

2. Select and delete one test pattern pair v_n' from V', which is compatible with v_i with the minimum shift times. Add v_n' to V as v_{i+1}; i++.

3. If $V' = \varnothing$, stop; else go to Step 2.

4.3.4. Test Application Time and Test Data Volume

The test application time T is the sum of the scan shift time of test data TS and the measurement result read out time TR. Here, we use time normalized as clock cycles. By considering the implementation of LOS test, we have the scan shift time of test data:

$$T_S = \sum_{i=0}^{n-1} (s_i + 1), \qquad (1)$$

where n is the number of the test pattern pairs. Let T_D be the read out time of the DVMC, and the measurement result read out time appears as:

$$T_R = \sum_{i=0}^{n-1} T_D \ . \qquad (2)$$

Therefore, the test application time is:

$$T = \sum_{i=0}^{n-1} (s_i + 1 + T_D) \ . \qquad (3)$$

The test data volume is the sum of data volume of of $V(V_V)$, $S(V_S)$ and $D(V_D)$. By considering the implementation of LOS test, we have the data volume of test data:

$$V_V = \sum_{i=0}^{n-1} (s_i + 1) \ . \qquad (4)$$

When we implement the scan shift of LOS, the maximum shift number is equal to the length of scan chain. Thus, the data volume of the shift time and the data volume of the control data of SSG are:

$$V_S = V_D = n \log_2 N \ , \qquad (5)$$

where N is the length of the scan chain. Therefore, the test data volume is:

$$V_{total} = \sum_{i=0}^{n-1} (s_i + 1) + 2n \log_2 N \ . \qquad (6)$$

4.4. Experimental Result

In this section, we show experimental results of the proposed test compaction method. In this evaluation, we use ISCAS 89 benchmark circuits. The initial test sets are constructed from the LOS test sets of [45]. The test set detects all the detectable transition faults under the single-path sensitization condition. A FF is inserted to each primary input, and arbitrary values can be assigned to each register with scan-in operation. We use the DVMC which has 14bit registers. Thus, we need 14 clock cycles to read out the result of the DVMC. First, we evaluate the test application time reducing effect of the proposed procedure. Next, we evaluate the data volume compaction effect of the proposed procedure. In the proposed procedure, we get a test pattern pair v_m from the original test set (the LOS test set of [45]). Then, we add v_m to the new test set and delete it from the original test set. After that we find another test pattern pair v_n which is compatible with v_m with the minimum shift times, add v_n to the new test set and delete it from the original test set. We repeat the above steps until the original test set is empty. We also compare the results of methods with/without reordering test patterns. In the procedure without pattern reordering, we get a test pattern pair v_m from the original test set (the LOS test set of [45]). Then, we add v_m to the new test set and delete it from the original test set. After that we get another test pattern pair v_{m+1} which is the next pair to v_m in the original test set. We find the minimum shift times which let v_{m+1} compatible with v_m, add v_{m+1} to the new test set and delete it from the original test set. We repeat the above steps until the original test set is empty. This thesis just proposed a test compaction method by optimizing the test pattern. Note that the proposed method does not change the area overhead of the conventional on-chip delay measurement. The area overhead of the on-chip delay measurement compared to conventional scan design are 12~20% for some large ISCAS89 circuits [19],[45].

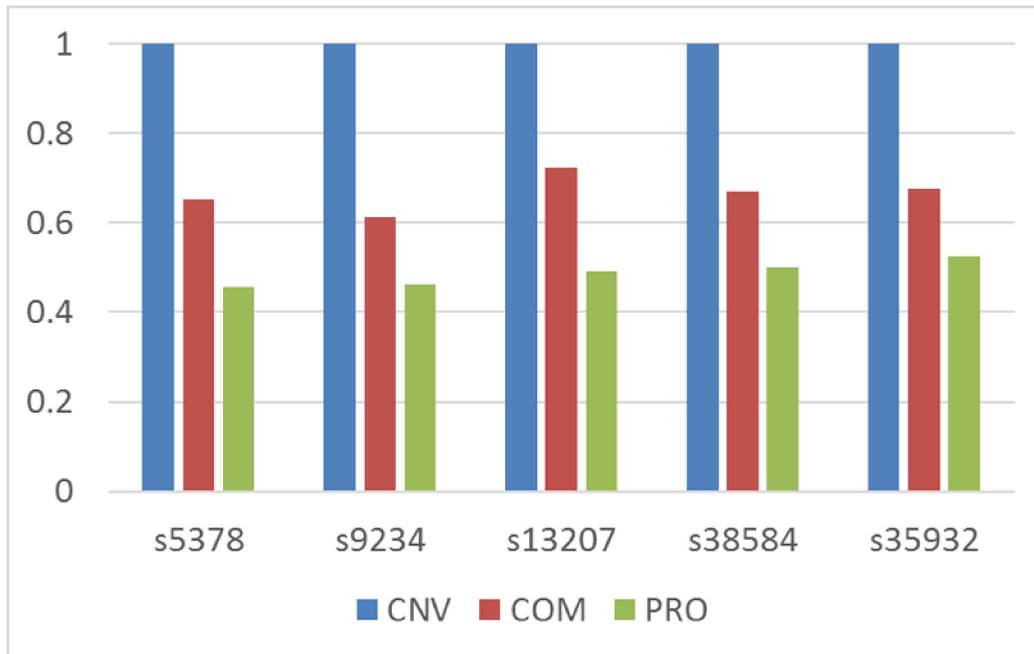


Figure 4.6 Effective of test application time reduction

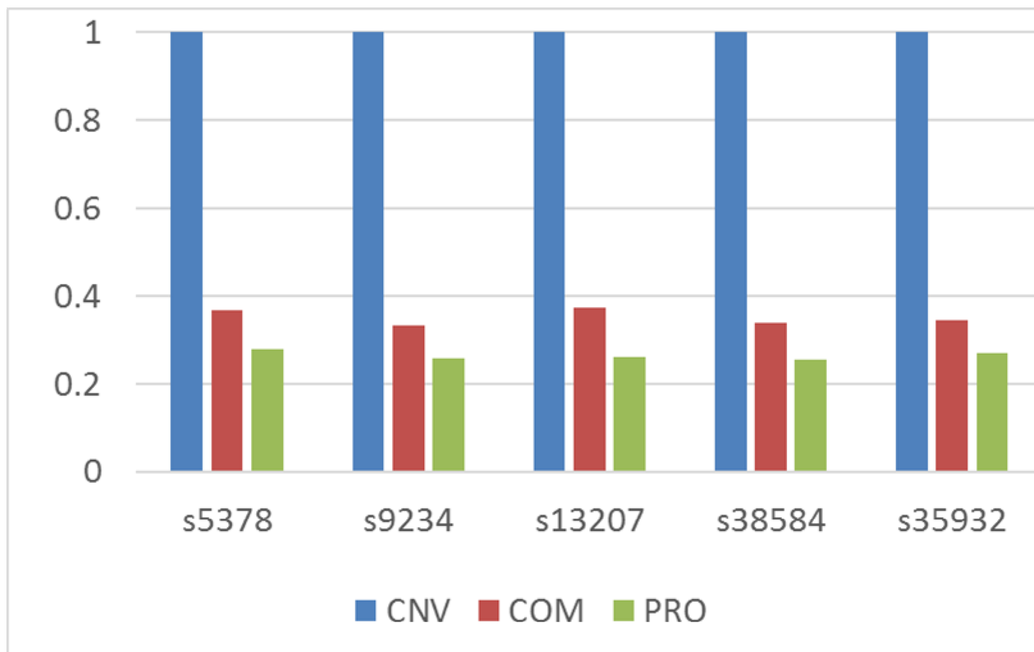


Figure 4.7 Effective of test data volume reduction

Table 4.1 shows the test application time of ISCAS89 benchmark circuits by using the conventional method [45] and the proposed method. Table 4.2 compares the test application time of methods with/without reordering test patterns. Here, test

application time is calculated by using formula (3). In these Tables, we show the circuit name with *circuit*. The column *CNV* shows the results of the conventional method in 10^4 clock cycles. The column *PRO* shows the results of the method using the proposed procedure1 (in 10^4 clock cycles). The column *COM* shows the results of the method using scan based pattern merging without reordering test patterns (in 10^4 clock cycles). The columns T_S and T_R show the scan shift time of test data and the measurement result read out time, respectively. The column T shows the test application time. The column T_{RED} shows the percentage of test application time reduction of each circuit using our method. From results of Table 4.1 we can find that the proposed method reduce the test application time very significantly for each benchmark circuits. Specifically, the test application time is reduced by 6.89~62.67% after the proposed compaction procedure. From Table 4.2, we notice that we reduce the test application time by 2.14~49.01% using scan based pattern merging without reordering test patterns. Our proposed procedure has better effect in test application time compaction than the method using only scan based pattern merging without reordering test patterns. Figure 4.6 shows the effective of teat application time reduction on large circuits, results show that our proposed method has better effective on large circuits.

