Studies on Test Application at Field Test and Low Power Logic-BIST

Kyushu Institute of Technology

Doctoral Thesis Mar. 2014

Studies on Test Application at Field Test and Low Power Logic-BIST

フィールドにおけるテスト印加と低電力論理 BIST に関する研究

A thesis submitted in partial fulfillment of the requirements

for the degree of doctor of philosophy

By

Senling Wang

Supervisor: Prof. Seiji Kajihara

Department of Information Systems

Graduate School of Computer Science and Systems Engineering

Abstract

Advances in semiconductor process technology have resulted in various aging issues in field operation of Very Large Scale Integration (VLSI) circuits. For example, HCI (Hot carrier injection), BTI (Bias Temperature Instability), TDDB (Time Dependent Dielectric Breakdown) are well-known aging phenomena, and they can increase the circuit delay resulting in serious reliability problems. In order to avoid system failures caused by aging, recent design usually sets a certain timing margin in operational frequency of the circuit. However, it is difficult to determine the size of the proper timing margin because of the difficulty of prediction of its aging speed in actual use that is related to operational environment. Pessimistic prediction may result in performance sacrificing although it will improve the reliability of the system.

BIST-based field test is a promising way to guarantee the reliability of the circuit through detecting the aging-induced faults during the circuit operation. However, the field test has a limitation on test application time, which makes it difficult to achieve high test quality. Therefore an effective test application method at field is required.

In addition to the requirement of short test application time, the BIST-based field test requires performing at-speed testing in order to detect timing-related defects. However, it is well known that power dissipation during testing is much higher than that in normal circuit operation. Because excessive power dissipation causes higher IR-drop and higher temperature, it results in delay increase during testing, and in turn, causing false at-speed testing and yield loss. While many low power test methods have been proposed to tackle the test power issue, inadequate test power reduction and lower fault coverage still remain as important issues. Moreover, low power testing that just focuses on power reduction is insufficient. When the test power is reduced to a very low level, a timing-related defect may be missed by the test, and a defective circuit will appear to be a good part passing the test. Therefore, appropriate test power control is necessary though it was out of considering in the existing methods.

In this dissertation, we first proposed a new test application to satisfy the limitation of short test application time for BIST-based field test, and then we proposed a new low power BIST scheme that focuses on controlling the test power to a specified value for improving the field test quality.

In chapter 3, a new field test application method named “rotating test” is presented in which a set of generated test patterns to detect aging-induced faults is partitioned into several subsets, and apply each subset in one test session at field. In order to maximize the test quality for rotating test, we proposed test partitioning methods that refer to two items: First one aims at maximizing fault coverage of each subset obtained by partitioning. Second one aims at minimizing the detection time interval of all faults in rotating test to avoid system failures. Experimental results demonstrated the effectiveness of the proposed partitioning methods.

In chapter 4, we proposed a new low power BIST scheme which can control the scan-in power, scan-out power and capture power while keeping test coverage at high level. In this scheme, a new circuit called pseudo low-pass filter (PLPF) is developed for scan-in power control, and a multi-cycle capture test technique is employed to reduce the capture power. In order to control scan-out power dissipated by test responses, we proposed a novel method that selects some flip-flops in scan chains at logic design phase, and fills the selected flip-flops with proper values before starting scan-shift operation so as to reduce the switching activity associated with scan-out. The experimental results for ISCAS-89 and ITC-99 benchmark circuits show that significant scan-in power reduction rate (the original rate of 50% is reduced to 7~8%) and capture power reduction rate (the original rate of 20% is reduced to 6~7%) were derived. With the scan-out controlling method, the scan-out power can be reduced from 17.2% to 8.4%, which could not be achieved by the conventional methods. Moreover, in order to control the test power to the specified rate to accommodate the various test power requirements. A scan-shift power controlling scheme was also discussed. It showed the capability of controlling any scan-shift toggle rate between 6.7% and 50%.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

1. Introduction

1.1. Background

Large Scale Integration (LSI) macro-chip plays a major role in modern human society that has been widely used in many types of electronic systems including computer, cell phone, motor vehicle, airplane, electrical appliances, etc. A very small defect on the chip can easily result in system failure and then lead to the loss of system reliability or even a big damage. Test plays an important role for the assurance of high level system reliability. Typically, test applies a set of test pattern to the input of circuit under test (CUT) and compares the output responses with the expected responses to find the defect-free chip as the responses match. In recent years, with the advances of semiconductor process technology, the transistor scale is shrinking, and more transistors are integrated onto the chip that circuit becomes complex. According to Moore's law, the number of transistors on a chip roughly doubles every couple of years. The most modern microprocessors (Intel's 62-Core Xeon Phi) in 2013 used a 22nm process and contained more than five billion transistors. The reduction of transistor scale and the huge amount of transistors on a chip increase the probability that manufacturing defects on the circuit resulting in faulty-chip. While faulty-chips can be removed by the production test and good chips will be shipped to the customers, when a good chip is put in the actual use for a long time or works at the severe environment, various aging phenomena such as HCI (Hot carrier injection), BTI (Bias Temperature Instability), TDDB (Time Dependent Dielectric Breakdown) might cause the increase of leakage current or open/short faults which can increase the delay resulting serious reliability problems. In order to avoid system failures caused by aging, recent design usually set a certain timing margin in operational frequency. However, it is difficult to determine the size of timing margin due to the difficulty of prediction of its aging speed in actual use that is related to operational environment. Pessimistic prediction may result in performance sacrificing although it will improve the reliability of the system.

BIST-based (Built-In Self Test) field test is a promising way in guaranteeing the reliability of the circuit because it can detect the aging-induced faults during circuit operation. However, since the time allowed for testing during circuit operation usually be very short (e.g.: 10msec), which makes it difficult to achieve high test quality.

In addition to the requirement of short test application time, the BIST-based field test requires performing at-speed testing in order to detect the timing-related defects. However, it is well known that power dissipation during testing is far higher than that in normal circuit operation. Excessive test power dissipation can cause higher IR-drop and higher temperature that result in delay variation during testing, and in turn, causing false at-speed testing and yield loss.

1.2. Objective

As described in the previous section, BIST-based field test is a promising way for guaranteeing the reliability of LSI by detecting the aging-induce faults in the field. However, the limitation of test application time of field testing and the high power dissipation during test are the big challenges for enhancing the test quality of field test. In this dissertation, these two problems will be discussed as the current issues and their solutions will be given.

In order to satisfy the limitation of short test application time for BIST-based field test for aging-induce faults detection, a BIST-based field rotating test will be presented. A set of generated test patterns to detect the aging induced faults is partitioned into several subsets, and apply each subset in one test session at field by rotating. The number of test patterns of each subset will not exceed the pattern limit derived from the upper bound of test application time. As the number of test patterns applied to each test is decreased, it will cause test quality loss. For maximizing the test quality for rotating test, two test partitioning methods that refer to two items will be presented: First one aims at maximizing fault coverage of each subset obtained by partitioning. Second one aims at minimizing the detection time interval of all faults in rotating test to avoid system failures.

For the power reduction of BIST-based field test, while many approaches have been proposed, existing methods only take into account reducing some parts of test power, power reduction is inadequate, and also lower fault coverage still remains as important issue. In addition, existing methods just focus on power reduction, however, if the test power is excessively reduced, some timing-related defects may be missed by the test and a defective circuit will appear to be a good part passing the test. Therefore, controlling the test power to a specified level is important. In this dissertation, we proposed a new low power BIST scheme to tackle these problems. In this scheme, for reducing the scan shift power, scan-in power (refer to the power dissipated by shifting test pattern into the scan chain) and scan-out power (refer to the power dissipated by shifting test responses out the scan chain) are considered individually. A new circuit called pseudo low-pass filter (PLPF) is developed to make the test patterns smoother with low switching activity for scan-in power reduction. A method that selects some flip-flops in scan chains at logic design phase, and overwrites the selected flip-flops' values before starting scan-shift so as to make the test responses smoother with low switching activity is proposed to reduce the scan-out power. For reducing the capture power (refer to the instantaneous power consumed by captured vectors at capture time), a multi-cycle test scheme is employed, where many capture clock cycles are executed in the capture mode during test. Furthermore, using PLPFs with different scan-in power reduction capability can control the scan-shift power to the user specified level.

1.3. Structure of this Dissertation

This dissertation is organized as follows:

Chapter 2 introduces some important concepts in LSI test related to this study.

Chapter 3 introduces BIST-based field rotating test for the aging-induced fault detection.

Chapter 4 introduces a new low power Logic-BIST scheme.

Chapter 5 outlines a summary of the methods proposed in this study.

Chapter 2

2. Preliminary

In this chapter, some important concepts in LSI test related to this study will be described including the principle of LSI test, design for testability (DFT) technique, fault modeling and the concept of fault simulation.

2.1. LSI Test

In the manufacturing process of LSI, process variations such as impurities in wafer material and chemicals, dust particles or in the projection system, etc., can produce physical defects on the chip [1]. Typical defects are broken conductors, missing contacts, bridging between conductors and many other phenomena that can make the chip fail. Since defects produced during manufacturing process are unavoidable, as a result, a fabricated chip may be non-conformance to the specification decided by the designer that is faulty and cannot be shipped to the market. In order to find the defect-free chip, test is required.

Figure 2.1 illustrates the basic principle of LSI testing. A set of binary stimuli (test patterns) is applied to the inputs of circuit under test (CUT), and compares the output responses with the expected responses decided by the designer. The circuit is considered to be good if the responses match.

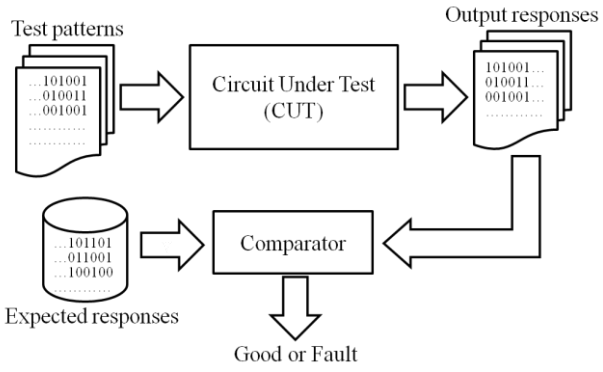


Figure 2.1 Principle of LSI testing [1]

2.2. Design for Testability (DFT)

Conventional test is to set the primary input of a circuit to the desired values and observe the test responses from the primary output. This approach worked well for the small circuit that mainly consisted of combinational logic. However, with the advance of manufacturing technology, more capabilities are integrated onto a chip, and the circuit becomes complicated. A large circuit contains not only combinational logic but also sequential elements such as flip-flops or latches. For the high test quality, sequential elements in the circuit need to be set to the desired values, test pattern generation needs to create test sequences over many clock cycles to justify desired assignments to circuit inputs. As a result, run times and complexity of the test generation increases.

Design for testability (DFT) refers to design techniques that make products easier to test. DFT techniques add some hardware used for test to the CUT to make the internal state of the circuit easier to be controlled and observed from external access. The most popular DFT techniques for LSI test include scan design, Logic Built-In Self-Test. In the following sub-sections, they will be described briefly.

2.2.1. Scan design

Scan design is currently the most common DFT technique [2]. The main idea in scan design is to obtain the controllability and observability for the sequential elements in the circuit. Typically, this is done by adding a test mode to the circuit and all flip-flops functionally form one or more shift registers called scan chains. Scan chains have a unique scan-input per scan segment. Thus, the length of scan chains is equal to the number of scan flip-flops divided by the number of scan inputs specified by the designer. Figure 2.2 shows a scan design schematic. Scan design can operate in three modes: normal mode, shift mode, and capture mode. In the normal model, all test signals are turned off, and the scan design operates in the functional configuration. In both shift mode and capture mode, using a test mode signal (TC), all flip-flops can be set to any desired states by shifting those logic states into the scan chain. Similarly, the captured test responses stored in flip-flops can be observed by shifting the contents of the scan chain out.

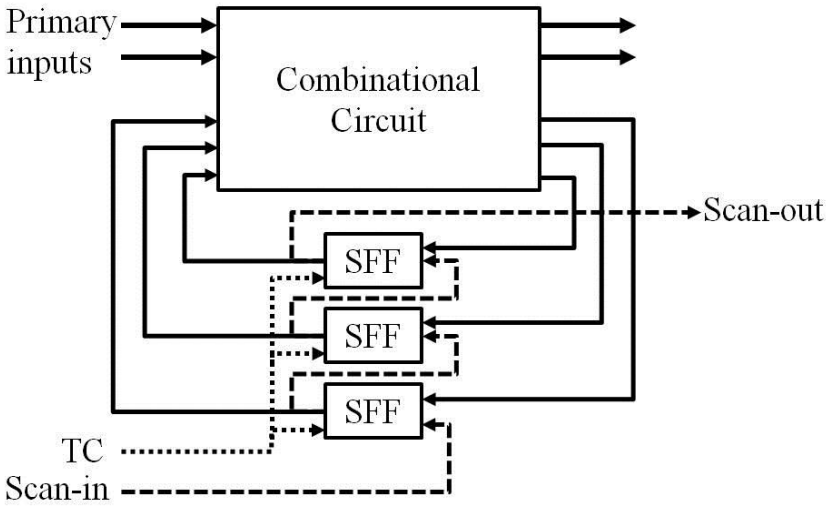


Figure 2.2 A scan-design schematic

Scan design requires converting the selected sequential elements in the circuit into scan cell. There are more than one possible implementations of a scan cell. The most widely used scan cell is Muxed-D as shown in Figure 2.3. Muxed-D scan cell is composed of a D flip-flop and a multiplexer. The multiplexer uses a scan enable (SE) input to select between the data in (DI) and the scan in (SI).

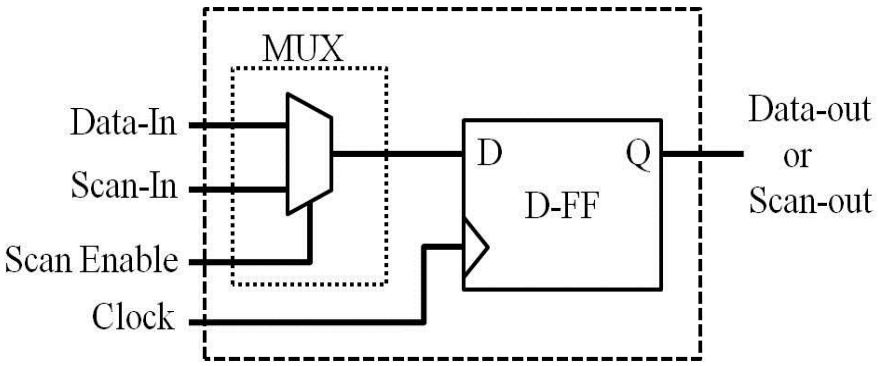


Figure 2.3 Structure of Muxed-D scan cell

2.2.2. Logic Built-In Self Test

Logic-BIST (Built-In Self Test) refers to a DFT technique which currently be widely used for system debug or field test because it can provide self-test ability and easily to conduct at-speed testing for timing-related defects to obtain high test quality [2].

In Logic-BIST, circuits that generate test patterns and analyze the output test responses are embedded on the chip. Figure 2.4 shows a typical Logic-BIST architecture. An on-chip test pattern generator (TPG) automatically generates test patterns for application to the inputs of the circuit under test (CUT). The output test responses are loaded into a signature analyzer (SA) to examine the response to the test patterns. Specific test control signal are generated by the BIST controller for coordinating the Logic-BIST operation among the TPG, CUT, and SA. The signature analyzer has an output to indicate if the circuit has passed or failed the test.

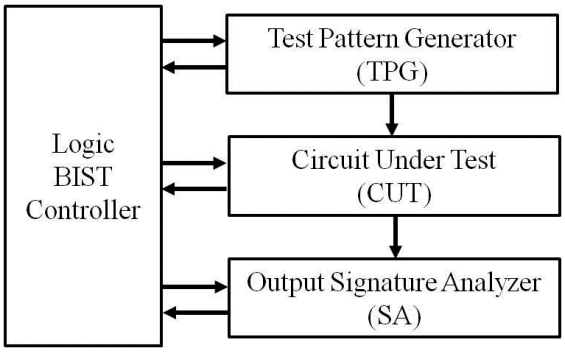


Figure 2.4 Logic-BIST architecture [2]

In most Logic-BIST architectures, linear feedback shift registers (LFSRs) is used as a TPG because it can generate sequence of good random property with little area overhead. Figure 2.5 shows a structure of four-stage LFSR which is composed of elements (latches or flip flops) and exclusive OR (XOR) gates. The signature analyzers (SAs) are commonly constructed from multiple-input signature registers (MISRs) as shown in Figure 2.6. The MISR is basically an LFSR that has an extra XOR gate at the input of the flip-flop for compressing the output responses of the CUT into the LFSR during shift operation.

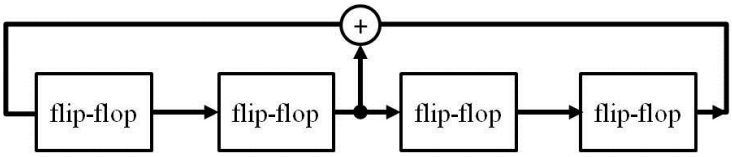


Figure 2.5 Four-stage LFSR

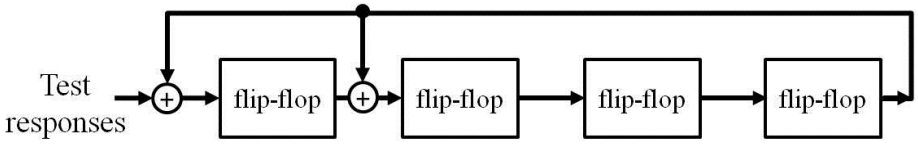


Figure 2.6 Four-stage MISR

Logic-BIST can easily apply a large number of test patterns with random property. As a result, more defects, either modeled or un-modeled, can be detected. In addition, Logic-BIST makes it easy to conduct at-speed testing for timing-related defects.

2.3. Fault Models

Various defect would be produced in the manufacturing process, it is difficult to generate tests for the real defects. For an accuracy test, it requires accurate description of the circuit behavior containing the physical defects, then, fault modeling is necessary.

There are many fault models related to various defect. In the following sub-sections the currently most popular fault model, the stuck-at fault model and the transition fault model will be introduced.

2.3.1. Stuck-at fault model

A stuck-at fault occurs when a signal line in the circuit is stuck at a constant logic value, either a logic 0 or logic 1, as shown in Figure 2.7. Each signal line can have two types of stuck-at faults: stuck-at-0 and stuck-at-1. For a circuit with n signal line, the total number of stuck-at faults probably exist in the circuit is 2n. This number can be further reduced by fault collapsing technique [1].

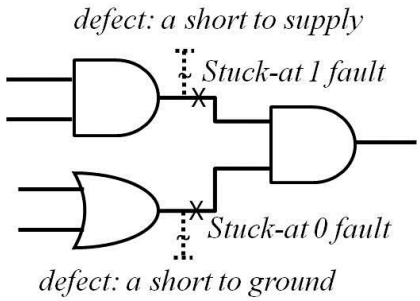


Figure 2.7 Stuck-at-0 fault and stuck-at-1 fault

Research has shown that stuck-at fault model can cover many other physical defects within a circuit. However, with the transistor scale continuously shrinking and the operation frequency increasing, other types of defects are beginning to appear, which cannot be covered by stuck-at fault model, such as the transition faults which will be introduced next.

2.3.2. Transition fault model

During the operation of a circuit, logic function and internal signal transition is performed within a specified time limit. For a correct operation the delay of signal transition should not exceed the time limit. In the manufacturing process, certain types of defects may cause the circuit gates to have a higher than normal delay. This un-expected delay can make the signal transition at the gate slower than normal speed. A transition fault will occur when the output of a gate switching from 0(1) to 1(0) takes longer time than normal [3]. If the delay is large enough so as to exceed the specified time limit, its effects can be propagated to the circuit outputs and the circuit will operate with a faulty behavior.

Transition fault model assumes that the delay fault affects only one gate in the circuit. There are two transition faults associated with each gate: a slow-to-rise fault and a slow-to-fall fault [3] as shown in Figure 2.8. A slow-to-rise (slow-to-fall) fault means that the transition from 0 to 1 (1 to 1) will not reach any output within the stipulated time.

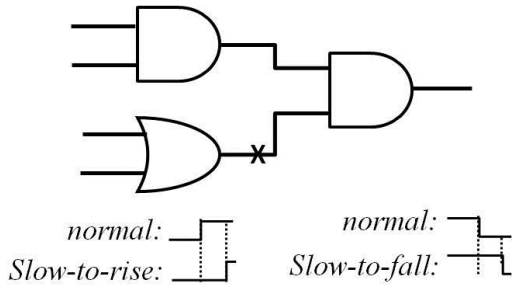


Figure 2.8 Slow-to-rise fault and slow-to-fall fault

To detect a transition fault in a circuit, it is necessary to apply two input vectors (V1, V2). The first vector V1 initializes the transition fault in the circuit. The second vector V2 launches the transition and propagates its effect toward the output or some observation points and captures the effects there at a specified time determined by the system clock speed. If the captured response indicates that the logic involved did not transition as expected during the clock cycle time, it is considered to contain a defect. There are two types of transition test method depending on how the transition is launched and captured: launch-off-shift (skewed load) [4] and launch-off-capture (broadside) [5].

2.3.2.1. Launch-off-Shift (LoS)

In the launch-off-shift (LoS) method [4], the transition is launched in the last shift of the scan chain load. Figure 2.9 shows the waveform of launch-off-shift. The critical timing is the time from that last shift (or launch) clock to the capture clock. The launch clock is a part of the shift operation and is immediately followed by a fast capture clock. The scan enable signal (SE) is high during the last shift and must go low very quickly to enable response capture at the capture clock edge. Since the capture clock is applied at system frequencies after the launch clock, the scan enable signal which typically drives the scan chains should also switch at system frequencies. This is a limitation because most scan shifting is done at lower frequencies. If the chains are shifted and tested at-speed (at system frequencies), a sophisticated buffer tree or strong clock buffer is required which results in high test cost.

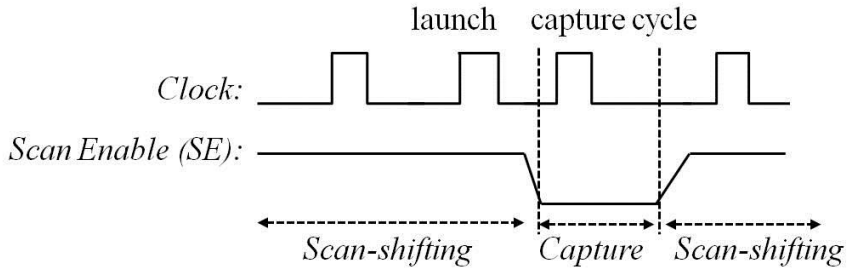


Figure 2.9 Waveform of launch-off-shift

2.3.2.2. Launch-off-Capture (LoC)

In the launch-off-capture (LoC) method [5], launch clock is separated from the shift operation, a pair of at-speed clocks is applied to launch and capture the transition in functional mode. Figure 2.10 shows launch-off-capture (LOC) method waveforms. This approach relaxes the speed constraint on the scan enable (SE) signal that test pattern scan shifting can be done at slow speeds, and also, the captured test responses can be shifted out slowly in test mode.

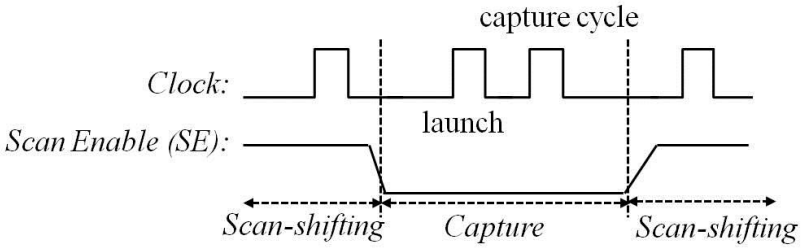


Figure 2.10 Waveform of launch-off-capture

2.4. Fault Simulation

Fault Simulation [1] is used to verify the quality of the test pattern generated for a given fault model. A typical application of fault simulation is shown in Figure 2.11. Fault simulation is executed after the design verification where the verified circuit net-list and verification stimuli (test patterns) are available. A fault list can be generated by the fault simulator for the specified fault model. As each fault is inserted, the fault simulator runs test patterns. If the fault simulation shows that the responses of the faulty circuit are different with the expected responses, fault is detected and will be deleted from the fault list.

At the end of fault simulation, fault coverage is computed to evaluate the quality of the supplied test patterns. Fault coverage is defined as the ratio of the number of detected faults to that of faults in the fault list as shown following.

$$Fault\ Coverage = \frac{Number\ of\ detected\ faults}{Total\ number\ of\ faults}$$

With the help of other program (e.g.: test generator), fault simulation also can be used for test pattern generation if required. As shown in Figure 2.11, if expected fault coverage cannot be achieved through the original test patterns, the undetected faults in the fault list will be applied to a test generator to produce new test vectors.

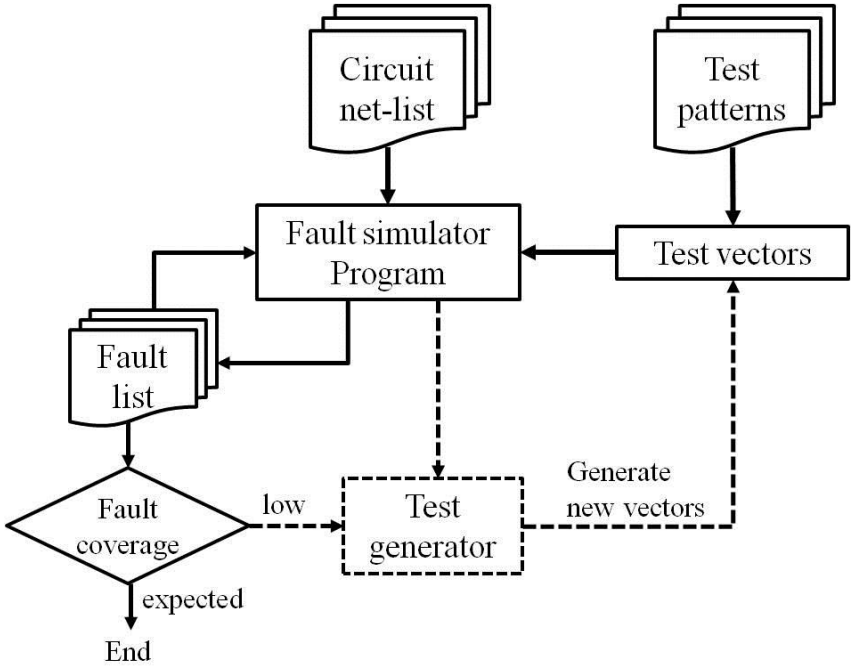


Figure 2.11 Fault simulation scheme [1]

Because the number of faults simulated for fault detection analysis is very large, fault simulation will spend much greater time than design verification. Many improved approaches have been developed, such as parallel fault simulation [6], deductive fault simulation [7] and concurrent fault simulation [8].

Chapter 3

3. BIST-based Field Rotating Test for Aging-induced fault detection

The continuous scaling down of transistor causes aging phenomena such as HCI, NBTI, TDDB, etc. [9-10]. These aging effects might cause the increase of leakage current or open/short faults which can increase the delay resulting serious reliability problems. This chapter presents a BIST-based rotating test [27] to detect the aging-induced faults for the assurance of the reliability of LSI. The proposed method partitions a set of pre-generated test patterns used to detect the aging induced faults into several subsets, and apply each test subset in one field test chance by rotation. As the number of test patterns applied to each test is decreased, test quality will be lost. In order to improve the test quality, two effective test pattern partition methods are also proposed.

The rest of this chapter is organized as following: Section 3.1 introduces the reliability challenge caused by aging degradation. Section 3.2 shows the field test for aging-induced fault. Section 3.3 describes the concept of the proposed rotating test. In Section 3.4, two pattern partition methods for rotating test and their experimental results will be discussed. Finally, concluding remarks are given in Section 3.5.

3.1. Reliability challenges: Aging

Reliable LSI plays an important role in electronic application systems which require high field reliability, such as medical devices, national defense, communications, and aircraft and automobile safety. Reliability of LSI depends on the failure rate during its life [12].

In the LSI life cycle, the failure rate varies with time. It can be modeled by a “bathtub” curve which is widely used in reliability engineering as shown in Figure 3.1. The bathtub curve can be divided into three stages: the ”infant mortality” stage with decreasing failure rate, the “normal operation” stage with random failure rate which is near constant, and the “wear-out” stage with increasing failure rate. During LSI’s early life the failure rate is very

high due to the manufacturing defects. In practice, the defective LSI or that with a high potential for failure are eliminated in this stage by manufacturing test involves burn-in test or voltage stress test, and etc. When a LSI is shipped to the market, the failure rate is relatively small and constant. Failures are caused by mechanisms inherent in the circuit. In the wear-out stage precedes the end of the LSI life, the probability of failure increases with time.

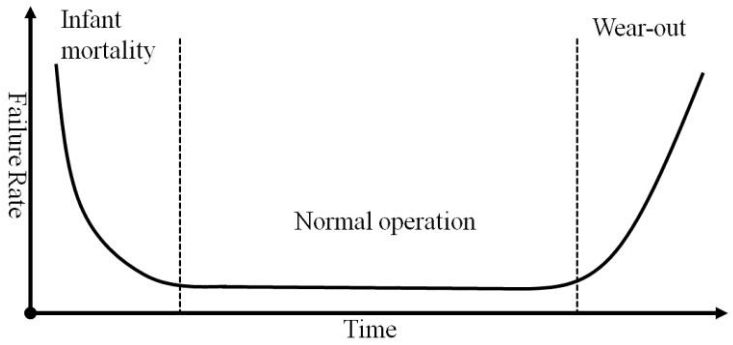


Figure 3.1 Bathtub curve for LSI's reliability [12]

With the advances in miniaturization of LSI, the feature size of transistor becomes smaller and smaller. The small feature size of the transistors causes aging phenomenon such as Hot-Carrier Injection (HCI), Negative Bias Temperature Instability (NBTI) and Time Dependent Dielectric Breakdown (TDDB) [9, 10]. These aging effects might cause the increase of leakage current or open/short faults which can increase the delay resulting serious reliability problems when the circuit works for a long time or be put in the severe environment, and it should be strictly avoided in high reliability electronic applications.

To guarantee the reliability of LSI, manufacturing test such as burn-in [13] test or stress test [13] are used through applying high temperature or over-voltage supply, aiming to accelerate the failure mechanisms and shorten the time to failure process. The chip with a high potential for failure would likely fail in the infant stage under high frequency or high temperature test. However, due to the small feature sizes and the low operation voltage, excessive burn-in test or stress test would deteriorate the good chips and result in their short lifetime [15].

Recent designs usually set a certain margin (timing) in operational frequency to avoid system failure caused by aging-induced faults [11]. However, timing margin is difficult to be determined by predicting its aging speed in actual use. If a circuit is used more frequently or in higher temperature, then an aging occurs earlier. In addition, the environment where the circuit operates often relates to its aging speed. For example, NBTI-induced delay degradation is significantly accelerated in high temperature. Therefore, the timing margin, which is derived from the worst case estimation, may results in performance degradation.

3.2. Field test for aging-induced fault

Field test is a promising approach in guaranteeing the reliability of the circuit through detecting the aging-induced faults [16]. It can be classified into two categories: concurrent test and non-concurrent test.

Concurrent test continuously detects the fault while the circuit is in normal operation mode, thus both the permanent defects (e.g. aging induced defects) and transient defects (e.g. soft errors) can be detected. Self-checking design [17], on-line monitoring [18], signature monitoring techniques [19] and concurrent self-test [20-22] fall into this category. On the other hand, concurrent testing requires large overhead or performance degradation due to special circuit architecture or inserted redundancy in terms of hardware, time, or information. For example, in the self-checking design [17], a complex circuit is partitioned into its constituent functional blocks and adding a checker circuit to check the output of each block, then can detect the concurrent error. The additional checker increases the hardware overhead and also results in performance degradation. Therefore, Concurrent test might not be accepted for general designs widely.

Recently non-concurrent test began to be used for some systems that need high reliability such as automotive, communication, medical, etc. [23-25]. Non-concurrent test is executed at test mode while the system is in idle state, suspended or power-on/off. Therefore, it has less impact on system performance. In addition, usually a built-in self-test (BIST) architecture is adopted to apply test patterns to the circuit under test (CUT) and

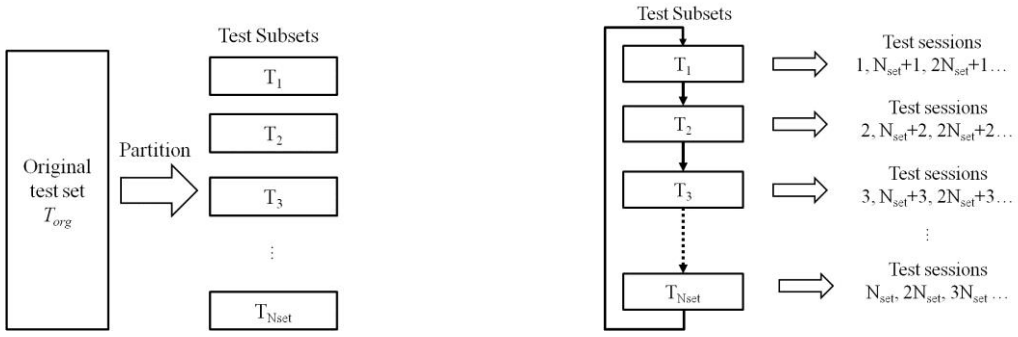
analyze the test responses. The impact on area overhead is relatively small. In [26], architecture for field test was proposed which has a BIST-based architecture. Even for non-stop systems, which do not often restart, the systems can run at test mode periodically in field. Thus field test like power-on test would be a promising method to detect aging-induced faults.

The major differences between the field test and the manufacturing test exist in the limitation of test application time and the test opportunity. Due to the requirement of the systems, test application time of the field test is very short, e.g. 10 msec. Therefore, if the number of test patterns is large, it may be impossible to apply all the patterns to the circuit within the required test time. On the other hand, since the BIST-based manufacturing test does not have any technological requirements on test time, its limitation for test application time is less than the field test apart from an economical reason.

Regarding the test opportunity, the power-on test has a peculiar feature. Usually the opportunity of the production test is once just after manufacturing the chip. But since the power-on test is executed every time the system is starting up, its opportunities are more than once. This can be an advantage of the field test as described in the following.

3.3. Rotating Test

In this work, a test partition with rotating test [27] is proposed to satisfy the constraint on application time for field test. Figure 3.2 shows the concept.



a. partitioning a given test set b. applying each subset by rotating

Figure 3.2 Test partitioning and rotating test

If a given test set is too large to apply in one field test (test session), it is needed to reduce the number of test patterns that will cause test quality loss because of the missing test patterns. However, the test patterns that were not applied at the test session can be applied at the next or later test session, because the field test is executed repeatedly every time the system runs at test mode. Therefore, we partition the original test set into some test subsets as illustrated in Figure 3.2(a), and applying one subset for the circuit at one test session. The original test set is partitioned so that the number of test patterns of each subset never exceeds the pattern limit derived from the upper bound of test time, and the number of subsets is as small as possible. Therefore, the number of test patterns of each subset should be N_{org}/N_{set} or N_{org}/N_{set}, where N_{org} and N_{set} are the number of test patterns of the original test set and the number of the subsets obtained by partitioning, respectively.

All the test patterns of the original test set can be applied through N_{set} opportunities of the tests. Each subset of test patterns is rotated and applied again at future test sessions, as shown in Figure 3.3(b). We call it "rotating test". In the example of Figure 3.2(b), sub test set T_1 is applied not only at the first test session but also at the $(i \times N_{set}+1)_{th}$ test session, where $i = 1, 2, 3, \ldots$

3.4. Pattern Partition algorithms for rotating Test

In rotating test, since the original test set needs to be partitioned into several smaller subsets to meet the test application time requirement. Once an aging-induced fault occurs during the circuit operation it might not be detected immediately until the test is applied that would cause a system failure. To guarantee the reliability, it is necessary to improve the fault coverage for each test subset as far as possible.

In addition, because of the missing test patterns of each subset a fault may not be detected at the followed test session right after it occurs. The fault effect would be propagated during a time interval from its occurrence to the detection. Although a system failure is not caused necessarily as soon as a fault excites, the longer detection time interval would cause the higher probability of a failure appears. Therefore, shorten the detection time interval for each fault is also necessary.

In this work, we proposed two pattern partition algorithms to tackle these problems.

3.4.1. Pattern partition for fault coverage improvement

In this section, the proposed pattern partition algorithm focuses on improving the fault coverage for each test subset as far as possible.

3.4.1.1. Quality of test partition for rotating test

Assume that all the subsets obtained by partitioning have the same size as N_{sub}. Then there are $(N_{org}-1)!/(N_{sub}!)N_{set}$ combinations of partition. Even though the original test set is the same, test quality of the rotating test could be generally different depending on partitions. An example is shown below.

Suppose that we partition a test set $T_{org} = \{t_1, t_2, t_3, t_4, t_5, t_6\}$ which are generated for nine faults f_1 to f_9 into three subset, *i.e.:* $N_{org} = 6$, $N_{set} = 3$. Hence every subset size is 2 calculated from N_{org}/N_{set}. Each test pattern detects faults as shown in Table 3.1; for example, t_1 detects three faults f_1, f_2 and f_5. We consider a test partition P_1 as shown in Table 3.2(a). The subsets T_1, T_2 and T_3 consist of $\{t_1, t_2\}$, $\{t_3, t_4\}$ and $\{t_5, t_6\}$. Individual fault coverage of subsets is 44% (= 4/9) for T_1 and T_2, and 33% for T_3, hence the average fault coverage of P_1 is 40.3%. Note that the sum of individual fault coverage for all subsets is more than 100% because faults f_5 and f_8 are detected in two subsets.

Next we consider an alternative test partition P_2 for T_{org} as shown in Table 3.2(b). In this case, individual fault coverage of subsets are 55%, 55%, 44% for T_1, T_2 and T_3, respectively, and the average fault coverage of partition P_2 is 51.3%, which is larger than that of P_1. It means that P_2 has higher test quality than P_1 because the field test aims at detecting aging-induced faults unlike the manufacturing test. Even if a fault has not occurred yet in a test session, the fault may occur before the next test session. Although a system failure is not caused necessarily as soon as a fault excites, high test quality would be derived if each fault can be detected by more test sessions frequently. Therefore, it is important for the rotating test to find a test partition so that the average fault coverage of individual subsets is as high as possible.

Table 3.1 Test pattern and detected faults

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9
t_1	o	o			o				
t_2		o	o						
t_3				o	o			o	
t_4					o	o		o	
t_5							o	o	
t_6								o	o

Table 3.2 Examples of test partitioning

(a): test partition P_1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	Flt.cov.
$T_1(t_1,t_2)$	o	o	o		o					44%
$T_2(t_3,t_4)$				o	o	o		o		44%
$T_3(t_5,t_6)$							o	o	o	33%

(b): test partition P_2

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	Flt.cov.
$T_1(t_1,t_5)$	o	o			o		o	o		55%
$T_2(t_2,t_4)$		o	o		o	o		o		55%
$T_3(t_3,t_6)$				o	o			o	o	44%

3.4.1.2. Problem formulation

From the above observations on test quality in test partitioning, we formulate the following problem:

[Test Pattern Partitioning Problem] Given a test set T_{org} consisting of N_{org} patterns and the number of sub test sets N_{set}, find a partition such that

(1) the number of test patterns of each sub test set is N_{org}/N_{set} or N_{org}/N_{set}, and

(2) the average fault coverage of individual sub test sets is maximized.

It is easy to satisfy condition (1) because the number of patterns of each subset is uniquely calculated from N_{org} and N_{set}. Therefore our discussion focuses on condition (2) below.

3.4.1.3. Partition Algorithm

The algorithm consists of two phases: First is test pattern partitioning, we are going to partitioning a given test set T into N_{set} sub test sets. Second is test pattern replacement, improving the pattern partitioning by exchanging test patterns between different sub test sets after pattern partitioning.

Phase 1: Test pattern partitioning

For pattern partitioning, we first define a terminology. Given test set T, for a couple of test patterns t_i and t_j, if they can detect more the same faults, t_i and t_j are more similar, the number of these faults is defined as the similarity between t_i and t_j. For example of test patterns and faults in Table 3.1, f_2 is a detected fault of t_1, it also be detected by t_2. The similarity between t_1 and t_2 is 1. The same as t_1 and t_2, the similarity between t_2 and t_3 is 0.