Table 4.3 shows the test data volume of ISCAS89 benchmark circuits by using the conventional method [45] and the proposed method. Table 4.4 compares the test data volume for methods with/without reordering test patterns. Here, test data volume is calculated by using formula (6). In these tables, the column *circuit* shows the circuit name. The column *CNV* shows the results of the conventional method in 10^4 bits. The column *PRO* shows the results of the method using the proposed procedure1 (in 10^4 bits). The column *COM* shows the results of the method using scan based pattern merging without reordering test patterns (in 10^4 bits). The columns V_S, V_D and V_V show the data volume of the shift time, the data volume of the control data of SSG and the data volume of test patterns V. The column V_{total} shows the test data volume of each circuit. The column V_{RED} shows the percentage of test data volume reduction of each circuit by using our method. From Table 4.3 we found that the proposed method reduce the test data volume very significantly in most of the benchmark

circuits. Specifically, the test data volume is reduced by 46.39~74.86% after the proposed compaction procedure. Table 4.4 shows that we reduce the test data volume by 42.18~68.65% using scan based pattern merging without reordering test patterns. The proposed procedure has better effect in test data volume compaction than the method using only scan based pattern merging without reordering test patterns. Figure 4.7 shows the effective of teat data volume reduction on large circuits, results show that our proposed method has better effective on large circuits.

From the results of these Tables, we note that our proposed method can reduce the test application time and test data volume in each ISCAS89 benchmark circuits. Moreover, it has better effect on large circuits. The reason is that large circuits have more test patterns. We have more choice to minimize the shift times. Then, we have better compaction effect on large circuits.

Table 4.1 Test application time of ISCAS89 benchmark circuits.

circuit	*CNV*(10^4*clocks*)			*PRO*(10^4*clocks*)			
	T_S	T_R	T	T_S	T_R	T	T_{RED}(%)
s298	0.15	0.14	0.29	0.06	0.14	0.20	29.48
s344	0.26	0.23	0.49	0.16	0.23	0.38	21.46
s349	0.26	0.23	0.49	0.15	0.23	0.38	21.76
s382	0.34	0.22	0.56	0.14	0.22	0.36	35.33
s386	0.07	0.15	0.22	0.05	0.15	0.20	9.04
s444	0.29	0.18	0.47	0.11	0.18	0.29	38.39
s510	0.06	0.12	0.18	0.05	0.12	0.16	6.89
s526	0.38	0.24	0.62	0.13	0.24	0.37	40.87
s641	0.66	0.46	1.12	0.32	0.46	0.78	29.85
s713	0.42	0.29	0.71	0.17	0.29	0.47	34.64
s820	0.10	0.24	0.34	0.06	0.24	0.30	11.11
s832	0.10	0.23	0.33	0.06	0.23	0.29	11.23
s953	0.71	0.33	1.04	0.36	0.33	0.69	34.18
s1196	1.74	1.28	3.01	1.03	1.28	2.31	23.39
s1238	1.75	1.29	3.03	1.06	1.29	2.34	22.81
s1423	4.74	0.88	5.62	1.22	0.88	2.10	62.67
s1488	0.14	0.29	0.43	0.08	0.29	0.37	14.20
s1494	0.14	0.28	0.42	0.08	0.28	0.36	14.88
s5378	25.81	2.01	27.82	10.67	2.01	12.68	54.44
s9234	34.56	2.11	36.67	14.86	2.11	16.97	53.71
s13207	133.66	2.79	136.46	64.07	2.79	66.87	51.00
s38584	1823.95	17.57	1841.52	905.14	17.57	922.72	49.89
s35932	1890.83	15.31	1906.14	985.42	15.31	1000.73	47.50

Table 4.2 Test application time of with/without reordering test patterns.

circuit	COM(10^4clocks)				PRO(10^4clocks)			
	T_S	T_R	T	T_{RED}(%)	T_S	T_R	T	T_{RED}(%)
s298	0.11	0.14	0.25	13.06	0.06	0.14	0.20	29.48
s344	0.21	0.23	0.43	11.25	0.16	0.23	0.38	21.46
s349	0.21	0.23	0.44	10.15	0.15	0.23	0.38	21.76
s382	0.25	0.22	0.46	16.98	0.14	0.22	0.36	35.33
s386	0.07	0.15	0.21	2.64	0.05	0.15	0.20	9.04
s444	0.21	0.18	0.39	15.52	0.11	0.18	0.29	38.39
s510	0.05	0.12	0.17	3.05	0.05	0.12	0.16	6.89
s526	0.25	0.24	0.49	21.94	0.13	0.24	0.37	40.87
s641	0.44	0.46	0.90	19.60	0.32	0.46	0.78	29.85
s713	0.28	0.29	0.58	19.00	0.17	0.29	0.47	34.64
s820	0.08	0.24	0.32	5.05	0.06	0.24	0.30	11.11
s832	0.08	0.23	0.31	5.51	0.06	0.23	0.29	11.23
s953	0.51	0.33	0.84	19.35	0.36	0.33	0.69	34.18
s1196	1.38	1.28	2.66	11.74	1.03	1.28	2.31	23.39
s1238	1.41	1.29	2.70	11.14	1.06	1.29	2.34	22.81
s1423	1.98	0.88	2.87	49.01	1.22	0.88	2.10	62.67
s1488	0.12	0.29	0.41	6.05	0.08	0.29	0.37	14.20
s1494	0.11	0.28	0.39	6.18	0.08	0.28	0.36	14.88
s5378	16.17	2.01	18.18	34.66	10.67	2.01	12.68	54.44
s9234	20.37	2.11	22.48	38.70	14.86	2.11	16.97	53.71
s13207	95.94	2.79	98.73	27.65	64.07	2.79	66.87	51.00
s38584	1216.46	17.57	1234.03	32.99	905.14	17.57	922.72	49.89
s35932	1271.48	15.31	1286.78	32.49	985.42	15.31	1000.73	47.50

Table 4.3 Test data volume of ISCAS89 benchmark circuits.

circuit	*CNV*(10^4*bit*)				*PRO*(10^4*bit*)				
	V_S	V_D	V_V	V_{total}	V_S	V_D	V_V	V_{total}	V_{RED}(%)
s298	0.04	0.04	0.33	0.41	0.04	0.04	0.08	0.16	60.59
s344	0.06	0.06	0.78	0.91	0.06	0.06	0.29	0.42	54.22
s349	0.06	0.06	0.78	0.91	0.06	0.06	0.28	0.41	54.39
s382	0.08	0.08	0.74	0.89	0.08	0.08	0.18	0.33	63.15
s386	0.03	0.03	0.27	0.33	0.03	0.03	0.12	0.18	46.39
s444	0.06	0.06	0.62	0.75	0.06	0.06	0.13	0.26	65.04
s510	0.03	0.03	0.42	0.47	0.03	0.03	0.20	0.25	47.11
s526	0.09	0.09	0.83	1.00	0.09	0.09	0.16	0.33	66.62
s641	0.16	0.16	3.54	3.87	0.16	0.16	1.44	1.77	54.33
s713	0.10	0.10	2.26	2.47	0.10	0.10	0.88	1.09	55.69
s820	0.05	0.05	0.78	0.88	0.05	0.05	0.35	0.45	48.45
s832	0.05	0.05	0.75	0.85	0.05	0.05	0.34	0.44	48.50
s953	0.12	0.12	2.12	2.36	0.12	0.12	0.71	0.95	59.96
s1196	0.46	0.46	5.84	6.76	0.46	0.46	2.22	3.13	53.65
s1238	0.46	0.46	5.88	6.80	0.46	0.46	2.25	3.17	53.39
s1423	0.44	0.44	11.48	12.37	0.44	0.44	2.23	3.11	74.86
s1488	0.06	0.06	0.57	0.70	0.06	0.06	0.23	0.35	49.87
s1494	0.06	0.06	0.56	0.68	0.06	0.06	0.22	0.34	50.29
s5378	1.15	1.15	61.33	63.63	1.15	1.15	15.54	17.83	71.97
s9234	1.21	1.21	74.50	76.91	1.21	1.21	17.58	19.99	74.01
s13207	1.99	1.99	279.16	283.15	1.99	1.99	70.06	74.04	73.85
s38584	13.81	13.81	3675.23	3702.84	13.81	13.81	918.95	946.57	74.44
s35932	12.03	12.03	3855.68	3879.74	12.03	12.03	1022.60	1046.65	73.02

Table 4.4 Test data volume of with/without reordering test patterns.

circuit	*COM*(10^4*bit*)					*PRO*(10^4*bit*)				
	V_S	V_D	V_V	V_{total}	V_{RED}(%)	V_S	V_D	V_V	V_{total}	V_{RED}(%)
s298	0.04	0.04	0.13	0.21	49.20	0.04	0.04	0.08	0.16	60.59
s344	0.06	0.06	0.34	0.47	48.74	0.06	0.06	0.29	0.42	54.22
s349	0.06	0.06	0.34	0.47	48.15	0.06	0.06	0.28	0.41	54.39
s382	0.08	0.08	0.28	0.43	51.72	0.08	0.08	0.18	0.33	63.15
s386	0.03	0.03	0.13	0.19	42.18	0.03	0.03	0.12	0.18	46.39
s444	0.06	0.06	0.24	0.37	50.78	0.06	0.06	0.13	0.26	65.04
s510	0.03	0.03	0.21	0.26	45.66	0.03	0.03	0.20	0.25	47.11
s526	0.09	0.09	0.28	0.45	54.83	0.09	0.09	0.16	0.33	66.62
s641	0.16	0.16	1.55	1.88	51.37	0.16	0.16	1.44	1.77	54.33
s713	0.10	0.10	1.00	1.20	51.17	0.10	0.10	0.88	1.09	55.69
s820	0.05	0.05	0.37	0.47	46.12	0.05	0.05	0.35	0.45	48.45
s832	0.05	0.05	0.36	0.46	46.30	0.05	0.05	0.34	0.44	48.50
s953	0.12	0.12	0.86	1.10	53.42	0.12	0.12	0.71	0.95	59.96
s1196	0.46	0.46	2.57	3.48	48.46	0.46	0.46	2.22	3.13	53.65
s1238	0.46	0.46	2.60	3.52	48.19	0.46	0.46	2.25	3.17	53.39
s1423	0.44	0.44	2.99	3.88	68.65	0.44	0.44	2.23	3.11	74.86
s1488	0.06	0.06	0.26	0.38	44.83	0.06	0.06	0.23	0.35	49.87
s1494	0.06	0.06	0.25	0.37	44.91	0.06	0.06	0.22	0.34	50.29
s5378	1.15	1.15	21.04	23.34	63.32	1.15	1.15	15.54	17.83	71.97
s9234	1.21	1.21	23.08	25.49	66.85	1.21	1.21	17.58	19.99	74.01
s13207	1.99	1.99	101.92	105.91	62.60	1.99	1.99	70.06	74.04	73.85
s38584	13.81	13.81	1230.27	1257.88	66.03	13.81	13.81	918.95	946.57	74.44
s35932	12.03	12.03	1308.65	1332.71	65.65	12.03	12.03	1022.60	1046.65	73.02

4.5. Conclusion

This chapter proposed a test compaction method for on-chip delay measurements. To reduce test application time and test data volume of the on-chip delay measurement, this chapter presented a method that uses scan-based test pattern merger. Experimental results on ISCAS89 benchmark circuits showed that the proposed method reduced the test application time by 6.89~62.67% and the test data volume by 46.39~74.86%.

In this work, we proposed a method to reduce test application time and test data volume by using only scan-based test pattern merging. By analyzing the results in Table 4.1, we noticed that the measurement result read out time occupied a considerable part of the total test time. In our future work, we will consider a new method to reduce the measurement result read out time. In this work, we used an in-house ATPG. As our future work, we also try to use a commercial ATPG for efficient test generation.

5. Fault Coverage Improvement and Test Compaction under LOS+LOC Test

5.1. SUMMARY

To detect small delay defects, on-chip delay measurement which measures the delay time of paths in the circuit under test (CUT) was proposed. However, small-delay defect coverage of on-chip delay measurement method is very low. We have proposed methods of small-delay defect coverage improvement (in section 3) and test compaction (in section 4) for on-chip delay measurement method. Due to LOS test has several advantages (higher fault coverage, smaller test pattern sets, and lower test generation cost. Results of the pre-simulation tests which confirm advantages of LOS test will be shown in section 5.4) over LOC test, we used conventional methods under LOS test. Normally, LOS test can achieve higher defect coverage with smaller test pattern set over LOC test. Test patterns generated by the LOS test have more don't cares than those generated by the LOC test. This implies that test patterns generated by the LOS test have more room for compaction. In addition, a fast scan enable is also not necessary for the on-chip delay measurement. For these reasons, we proposed conventional methods under LOS test. However, with the increasing of scan segment's and test points numbers, the effect for the coverage improvement is not very notable for some circuits. It means that this method leads to high area overhead for high defect coverage. In addition, some faults are not testable by the LOS test.

To improve the small-delay defect coverage of on-chip delay measurement method with small hardware overhead, this study presents a method using LOS+LOC based on a conventional method. We also propose a test compaction procedure under LOS+LOC that reduces scan shift time and test data volume using test pattern merging. The evaluation results show that, compare with the conventional LOS+LOC method, the proposed method reduces the test application time by 47.87~54.02% and test data volume by 71.72~74.50%. Compare with the conventional LOS based method, the proposed procedure can provide similar or higher defect coverage with very small hardware overhead. Specifically, the hardware overhead is 9.27~35.21% smaller than the conventional method. The proposed test compaction procedure reduces the test application time by 4.47~29.29% and test data volume by 4.46~

29.96%.

The rest of the paper is organized as follows. Section 5.2 compare the LOS and LOC test. Section 5.3 introduces the proposed coverage improvement method. Section 5.4 explains the proposed test compact procedure. Section 5.5 evaluates the proposed methods. Finally, section 5.6 concludes the chapter..

5.2. LOS Test vs. LOC Test

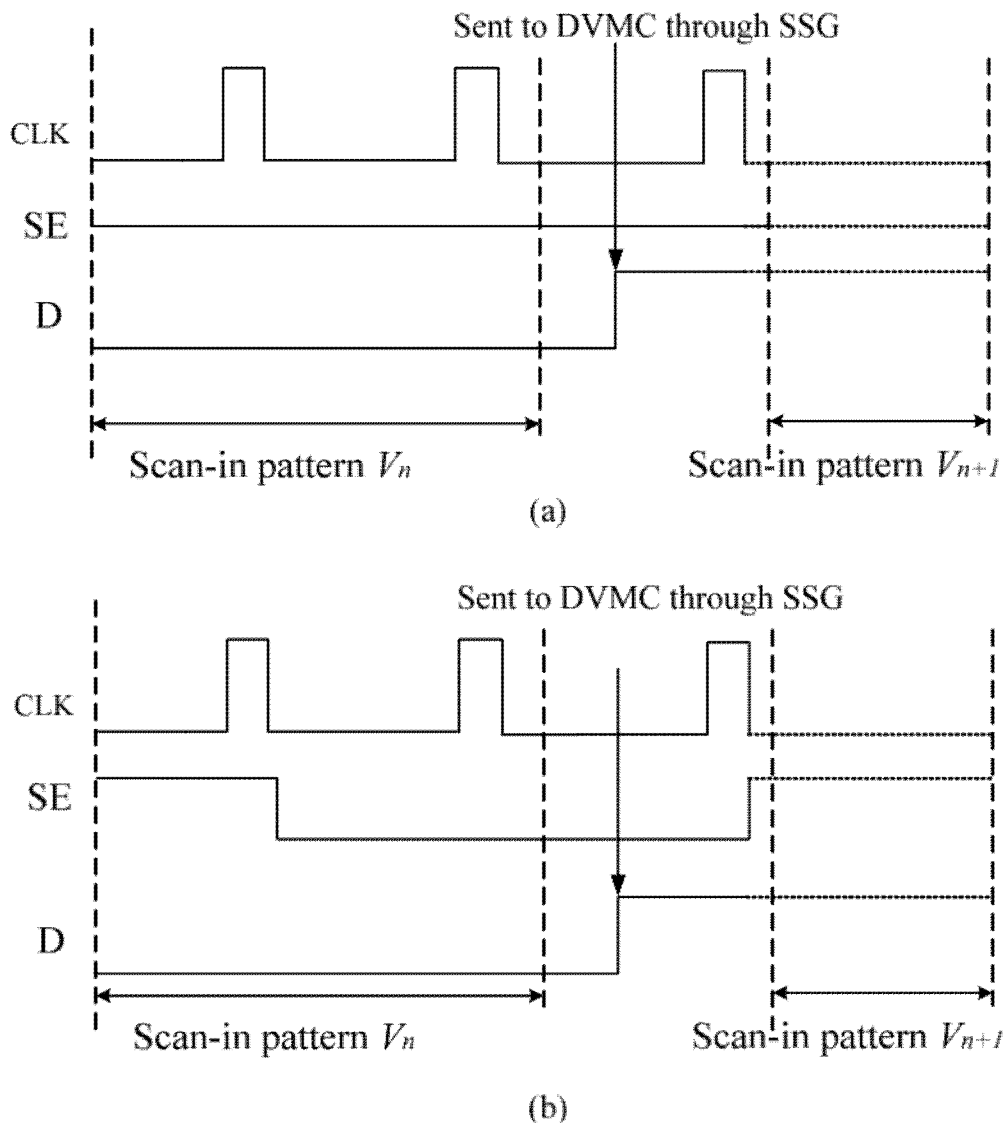


Figure 5.1 Operation of (a) LOS test and (b) LOC test of on-chip delay measurement.