In order to maximize average fault coverage of subsets, we need to partition the test set into subsets so that the test patterns in the same subset do not detect the same faults and the different subsets can detect more the same faults as far as possible. The definition of similarity shows that: the smaller similarity of a couple of test patterns, the more different faults can be detected. During the partition, we need to comply with a rule: while distributing a pattern to a subset, the similarity between the new pattern and the patterns which already exist in the subset must be the smallest, in the meantime, the similarity between subsets must be the biggest. If we create a complete table as Table 3.1 shown with respect to test patterns and detected faults, we would have enough information to calculate the similarity. However, it is not efficient on both time and memory usage because it requires fault simulation without fault dropping and a table whose size is $O(N_{org} \times N_{flt})$ where N_{flt} is the number of faults of the circuit.

In order to calculate the similarity of test patterns, we employ fault simulation with fault dropping after a fault is detected N-times where N can be set arbitrarily. We consider that for N_{set} pattern partition, N_{set}-times fault dropping simulation can get enough information. Below is the outline of the test partitioning algorithm for given test set T and the number of subsets N_{set}:

Step 1: For *T*, perform fault simulation with fault dropping after *N*-times detection in an arbitrary order of test patterns, and for each fault record the ID of the first *N* patterns which detect the fault.

Step 2: For every pair of test patterns t_i and t_j ($i \neq j$), count the number of faults which are detected by the two test patterns simultaneously, and then create a two-dimensional table, as table 3.3.

Table 3.3 Similarity of test vector pairs

	t_1	t_2	t_3	t_4	t_5	t_6
t_1	-	1	1	1	0	0
t_2	1	-	0	0	0	0
t_3	1	0	-	2	1	1
t_4	1	0	2	-	1	1
t_5	0	0	1	1	-	1
t_6	0	0	1	1	1	-

Step 3: According with the two-dimensional table of similarity, distribute a test pattern into a subset by the following two criteria:

1): The similarity between subsets is large.

For N_{set} pattern partitions, find out N_{set} test patterns which with the largest similarity between them from the two-dimensional table of similarity, and distribute each of them to different subsets. If M ($M > N_{set}$) test patterns exist, calculate the sum of similarity for t_i ($i \in N_{set}$) and each $t_j \in T$ ($j \neq i$), respectively. Then, find out N_{set} test patterns which with the smaller sum of similarity value, and distribute one of them to a subset.

2): The similarity between test patterns in each subset is small.

While distributing a test pattern to a subset, from the two-dimensional table of similarity, seek one pattern which with the smallest similarity between this pattern and the patterns which already exist in the subset. If more than one pattern is qualified, calculate the sum of similarity for the pending pattern and the patterns already exist in the other subsets. Distribute the pattern which with the biggest sum value of similarity to the subset.

Step 4: According to the two criteria shown in Step 3, perform the partition procedure until all given test patterns are distributed.

In the fault simulation at Step 1, a fault is dropped from the target fault list when the N-times detection pattern for the fault was found. By fault dropping simulation, a detected fault list for every test pattern can be created. We give an example for test patterns in Table 3.1. By comparing the fault list of each pair of test patterns, we create a two-dimensional table for example as table 3.3 to record the similarity of every couple of patterns at Step 2. At Step 3, pattern partition must meet the two formulas simultaneously.

Phase 2: Test pattern replacement

After test pattern partitioning, we consider that for a test set T with large number of test patterns, during test pattern partitioning, since fault dropping simulation dropped too much faults information, test pattern partitioning may be not accurate, capability of improvement in fault coverage still exists. Therefore, we perform test pattern replacement for partitioned sub test sets. First, we define a terminology. Given test set T, if fault f is detected by t_i in T, but not detected by any test vector in T-{t_i}, f is called *an essential fault* of t_i [28]. As the example shown in Table 3.1, f_4 is an essential fault of t_3.

For a sub test set T_{sub}, some test patterns can detect more essential faults, we consider that these patterns are important for T_{sub}, removing one of them affects the fault coverage severely. Some test patterns can't detect anyone essential fault, removing some of them has less impact on fault coverage. We give an example for fault list of a sub test set T_1 in table 3.4. Sub test set T_1 detects 7 faults of all the 9 faults, fault coverage is 77%. Fault f_1 and f_8 are essential faults for T_1, detected by t_1 and t_4 respectively. Test pattern t_3 detects {f_3, f_5}, t_5 detects {f_4, f_6}. If remove t_3 and t_5 from T_1, no change in the fault coverage of T_1. From the *number of essential faults* (*#ess*) and *unessential faults* (*#uness*) detected by each test pattern, we can calculate *the number of rank* (*#rank*) by

$$\#rank = \#ess * 2 + \#uness / 2 \quad (1)$$

(Value of #uness/2 is rounded)

Table 3.4 Fault list of a sub test set T_1

	f_1	f_2	f_3	f_4	f_5	f_6	f_7	f_8	f_9	*Flt.cov(%)*	*#ess*	*#uness*	*#rank*
T_1	o	o	o	o	o	o		o		77	-	-	-
t_1	o	o	o		o					44	1	3	4
t_2		o	o	o						33	0	3	2
t_3			o		o					22	0	2	1
t_4					o	o		o		33	1	2	3
t_5				o		o				22	0	2	1

For test pattern replacement, we focus on test patterns with lower rank. We consider that a test pattern with low rank in a subset may detect the faults which are not detected by other subsets. For example as Figure 3.3 shows, given 2 subsets T_1 and T_2, f_2 and f_1 are undetected faults, respectively. For T_1, test pattern t_2 has the lowest rank and detects fault f_1, for T_2, t_8 detects f_2 with the lowest rank, t_2 and t_8 are not useful for T_1 and T_2, but if exchange them, f_2 and f_1 become detectable for T_1 and T_2.

Therefore, we consider that test pattern replacement can improve the test quality for partitioning method proposed in phase 1. The outline of test pattern replacement is shown as below:

Step1: For every test pattern of subset T_i ($i \in N_{set}$), count the number of essential faults and unessential faults (all fault information is obtained during fault simulation in phase 1), record the undetected faults by T_i in a undetected fault list.

Step2: Calculate the rank of every test pattern by expression (1), sort test patterns for each subset by rank from low to high.

Step3: Set a threshold of rank for subset T_i, compare the undetected fault list of T_i with the fault list of every test pattern t_a (*#rank* of t_a < threshold) in T_j ($j \in N_{set}$, $j \neq i$), if t_a detects more faults which were not detected by T_i, exchange t_a and test pattern which with the lowest rank in T_i.

Step4: Update the undetected fault list for T_i and sort the test patterns for each subset by rank from low to high again, threshold minus 1. Return to step 3 until threshold=0.

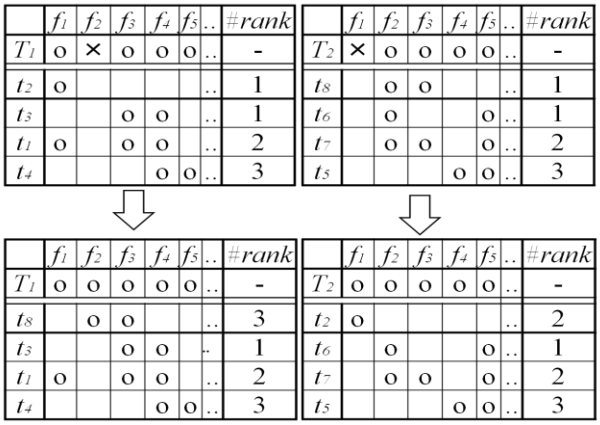


Figure 3.3 Example for test pattern replacement

3.4.1.4. Experimental results

To demonstrate the effectiveness of the proposed test partition algorithm, we implemented it using C language and performed experiments for *ISCAS'89* and *ITC'99* circuits on a PC with *Core(TM)2 Duo 2.66GHz, 1.99GB RAM*. In these experiments, we used test patterns generated for single stuck-at faults by an in-house *ATPG* program.

In the first experiment, we performed *Phase 1* of proposed test partition algorithm, partitioned the given compacted test set into eight subsets. Results are given in Figure 3.4 in which we draw the curve of average coverage for some circuits to compare our method with random partitioning. Y axis shows the average fault coverage, X axis shows the different partitions. Different curves show the result for different circuits. The origin of each curve shows the average fault coverage of subsets partitioned randomly, the other points show the average fault coverage of subsets partitioned by proposed method. We set the detection time of dropping simulation from 2 to 10. From the curves, we can see that for proposed partition method, with the detection time for dropping simulation increase, the average coverage increased. After 4 times dropping simulation the curves become gently, while the detection time for dropping simulation is set to 10, for most circuits, the proposed partition method found the best partition.

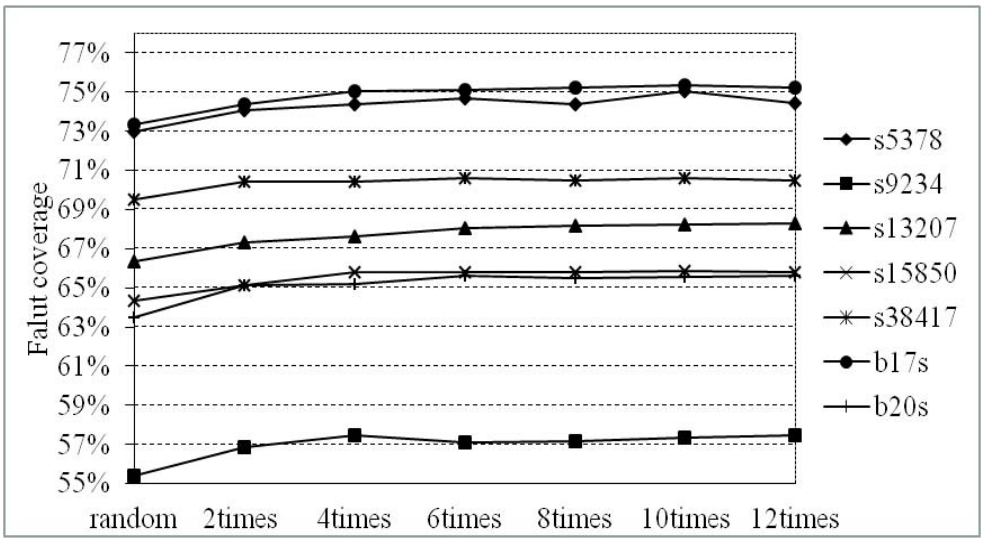


Figure 3.4 Experimental result for Nset=8

Table 3.5 gives the result of proposed method using 10 times fault dropping simulation. The first two columns show circuit names and the numbers of the given test patterns. The third and forth columns show the average fault coverage of eight subsets for our partitioning and random partitioning, respectively, and the fifth column gives the difference between two methods. These results show that it is meaningful for high quality power-on test to find better partition for the rotating test. The last column of the table gives the computing time in second. Since the algorithm is based on fault dropping simulation, test size, circuit size and detection time for dropping simulation dominate the computing time. Table 3.6 gives the application time of proposed partition while setting the detection time of dropping simulation from 2 to 10. The first column shows the circuit names. From the second column we show the application time while setting the detection time of fault dropping simulation from 2 to 10 respectively. We can see that comparing with 2-times fault dropping simulation, proposed partition by 10-times dropping simulation cost almost 1.5 times application time. Because more accurate similarity table requires more faults information, and more computing time is consumed.

Table 3.5 Experimental result for random partition and the proposed partition with N_{set} = 8

Circuit	#pattern	Random (%)	Proposed (%)	Diff(%)	Time(s)
s5378	100	72.983	75.057	2.074	1.592
s9234	111	55.427	57.361	1.933	2.177
s13207	235	66.333	68.216	1.883	6.526
s15850	97	64.314	65.867	1.553	5.847
s38417	87	69.534	70.586	1.052	37.57
b17s	1250	73.336	75.334	1.999	359.459
b20s	989	63.514	65.546	2.032	65.367
Average	-	-	-	1.789	-

Table 3.6 Run time

Circuit	Runtime (sec)					
	2times	4times	6times	8times	10times	12times
s5378	0.75	0.92	1.08	1.33	1.59	1.81
s9234	1.18	1.37	1.61	1.92	2.18	2.39
s13207	3.9	4.39	4.91	5.74	6.53	7.41
s15850	2.32	2.89	3.74	4.73	5.85	7.13
s38417	8.7	13.86	20.37	28.53	37.57	50.2
b17s	216.64	235.83	266.09	307.63	359.46	411.81
b20s	56.53	55.42	57.6	63.73	65.37	70.35

In the second experiment, we performed test pattern replacement for subsets which partitioned by the proposed test pattern partitioning method using 10 times fault dropping simulation, results are given in Table 3.7. The third and forth column shows the average fault coverage of eight subsets for the proposed test pattern partitioning method and performed test pattern replacement, respectively, and the fifth column gives the difference between two methods. These results show that test pattern replacement improved the average fault coverage of subsets after the test pattern partitioning.

Table 3.7 Experimental result for the proposed partition algorithm with N_{set} = 8

Circuit	#pattern	Test pattern partitioning algorithm		
		Phase 1: Pattern partitioning (%)	Phase 2: Pattern replacement (%)	Difference (%)
s5378	100	75.057	75.576	0.519
s9234	111	57.361	58.128	0.767
s13207	235	68.216	69.25	1.034
s15850	97	65.867	66.614	0.747
s38417	87	70.586	71.013	0.427
b17s	1250	75.334	76.11	0.776
b20s	989	65.546	66.297	0.751
Average	-	-	-	0.717

3.4.2. Pattern partition for reducing the probability of failure appearance

For rotating test, since the original test set needs to be partitioned into several smaller subsets to meet the test application time requirement. Fault coverage of each subset would be reduced due to the missing test patterns so that many faults cannot be detected in test sessions and the detection time interval of faults becomes longer. Although a system failure is not caused necessarily as soon as a fault excites, the longer detection time interval would cause the higher probability of a failure appears. Therefore, reduce the probability of failure caused by faults is required for guaranteeing the system reliability.

3.4.2.1. Failure Appearance Probability (FAP)

We define *Failure Appearance Probability (FAP)* as the probability of a failure appears during system operation. The computation for *FAP* is based on two assumptions: 1) no fault occurs during test application; 2) failure caused by a fault can be avoided when the fault is detected.

To compute the *FAP*, we assume that executing field test per unit time. Suppose that existence probability of an aging-induced fault during unit time is p and the probability of failure caused by the fault in a time unit is *q*. Hence, the *FAP* of a fault per unit time is *pq*. If the fault also cannot be detected in the next test session, the detection time interval becomes two units of time. The existence probability of the fault is *2p*, and failure would be

caused by the faults more easily since the detection time interval becomes longer. Therefore, *FAP* of the fault undetected at two test session is 2^2pq. We give an example to explain the *FAP* computation for rotating test below.

Suppose that a given test set can be partitioned into 4 subsets $\{T_1, T_2, T_3, T_4\}$ and rotate them in field. Consider a fault *f* detected by T_1, if *f* can be detected by T_2 but undetected by T_3 and T_4. During the period between T_1 and T_2, the time interval is 1 so that the failure rate of *f* is *pq*. After T_2 is applied, *f* will be undetected in three consecutive test sessions until the next test T_1 is applied, so that the detection time interval is 3 and the failure rate of fault *f* during the period between test T_2 and the next T_1 is 3^2pq. If we assume that $pq=1$, the *FAP* of fault *f* during a rotating test is the sum of the *FAP* in different test sessions divided by the number of subsets that is (1+9)/4=2.5. For all faults detected by T_1, *FAP* can be formulated as:

$$FAP(T_1) = pq*[(\Delta f_{1,1} - \Delta f_{1,2}) + 4(\Delta f_{1,2} - \Delta f_{1,3}) + 9(\Delta f_{1,3} - \Delta f_{1,4}) + 16\Delta f_{1,4}]$$

$$= pq(\Delta f_{1,1} + 3\Delta f_{1,2} + 5\Delta f_{1,3} + 7\Delta f_{1,4}) \quad (1)$$

Where, $\Delta f_{1,1}$ denotes the fraction of faults detected by T_1, $\Delta f_{1,2}$ denotes the fraction of faults detected by T_1 missed by T_2, $\Delta f_{1,3}$ denotes the fraction of faults detected by T_1 missed by T_2 and T_3, and $\Delta f_{1,4}$ denotes the fraction of faults detected by T_1 missed by T_2, T_3 and T_4 (only detected by T_1). Note that (1) only gives the failure rate for the case that faults detected by T_1. The general case for computing the failure rate for the faults detected by T_i can also use the same equation (1) where $\Delta f_{i,j}$ becomes the fraction of faults detected by T_i and undetected by other subsets $T_{i+1} \ldots T_{i-1}$ during a rotating test. For the case that a given test set is partitioned into 4 subsets, the failure rate for rotating test can be expressed as:

$$FAP(P:4 \mid \{T_1, T_2, T_3, T_4\}) = \frac{pq}{4}[FAP(T_1) + FAP(T_2) + FAP(T_3) + FAP(T_4)] \quad (2)$$

Where $FAP(T_i)$ is the *FAP* of the faults detected by T_i. We can write the FAP for the rotating test with 4 subsets as:

$$FAP(P:4|\{T_1,T_2,T_3,T_4\}) = \frac{pq}{4}\sum_{i=1}^{4}\sum_{j=1}^{4}(2j-1)\Delta f_{i,j} \quad (3)$$

The *FAP* calculation can be extended to the rotating test with N_{set} subsets by rewriting (3) into (4).

$$FAP(P:N_{set}|\{T_1,T_2,...T_{N_{set}}\}) = \frac{pq}{N_{set}}\sum_{i=1}^{N_{set}}\sum_{j=1}^{N_{set}}(2j-1)\Delta f_{i,j} \quad (4)$$

From the above discussions, we can observe that the *FAP* of rotating test is depending greatly on test partitioning. We show an example below. Suppose that we need to partition a test set into 4 subsets {T_1, T_2, T_3, T_4}, a fault *f* can be detected by two patterns t_1 and t_2. Let's distribute t_1 and t_2 to subsets in three cases: 1) distribute t_1 and t_2 to the same subset T_1; 2): distribute t_1 to subset T_1 and t_2 to subset T_2; 3) distribute t_1 to subset T_1 and t_2 to subset T_3; In case 1, fault *f* can only be detected by subset T_1, the detection time interval is 4. If we suppose that the factor *pq* in equation (1) is 1, hence average *FAP* of *f* is 16/4=4. In case 2, fault *f* can only be detected by two consecutive subsets T_1 and T_2, however it will be missed at the next two consecutive subsets T_3 and T_4 so that the detection intervals are 1 and 3, respectively. Average *FAP* of *f* is (1+9)/4=2.5. In case 3, fault *f* is detected by T_1 and T_3, missed at the T_2 and T_4. The detection interval is 2, and the average *FAP* of *f* is (4+4)/4=2. Note that distributing test patterns t_1 and t_2 in case 3 can derive the minimum FAP.

The above example shows that for a fault *f* detected by *N* (N>1) test patterns, evenly distributing the *N* test patterns to subsets can minimize the *FAP* for rotating test. However, in general, it is difficult or even impossible to have an optimum partition on the original test sets for rotating test so that the detection patterns of every fault can be distributed evenly. This can be seen by an example. Suppose that a given test set is partitioned into 8 subsets and we have 4 test patterns {t_1, t_2, t_3, t_4} for 3 faults {f_1, f_2, f_3}. Fault f_1 is detected by 4 test patterns, f_2 is detected by t_1 and t_3 and f_3 is detected by t_1 and t_2. For fault f_1, we distribute all its test pattern evenly with T_1 = {t_1}, T_3 = {t_2}, T_5 = {t_3} and T_7 = {t_4}. For fault f_2 there is no problem as its test patterns t_1 and t_3 are distributed evenly. However for fault f_3, we

cannot find a way to assign t_1 and t_2 so that the *FAP* of f_3 is minimized while not increasing the *FAP* of other faults.

In [29], a genetic algorithm (GA) based test partition method is proposed that can find a near optimal partitioning that detection test patterns of all faults are partitioned as evenly as possible. However, since the GA-based test partition method distributes each test pattern to only one subset so that the faults detected by only one test pattern can be detected at one test session but undetected at the other test sessions during one rotating, thus the *FAP* cannot be reduced. The *FAP* of these faults would be very high if there are too many test sessions during a rotating test. Therefore, for minimizing the *FAP* of rotating test, we need to find a partitioning that can reduce the *FAP* for the faults detected by only one pattern. In the next section we formulate a problem for test partitioning and propose a new test partition approach for rotating test aiming at the *FAP* minimization.

3.4.2.2. Problem formulation

From the above discussions on *FAP* in test partitioning for rotating test, we formulate the following problem:

[Test Pattern Partitioning Problem] Given a test set T_{org} consisting of N_{org} patterns and the number of test patterns of each subset N_{test}. Note that T_{org} can detect all the faults of circuit. Then distribute test patterns in T_{org} into subsets. The objective is to minimize the *FAP* of a rotating test with obtained subsets.

For the problem solved in sub-section 3.4.1, the number of subsets is determined from N_{org} and N_{test}. Unlike that problem, because this formulation allows a test pattern of T_{org} to be included in more than one subset, the number of obtained subsets may be larger.

3.4.2.3. Partition algorithms

The partitioning algorithm focuses on two points for failure rate reduction. 1: distribute the detection test patterns of every fault as evenly as possible; 2: for the faults detected by only one pattern, repeatedly distribute the detection pattern to different subsets.

In order to evenly distribute the detection test patterns of faults, every time distributing a new test pattern to a subset we need to select the pattern which can detect more faults only detected by the previous subsets. For example: select a new pattern t_{new} is distributed to subset T_i, t_{new} should detect more faults undetected by other patterns in T_i but (only) detected by $T_i \ldots T_{i-1}$. In this way, the faults that were detected in the previous test session but undetected in the followed test sessions can be detected by applying the new subset. Hence the *FAP* of the faults can be reduced.

For the faults detected by only one pattern, *FAP* can be reduced by repeatedly distributing the test pattern to different subsets. For example, suppose that a test set is partitioned into 4 subsets $\{T_1, T_2, T_3, T_4\}$, fault f is only detected by test pattern t_1 which was distributed to subset T_1. If suppose that $pq = 1$, the *FAP* of fault f is 16/4=4. On the other hand, if distribute t_1 to both subset T_1 and T_3, *FAP* of f can be reduced to (4+4)/4=2.

However, repeatedly distributing test patterns to subsets would cause the number of subsets increase and the *FAP* of some faults would become larger. This can be seen by an example: suppose that partitioning a test set with 8 test patterns into subsets, and the size of each subset is 2. Then the test set can be partitioned into 4 subsets $\{T_1, T_2, T_3, T_4\}$ without repeatedly distributing any test patterns. Suppose that 2 test patterns of the test set $\{t_1, t_2\}$ detect 3 faults $\{f_1, f_2, f_3\}$, where t_1 detects f_1, t_2 detects $\{f_2, f_3\}$. If distribute t_1 to T_1 and t_2 to T_2, the *FAP* of f_1 is 16/4=4 and f_2 is (1+9)/4=2.5. In order to reduce the *FAP* of f_1, we can distribute t_1 to both T_1 and T_3. But, since the size of each subset is 2, repeatedly distributing t_1 needs to increase the number of subsets to 5, and the average *FAP* of f_1 can be reduced to (4+9)/5=2.6. However, for fault f_2, because the number of subsets is increased to 5, the detection time interval between T_3 and the next subset T_2 is 4 so that the average *FAP* of f_2 becomes (1+16)/5=3.4, it is larger than the case of 4 subsets. Therefore, in order to improve the *FAP* for all faults, we need to make the number of subsets as small as possible. The outline of the proposed test partitioning is given below:

Step1: Perform fault simulation in an arbitrary order of the original test patterns, create a fault list "*TarFList*" to record the detectable faults of the given test patterns.

Step2: Create a subset T_1. Select a test pattern which can detect the most faults in "*TarFList*". Then update the fault list "*TarFList*" to remove the faults detected by T_1. Repeat step2 until N_{sub} patterns were distributed to T_1.

Step3: Create a new subset T_i *(i>1)* and a fault list "*OnlyDetFlist*" to store the faults that were detected by the past subset T_j ($j<i$) but undetected by the subsets T_x ($j<x<i$). Select N_{sub} test patterns which can detect the most faults in "*OnlyDetFlist*" and more additional faults in "*TarFList*" to T_i. Every time selected a pattern to T_i update the fault list "*TarFList*" and "*OnlyDetFlist*". Repeat step3 until no faults exist in "*TarFList*".

At step 3, in order to evenly distribute test patterns, we use a fault list "*OnlyDetFlist*" to record the faults that are detected by each of the past subset but undetected by the followed subset, select patterns which can detect more faults in "*OnlyDetFlist*" to the new subset, so that the faults that were detected in the previous test sessions but undetected in the followed sessions could be detected again by applying the new subset. To reduce the failure rate for the faults detected by only one pattern, we set a threshold for detection interval, when selecting a new pattern for a subset T_i, count the number of faults that are only detected by the patterns of subset T_j ($j=i$-*threshold of detection interval*), select the pattern in T_j that can detect the most faults detected by only one pattern to T_i again. Since repeatedly distributing patterns to subsets would increase the number of subsets, in the partitioning procedure we do not set a limit for the number of subsets. In order to make the number of subsets as small as possible, every time selecting new patterns to a subset the new patterns also should can detect more additional faults undetected by all the previous subsets, and the partitioning procedure is stopped if all the detectable faults of the original test patterns are detected by subsets.

3.4.2.4. Experimental Results

We performed experiments on ISCAS'89 circuits and test patterns generated for single stuck-at faults. In the experiments, we set the maximum number of patterns of each subset denoted by N_{test} to 10 and 20. Number of test patterns distributed to every subset never exceeds N_{test}. In order to verify the effectiveness of the proposed approach in improving the

failure occurrence probability for rotating test, we also performed experiments for random partition and the genetic algorithm (GA) based test partition [29].

In the first experiment, N_{test} was set to 10 and the results are given in Table 3.8. The first column gives the name of the circuit, followed by the number of test patterns T_{org}. Columns headed by "*#Subset*" give the number of subsets partitioned by three partition methods, respectively. The columns denoted by "*AveDetInt*" show the results of average detection time interval of all faults in a rotating test for the three partitioning methods. Results of *failure occurrence probability* for different partition methods are shown by the column "*FAP*", respectively. From comparison of the results of average detection time interval and *FAP* among the three partition methods, we can find that the GA based method derived shorter detection time interval of faults and lower *FAP* than the random partitioning because of distributing test patterns evenly. The proposed partition method shorted the detection time interval for all faults mostly and achieved the maximum reduction in *FAP*. This is because the proposed method not only distribute patterns as evenly as possible but also focus on reducing *FAP* for the faults detected by unique pattern through repeatedly pattern distributing. This proves the effectiveness of the proposed test partitioning on minimizing *FAP* for rotating test.

In the second experiment, we increased the size of each subset N_{test} = 20. Experimental results are shown in Table 3.9. From the results we can observe that *FAP* decreased due to the shorter detection time interval of faults as the number of subsets decreased. The effectiveness of the proposed test partitioning also can be observed from the results. In Table 3.10 we give the runtime of the GA based partition method and the proposed partition method when set the size of each subset to 10 and 20, respectively. From the these results we can see that the proposed method take less time to find a better partitioning than the GA based method did.

Table 3.8 Experimental results with N_{test}=10

Circuits	#Pattern	Random Partitioning			GA-based Partitioning			Proposed Method		
		#Subset	*AveDetInt*	*FAP*	*#Subset*	*AveDetInt*	*FAP*	*#Subset*	*AveDetInt*	*FAP*
s1488	101	11	2.688	6.144	11	2.673	5.502	14	2.284	5.424
s5378	100	10	1.44	2.857	10	1.422	2.707	11	1.397	2.677
s9234	111	12	1.958	4.893	12	1.926	4.615	14	1.8	4.494
s13207	235	24	1.906	8.39	24	1.904	7.307	29	1.639	7.05
s15850	97	10	1.601	3.715	10	1.59	3.414	12	1.474	3.271
AVE	-	-	1.845	4.823	-	1.903	4.709	-	1.671	4.273

Table 3.9 Experimental results with N_{test}=20

Circuits	#Pattern	Random Partitioning			GA-based Partitioning			Proposed Method		
		#Subset	*AveDetInt*	*FAP*	*#Subset*	*AveDetInt*	*FAP*	*#Subset*	*AveDetInt*	*FAP*
s1488	101	6	1.926	3.47	6	1.897	3.089	7	1.638	2.631
s5378	100	5	1.235	1.777	5	1.219	1.659	6	1.204	1.644
s9234	111	6	1.526	2.757	6	1.518	2.509	7	1.445	2.397
s13207	235	12	1.622	4.473	12	1.610	3.885	14	1.458	3.556
s15850	97	5	1.345	2.142	5	1.325	1.975	6	1.281	1.871
AVE	-	-	1.489	2.757	-	1.514	2.624	-	1.374	2.287

Table 3.10 Run time for GA and Proposed partition

Circuits	Runtime (sec.) for N_{test}=10		Runtime (sec.) for N_{test}=20	
	GA based partition	*Proposed method*	*GA based partition*	*Proposed method*
s1488	214	0.6	180.7	0.5
s5378	4878.9	6.1	4218.4	5.9
s9234	7344.4	20.7	5992.8	17
s13207	37604.5	168.2	30671.7	126.7
s15850	24723.2	45.9	20523.6	38.1
AVE	14953	48.3	12317.4	37.6

3.5. Conclusions

In this chapter, a rotating test with test pattern partition is presented to meet the limitation of short test application time in field for aging-induced fault detection. In order to improve the test quality for rotating test. First, we proposed test partition algorithm focus on maximizing fault coverage of each subset obtained by partitioning. It consists of two phases, test pattern partitioning and test pattern replacement. Experimental results showed that test partitioning for the rotating test is effective for high quality field test.

Second, we focus on shorting the detection time interval of all faults in rotating test in order to avoid the system failure. A novel method is proposed to estimate the Failure Appearance Probability (FAP) for rotating test, and also proposed a test partitioning approach to minimize the FAP by distributing patterns repeatedly. Experimental results demonstrated the effectiveness of the proposed partitioning approach.

Chapter 4

4. Test Power Reduction for Logic-BIST

Logic-BIST (Built-In Self Test) currently is widely used for system debug or field test because it can provide self-test ability and easily to conduct at-speed testing for timing-related defects to obtain high test quality. However, high power dissipation during test is a vital issue. In Logic-BIST, random test patterns are applied to the test that result higher power dissipation compared to the normal circuit operation since the low correlation between the consecutive random test patterns. Excessive power dissipation can cause higher IR drops and heating of the chip resulting delay variation, and in turn lead to false test or yield loss.

In order to guarantee the reliability of LSI, low power test for Logic-BIST is strongly required. While many approaches have been proposed to tackle the test power problems, inadequate test power reduction and lower fault coverage still remain as vital issues. In addition, existing methods just focus on the test power reduction, however, if the test power is extremely reduced, some small delay may be missed by the test resulting in test quality loss. Therefore, reducing the test power to a specified level is also necessary. In this chapter, a low power Logic-BIST scheme is proposed which not only can significantly reduce the whole test power, but also can control the test power to the specified level.

The rest of this chapter is organized as following: Section 4.1 discusses the power dissipated in LSI. Section 4.2 describes power issues during test. Section 4.3 introduces some low power test technologies. The objective and test power metrics used in this work are described in Section 4.4 and 4.5, respectively. In Section 4.6, the proposed scan-in, capture and scan-out power reduction methods for Logic-BIST are presented, respectively, and also discussing the scan-shift power controlling in this section. The chapter concludes in Section 4.7.

4.1. Power dissipation in LSI

Continuous scaling down of the feature size of complementary metal oxide semiconductors (CMOS) increases the transistor density enabling more functionality to be integrated on a chip and high speed circuit operation. The growth in transistor density results in rapid increase of the power density on the chip since the supply voltage cannot be reduced adequately as the difficulty in keeping its noise margin [30]. Then, power problem becomes a vital issue.

There are two major components of power dissipation in a CMOS circuit: static power and dynamic power [31].

Static dissipation: power is dissipated by the leakage current or other currents drawn continuously from the power supply when the gate is inactive.

Dynamic dissipation: The dominant source of power dissipation in CMOS circuits. Power is dissipated by the current required for charging and discharging the load capacitances when the gate is active. Figure 4.1 shows the power dissipation in a CMOS inverter. When the output switches from 0 to 1 (1 to 0) the PMOS (NMOS) is turned on and the NMOS (PMOS) is turn off, the capacitor (C_L) starts charging (discharging) and dissipates power as a heat. The higher switching activity at the output of the gate is, the more power will be dissipated and generating more heat.

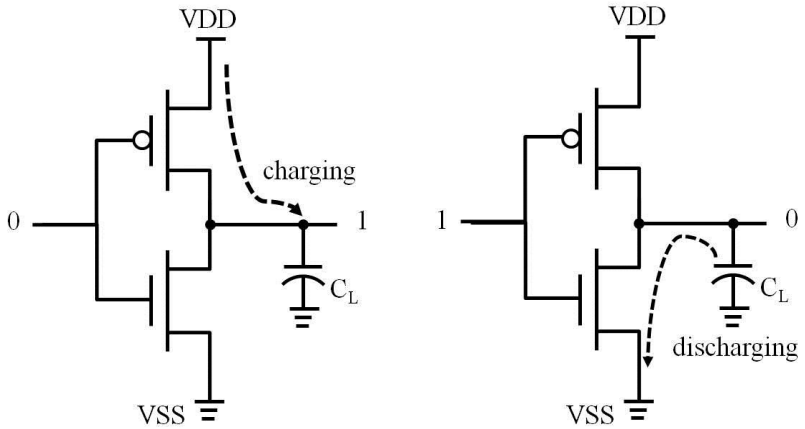


Figure 4.1 Power dissipation in COMS gate

4.2. Test Power Issues

With the ever increasing integration capability of semiconductor technology, more capabilities are integrated onto a chip, and the circuit becomes complicated. Scan design is widely used in order to improve the testability of a circuit, and in turn, achieving the high test quality. In the mean time, as the continuous shrinking in the feature size of transistor and the increase of circuit speed, timing-related defects are high proportion of the total chip defects and at-speed testing is crucial. However, high power dissipation during scan test is a vital issue which limits at-speed testing.

Previous studies have confirmed that switching activity during testing is much higher than that during functional operation which can result in higher dynamic power dissipation during the test.

The source of power dissipated during scan-based test mainly consists of following three parts:

1) Scan-in power: The power dissipated when test patterns are fed into scan chains. Generally, test patterns are generated by an *automatic test pattern generator (ATPG)* for scan-based test or an on-chip *liner feedback shift register (LFSR)* for Logic-BIST. Unlike the normal functional operation, the correlation between successive test patterns is very low. Many toggles (0 to 1, or, 1 to 0) exist in the test patterns. The high toggle rate will produce more switching activities in test than that in normal operation and cause high power dissipation.

2) Scan-out power: The power dissipated when scan out the test responses are dumped from scan chains. Because the goal of test is to activate as many nodes as possible in the circuit, toggle rate of test responses should be very high that can cause excessive switching activities during shift operation.

It should be noted that scan-in and scan-out is done concurrently, scan shift power is the sum of scan-in power and scan-out power.

3) Capture power: The instantaneous power dissipated by captured vectors at capture time. Since test patterns applied to the circuit have non-functional characteristics compare with the input patterns in the functional operation, it tends to cause high power dissipation.

High power dissipation during test causes a high rate of current flowing in power and ground lines leading to excessive peak supply current. Excessive peak supply currents may cause higher IR drops (voltage droop), which tend to increase signal propagation delays of gates. Also, high power dissipation during test would cause the excessive temperature and noise that result in the delay variations which can erroneously change the logic state of circuit resulting in incorrect operation of circuit gates that may cause some good circuits to fail the test and consequently yield loss [32], or the faulty circuits to pass the test resulting in test quality loss. In the mean time, at-speed testing which captures the test response of the scan design at the rated clock-speed is required for detecting the timing-related defects in the very large scale circuit. The excessive power dissipation limits the application of at-speed testing and can result in test-induced yield loss [33].

4.3. Low power test approaches

Low-power test technology has been investigated deeply to tackle the test power issues, and many low power approaches were proposed for scan test and Logic-BIST. Following, a brief review of some of these approaches are presented.

4.3.1. Low power scan test

There exist many low power approaches for scan test that can be classified into two approaches: software-based approach and hardware-based approach. The software-based approach such as utilizing don’t-care (X) bits reduces the test power by modifying the deterministic test patterns generated by automatic test pattern generator (ATPG) [33]. The hardware-based approach involves blocking circuitry inserting [34, 35], scan segmentation technique [36], vector inhibition, selection techniques [37-39] and etc.

4.3.2. Low power Logic-BIST

Logic-BIST currently becomes vital for system debug or field test because it can test the logic circuit itself and easily conduct at-speed testing for timing-related defects to achieve a high test quality. Its test power should be carefully controlled.

Although many sophisticated methods are proposed for scan-test, there are not so many for Logic-BIST because its uncontrollable randomness makes software-based low power approach [33] for scan-test difficult. Hardware approaches for scan-test still might be available for Logic-BIST. However, the blocking circuitry inserting technique requires delay penalty for user paths and the power consumed at FFs, which might be 30-50% of the total area, is not reduced. The scan segmentation technique requires complex clock controlling and power density on a segmented chain might not be reduced. The combination approaches of software and hardware [37-39] propose the vector inhibition and selection techniques that focus on the ratio of care bits on a scan chain or on a block and ineffective ones are enabled with a mask logic or turning-off the clock. These require huge simulation efforts and sophisticated clock controlling.

Other approaches are that control the scan-in test patterns so that their toggle rate be very low [40-43]. In [40] authors proposed a low-transition random TPG (LT-RTPG) and insert it between the random pattern generators (e.g. LFSR: linear feedback shift register) and scan chain inputs to reduce the number of toggles on the test pattern. Figure 4.2 shows the concept of LT-RTPG. The output N bits from LFSR go through an AND gate and toggle flip-flop (T-FF). This means the scan-in bit toggles when all of the N bit values are 1, which is small probability. This method can significantly reduce the scan-in power, and also plays some effects on scan-out power reduction. However, since the generated low scan-in power patterns are low correlation to the original random patterns, fault coverage loss cannot be avoided. Figure 4.3 shows the comparison of LT-RTPG (2) and (3), and the original LFSR. The shadowed cells show value 1 and white cells show value 0. Using an 8-bit LFSR, their output bits are plotted. Although LT-RTPG (2) and (3) reduce toggles, it is seen that frequent toggles remain in some part. Furthermore, as almost half of the original bits are changed in LT-RTPG (2 or 3), there is little correlation between its vectors and the original ones which causes fault coverage loss.

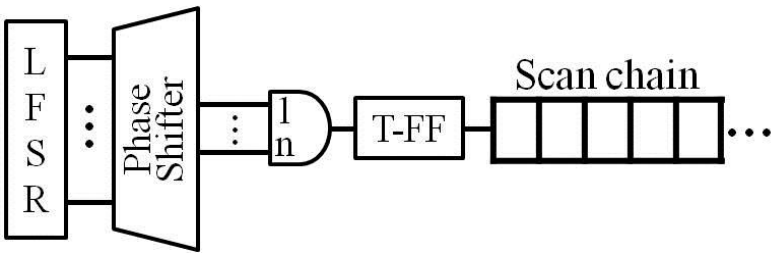


Figure 4.2 Structure of LT-RTPG [40]

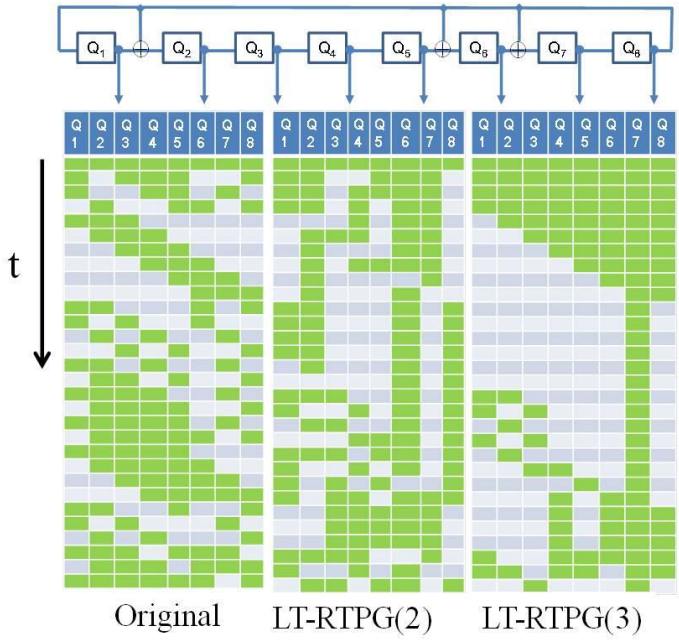


Figure 4.3 Comparison of LT-RTPG and the original LFSR patterns

In [41] authors improved the LT-RTPG and proposed adaptive low shift power test pattern generator (ALP-RTPG). Figure 4.4 shows the concept. The essential improvement is that there is feedback from the last two scan flip-flops (S-FF) on a scan-chain. This feedback mechanism can control the scan-out power. However, the effect of scan-out power is not directly analyzed in the work.