As intruded in section 3 and section 4, for circuits using scan, there are two approaches for delay fault testing: Launch-off-Shift (LOS), and Launch-off-Capture (LOC) methods. In the LOS method, the second pattern for two-pattern testing is generated by an one-bit shift of the first pattern. In the LOC method, the second pattern is obtained from the circuit response to the first pattern.

Normally, the LOC test is the first choice of scan-based test method in many cases due to difficulty meeting at speed enable signal switching of the LOS test. However,

enable signal switching under system clock is not necessary in on-chip delay measurement. LOS test has several advantages (higher fault coverage, smaller test pattern sets, and lower test generation cost) over LOC test [1]. Test patterns generated by the LOS test have more room for compaction than those generated by the LOC test. For these reasons, we proposed conventional methods under LOS test. Figure 5.1 shows the waveforms of LOS test and LOC test of the on-chip delay measurement. When using the on-chip delay measurement method to detect small-delay defects on path p (from FF_i to FF_j), we set the SSG to detect transition on the input of FF_j. At the moment the transition reaches the input D of FF_j, the transition is sent to stop of the DVMC through the SSG. Then, the DVMC stops the measurement. In this process, as shown in Figure 5.1 (a), the capture operation (at speed enable signal switching) of LOS test is unnecessary unlike the conventional LOS test. On the other hand, as shown in Figure 5.1 (b), the LOC operation of the on-chip delay measurement is the same with the conventional LOC test.

5.3. Proposed Coverage Improvement Method

A method for improving small-delay defect coverage of on-chip delay measurement has been proposed in section 3. The conventional work used segmented scan and test point insertion under LOS test. This method can improve the defect coverage. However, the area overhead is high.

As the conventional work which using segmented scan and test point insertion under LOS test, leads to high area overhead. The simulation results in section 3 show that, LOS+LOC test achieve higher defect coverage than LOS or LOC. In addition, some faults that are not testable by the LOS test. If we insert the same number of test points, the defect coverage of LOS+LOC also will be higher than only using LOS. To achieve high defect coverage with small area overhead, we consider using LOS+LOC based on the conventional method in [16].

Figure 5.2 shows the path list and test pattern generation flow. Firstly, segmented scan and test point insertion will be used to modify a circuit under test. The values of N_S, N_C and N_O are decided by results of hundreds of pre-simulation tests. We can use these values to get a higher coverage with an acceptable hardware overhead. Note that, if we insert a control point on a line *l* that lies on a critical path, then inserting a control point on *l* may degrade into signal lines that are on a critical path. Before the control point insertion procedure, we identify all signal lines that lie on a critical path. We delete these signal lines from the potential test point set.

Next, we delete the paths that cannot be sensitized under the single-path sensitization using LOS+LOC. We get the new path list *pl* and fault list TF. LOS test has several advantages (higher fault coverage, smaller test pattern sets, and lower test generation cost) over LOC test. A fast scan enable is also not necessary in the on-chip delay measurement. The effect of LOS using the proposed method is better than LOC [45]. Normally, test patterns generated by the LOS test have more don't cares, i.e. fewer specified bits, than those generated by the LOC test. This implies that test patterns generated by the LOS test have more room for compaction. Therefore, when we try to sensitize a path, LOS test has priority over the LOC test.

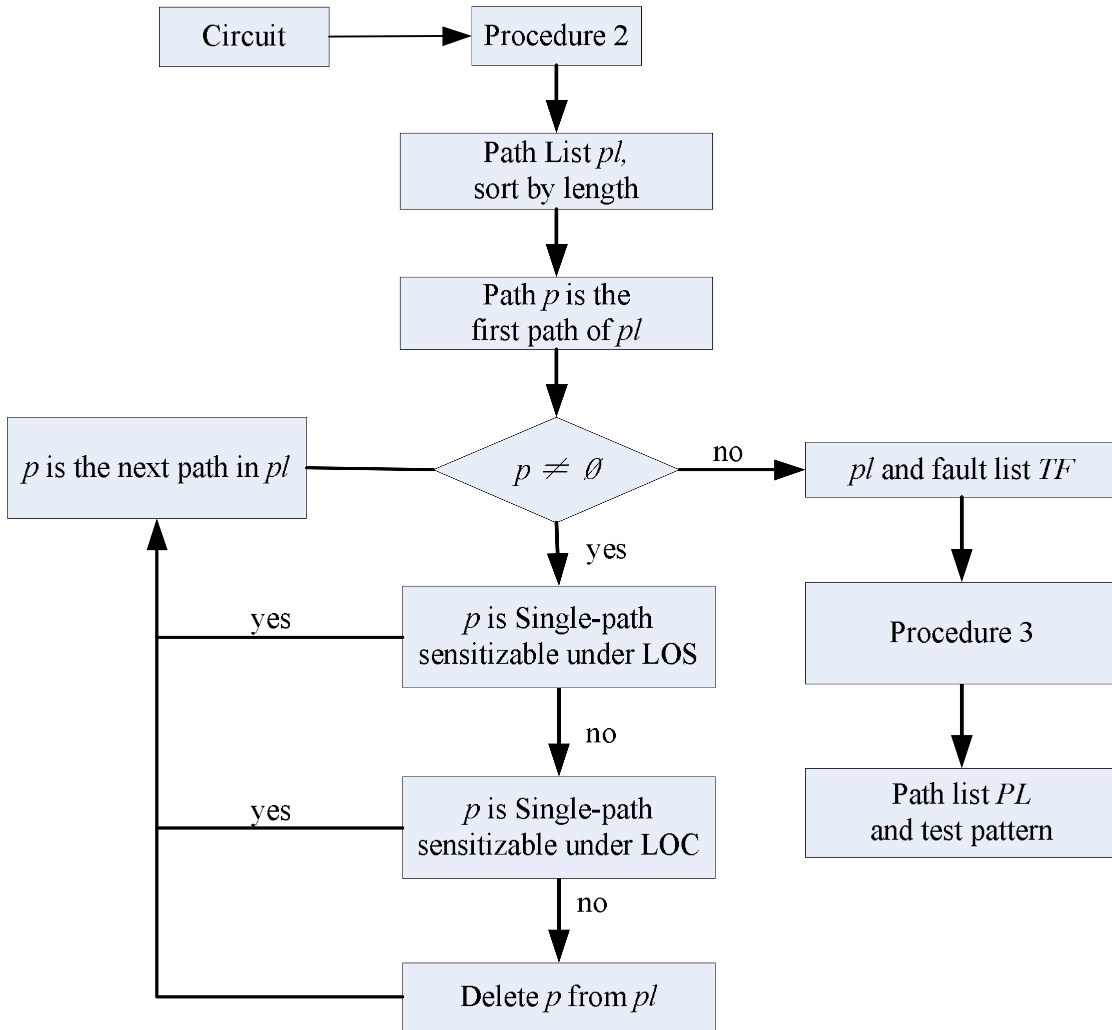


Figure 5.2 Pathlist and test pattern generation flow.

Finally, we select a minimal set of paths applied for TF by using a greedy covering procedure. To achieve higher defect coverage with minimal paths, we should consider the number of faults that can be detected by one path. In other words, path which detects the largest number of faults will be first selected. The procedure for path selection is given next as Procedure 1.

Procedure 1: Path Selection

1) Let TF, *pl* be the set of target faults and path list. Let PL be the set of paths that be used for testing and set PL = ∅.

2) For every line path p_i in PL, let F(p_i) be an empty fault set. Find the set of faults

F(p_i) on the path p_i.

3) Select a path p_j such that F(p_j) has the largest number of faults $tf_j \in$ TF.

4) Add the path p_j to PL. Remove faults $tf_j \in$ F(p_j) from TF.

5) If TF = ∅, stop; else go to Step 3.

After this procedure, we can get the path list PL and the corresponding test pattern pairs. To achieve more effective defect coverage with the same hardware overhead, we try to set the area ratio of observation point and control point that in the overall hardware overhead. We will show the data in the next section.

5.4. Proposed Test Compact Procedure

We can achieve high fault coverage using LOS+LOC based on the conventional method. However, it also increases the amount of paths and test pattern pairs. On-chip delay measurement incurs high test cost because it uses scan design, which brings about long test application time due to scan shift operation. Thus, a method reducing test application time is strongly required.

In LOS testing of on-chip path delay measurement, the capture operation is unnecessary unlike the conventional delay testing. Thus, FFs keep the transition pattern (denoted as $v_{m,1}$) of the test pattern pair sensitizing a PUM p even after the measurement of p. If $v_{m,1}$ can be used as the initial pattern (denoted as $v_{n,0}$) of another test pattern pair (v_n which sensitizes another path p'), we can sensitize p' by just shifting 1 bit of the transition pattern (under LOS test). The proposed method uses this characteristic. LOC test is different from LOS test. In LOC testing of on-chip path delay measurement, FFs keep the circuit response of the transition pattern of one pattern pair sensitizing a PUM p. If these values can be used as the initial pattern of another test pattern pair which sensitizes another path, we can reduce the shift time. When generated test patterns in Section 5.3, we considered that LOS test patterns have more room for compaction. Therefore we can get a good test compaction result.

In one circuit, the set of paths under measurement is denoted by P (includes paths $p_0,p_1, ...,p_{(m}-1)$). Let D (which includes $d_0,d_1, ...,d_{(m}-1)$) be the control data of SSG. The data d_i selects the path p_i as the PUM. Let V (which includes test pattern pairs $v_0,v_1, ...,v_{(m}-1$) for sensitizing paths $p_0,p_1, ...,p_{(m}-1)$) be the test data. We also need to know the control data of shift times (denoted as S which includes $s_0,s_1, ...,s_{(m}-1$) for controlling the shift time of test pattern pairs $v_0,v_1, ...,v_{(m}-1)$).

We describe the procedure for test application time and test data volume reduction. Specifically, we introduce the generation of the test data, the corresponding control data of SSG and the control data of shift times.

We introduce the proposed test compaction method using an example in Table 5.1 and Table5.2. Table 5.1 shows the initial test patterns generated in section 5.3, and

Table 5.1 Initial Test Patterns

No. of *v*	*vn,*0	*vn,*1	value of FFs	*S*	*D*
*v*0	X10XXX	10XXXX	10XXXX	6	1
*v*1	111XXX	11XXX1	11XXX1	6	2
*v*2	X011XX	011XXX	011XXX	6	0
*v*3	100X01	X111XX	10011X	6	2
*v*4	110XX1	X101XX	1X100X	6	0

Table 5.2 Example of Test Compaction.

No. of *v*	*vn,*0	*vn,*1	value of FFs	*S*	*D*
*v*0	X10XXX	10XXXX	10XXXX	6	1
*v*2	X011XX	011XXX	011XXX	0	0
*v*1	111XXX	11XXX1	11XXX1	1	2
*v*4	110XX1	X101XX	1X100X	0	0
*v*3	100X01	X111XX	10011X	2	2

Table 5.2 shows the compacted result using the proposed procedure. As shown in Table 5.1, we assume that the CUT contains five PUMs: p_0, p_1, p_2, p_3, p_4. Path p_0 ends in FF_1, p_1, p_3 end in FF_2 and p_2, p_4 end in FF_0. Test pattern pairs for sensitizing paths are shown in Table 5.1 as v_0~v_4. In which $v_{n,0}$ shows the initial pattern, and $v_{n,1}$ shows the transition pattern for sensitizing path p_n. As introduces in section 5.3, when we try to sensitize a path, LOS test has priority over the LOC test. After the test generation flow in section 5.3, test patterns have two parts: patterns under LOS test and patterns under LOC test. If we try to insert a LOS (LOC) pattern into a series of LOC (LOS) patterns, the test will be very difficult for controlling the Scan Enable signal. Therefore, when we testing a circuit using these test patterns, we use patterns under LOS test firstly, after that we use patterns under LOC test. This rule also be considered in the test compaction. Here we assume paths p_0~p_2 are sensitized under LOS test, and paths p_3~ p_4 are sensitized under LOC test.

We introduce the proposed test compaction method using Table 5.1. There have two steps to compact test patterns. In the first step we compact patterns under LOS test. At

first we need to decide the first path to be sensitized, here we choose p_0. Because p_0 ends in FF_1, the control data of SSG d_0 is set to 1 (01). The control data of shift time s_0 is 6 (110), which equals to the length of the scan chain. After the sensitization of p_0, the data stored in the FFs are $v_{0,1}$ = 10XXXX. Next, we try to sensitize p_2. The reason why we do not select p_1 is to reduce the test application time (in a greedy way). Here, v_0 and v_2 are compatible, and v_0 and v_1 are compatible with 2 bit shift. This means that sensitizing p_1 requires shifting of 3=2+1 bits while sensitizing p_0 requires only 1 bit shift. Here, the control data of SSG d_1 is set to 0 (00), and the control data of shift time s_1 is set to 0 (000). Next, we sensitize p_1, when the control data of SSG d_2 is set to 2 (10), and the control data of shift time s_2 is set to 1 (001) (v_2 and v_1 are compatible with 1 bit shift).

After compact patterns under LOS test, we try to compact patterns under LOC test. Here, v_1 and v_4 are compatible, and v_1 and v_3 are compatible with 4 bit shift. Therefore, we try to sensitize p_4 before p_3. the control data of SSG d_3 is set to 0 (00), and the control data of shift time s_3 is set to 0 (000). At last, we sensitize p_3, when the control data of SSG d_4 is set to 2 (10), and the control data of shift time s_4 is set to 2 (010) (v_4 and v_3 are compatible with 2 bit shift).

After all the steps, we get the compacted test data V, the corresponding D (the control data of SSG) and S (the control data of shift times) as shown in Table 5.1. The procedure for reducing test application time and test data volume is given as follows. Firstly, we use the procedure to compact test patterns under LOS test. After that we use the procedure to compact test patterns under LOC test.

Procedure 4: test application time and test data volume reduction.

1) Let V' be a set of test pattern pairs without applying the proposed method. It consists of two parts: test patterns under LOS test V_{LOS}', and test patterns under LOC test V_{LOC}'. Let V be an empty set (The objective compacted data will be obtained as V). Let i be an integer, and set i=0. Select and delete one test pattern pair v_m' from V_{LOS}'. Add v_m' to V as v_i.

2) Select and delete one test pattern pair v_n' from V_{LOS}', which is compatible with v_i with the minimum shift times. Add v_n' to V as v_{i+1}; i++. 3) If $V_{LOS}' = \emptyset$, go to Step 4; else go to Step 2.

4) Select and delete one test pattern pair v_p' from V_{LOC}', which is compatible with v_i with the minimum shift times. Add v_p' to V as v_{i+1}; i++.

5) If $V_{LOC}' = \emptyset$, stop; else go to Step 4.

5.5. EVALUATION

In this section, we give results of the pre-simulation tests which confirm the fact that LOS test has several advantages over LOC test. We study effects of these introduced techniques. Firstly, we study the area reduction effects of these introduced techniques on the set of faults undetectable by LOS+LOC test under the single-path sensitization condition. The hardware overhead will be evaluated on similar or higher defect coverage with the conventional method. We provide experimental results of the proposed test compaction method. In this evaluation, we use ISCAS89 benchmark circuits. The test patterns are generated with in-house ATPG based on what is used in [45].

5.5.1. LOS Test vs. LOC Test

As we know, the defect coverage increases with the increase of test application time. Figure 5.3 shows the evaluate results of LOS/LOC test under the conventional robust test, and Figure 5.4 shows the result of LOS/LOC test under the proposed method (using s5378). In these Figures, the y-axis shows the defect coverage, and the x-axis shows the test application time. From the experiment result, the defect coverage is improved with the increases of the test application time. In all cases, the defect coverage of LOS test is better than the LOC test. We also observed that the test application time of LOC test is longer than the LOS test. Due to the launch patter should be calculated from the response of a CUT at the capture clock (because launch and capture clock are applied while scan enable is low), LOC test set suffers from large test set size and low fault coverage compared to LOS test set. In addition, LOC test requires more ATPG computation and restrictions than LOS test set whose launch pattern is simply shifted in. In the robust test, the defect coverage increasing with the test application time closes to linear increase. The reason for that is we didn't consider test compaction in Figure 5.3. In Figure 5.4, which we considered test compaction, the defect coverage increasing with the test application time of LOS test is much faster than the LOC test. It shows that test patterns generated by the LOS test have more room for compaction. The reason is that test patterns generated by the LOS test have more don‘t cares than those generated by the LOC test.

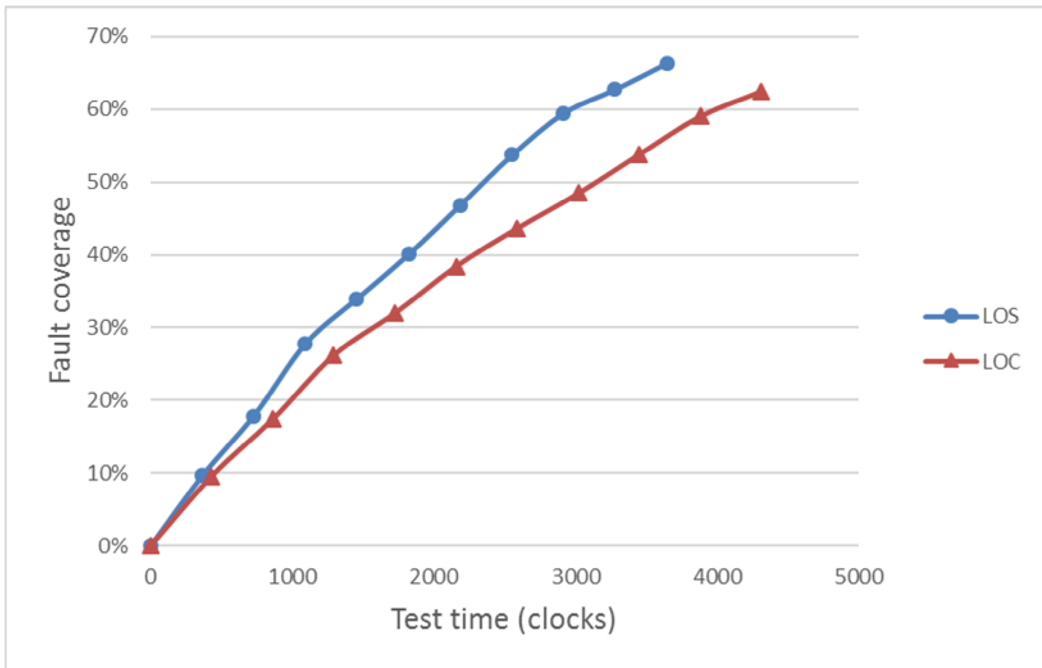


Figure 5.3. Defect coverage vs. test time of robust test (s5378).