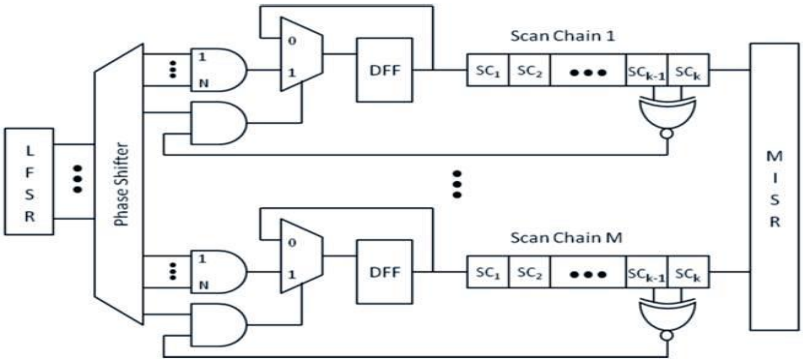


Figure 4.4 Structure of ALP-RTPG [41]

4.4. Objectives of this work

Some approaches have been proposed to tackle the test power problems for Logic-BIST as described in the previous section. Existing methods only take into account the scan-in power reduction, whole power reduction is inadequate. In [40] and [41], both focus on controlling the scan-in power by modifying test patterns with low switching activity that achieved significant scan-in power reduction. However, scan-out power control is out of considering as well as the capture power. And also, fault coverage has been lost as the low correlation of the modified test patterns to the original random patterns, to achieve the desired fault coverage large number of test patterns is required that increases the test application time. In addition, existing methods just focus on the test power reduction. Actually, if the test power is extremely reduced that even much lower than the power in normal operation, some small delays will be missed in the test and results test quality loss.

The objectives of this work are as following:

1): a scan-in control method that can generate low power test patterns with strong correlation to the original random patterns.

2): a scan-out power control method that can reduce the scan-out power directly.

3): a method that can control the capture power.

4): a method that can control the test power to the user specified level.

4.5. Test power metric in this work

In this work, power dissipation during test is divided into three parts:

1): Scan-in power: The power dissipated when test patterns are fed into scan chains.

2): Scan-out power: The power dissipated when scan out the test responses are dumped from scan chains.

3): Capture power: The instantaneous power dissipated by captured vectors at capture time.

It should be noted that scan-in and scan-out is done concurrently, scan shift power is the sum of scan-in power and scan-out power.

To evaluate the power dissipation effectively, we utilize weighted transition metrics (WTM) [33] and set the following metrics for our evaluation. Although it is shown for a scan-chain, it can be easily extended for a parallel scan design.

Scan-in power: The formulation for a scan-in test vector t is as follows.

$$WTM_{in}(t) = \frac{\sum_{i=1}^{L-1}(t_i \oplus t_{i+1}) \times i + (t_L \oplus r_1) \times L}{\sum_{i=1}^{L} i} \quad (1)$$

where L is the scan chain length, t_i is the i_{th} bit of t and t_L is the last bit of t. r denotes the captured test response in the scan chain and the r_1 is the first bit of r. WTM_{in} is the average of (1) for all the test vectors.

Scan-out power: The formulation for a test response r is given as follows.

$$WTM_{out}(r) = \frac{\sum_{i=1}^{L-1}(r_i \oplus r_{i+1}) \times (L-i)}{\sum_{i=1}^{L-1}(L-i)} \quad (2)$$

where L is the scan chain length and r_i is the i_{th} bit of r. WTM_{out} is the average of (2) for all the test response vectors.

Scan-shift power: The sum of WTM_{in} and WTM_{out} is used as the metric of the total scan-shift power. This is the same as average weighted transition metric. The metric for test vector t and test response r is as follows.

$$WTM(t,r) = \frac{1}{2}[WTM_{in}(t) + WTM_{out}(r)] \quad (3)$$

WTM is the average of (3) for all the test vectors and the test responses.

Capture power: A simple metric that measures the toggle rate at FFs is used for our evaluation. The metric for a test vector t and test response r will be as follows.

$$CTM(r_i) = \frac{1}{L}\sum_{j=1}^{L}(t_j \otimes r_j) \quad (4)$$

where L is the scan chain length and t_j (r_j) is the j th bit of $t(r)$. *CTM* is defined as the average of (3) for all the test vectors and the test responses.

4.6. Proposed Power Reduction Methodologies

4.6.1. Low power for Scan-in and Capture

In [34], power reduction methods for scan-in and capture are proposed. Where, a new low power test pattern generation circuit called *pseudo low-pass filter* (PLPF) is utilized to modify the random test patterns generated by LFSR so as to make the scan-in patterns smoother with low switching activity, and in turn reduce the scan-in power. In order to reduce the capture power, a multi-cycle test scheme is employed. Following, these methods will be presented in detail.

4.6.1.1. Scan-in power reduction with PLPF (Pseudo Low-Pass Filter)

Scan-in power reduction requires reducing the number of toggles in scan-chain. Let define f_1 as a pattern of repeated bits "01010101…," which has the most toggles. In the same way,

f_2 be "001100110011…," which has the secondary many toggles in vectors of repeated bits. Seeing the vector in Figure 4.5, some parts of f_1 patterns or f_2 patterns are found. It is apparent if these high frequency parts are removed from the vector, the number of toggles will be reduced in a convincing way. It suggests a kind of low-pass filter will be effective.

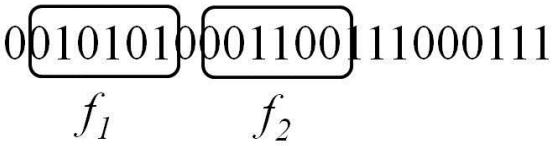


Figure 4.5 High frequency in a test vector

Figure 4.6 shows the overall proposed structure for scan-in power reduction. We use a conventional linear feedback shift register (LFSR), which can be an internal type or an external type. The pseudo low-pass filter (PLPF) is proposed, which is a combinational circuit and generates a modified bit of S_i (i=1, L) from the 2k+1 bits inputs $S_{i,j-k}$, $S_{i,\,j-k+1}$, , , $S_{i,\,j-1}$, $T_{i,\,j}$, $T_{i,\,j+1}$, , , $T_{i,\,j+k-1}$, $T_{i,\,j+k}$. The past bits $S_{i,j-k}$, $S_{i,\,j-k+1}$, , , $S_{i,\,j-1}$ come from the scan chain connected to the PLPF through feedback loops and the current and future bits $T_{i,\,j}$, $T_{i,\,j+1}$, , , $T_{i,\,j+k-1}$, $T_{i,\,j+k}$ come from a kind of phase shifter (PSF: the phase shifter for filter). It is well-known that the future bits can be extracted using the current bits with a combinational logic. However, in case of 2k+1=3 or 5, PSF is very simple as shown later.

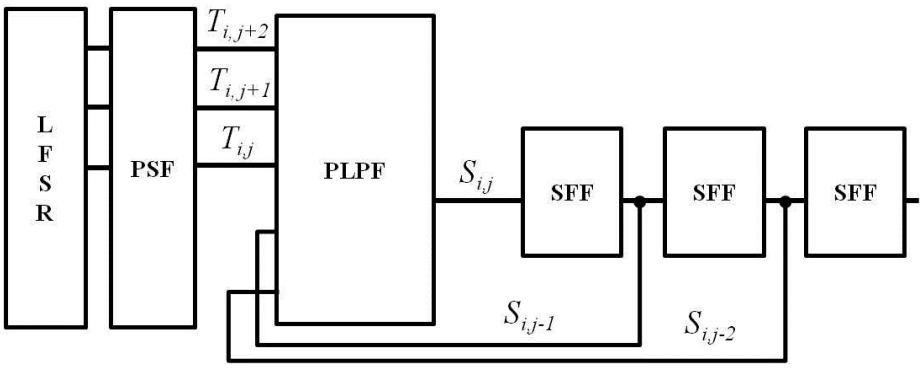
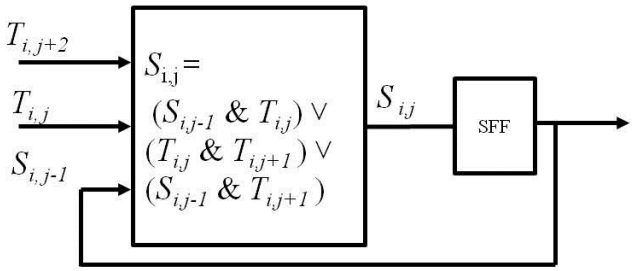


Figure 4.6 Structure for scan-in power reduction



$T_{i,j+1}$	$T_{i,j}$	$S_{i,j-1}$	$S_{i,j}$
0,	0,	0	$(0+0+0)/3 = 0/3 \rightarrow 0$
0,	0,	1	$(0+0+1)/3 = 1/3 \rightarrow 0$
0,	**1**,	0	$(0+1+0)/3 = 1/3 \rightarrow \underline{\mathbf{0}}$
0,	1,	1	$(0+1+1)/3 = 2/3 \rightarrow 1$
1,	0,	0	$(1+0+0)/3 = 1/3 \rightarrow 0$
1,	**0**,	1	$(1+0+1)/3 = 2/3 \rightarrow \underline{\mathbf{1}}$
1,	1,	0	$(1+1+0)/3 = 2/3 \rightarrow 1$
1,	1,	1	$(1+1+1)/3 = 3/3 \rightarrow 1$

Figure 4.7 Detail Structure of PLPF (3)

We define PLPF (2k+1) as a PLPF with 2k+1 bits. Figure 4.7 shows the detailed structure of PLPF (3). The output bit $S_{i,j}$ is defined as the moving average (a low-pass filter) of the three input bits $S_{i,j-1}$, $T_{i,j}$, $T_{i,j+1}$. The moving average is the average of the past, current and future, which makes the sequence smoother and remove high frequency factors. Using this filter, f_1 (i.e. 010 or 101) components are completely eliminated. This function can be implemented based on the following equation.

$$S_{i,j} = \left(S_{i,j-1} \& T_{i,j}\right) | (T_{i,j} \& T_{i,j+1}) | (S_{i,j-1} \& T_{i,j+1})$$

In the same way, f_2 (i.e. 001100 or 110011) components can be eliminated by PLPF (5), and f_k components can be eliminated by PLPF (2k+1) as proved in the following.

Definition 1: The output bit S_j of PLPF (2k+1) with inputs S_{j-k}, S_{j-k+1}, ... , S_{j-1}, T_j, T_{j+1}, ... , T_{j+k-1}, T_{j+k} is defined as follows.

If sum of all input bits > k, then S_j =1; else S_j =0;

Theorem 1: *The output vectors constructed of S_j of PLPF (2k+1) contain no f_l ($l < k+1$) components once it satisfies the following initial condition at some j_0.*

Ini_Con: The sequence of {S_{j-l} (l=1 to k+1)} has one or no toggle.

Proof: In case of $T_j = S_{j-1}$: When all of S_{j-l} (l=1 to k+1) are the same, there are more than k+1 same values around (i.e. $l = j-k, j-k+1, \ldots, j-1, j, j+1, \ldots, j+k$) T_j, therefore, $S_j = S_{j-l}$ (l=1 to k+1). When $S_{j-m} \neq S_{j-m-1}$ ($m < k+1$), $S_{j-1} = S_{j-2} = \ldots = S_{j-m}$ and $S_{j-m-1} = S_{j-m-2} = \ldots = S_{j-k-1}$ from *Ini_Con*. As there are more than k+1*the* same values around S_{j-1}, they also exist around T_j. Therefore, $S_j = S_{j-l}$ (l=1 to m). In both case, *Ini_Con* is kept.

In case of $T_j \neq S_{j-1}$: When all of S_{j-l} (l=1 to k+1) are the same, it is apparent that only the case of $T_j = T_{j+l}$ (l=1 to k), $S_j = T_j$ (l=1 to k) and $S_{j+l} = S_j$ (l=1 to k). For S_{j+l} (l=1 to k), *Ini_Con* is kept. When $S_{j-m} \neq S_{j-m-1}$ ($m < k$), $S_{j-1} = S_{j-2} = \ldots = S_{j-m}$ and $S_{j-m-1} = S_{j-m-2} = \ldots = S_{j-k-1}$ from *Ini_Con*. As there are more than k+1 same values around S_{j-1}, they also exist around T_j. Therefore, $S_j = S_{j-l}$ (l=1 to m). In both cases, *Ini_Con* is kept. From the above discussion, it is shown that S_j toggles in only the case of k consecutive bits. *Q. E. D.*

Regarding the initial condition, as it is a loose restriction, it is satisfied soon in our experience. Figure 4.8 and 4.9 show examples of PSF circuits for PLPF (3) and for PLPF (5), respectively. In PLPF (3), the PSF constructs of only wire connections. Even in PLPF (5), the PSF constructs of small number of gates.

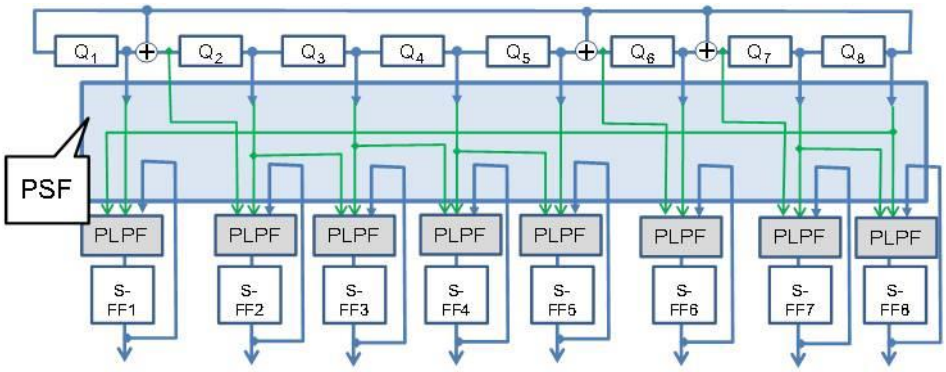


Figure 4.8 PSF for PLPF(3)

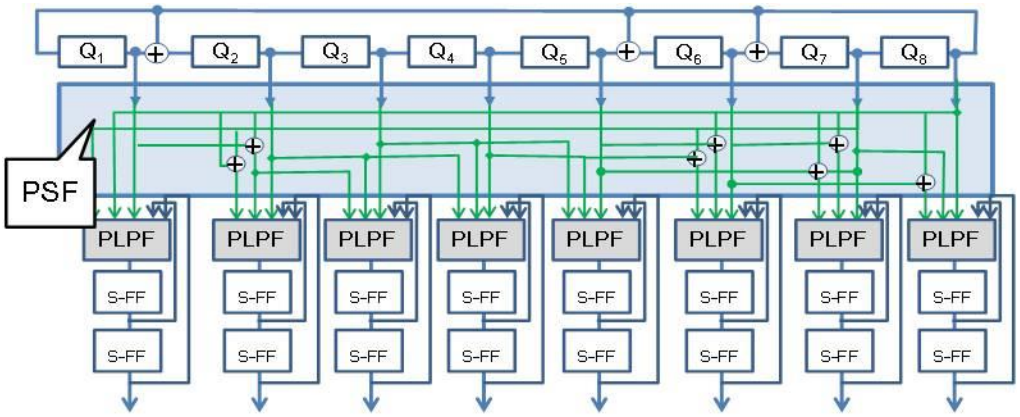


Figure 4.9 PSF for PLPF(5)

Figure 4.10 shows the comparison with the original vector and the vector modified by PLPF (3). It is seen that the new vector is quite similar to the original one and only 17% bits are changed whereas nearly 50% bits are changed by LT_RTPG(2) and (3). The ratio of 0 and 1 value are kept almost the same. These features are preferable because the experimental knowhow with the original vectors regarding such as fault coverage, power or reseeding information might be kept even in the new vectors.

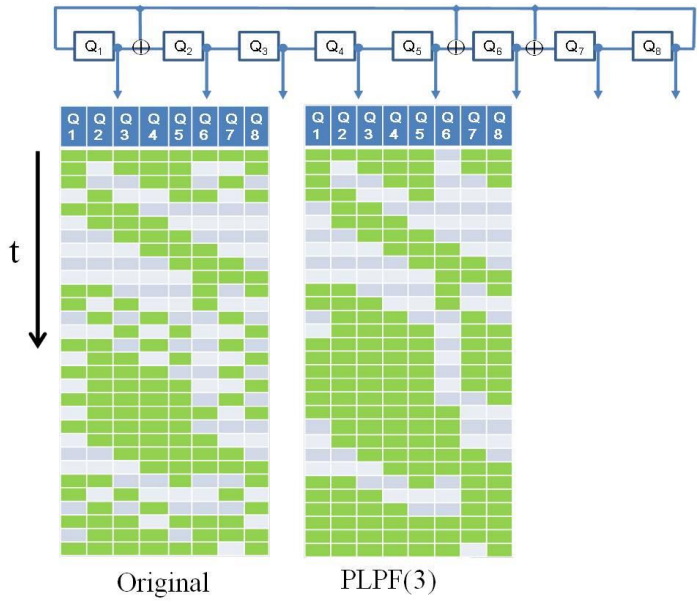


Figure 4.10 Vector Comparison with PLPF and LFSR

4.6.1.2. Capture power reduction with Multi-cycle test

To reduce the capture power in Logic-BIST, multi-cycle test scheme is utilized. In [45], authors reported that many capture cycles can reduce the capture power. As an excessive capture power causes timing issues during at-speed test, many captures with a slow timing might reduce the capture power without causing timing issues. Figure 4.11 shows an improved capture timing scheme. The first M captures are applied with a slow timing and the last capture is applied with at-speed timing. However, a significant decrease of fault coverage is a concern. To tackle this problem, we utilize the multi-cycle BIST scheme [46] in Figure 4.12. In the scheme, a part of FFs are directly observed using a compactor during many captures, which prevent the decrease of fault coverage. In the work, the observation of 20% FFs are recommended with 2% area penalty. Figure 4.13 is the proposed capture timing. The first M capture clocks are applied with a slow speed and the following N captures are applied with at-speed timing. A part of FFs are directly observed during M+N captures (in this case, the stuck-at faults can be detected in the first M captures while the delay faults are detected in the following N captures) or during the last N captures (in this case, only the delay faults are focused).