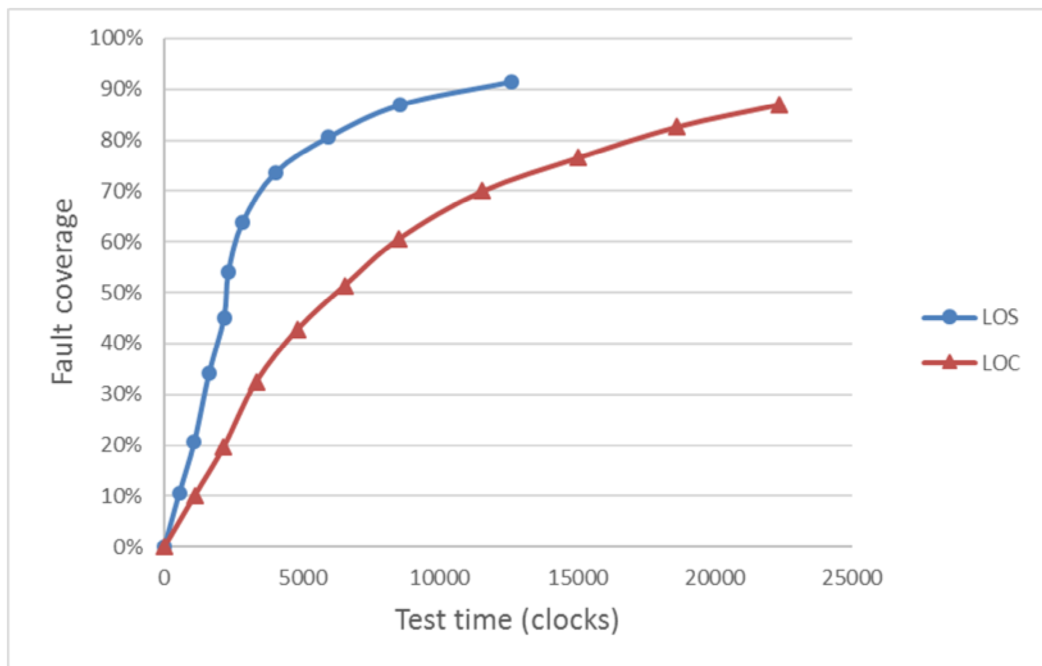


Figure 5.4. Defect coverage vs. test time of proposed method (s5378).

Experiment results in Figure 5.3 and Figure 5.4 show that LOS test has several advantages (higher fault coverage, smaller test pattern sets, easier for test

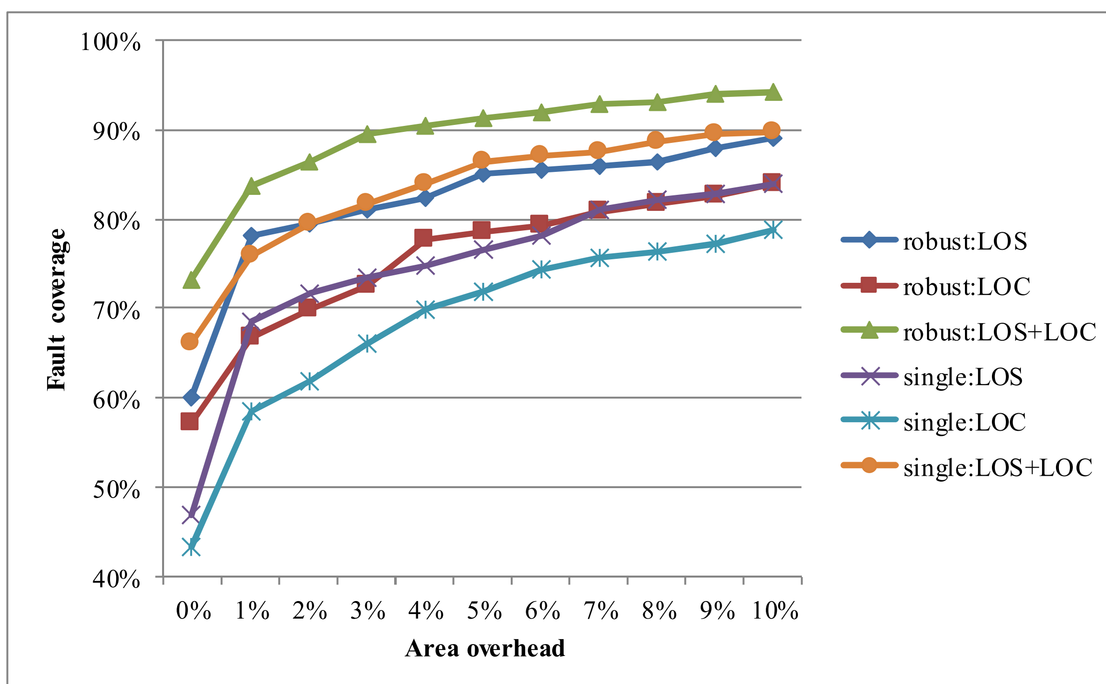


Figure 5.5. Effect comparison of the proposed method between LOS/LOC/LOS+LOC under robust sensitization and single-path sensitization of s9234.

compaction) over LOC test. In the conventional research, we only considered test under LOS test. However, it lead to high area overhead for high defect coverage. To improve the small-delay defect coverage of on-chip delay measurement method with small hardware overhead, this study presents a method using LOS+LOC test.

5.5.2. The Defect Coverage Improvement Effect

Figure 5.5 shows the effect comparison of the proposed method for LOS/LOC/LOS+LOC under robust sensitization and single-path sensitization of s9234. The y-axis shows the defect coverage, and the x-axis shows the area overhead. From the experiment result, the proposed method is effective for defect coverage improvement. We find that, for all cases, the defect coverage is improved with the increases of the area overhead. In the same sensitization condition, the LOS+LOC test is the most effective. And the LOS test is more effective than the LOC test. For example, in single-path sensitization, when we set the area overhead to 5%, the defect

coverage of LOC, LOS and LOS+LOC test are 71.83%, 76.49% and 86.42%, respectively. We can find similar results in the robust test. Therefore, when we try to sensitize a path in the test pattern generation flow, LOS test has priority over the LOC test.

Table 5.3 and 5.4 show the defect coverage improvement effects of the proposed method. In these Tables, the column *Circuit* shows the circuit name. Columns LOS+LOC, *CON* and *Proposed* show the evaluation results of the conventional LOS+LOC method, conventional method and the proposed method. The column N_T gives the number of the test pattern pairs (As only one path is selected to be measured for each test pattern pair, N_T also gives the number of measurements). Columns N_S, N_C and N_C show the numbers of scan segments, inserted control points and inserted observation points, respectively. Columns $C_0(\%)$, $C_1(\%)$ and $C_2(\%)$ report the defect coverage. Columns $C_{IMP1}(\%)$ and $C_{IMP2}(\%)$ report the effect of defect coverage improvement by using defect coverage improving techniques (Compare with LOS+LOC and conventional method, respectively.). Columns $S_0(mm^2)$, $S_1(mm^2)$ and $S_2(mm^2)$ report the area. The column AO reports the area overhead, which is calculated by $AO = (S_1 - S_0)/S_0 \times 100(\%)$. As shown in the results, we got an acceptable defect coverage with a small area overhead. Compare with the original LOS+LOC, the proposed procedure improved the defect coverage 16.21~28.23% with 2.60~5.24% of hardware overhead. With similar or higher defect coverage, the area overhead of the proposed system is 9.27~35.21% lower than the method from section 3. For example, the defect coverage of s13207 can be improved to 91.13% with 19.37% of area overhead by using method from section 3. The proposed method provides similar defect coverage (90.84%) with only 4.20% of area overhead.

5.5.3. The Test Compaction Effect

We provide experimental results of the proposed test compaction method. First, we evaluate the test application time reducing effect of the proposed procedure. Next, we evaluate the data volume compaction effect of the proposed procedure. Columns LOS+LOC, *CON* and *Proposed* show the evaluation results of the conventional LOS+LOC method, conventional method in section 4 and the proposed method.