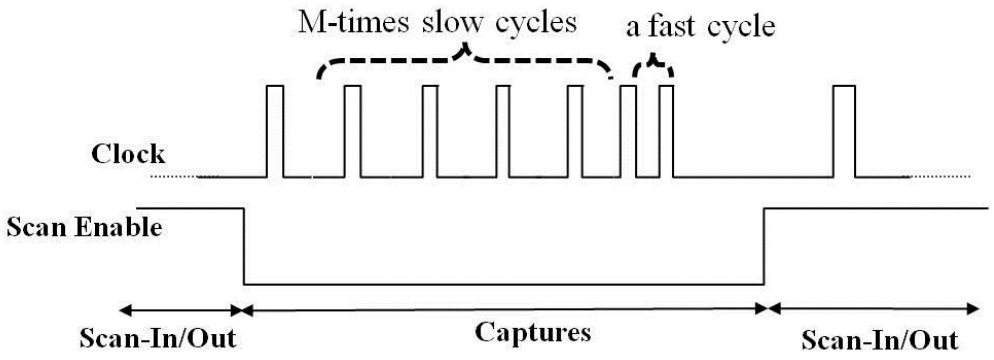


Figure 4.11 Multi-capture waveform

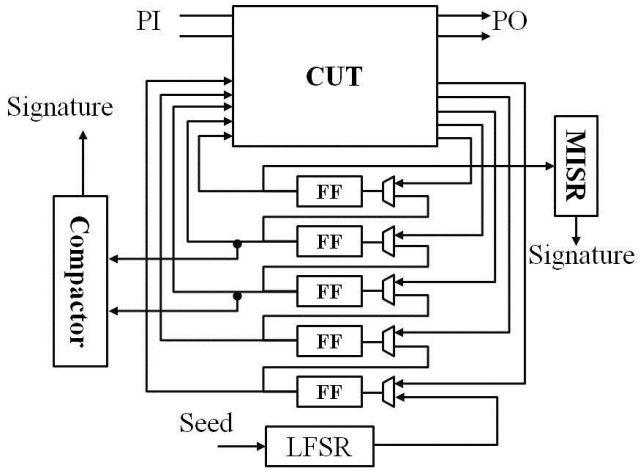


Figure 4.12 Multi-cycle BIST with partial observation [46]

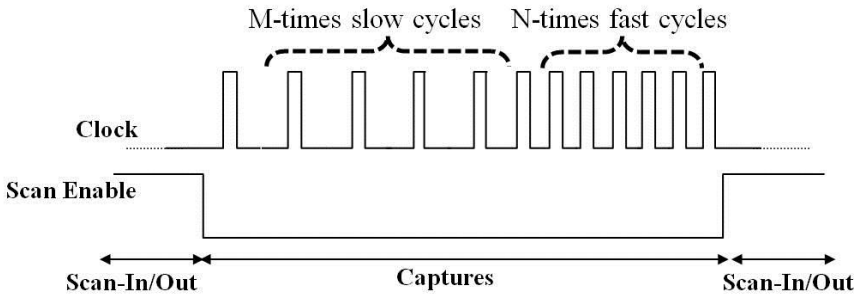


Figure 4.13 Proposed capture timing

4.6.1.3. Experimental results

The proposed technology was evaluated using the ISCAS89 and ITC99 benchmark. A 16-bit internal type LFSR (characteristic polynomial: $X^{16}+X^{15}+X^{13}+X^{4}+1$) and generated 30k vectors were used. A parallel scan structure with 100 FFs scan-chain is adopted. Primary inputs are also fed by a LFSR. Primary outputs are not observed during BIST. Multi-cycle BIST with *M* slow capture and *N* fast capture is used (we refer this scheme as Mul (*M*, *N*)). In the scheme, 20% of FFs are observed using the SCOPE-based selection [12]. An in-house fault simulator is used to calculate the single stuck-at fault coverage. For comparison purpose, consecutive bits of LFSR outputs are input to LT (*N*) (*N*=3, 4) or ALP (*N*) (*N*=2, 3) for the evaluation. Although inputs bits are not restricted to those in [40, 41], this is done for convenience.

Scan-in Power Reduction

Table 4.1 shows the comparison of scan-in power reduction using Mul (0, 1) scheme. Here, "IN", 'OUT" and "Ave." show WTM_{in}, WTM_{out} and WTM respectively. "Peak" means the maximum of WTM (t_i, r_i). LT (3), ALP (2) and PLPF (3) achieve from 13 to 17% rate of WTM_{in}. LT (4), ALP (3) and PLPF (5) achieve from 7 to 8% rate of WTM_{in}. This shows that these two groups of methods should be selected according to the required grade of reduction rate. It is seen that WTM_{in} is well-controlled with LT and PLPF in variation of less than 0.7%. However, it is up to 4.6% variation with ALP. WTM_{out} is also reduced. However, its amount differs greatly depending the circuit and usually is larger than WTM_{in} except s35932. The peak rate of WTM (t_i, r_i) is an important metric because the problem of IR-drop are caused vector by vector. PLPF (5) looks to have a good controllability of the peak power than others. However, it is still greater than twofold of WTM_{in}.

Table 4.1 Scan-in power reduction with Mul (0, 1)

Circuit	LFSR				LT (3)				LT (4)				ALP (2)				ALP (3)				PLPF (3)				PLPF (5)			
	IN	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak	*IN*	OUT	Ave.	Peak
s38417	***50.2***	44.7	47.5	55.7	***12.9***	25.8	19.3	28.1	***6.6***	22	14.3	23.3	***14***	22.7	18.3	26.9	***7.4***	16.7	12.1	19.1	***17***	28.7	22.8	29.8	***7.5***	23	15.3	21.2
s38584	***50.4***	47.5	49	60.5	***12.9***	22.5	17.7	30.9	***6.6***	18.4	12.5	25.4	***14.2***	21	17.6	29.5	***8***	16.3	12.1	21.9	***17.2***	25.4	21.3	29.6	***7.6***	19	13.3	22
s35932	***50.3***	25.2	37.7	59	***13***	9.4	11.2	27	***6.7***	5.9	6.3	19.4	***14.1***	9.5	11.8	26.9	***7.7***	5.6	6.7	18.1	***17.1***	11.6	14.3	26.6	***7.6***	6.4	7	16.2
b14s	***50.4***	48.1	49.3	65.5	***13.4***	16.7	15.1	27	***7.2***	11	9.1	25.8	***18.6***	20.5	19.5	37.2	***10***	12.9	11.4	24.3	***17.5***	20.3	18.9	32.6	***8.1***	11.7	9.9	23.1
b15s	***50.5***	47.7	49.1	62.6	***12.9***	13.6	13.2	26.6	***6.7***	8.1	7.4	18.6	***15.8***	15.6	15.7	29.9	***8.6***	9.5	9.1	18.2	***17.1***	17.3	17.2	27.1	***7.6***	8.9	8.2	15.6
b17s	***50.2***	47.8	49	58.6	***13.1***	13.4	13.3	23.2	***7***	7.9	7.4	16.2	***14.1***	14.5	14.3	22	***8***	9.4	8.7	15.7	***17.2***	17.1	17.2	22.7	***7.8***	8.6	8.2	13.3
b20s	***50.5***	48.6	49.5	63.9	***12.8***	16.9	14.9	28.1	***6.6***	10.7	8.6	21	***15.6***	18	16.8	33	***8.3***	10.8	9.6	21.9	***17***	20.9	18.9	29.1	***7.4***	12.1	9.8	19
b21s	***50.5***	48.6	49.5	63.2	***12.8***	17.1	15	28.1	***6.6***	11.1	8.8	21	***15.6***	18	16.8	33	***8.3***	10.7	9.5	21.9	***17***	20.9	18.9	29.1	***7.4***	12.2	9.8	18.9
b22s	***50.5***	48.5	49.5	62.1	***13***	17.4	15.2	29.6	***6.8***	11.5	9.2	22.1	***14.3***	18.2	16.2	29.7	***7.6***	12.7	10.1	19.9	***17.1***	21.4	19.3	28.6	***7.6***	12.9	10.2	17.6
Ave.	***50.4***	45.2	47.8	61.2	***13***	17	15	27.6	***6.8***	11.8	9.3	21.4	***15.1***	17.5	16.3	29.8	***8.2***	11.6	9.9	20.1	***17.1***	20.4	18.8	28.4	***7.6***	12.8	10.2	18.5

Capture Power Reduction

Table 4.2 shows the comparison of capture power reduction using Mul (N, 1) scheme, where N is set to 1, 15, 20 or 30. "Ave." shows CTM and "Peak" shows the maximum of CTM (r_i). The peak rate of CTM (r_i) is an important metric because voltage droop are caused vector by vector. It is easily seen that LT has a little effect of capture power reduction. ALP reduces CTM up to nearly 10%; however, it has also little effect of peak capture power reduction. Mul (N, 1) not only reduces CTM drastically (to 7.3% by PLPF (3) and 6.9% by PLPF (5)), but also reduces the peak capture power up to nearly 15%.

Table 4.2 Capture power reduction with Mul (1/15/20/30, 1)

Circuit	LFSR		LT (3)		LT (4)		ALP (2)		ALP (3)		PLPF (3)								PLPF (5)							
											Mul (1,1)		Mul (15,1)		Mul (20,1)		Mul (30,1)		Mul (1,1)		Mul (15,1)		Mul (20,1)		Mul (30,1)	
	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak	Ave.	Peak
s38417	27.6	35.3	22.7	36.6	21.9	36.2	15.1	27	11.8	25.7	23.1	34.2	2.5	10.5	2.6	10.8	2.4	10.1	22.1	35.6	2.1	9.1	2.2	9.8	2.1	9.8
s38584	37.5	59.7	19.6	32.6	16.8	31.6	16	28.8	11.7	26.4	21.3	31.2	8.7	14.8	7.6	15.3	5.9	14.7	17	32.2	7.4	13.8	7.2	15.2	5.9	14.6
s35932	50	59.8	46.8	72.2	44.6	74.7	25.6	64.9	18	58.9	47.2	65.6	18.3	55.3	17.7	55.7	16.5	55.7	46.5	74.3	16.7	58.8	16.1	58.8	15	58.1
b14s	13.1	41.2	12.9	44.1	14.7	48.6	10.5	44.1	9	43.7	12.8	41.6	10.2	28.2	10.1	27.3	10.1	28.6	13.9	51	10.1	28.2	10.2	28.6	10.1	27.8
b15s	5.9	19.6	7.1	22.9	7.7	25.4	6.6	24.7	6.7	16.9	6.9	21.8	1.2	13.4	1	12.7	0.9	11.4	7.5	21.6	1.1	13.4	1	18	0.9	11.6
b17s	5.8	13.9	6.8	15.5	7.3	17	7.2	17	7.2	16.2	6.6	16.5	0.9	7.3	0.7	7.1	0.6	7.6	7.1	17	0.8	6.9	0.7	5.7	0.6	6.9
b20s	13.1	37.3	13.1	36.9	14.5	45.5	9.3	41	6.7	43.5	12.9	36.5	9.6	26.3	9.6	26.5	9.6	24.7	14.2	43.5	9.1	27.6	9.1	26.5	9.1	25.3
b21s	13.1	37.6	13.5	39.2	15.5	43.9	9.6	42.2	6.9	42.7	13	35.9	9.6	26.3	9.6	26.7	9.6	26.5	14.5	40.8	9	28	9.1	28	9.1	29.6
b22s	13.1	30.9	13.9	36.3	16	42	9.5	35.4	7.5	28.6	12.9	32	9.8	21.1	9.8	22.3	9.8	24.1	14	37.4	9.4	26	9.4	23.4	9.4	22.9
Ave.	19.9	37.3	17.4	37.4	17.7	40.5	12.2	36.1	9.5	33.6	17.4	35	7.9	22.6	7.6	22.7	7.3	22.6	17.4	39.3	7.3	23.5	7.2	23.8	6.9	23

Figure 4.14 shows the reduction curve of Mul (*N*, 1) for each data. Although, the effect of the proposed method is large, there are some data whose peak power doesn't reduce so much such as s35932 or b14s. Figure 4.15 shows the CTM (r_i) distribution of b14 for 30k vectors using Mul (10, 1) scheme. Figure 4.15 (a) is CTM of 1st capture and Figure 4.15 (b) is CTM of 10th capture. It is seen that almost half vectors have low toggle rates, which contribute to reduce the average CTM. However, the remaining vectors have still high toggle rates. s35932 also has the similar behavior. If the behavior were the same as the original functional one, no other reduction method would be needed. If not, more improved technique is required for future.

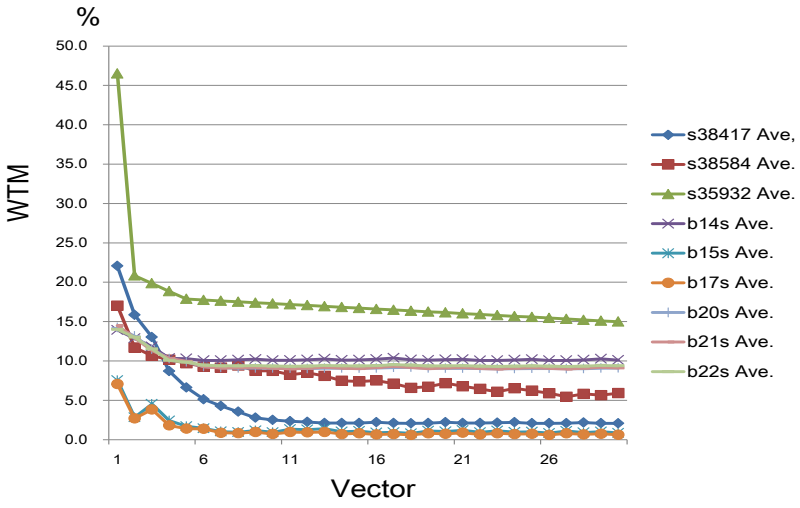


Figure 4.14 Capture-Power Reduction with Mul (N, 1)

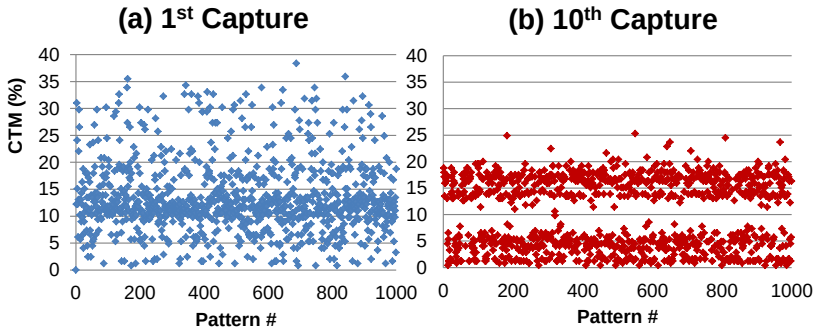


Figure 4.15 Capture toggle rate (CTM) of each vector

Test Coverage Estimation

Table 4.3 shows the comparison of test coverage. Here, "Scan", "Cap." and "TC" show WTM, CTM and test coverage respectively. As referred in section 4.2, LT (3), ALP (2) and PLPF (3) achieve the similar WTM rate each other and LT (4), ALP (3) and PLPF (5) achieve the similar WTM rate each other too. PLPF (3) with Mul (10, 10) scheme and PLPF (5) with Mul (10, 10) scheme achieve nearly 10% better test coverage (in average) than LT or ALP. It should be noted that it is better than the original LFSR's (in average) even at low shift-power. For applying Mul (10, 10) scheme, it is reported that 2% area penalty/investment is needed [12]. The proposed power BIST technology achieves low scan-sift power and low capture power with high test coverage.

Table 4.3 Test Coverage Evaluation

	LFSR			.LT (3)			ALP (2)			.PLPF (3) Mult (0,1)			.PLPF (3) Mult (10,10)			.LT (4)			ALP (3)			.PLPF (5) Mult (0,1)			.PLPF (5) Mult (10,10)		
Circuit	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*	Scan	Cap.	*TC*
s38417	47.5	27.6	***93.7***	19.3	22.7	***90.9***	18.3	16.1	***88.2***	22.8	23.1	***91.9***	19.7	2.6	***93.9***	14.3	21.9	***90.2***	12.1	11.8	***83.9***	15.3	22.1	***90.1***	13.3	2.2	***92.1***
s38584	49	37.5	***91.2***	17.7	19.6	***86.8***	17.6	16	***86.9***	21.3	21.3	***87.1***	18.7	7.6	***88***	12.5	16.8	***86***	12.1	11.7	***81.9***	13.3	17	***83.7***	13.3	7.2	***86***
s35932	37.7	50	***86.7***	11.2	46.8	***86.7***	11.8	25.6	***86.7***	14.3	47.2	***86.7***	17.2	17.7	***83.8***	6.3	44.6	***86.7***	6.7	18	***86.7***	7	46.5	***86.7***	11.4	16.1	***83.8***
b14s	49.3	13.1	***85***	15.1	12.9	***80.5***	19.5	10.5	***86.3***	18.9	12.8	***82***	20.7	10.1	***90.1***	9.1	14.7	***79***	11.4	9	***81.2***	9.9	13.9	***77.3***	12.8	10.2	***89.3***
b15s	49.1	5.9	***75.2***	13.2	7.1	***43.6***	15.7	6.6	***58***	17.2	6.9	***42.9***	19.2	1	***92.9***	7.4	7.7	***40.8***	9.1	6.7	***52.2***	8.2	7.5	***39.8***	11	1	***92.9***
b17s	49.5	5.8	***84.3***	14.8	6.8	***83.8***	16.8	7.2	***85.4***	18.9	6.6	***81.8***	20.9	0.7	***90.8***	8.6	7.3	***83.9***	9.6	7.2	***81.1***	9.7	7.1	***81.5***	12.7	0.7	***91.7***
b20s	49	13.1	***80.5***	13.3	13.1	***59.7***	14.3	9.3	***54.8***	17.2	12.9	***42.1***	18	9.6	***73.4***	7.4	14.5	***56.4***	8.7	6.7	***49.6***	8.2	14.2	***39***	9.4	9.1	***67.6***
b21s	49.5	13.1	***86***	15	13.5	***85.2***	16.8	9.6	***87.4***	18.9	13	***83.2***	20.8	9.6	***91.7***	8.8	15.5	***84.9***	9.5	6.9	***82.8***	.9.8	14.5	***82.7***	12.7	9.1	***92.4***
b22s	49.5	13.1	***85.3***	15.2	13.9	***84.3***	16.2	9.5	***85.9***	19.3	12.9	***82.1***	21.3	9.8	***90.6***	9.1	16	***83.2***	10.1	7.5	***81.4***	10.2	14	***80.4***	13.3	9.4	***91.4***
Ave.	47.8	19.9	***85.3***	15	17.4	***77.9***	16.3	12.3	***80***	18.8	17.4	***75.7***	19.6	7.6	***88.4***	9.3	17.7	***76.8***	9.9	9.5	***75.6***	9.1	17.4	***73.5***	12.2	7.2	***87.5***

4.6.2. Scan-in power controlling

4.6.2.1. Requirement of power controlling

High power dissipation during test can increase the delay resulting in some good circuits to fail the test and consequently yield loss. Thus, test power reduction is necessary. On the other hand, if the test power is extremely reduced that even lower than the power dissipated in the normal operation, the effect of some small timing-related defects in the circuit likely cannot be observed at the output during test. This can cause a defective circuit appears to be a good part passing the test, and leading to reliability degradation of LSI. Therefore, controlling the test power to a specified rate accommodate to the requirement of design is required.

4.6.2.2. Scan-in power controlling scheme

In section 4.6.1, a pseudo low-pass filter has been presented for scan-in power reduction. Changing the number of input bit fed from the LFSR for PLPF can reduce the scan-in pattern toggle rate to different level. Using the conditional probability calculation, it is easy to see that PLPF (3) generates vectors of 12.5% toggle rate and PLPF (5) generates vectors of 6.7% toggle rate under the assumption that the original vectors have complete randomness with 50% toggle rate. This suggests that any toggle rate λ of vectors between 6.7% and 50% are realizable assigning the values of α, β and γ in the following equations.

$$\lambda = \alpha T_1 + \beta T_2 + \gamma T_3 \tag{6}$$

$$\alpha + \beta + \gamma = 1 \tag{7}$$

where T_1 (=50%) is the rate of LFSR, T_2 (=12.5%) is that of PLPF (3) and T_3 (=6.7%) is that of PLPF (5).

Figure 4.16 is an example of controlling circuit. It is apparent that the solution of (α, β, γ) that satisfies (6) and (7) are not unique. In our experience, the remained randomness after reducing power affects the fault coverage. Then, assigning α as a large number will be more effective.

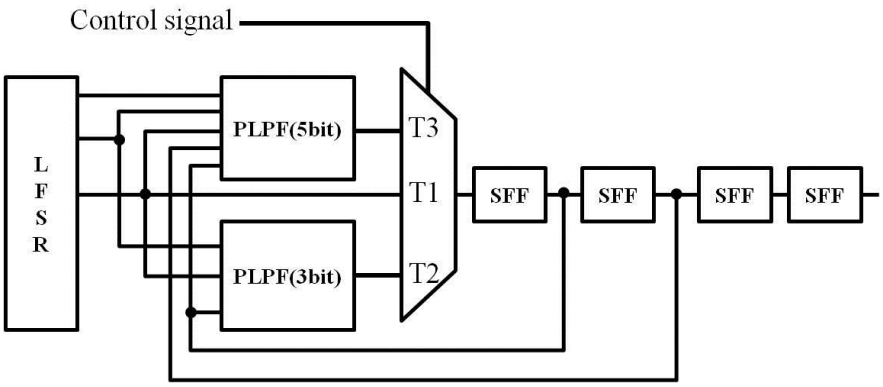


Figure 4.16 Power controlling scheme

As discussed above, this scan-in power controlling scheme has the capability to generate a test pattern with the toggle rate between 6.7% and 50% so as to control the scan-in power to a user specified rate. Since the scan-out power that dissipated by the test responses was not taken into account, accurate shift-power controlling should be still difficult. An effective scan-out power reduction method is strongly required which will be discussed in the following sections.

4.6.3. Scan-out power reduction by outputs overwriting

4.6.3.1. Scan-out power issues

As described in section 4.6.1, we proposed a low power BIST scheme that can reduce the scan-in power and capture power with little loss of fault coverage by using a *Pseudo Low-Pass Filter* (PLPF) and multi-cycle capture technique. Yet, as earlier low power methods done (LT-RTPG, ALP-RTPG), the scan-out power is just reduced as bi-product of scan-in power reduction. The amount of reduction is not enough compared to that of scan-in power. Figure 4.17 shows the comparison of the scan-in power and the scan-out power reductions in LT-RTPG, ALP-RTPG and PLPF for 5 ITC99 benchmark circuits. It can be seen that scan-out power has been reduced as the effect of scan-in power reduction. However it is still almost 2-times higher than scan-in power. It is difficult to control the total scan-shift power only from scan-in power control. Therefore, scan-out power control still remains as important issue.

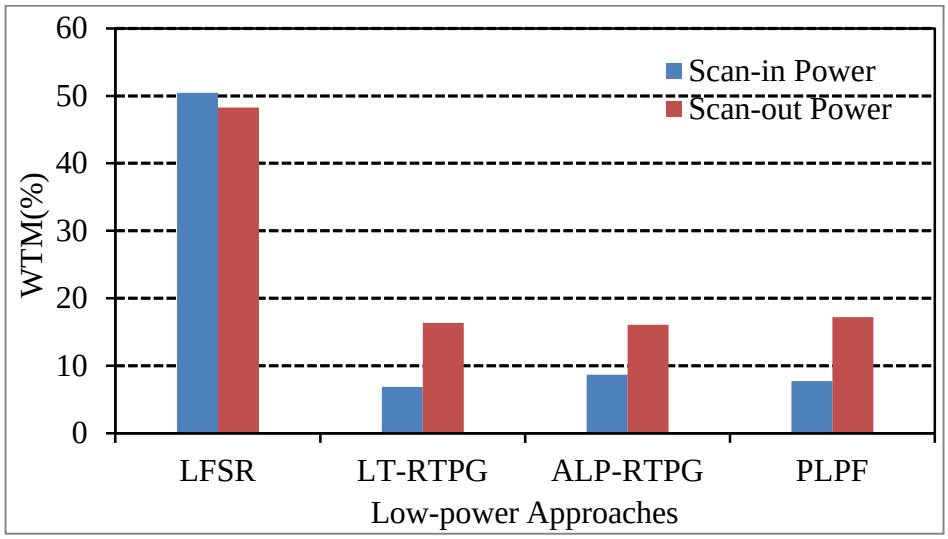


Figure 4.17 Comparison of scan-in/out power reduction

In this work, we employ the low power multi-cycle BIST scheme described in section 4.6.1 and proposed a novel scan-out power reduction method [47, 48]. Following, it will be described in detail.

4.6.3.2. Proposed scan-out power reduction method

Figure 4.18 shows the idea of the proposed scan-out power reduction method. First, the method selects some FFs for scan out control at logic design phase. We call the FFs "control FFs" (denoted as "C"). In scan testing, it fills the control FFs with proper values after the last capture before starting scan-shift operation using additional control circuits to make the scan-out vector smoother so as to reduce the switching activity associated with scan-out vectors.

It should be noted that because capture power caused by launch affects delay propagation in the capture cycle, value-filling after the last capture will not affect the capture power. Since filling new values to the control FFs before observing the original captured values would affect the fault coverage. For compensating fault coverage loss, we employ multi-cycle scan test with partial observation [46], where the responses of CUT captured into the control FFs are compressed directly into an additional compactor (*compactor B*) during many captures (except the last capture) and the output (signatures) of *compactor B* are observed in shift mode as well as *compactor A*. Multi-cycle test consists of

many capture cycles, which brings more chances of fault sensitization [49], and capturing the values of control FFs into *compactor B* during multiple clock cycles increases the number of fault detection chances. Therefore, even if the captured values of the control FFs are modified after the last capture, the loss of fault coverage could be compensated.

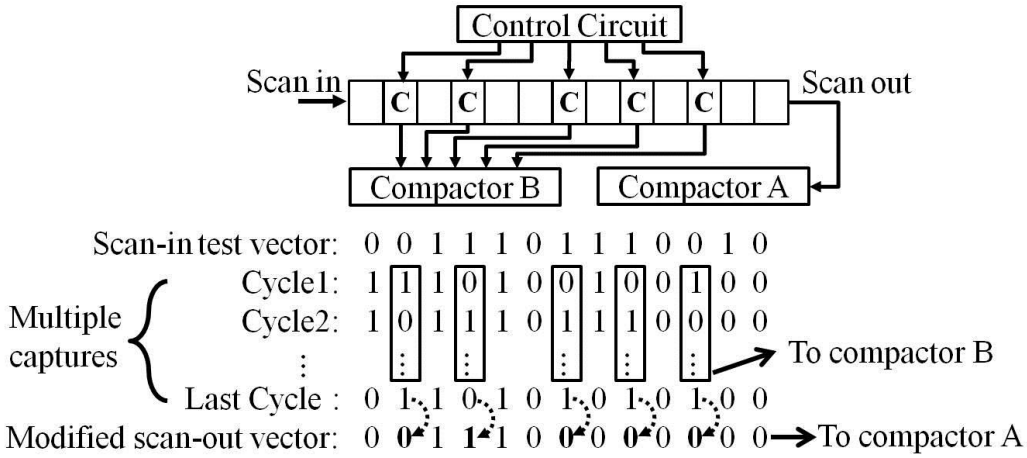


Figure 4.18 Idea of scan-out power reduction

The proposed scan-out power reduction method has two issues to be solved as follows.

1. Control FF selection: It is needed to determine FFs that could reduce the scan-out power mostly by value filling and with less fault coverage loss by partial observation.

2. Value filling method: For the selected control FFs, we need to determine that what value to fill them could achieve the largest scan-out power reduction and control circuit will cause less hardware overhead increase.

4.6.3.3. Control FFs' Selection

When controlling the scan out power, we need to consider not only scan-out power reduction but also fault coverage loss simultaneously. Although a part of fault coverage loss caused by filling new logic values in the control FFs might be compensated by multi-cycle test and partial observation, it is still needed to reduce fault coverage loss as far as possible. In this work, we first theoretically analyze factors for the scan-out power reduction and the fault coverage improvement independently, and then propose a control FFs selection method that takes the factors into consideration simultaneously.

a. Factor for scan-out power reduction

In multiple captures test, *high frequency bits* in the test responses (i.e., "101010" or "010101") that often produce high switching activity in shift operation appear intensively in some groups of FFs in scan chains. Figure 4.19 shows the state of a scan chain after the last capture for b14 benchmark circuit. Each row denotes the state of the scan chain after the last capture in different tests, and each cell denotes a scan FF in the scan chain (gray: state 1, white: state 0). We can see that a part of FFs that locates at the scan-in side of the scan chain toggles frequently during test, and *high frequency bits* concentrate there. Other parts of FFs in the scan chain almost do not toggle for most of the tests. Since *high frequency bits* have large number of toggles which result in big power [44], controlling the FFs that often produce *high frequency bits* should be effective for the scan-out power reduction.

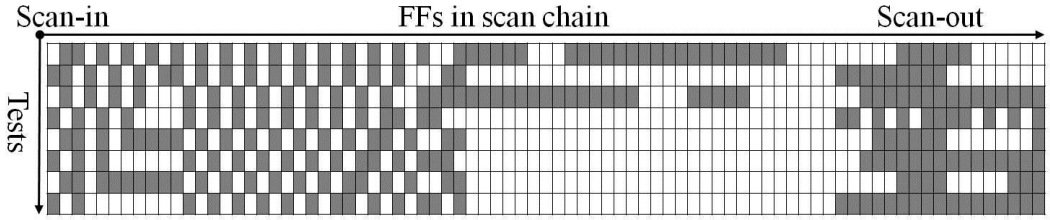
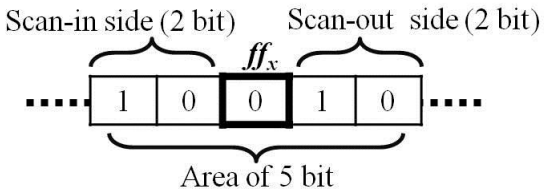


Figure 4.19 State of a scan chain of b14

We use *toggle density* to determine the FFs where *high frequency bits* concentrate in the scan chain. *Toggle density* is defined as the average toggle rate in the area of *N* bit adjacent of each FF at the last capture before scan-out operation. For a FF, the toggle density is computed by dividing the toggle number of the current vector by the maximum toggle number (e.g., vector as "…10101…") in the area of *N* bit adjacent FFs ((*N-1)*/2 bit adjacent FFs on the scan-in side and scan-out side, respectively). An example for toggle density computation is shown in Figure 4.20. Suppose the state of a part of FFs in the scan chain after the last capture is "10010", toggle density of ff_x is 3/4=0.75 as *N* is set to 5 bits. It should be noted that for the FFs on the head of scan-in or scan-out side of a scan chain, toggle density is the average toggle rate in the area of (*N-1)*/2 bit adjacent FFs on the scan-out side or the scan-in side, respectively. As shown in Figure 4.21, ff_1 and ff_x are the first scan FFs from the input and output side of the scan chain, respectively. The toggle density

of ff_1 is the average toggle rate in the area of $\{ff_1, ff_2, ff_3\}$, and toggle density of ff_x is the average toggle rate in the area of $\{ff_{x-2}, ff_{x-1}, ff_x\}$ as N is 5. For ff_2 and ff_{x-1}, toggle density is the average toggle rate in the area of $\{ff_1, ff_2, ff_3, ff_4\}$ and $\{ff_{x-3}, ff_{x-2}, ff_{x-1}, ff_x\}$, respectively. Figure 4.22 shows the toggle density of FFs in a scan chain for b14 circuit computed by logic simulation using 30k test vectors generated by PLPF [44]. Note the state of scan chain in Figure 4.19, FFs that *high frequency bits* are concentrated are well denoted by toggle density.



Toggle number: 3

Maximum toggle number: 4 (for vector "10101")

Toggle density: ¾=0.75

Figure 4.20 Computation of toggle density

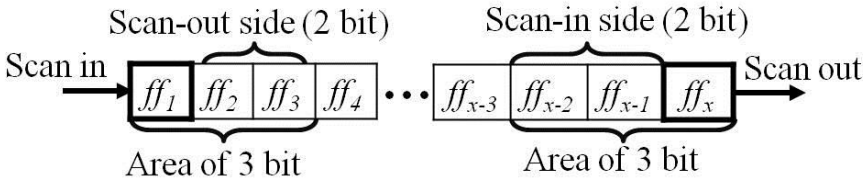


Figure 4.21 Special cases of toggle density computation

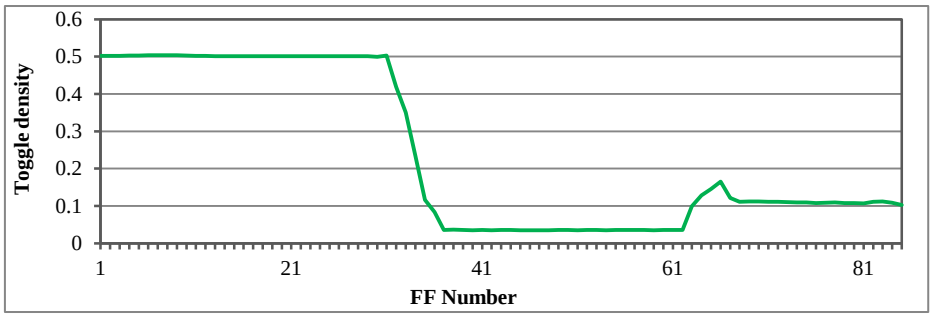


Figure 4.22 Toggle density of a scan chain of b14

It should be noted that elimination of the *high frequency bits* close to the scan-in side of the scan chain should be more effective for power reduction than that of close to the scan-out side. In order to determine FFs that are most effective for power reduction, we compute the *toggle density weighted by location numbers* for each FF by multiply their toggle density by the location number in the scan chain. Here, the location number of FFs is counted from the first scan-out FF (i.e., =1) to the first scan-in FF (i.e., =the scan chain length) in scan chain which shows the contribution to the scan-out power of the scan chain in which the FF belongs, because the number of scan-out clocks of the FF is proportional to its location number.

We consider that select the FFs with lager *toggle density weighted by location numbers* for scan-out control could achieve the most scan-out power reduction.

b. Factors for fault coverage improvement

While fault coverage loss caused by value filling might be relaxed by partial observation of FFs, resulting fault coverage depends on how to select FFs for partial observation. We select the FFs that have large fault coverage contribution for partial observation. A fault simulation-based method could maximize the fault coverage for a specified ratio of FFs. However, it requires huge computation time because sequential fault simulation without fault dropping for pseudo random patterns is time-consuming. In this work, we evaluate the fault coverage contribution of each FF by analyzing the fault observation capacity and fault propagation capacity for each FF using logic cone analysis that is structural circuit analysis. The collapsed single stuck-at fault model is used in the evaluation because it has less number of faults that requires less computation time than transition delay fault model. In addition, circuits that have high stuck-at fault coverage should also have high transition fault testability. Evaluating the fault observation/propagation capacity based on stuck-at fault model should be appropriate for transition fault coverage estimation.

Fault observation capacity of a FF is defined as the number of faults that can be observed at the FF without passing through any another FF. Figure 4.23 shows the logic

cone analysis for fault observation capacity. In the combinational logic of a sequential circuit, each FF has possibility of detection of faults in a specific region (e.g., region *I* for ff_1, region *II* for ff_2) denoted by an input cone. Faults in region *I* can be observed at ff_1 as effects of the faults may be propagated to ff_1. Some faults in region *I* also can be observed at ff_2 due to the region overlapping with *II*. If faults that can only be observed at ff_1 are more than those at ff_2 (i.e., number of faults in the non-overlapped region of *I* is larger than that of *II*), fault observation capacity of ff_1 should be higher than ff_2. On the other hand, if the number of faults that are only observed at ff_1 is small, fault observation capacity of ff_1 should be lower.

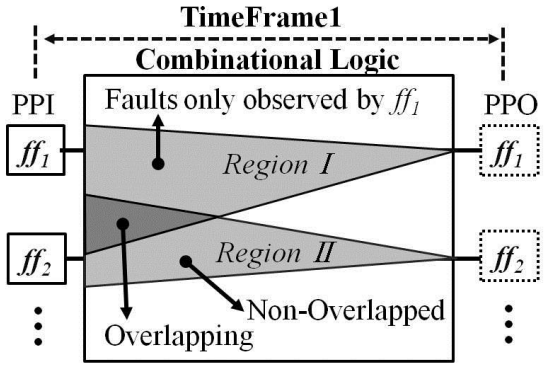


Figure 4.23 Fault observation capacity of FFs

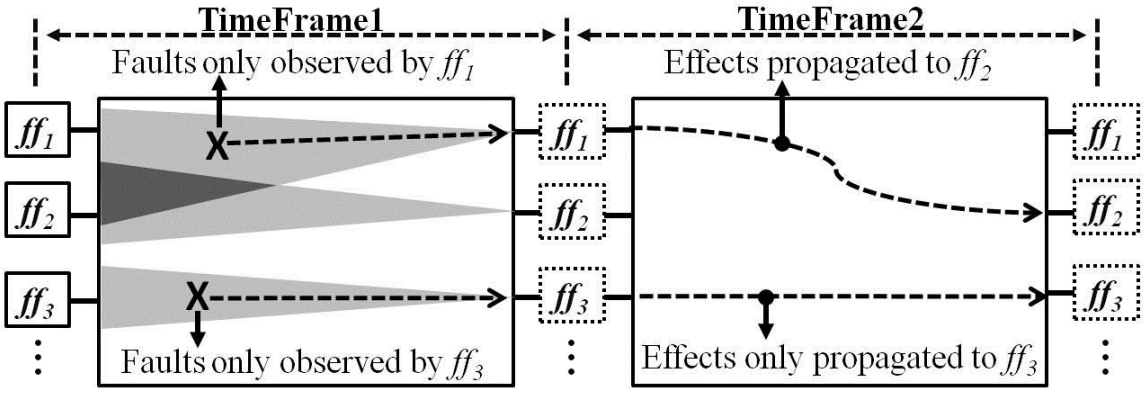


Figure 4.24 Fault propagation capacity of FFs

Fault propagation capacity of a FF is defined as the number of faults that can be propagated to the FF without through any other FFs. Figure 4.24 gives an example for fault propagation capacity analysis. In a sequential circuit, the effect of a fault that is only observed by ff_1 might be propagated to another FF (e.g., ff_2) in the later time frames while propagation paths exist between them. These faults might be detected by another FF in the later capture cycles. However, if no propagation path exists between an FF and any other FFs (e.g., ff_3), the effect of a fault cannot be propagated to any other FFs. Such a fault cannot be detected at any FF except ff_3 even with multiple-cycle test. Therefore if more faults only can be propagated (no propagation paths from a FF to any other FF), fault propagation capacity of the FF should be higher.

FF with high fault propagation capacity and fault observation capacity indicates that more faults will be only observed (propagated) at (to) the FF. These faults should have less chance of detection due to the faulty value masking during multi-capture [46]. Observing the FF during multiple captures brings more chances for these faults detection. Therefore, we consider that select the FFs with higher fault propagation capacity and fault observation capacity for partial observation in multi-cycle test could increase the fault coverage.

c. Control FFs' selection method

In order to achieve larger scan-out power reduction with less fault coverage loss, the control FFs' selection method should take the factors of scan-out power reduction and fault coverage improvement into consideration, simultaneously. In this work, we employ a multiple criteria decision analysis TOPSIS (Technique for Order Preference by Similarity to Ideal Solution) in [50] to select the FFs for scan-out control. The selection procedure is given as follows.

Step1: Create an evaluation matrix consisting of M alternatives (*#of FFs*) and N criteria (*# of factors discussed above*) with the intersection of each alternative and criteria given as t_{ij}, we therefore have a matrix:

$$T = (t_{ij})_{M\times N}, \quad (i = 1,2,\cdots,M,\ j = 1,2,\cdots,N) \tag{4}$$

Step2: Normalize matrix *T* using following formula.

$$R = (r_{ij})_{M\times N},\ \ r_{ij} = t_{ij} \Big/ \sqrt{\sum_i^M t_{ij}^2} \tag{5}$$

Step3: Calculate the weighted normalized decision matrix v_{ij} by (6).

$$v_{ij} = w_j r_{ij},\ \ \sum_j^N w_j = 1 \tag{6}$$

Step4: Determine the worst alternative (v_j^-: minimum value of each factor) and the best alternative (v_j^+: maximum value of each factor), and calculate the distance between the target alternative *i* and the worst condition (S_i^+) and the distance between the alternative *i* and the best condition (S_i^-) by formula (7).

$$S_i^+ = \sqrt{\sum_{j=1}^N (v_{ij} - v_j^+)^2},\ \ S_i^- = \sqrt{\sum_{j=1}^N (v_{ij} - v_j^-)^2} \tag{7}$$

Step5: Calculate the similarity to the worst condition (C_i) for each alternative by (8):

$$C_i = \frac{S_i^-}{S_i^+ + S_i^-} \tag{8}$$

Step6: Select the FFs with large C_i for scan-out control.

4.6.3.4. Value filling methods

For the selected control FFs, we propose three value filling methods for achieving the largest scan-out power reduction.

a. Fixed-value filling

The selected control FFs can be filled with a fixed value (0 or 1) after the last capture. As 0 appears more often than 1 in scan-out empirically, 0-value is used as the fixed value. Figure 4.25 shows the control circuit structure for 0-filling. The capture clock (CLK) and a last capture signal (LCAP) go through a NAND gate and generate a reset signal to set the control FF to 0 before starting scan-shift operation. Here, LCAP is generated by a clock counter just before the last capture cycle in the capture mode of multi-cycle test. The

control structure for 0-filling is very simple and has small impact on area overhead, in addition, without delay penalty.