Columns T_S and T_R show the scan shift time of test data and the measurement result read out time, respectively. The column T shows the test application time. Columns T_{RED1} and T_{RED2} show the percentage of test application time reduction of each circuit using our method (Compare with LOS+LOC and conventional method in section 4, respectively.). Columns V_S, V_D and V_V show the data volume of the shift time, the data volume of the control data of SSG and the data volume of test patterns V. The column V_{total} shows the test data volume of each circuit. Columns V_{RED1} and V_{RED2} show the percentage of test data volume reduction of each circuit by using our method (Compare with LOS+LOC and conventional method in section 4, respectively.).

In this evaluation, we use ISCAS 89 benchmark circuits. The initial test sets are constructed from the LOS+LOC test sets of section 5.5.2. The test set detects all the detectable transition faults under the single-path sensitization condition. A register is inserted to each primary input, and arbitrary values can be assigned to each register with scan in operation. We use the ring oscillator based DVMC which has 14bit registers. Thus, we need 14 clock cycles to read out the result of the DVMC.

Table 5.5 shows the test application time of ISCAS89 benchmark circuits. Table 5.6 shows the test data volume of ISCAS89 benchmark circuits by using the conventional method in section 4 and the proposed method. From the results of Table 5.5 and Table 5.6, we note that the proposed method is effective for test compaction. The evaluation results show that, compare with the conventional LOS+LOC method, the proposed method reduces the test application time by 47.87~54.02% and test data volume by 71.72~74.50%. Compare with the conventional LOS based method, the proposed test compaction procedure reduces the test application time by 4.47~29.29% and test data volume by 4.46~29.96%.

Table 5.3 Effect of Defect Coverage Improvement.

Circuit	LOS+LOC		CON				*Proposed*			
	N_{T0}	C_0(%)	(N_S/N_C/N_O)	N_{T1}	C_1(%)	(N_S/N_C/N_O)	N_2	C_2(%)	C_{IMP1}(%)	C_{IMP2}(%)
s5378	243	68.28	32/50/100	943	91.75	8/10/20	773	97.11	28.83	5.36
s9234	324	66.03	64/100/400	1347	90.90	16/20/40	935	87.13	21.10	-3.77
s13207	517	69.14	64/200/200	1994	91.13	16/30/60	1571	90.84	21.70	-0.29
s38584	6544	75.54	128/200/500	12552	90.35	32/50/100	11349	95.65	20.11	5.30
s35932	5971	62.17	128/200/500	10935	73.61	32/50/100	10524	78.38	16.21	4.77

Table 5.4 Effect of Area Reduction.

Circuit	LOS+LOC	CON			*Proposed*			
	$S_0(mm^2)$	(N_S/N_C/N_O)	$S_1(mm^2)$	AO_1(%)	(N_S/N_C/N_O)	$S_2(mm^2)$	AO(%)	$AO-AO_1$(%)
s5378	0.118	32/50/100	0.143	21.46	8/10/20	0.123	4.24	17.22
s9234	0.191	64/100/400	0.268	40.45	16/20/40	0.201	5.24	35.21
s13207	0.357	64/200/200	0.426	19.37	16/30/60	0.372	4.20	15.17
s38584	0.889	128/200/500	1.004	12.85	32/50/100	0.915	2.92	9.93
s35932	0.963	128/200/500	1.077	11.87	32/50/100	0.988	2.60	9.27

Table 5.5 Test application time of ISCAS89 benchmark circuits.

Circuit	LOS+LOC*(10^6 clocks)*			CON*(10^6 clocks)*			*Proposed(10^6 clocks)*				
	T_S	*TR*	*T*	T_S	*TR*	*T*	T_S	*TR*	*T*	*TRED*1*(%)*	*TRED*2*(%)*
s5378	0.21	0.02	0.23	0.11	0.02	0.13	0.09	0.02	0.10	54.02	21.14
s9234	0.24	0.01	0.25	0.15	0.02	0.17	0.10	0.01	0.12	52.97	29.29
s13207	1.05	0.02	1.08	0.64	0.03	0.67	0.51	0.02	0.54	50.09	19.25
s38584	16.49	0.16	16.65	9.05	0.18	9.23	8.16	0.16	8.32	50.02	9.83
s35932	18.20	0.15	18.34	9.85	0.15	10.01	9.42	0.15	9.56	47.87	4.47

Table 5.6 Test data volume of ISCAS89 benchmark circuits.

Circuit	LOS+LOC*(10^6 bits)*				CON*(10^6 bits)*				*Proposed(10^6 bits)*					
	V_S	*VD*	V_V	*Vtotal*	V_S	*VD*	V_V	*Vtotal*	V_S	*VD*	V_V	*Vtotal*	*VRED*1*(%)*	*VRED*2*(%)*
s5378	0.01	0.01	0.50	0.52	0.01	0.01	0.16	0.18	0.01	0.01	0.13	0.15	71.72	15.87
s9234	0.01	0.01	0.52	0.53	0.01	0.01	0.18	0.20	0.01	0.01	0.12	0.14	73.59	29.96
s13207	0.02	0.02	2.20	2.23	0.02	0.02	0.70	0.74	0.02	0.02	0.56	0.59	73.37	20.31
s38584	0.12	0.12	33.23	33.48	0.14	0.14	9.19	9.47	0.12	0.12	8.29	8.54	74.50	9.78
s35932	0.12	0.12	37.11	37.34	0.12	0.12	10.23	10.47	0.11	0.11	9.77	10.00	73.21	4.46

5.6. CONCLUSION

To improve the small-delay defect coverage of on-chip delay measurement method with small hardware overhead, this study presents a method using LOS+LOC based on a conventional method. In addition, we proposed the corresponding test compaction procedure. Compare with the conventional LOS+LOC method, the proposed method reduces the test application time by 47.87~54.02% and test data volume by 71.72~74.50%. Compare with the conventional LOS based method, the proposed procedure can provide similar or higher defect coverage with very small hardware overhead. Specifically, the hardware overhead is 9.27~35.21% smaller than the conventional method. The proposed test compaction procedure reduces the test application time by 4.47~29.29% and test data volume by 4.46~29.96%.

6. CONSLUSION

On-chip delay measurement have been proposed to detect small-delay on VLSI chips. When using on-chip delay measurement method to detect small-delay defects, PUMs are sensitized by delay fault test patterns. However, this thesis reveals that the robust test patterns are not suitable for on-chip delay measurement. Specifically, they require test generation under the single-path sensitization condition, which causes its small-delay fault coverage to be very low. To improve fault coverage, this thesis introduces techniques which use segmented scan and test point insertion (TPI).

Evaluation results give evidence that, for improving small-delay fault coverage of on-chip delay measurement, the use of segmented scan and *test point insertion* (TPI) is efficient. Evaluation results indicate that we can get an acceptable fault coverage, by combining these techniques for *launch off shift* (LOS) testing under the single-path sensitization condition. Specifically, fault coverage is improved 27.02~47.74% with 6.33~12.35% of hardware overhead.

On-chip delay measurement incurs high test cost because it uses scan design, which brings about long test application time due to scan shift operation. Thus, a method reducing test application time is strongly required. In on-chip path delay measurement, the capture operation is unnecessary unlike the conventional delay testing. Thus, FFs keep the transition pattern (denoted as $v_{m,1}$) of the test pattern pair sensitizing a PUM p even after the measurement of p. If $v_{m,1}$ can be used as the initial pattern (denoted as $v_{n,0}$) of another test pattern pair (v_n which sensitizes another path p'), we can sensitize p' by just shifting 1 bit of the transition pattern (under LOS test). The proposed method uses this characteristic. This thesis presents a method reduces scan shift time and test data volume by using scan-based test pattern merging. We can also reduce the switching activity induced by the launch pulse. As a result, this also reduces excessive IR-drop in scan testing avoiding test-induced yield loss.

The proposed method reduces scan shift time and test data volume using test

pattern merging. Evaluation results on ISCAS89 benchmark circuits indicate that the proposed method reduces the test application time by 6.89~62.67% and test data volume by 46.39~74.86%.

To improve the small-delay defect coverage of on-chip delay measurement method with small hardware overhead, we also present a method using LOS+LOC based on a conventional method. In addition, we proposed the corresponding test compaction procedure. Compare with the conventional LOS+LOC method, the proposed method reduces the test application time by 47.87~54.02% and test data volume by 71.72~74.50%. Compare with the conventional LOS based method, the proposed procedure can provide similar or higher defect coverage with very small hardware overhead. Specifically, the hardware overhead is 9.27~35.21% smaller than the conventional method. The proposed test compaction procedure reduces the test application time by 4.47~29.29% and test data volume by 4.46~29.96%.

ACKNOWLEDGEMENT

I would like to first thank Professor Masato Kitakami and Professor Kazuteru Namba, my my advisors, for their constant encouragement and guidance during my Ph.D. study. Professor Kitakami and Professor Namba gave me bunches of advice and helped me conquer many difficulties in both my study and my life. I offer my sincere appreciation and gratitude to their patient advice, warm help, funding, and edits of the documents.

To my other committee members, Professor Osawa, Professor Kuroiwa and Professor Hoshino, I am very grateful for their probing questions and validation of the worthiness of my research. Thank you very much for your help and support.

I would like to express my heartfelt gratitude to Professor Hideo Ito, who led me into the world of VLSI testing. I would also thank you to all my friends in the Kitakami & Namba lab as well as in the Graduate School of Advanced Integration Science for the study and happy time together.

I am inspired by and thankful to all my friends in Chiba and all over the world. Although we are in different places pursuing our own dreams, your encouragement and friendship in these years make me realize my goals in study confidently.

This dissertation is dedicated to my parents, Chengxu Zhang and Junmei Chen. Thank you for consistently encouraging me to be brave, independent and optimistic, and I appreciate your support and love in these years.

Finally, but not lastly, I am grateful for my wife, Lirong Shi, for the happy and hard time together during in these years. Thank you very much for your understanding, patience, help and collaboration. I am so happy to grow up with you. Yunxi Zhang, you are the sunshine of my life.

Reference

[1] L.-T. Wang, C.-W. Wu, and X. Wen, VLSI Test Principles and Architectures: Design for Testability, Morgan Kaufmann, 2006.

[2] L.-T. Wang, Y.-W. Chang, and K.-T. Cheng. Electronic Design Automation: Synthesis, Verification, and Test (Systems on Silicon). Morgan Kaufmann, 2009.

[3] I. A. Grout, Integrated Circuit Test Engineering: Modern Techniques, Springer, 2005.

[4] L-T Wang, C. Stroud, and N. Touba, System-on-Chip Test Architectures, Morgan Kaufmann, 2007.

[5] N. Jha and S. Gupta, Testing of Digital Systems. New York: Cambridge Univ. Press, 2003.

[6] A. Krstic and K.T. Cheng, Delay fault testing for VLSI circuits, Kluwer Academic Publishers, 1998.

[7] A.D. Singh and G. Xu, "Output hazard-free transition tests for silicon calibrated scan based delay testing," Proc. IEEE VLSI Test Symp., pp.349–357, 2006.

[8] S.K. Sunter, "BIST vs. ATE: need a different vehicle?," Proc. IEEE Int'l Test Conf., p.1148, 1998.

[9] X. Qian and A.D. Singh, "Distinguishing resistive small delay defects from random parameter variations," Proc. IEEE Asian Test Symp., pp.325–330, 2010.

[10] Semiconductor Research Corporation, Research Challenges in Test and Testability, 2006.

[11] A.D. Singh, "Scan based testing of dual/multi core processors for small delay defects," Proc. IEEE Int'l Test Conf., pp.1–8, 2008.

[12] M. Tehranipoor and N. Ahmed, Nanometer Technology Designs: High-Quality Delay Tests, Springer Publishing Company, 2007.

[13] K. Noguchi, K. Nose, T. Ono, and M. Mizuno, "A small-delay defect detection technique for dependable LSIs," Proc. IEEE Symp. VLSI Circuits, pp.64–65, 2008.

[14] R. Datta, A. Sebastine, A. Raghunathan, and J.A. Abraham, "On-chip delay measurement for silicon debug," Proc. ACM Great Lakes Symp. VLSI, pp.145–148, 2004.

[15] H. Yotsuyanagi, H. Makimoto, and M. Hashizume, "A boundary scan circuit with Time-to-Digital Converter for delay testing," Proc. IEEE Asian Test Symp., pp.539–544, 2011.

[16] M.C. Tsai, C.H. Cheng, and C.M. Yang, "An all-digital highprecision built-in delay time measurement circuit," Proc. IEEE VLSI Test Symp., pp.249–254, 2008.

[17] S. Pei, H. Li, and X. Li, "A low overhead on-chip path delay measurement circuit," Proc. IEEE Asian Test Symp., pp.145–150, 2009.

[18] T. Tanabe, K. Katoh, K. Namba, and H. Ito, "A delay measurement for VLSI circuit by subtraction," IEICE Tran. Inf. & Syst., vol.93, no.4, pp.460–468, April 2010.

[19] K. Katoh, K. Namba, and H. Ito, "A low area on-chip delay measurement system using embedded delay measurement circuit," Proc. IEEE Asian Test Symp., pp.343–348, 2010.

[20] R. Datta, A. Sebastine, and J.A. Abraham, "Delay fault testing and silicon debug using scan chains," Proc. IEEE European Test Symp., pp.46–51, 2004.

[21] R. Datta, G. Carpenter, K. Nowka, and J.A. Abraham, "A scheme for on-chip timing characterization," Proc. IEEE VLSI Test Symp., pp.24–29, 2006.

[22] K. Namba and H. Ito, "Test sets for robust path delay fault testing on two-rail logic circuits," IEEE Trans. Comput., vol.60, no.10, pp.1459–1470, Oct. 2011.

[23] N. Ahmed, M. Tehranipoor, and V. Jayaram, "Timing-based delay test

for screening small delay defects," Proc. IEEE/ACM Design Automation Conf., pp.320–325, 2006.

[24] X. Fan, Y. Hu, and L.T. Wang, "An on-chip test clock control scheme for multi-clock at-speed testing," Proc. IEEE Asian Test Symp., pp.341–348, 2007.

[25] M. Collins and B.M. Al-Hashimi, "On-chip time measurement architecture with femtosecond timing resolution," Proc. IEEE European Test Symp., pp.103–110, 2006.

[26] X. Wang, M. Tehranipoor, and R. Datta, "Path-RO: a novel onchip critical path delay measurement under process variations," Proc. IEEE/ACM Int'l Conf. Computer-Aided Design., pp.640–646, 2008.

[27] A. Jain, A. Veggetti, D. Crippa, and P. Rolandi, "An on-chip flip-flop characterization circuit," Proc. Int'l conf. Integr. Circuit and Sys. Design, pp.41–50, 2010.

[28] M. Collins, B.M. Al-Hashimi, and N. Ross, "A programmable time measurement architecture for embedded memory characterization," Proc. IEEE European Test Symp., pp.128–133, 2005.

[29] S. Ghosh, S. Bhunia, A. Raychowdhury, and K. Roy, "A novel delay fault testing methodology using low-overhead built-in delay sensor," IEEE Trans. Comput.-Aided Des. Integr. Circuits Sys., vol.25, no.12, pp.2934–2943, Dec. 2006.

[30] A. Raychowdhury, S. Ghosh, and K. Roy, "A novel on-chip delay measurement hardware for efficient speed-binning," Proc. IEEE Int'l On-Line Testing Symp., pp.287–292, 2005.

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

[32] J. Savir "Skewed-load transition test: Part I, calculus", Proc. Int. Test Conf., pp.705 -713, 1992.

[33] J. Savir and S. Patil "On broad-side delay test", Proc. VLSI Test Symp., pp.284 -290, 1994.

[34] C. J. Lin and S. M. Reddy. "On Delay Fault Testing in Logic Circuits", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, CAD-6(5):694-703, Sept., 1987.

[35] S. Patil and J. Savir. "Skewed-load transition test: Part II, coverage", Proc. Int. Test Conf., pp.714 -722, 1992.

[36] K.-T. Cheng, "Transition fault testing in sequential circuits", IEEE Trans. Computer-Aided Design, vol. 12, pp.1971-1983, 1993.

[37] Y. Levendel and P. R. Menon, "Transition Faults in Combinational Circuits: Input Transition Test Generation and Fault Simulation", Proc. 16th. Int'l. Fault-Tolerant Computing Symp., pp.278 -283, 1986.

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

[39] Z. Zhang, S.M. Reddy, I.P. Pomeranz, J. Rajski, and B.M. AlHashimi, “Enhancing delay fault coverage through low power segmented scan,” Proc. IEEE European Test Symp., pp.21–28, 2006.

[40] J.S. Yang, B. Nadeau-Dostie, and N.A. Touba, “Test point insertion using functional flip-flops to drive control points,” Proc. IEEE Int’l Test Conf., pp.1–10, IEEE, 2009.

[41] H. Onodera, A. Hirata, T. Kitamura, and K. Tamaru, “P2lib: Process portable library and its generation system,” IPSJ Journal, vol.40, no.4, pp.1660–1669, 1999.

[42] B. Kruseman, A.K. Majhi, G. Gronthoud, and S. Eichenberger, “On hazard-free patterns for fine-delay fault testing,” Proc. IEEE Int’l Test Conf., pp.213–222, 2004.

[43] P. Nigh and A. Gattiker, “Test method evaluation experiments & data,” Proc. IEEE Int’l Test Conf., pp.454–463, 2000.

[44] H. Balachandran, K.M. Butler, and N. Simpson, “Facilitating rapid first silicon debug,” Proc. IEEE Int’l Test Conf., pp.628–637, 2002.

[45] W. Zhang, K. Namba, and H. Ito, "Improving small-delay fault coverage for on-chip delay measurement," Proc. IEEE Int'l Symp. Defect Fault Tolerance VLSI Syst., pp.193–198, 2012.