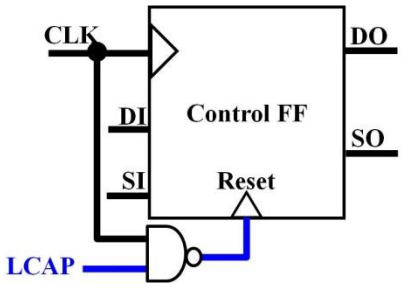


Figure 4.25 Control structure for 0-filling

A fixed-value filling method can achieve great scan-out power reduction when the selected control FFs are consecutive, however, is not so effective for the alternate control FFs. Figure 4.26 shows a part of the distribution of the control FFs (denoted as "C") in the scan chains of b14 circuit using the proposed selection method in sub-section 4.6.3.3. It shows that many selected FFs are consecutive and a part of the control FFs are alternate. Figure 4.27 and Figure 4.28 give examples of 0-filling for consecutive control FFs and alternate control FFs, respectively. In Figure 4.27, suppose that five control FFs (denoted as "C") in the scan chain are consecutive and a response "01110101" is captured at the last capture. After 0-filling, scan-out vector becomes "00000001" so that significant scan-out power reduction can be achieved. In Figure 4.28, we use the same original vector as Figure 4.27 and suppose two alternate control FFs are selected for 0-filling. The scan-out vector becomes "01010101". In this case, 0-filling does not eliminate the *high frequency bits* p_1, instead produces new *high frequency bits* p_2 in the scan-out vector which results in scan-out power increase.

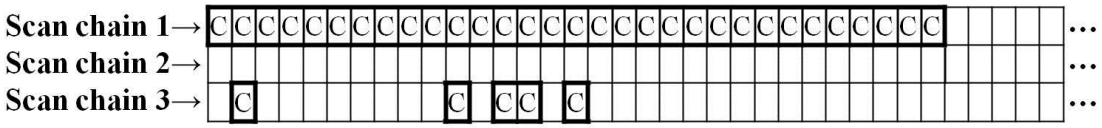


Figure 4.26 Distribution of control FFs in scan chains

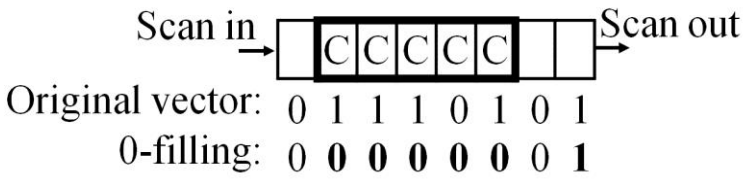


Figure 4.27 0-filling for consecutive control FFs

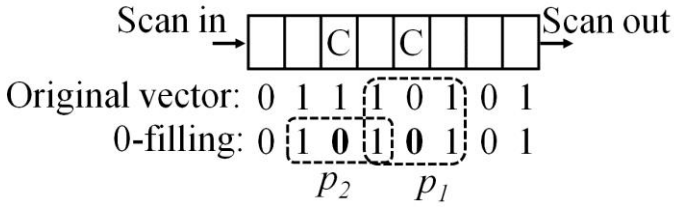


Figure 4.28 0-filling for alternate control FFs

b. Adjacent-value Filling (Ad-filling)

The selected control FFs also can be filled with the value of its adjacent FF that locates in the scan-in direction. Figure 4.29 shows the concept of the adjacent-value filling. Compared with the 0-filling method, the adjacent-value filling works better for the alternate control FFs, because the control FFs are dynamically filled with the values observed from the adjacent FFs so that *high frequency bits* such as "010" or "101" will always be eliminated and the original low frequency bits in scan chain will be unaffected. However, for the consecutive control FFs, the scan-out power reduction of the adjacent-value filling is smaller than that of the 0-filling.

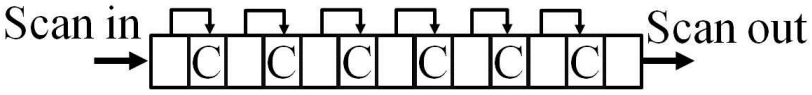


Figure 4.29 Concept of adjacent-value filling

Figure 4.30 and Figure 4.31 give examples, in which the same original vector as Figure 4.27 is used. In Figure 4.30, the position of control FFs is the same as Figure 4.28, but a filling method is changed to the adjacent-value filling. It shows that the *high frequency bits* p_1 in the original scan-out vector is eliminated, and also the original low frequency bits are not affected. In Figure 4.31, suppose five consecutive FFs can be

selected as control FFs for 0-filling method (in the area with thick line as the same as in Figure 4.27). The 2rd and 4th FF of the five consecutive FFs should not be assigned as control FFs in adjacent-value filling method because each control FF requires its adjacent FF for Ad-filling value feeding. When filling the control FFs (denoted as "C") with the adjacent FF values after the last capture, scan-out vector becomes "00110001". Although the *high frequency bits* in the original scan-out vector is eliminated, scan-out power reduction is not so effective than the 0-filling method where the modified scan-out vector is "00000001".

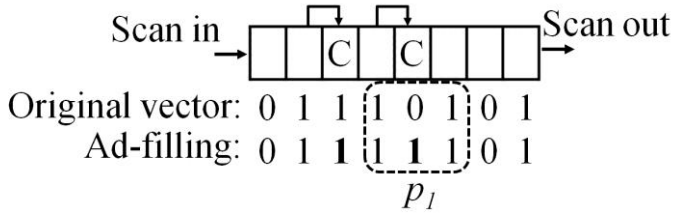


Figure 4.30 Ad-filling for alternate control FFs

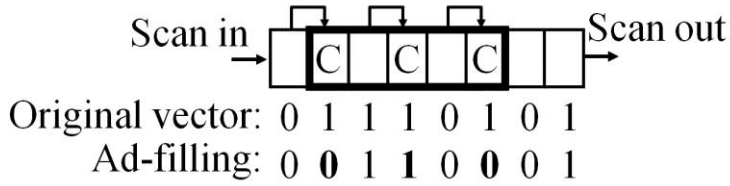


Figure 4.31 Ad-filling for consecutive control FFs

We view the distribution of control FFs in scan chains shown in Figure 4.26. Many selected control FFs are consecutive. If we could feed the consecutive control FFs with the same FF value, it should be more effective for scan-out power reduction. However, control the path delay between FFs would become difficult. Therefore, in this work we feed at most two control FFs by one adjacent FF value synchronously to avoid the affect of path delay as far as possible. An example is given in Figure 4.32, in which two consecutive control FFs are filled with the same value of the adjacent FFs of them synchronously after the last capture. It shows that value filling not only eliminate *high frequency bits* "010" but also eliminate "0110".

Figure 4.33 shows the control circuit structure for the adjacent-value filling. The capture clock (CLK) and the last capture signal (LCAP) generate a control signal through an AND gate to drive two NAND gates. The input DI of the adjacent FF is directly applied to one NAND gate and the other NAND gate though a NOT gate. The output of two NAND gates set the control FF to the same value as its adjacent FF's.

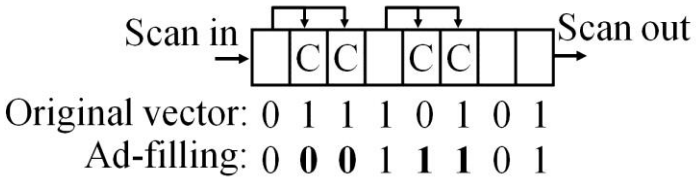


Figure 4.32 Feeding two control FFs by one FF

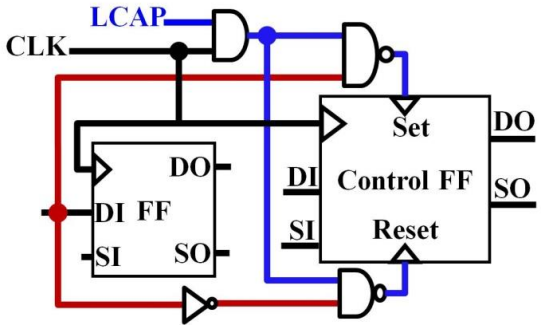


Figure 4.33 Control structure for Ad-filling

c. Hybrid value filling

Consider the merits and demerits of the 0-filling and the adjacent-value filling discussed above, combining with these two methods should be more effective for scan-out power reduction. Therefore, we propose a hybrid value filling method in which a threshold *N* of the length of consecutive control FFs is set to determine the value filling method for the selected control FFs. If a group of FFs are consecutive and the length is larger than the threshold *N*, they will be filled with a fixed value 0, otherwise filled with the adjacent value after the last capture. Here, the threshold *N* is equal to the specified bit number of the area for toggle density computation discussed in sub-section 4.6.3.3 in order to unite control FF selection and value filling. Figure 4.34 gives an example in which the threshold of the length of the consecutive control FFs is 5.

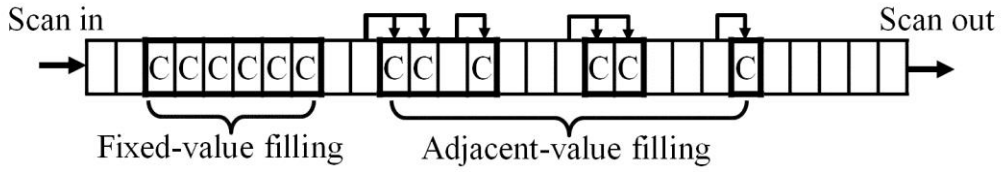


Figure 4.34 Structure of hybrid value filling

4.6.3.5. Experimental results

We evaluated the proposed methods using ITC99 benchmark data. A 16-bit internal type LFSR (characteristic polynomial: $X^{16}+X^{15}+X^{13}+X^{4}+1$) and a pseudo low-pass filter (PLPF) are used to generate 30k test vectors. A parallel scan structure with 100 FF-length of scan-chains is adopted (when # of FFs > 1600, 200 FF-length). Because the capture power of a circuit becomes stabilized at a rather low level as applying many capture cycles (i.e., 20 capture cycles) [45], it suggests that IR-drop-induced yield loss could be avoided. Therefore, a multi-cycle BIST with 10 slow-captures (focused on stuck-at faults) and 10 fast-captures (focused on transition delay faults) are used in the experiments. Experiments are executed in 2 cases. In *Case1*: We focus on achieving the most scan-out power reduction and fault coverage loss was ignored. We select a specified ratio of FFs for scan-out control only according to the *toggle density weighted by location numbers* discussed in sub-section 4.6.3.3 where the bit number of the area for toggle density computation is set to 5. In *Case2*: In order to compensate the fault coverage loss, we execute the experiments using the selection method proposed in sub-section 4.6.3.3 in which the factors of fault coverage improvements are considered simultaneously. To prevent the increase of area overhead, only 20% of FFs are selected for scan-out control. The weighted transition metric discussed in section 4.5 is used for power evaluation and a home-made fault simulator is used to estimate the single stuck-at fault coverage and transition fault coverage.

Scan-out power reduction

Table 4.4 shows the comparison of the 0-filling (*0*), the adjacent-value filling (*Ad*) and the hybrid value filling (*Hd*) for *Case1* and *Case2*. Here, "*SI*" and "*SO*" shows scan-in

power and scan-out power respectively. The results of scan-in power and the original scan-out power without scan-out control (*No-Ctrl*) are shown in the second and third columns for comparison. Peak power reduction is very crucial because greater IR-drop or crosstalk is caused by high peak current. Therefore, we also evaluated the peak scan-out power and peak shift-power in Table 4.5, which are denoted by "*P.Out*" and "*P.Shift*", respectively.

Table 4.4 Scan-in & Scan-out power evaluation

		SO						
			Case1			*Case2*		
Circuit	*SI*	*No-Ctrl*	*0*	*Ad*	*Hd*	*0*	*Ad*	*Hd*
b14	8.24	22.82	7.36	12.24	7.36	8.52	12.61	8.29
b15	7.74	13.06	8.76	9.54	8.46	8.59	10.69	8.59
b20	7.54	16.4	9.12	10.93	8.93	9.68	11.15	9.23
b21	7.54	16.4	9.09	10.77	8.89	9.67	10.99	9.23
b22	7.69	17.37	8.7	11.06	8.49	9.28	10.98	8.98
AVE	7.75	17.21	8.61	10.91	8.42	9.15	11.28	8.86

Table 4.5 Peak power evaluation

			Case1						*Case2*					
	No-Ctrl		*0*		*Ad*		*Hd*		*0*		*Ad*		*Hd*	
Circuit	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*	*P.Out*	*P.Shift*
b14	37.9	21.93	8.92	8.31	18.69	12.32	8.92	8.31	11.59	8.94	18.65	12.3	10.94	8.76
b15	18.23	11.98	9.57	8.7	11.34	9.21	9.17	8.5	11.59	9	14.4	10.07	11.59	9
b20	17.09	12.66	9.92	9.76	11.94	10.28	9.51	9.42	10.44	9.39	12.24	10.29	9.9	9.12
b21	17.18	12.75	9.8	9.79	11.54	10.31	9.36	9.45	10.45	9.36	11.78	10.19	9.86	9.1
b22	27.03	16.73	13	9.71	16.98	11.7	13	9.71	13.97	10.2	16.7	11.56	13.69	10.06
AVE	23.49	15.21	10.24	9.25	14.1	10.76	9.99	9.08	11.61	9.38	14.75	10.88	11.2	9.21

Table 4.4 shows the proposed value filling methods significantly reduce scan-out power. In *Case1*, the 0-filling could reduce the scan-out power of all circuit from the original 17.2% to 8.6% (i.e., 50% reduction) on average. The adjacent-value filling also achieves 37% scan-out power reduction (i.e., from 17.2% to 10.9%) which is smaller than that of 0-filling. This can be explained by Figure 4.30. Many control FFs are consecutive so that scan-out power reduction of the adjacent-value filling is smaller than the 0-filling. The most power reduction is achieved by the hybrid value filling in which the original 17.2% scan-out power is reduced to 8.4% (i.e., 51% reduction) on average. In *Case2*, scan-out

power also has been significantly reduced where the original 17.2% is reduced to 8.9% (i.e., 48% reduction) on average by hybrid value filling. Although the reduced scan-out power is bigger than *Case1* (i.e., 8.4%), the difference is very small (i.e., 0.4%) and fault coverage loss can be compensated which will be discussed later. Table 4.5 shows that the proposed methods significantly reduced the peak scan-out power and peak shift power. The most peak scan-out power reduction is achieved by hybrid value filling that the original 23.5% is reduced to 10.0% (i.e., 57% reduction) in *Case1*. Peak scan-out power also has been reduced from the original 23.5% to 11.2% (i.e., 52% reduction) by hybrid value filling in *Case2*. As the scan-shift power is already at a very low level, nearly 40% reduction of peak shift power is a big advantage, which is not achieved by the conventional methods.

Fault Coverage Estimation

Table 4.6 Fault coverage evaluation

			Case1						*Case2*					
	No-Ctrl		*0*		*Ad*		*Hd*		*0*		*Ad*		*Hd*	
Circuit	*SA*	*TD*	*SA*	*TD*	*SA*	*TD*	*SA*	*TD*	*SA*	*TD*	*SA*	*TD*	*SA*	*TD*
b14	89.27	70.03	89.27	68.13	89.27	68.81	89.27	68.13	89.27	68.98	89.27	69.71	89.27	68.98
b15	92.9	79.65	92.9	79.65	92.9	79.65	92.9	79.65	92.91	79.8	92.91	79.8	92.91	79.8
b20	91.7	82.94	91.7	82.84	91.7	82.87	91.7	82.84	91.7	83.24	91.7	83.27	91.7	83.24
b21	92.4	84.87	92.4	84.77	92.4	84.8	92.4	84.77	92.4	84.89	92.4	84.91	92.4	84.89
b22	91.39	82.52	91.39	82.45	91.39	82.47	91.39	82.45	91.41	83.02	91.41	83.03	91.41	83.02
AVE	91.53	80	91.53	79.57	91.53	79.72	91.53	79.57	91.54	79.99	91.54	80.14	91.54	79.99

Table 4.6 shows the comparison of fault coverage for stuck-at faults (*SA*) and transition delay faults (*TD*) using the proposed value filling methods. In *Case1*, it shows that no stuck-at fault coverage is lost and transition fault coverage loss is little that 0.43%, 0.28% and 0.43% is lost by the 0-filling, the adjacent value filling and the hybrid value filling, respectively. The adjacent value filling have less transition fault coverage loss. This is because it dynamically fills the control FFs with the values observed from the adjacent FFs after the last capture which has less impact on fault detection. In *Case2*, fault coverage of most of circuits is improved and transition fault coverage loss is only 0.01% on average by the hybrid value filling, and 0.42% fault coverage loss in *Case1* is compensated. It is because the control FFs' selection method takes the factors of fault coverage improvements

into consideration simultaneously, so that FFs with high fault coverage contribution are selected for scan-out control and are observed during multi-cycle test which brings more chances for detecting more faults.

Area overhead (investment) Estimation

Discussing hardware overhead of the proposed scan-out power control method on small circuits is not appropriate. We evaluated the area overhead imposed by the proposed value filling methods on a large data model that is based on the SoC model in the Test and Test Equipment chapter of ITRS2009 [51]. We consider that FFs in the SoC model are set-reset type flip-flops for area overhead computing. Since partial observation of FFs requires an additional compactor (consists of an XOR tree and a MISR) which causes 1.93% area overhead as 20% FFs observation [46], it will be included in our evaluation. In order to estimate the area overhead for the hybrid value filling, we evaluate the ratio of FFs that are controlled as the 0-filling and the adjacent value filling in the hybrid value filling for all circuits. Table 4.7 shows the results, where the number of control FFs is shown in the second column. We can see that approximately 97% of control FFs are controlled as the 0-filling and 3% are controlled as the adjacent value filling for the hybrid value filling. Therefore, we use ratio 97% for the 0-filling and ratio 3% for the adjacent value filling to compute the area overhead of the hybrid value filling on the evaluation model. Figure 4.35 shows the estimation result. It shows that the 0-filling increases very little area overhead (i.e., 0.01%) and the adjacent-value filling causes the most increase of area overhead (i.e., 1.4%). The hybrid value filling only causes 0.05% increase of area overhead because most of the selected FFs are controlled by the 0-filling method.

Table 4.7 FFs control in hybrid value filling

Circuit	*# of ctrl FFs*	*Case1*				*Case2*			
		0		*Ad*		*0*		*Ad*	
		#FF	*Ratio (%)*	*#FF*	*Ratio (%)*	*#FF*	*Ratio (%)*	*#FF*	*Ratio (%)*
b14	49	49	100	0	0	48	98	1	2
b15	89	87	97.8	2	2.2	89	100	0	0
b20	98	93	94.9	5	5.1	95	96.9	3	3.1
b21	98	93	94.9	5	5.1	95	96.9	3	3.1
b22	147	146	99.3	1	0.7	139	94.6	8	5.4
AVE	-	-	97.4	-	2.6	-	97.3	-	2.7

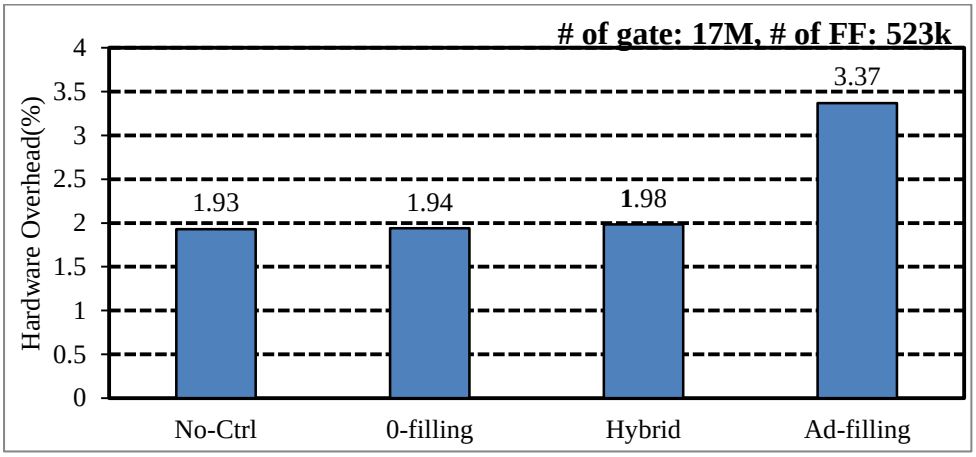


Figure 4.35 Area overhead (investment) estimation

4.7. Conclusions

In this chapter, a novel low-power Logic-BIST scheme, which controls scan-in power, scan-out power and capture power while keeping test coverage at high level, is proposed. The experimental results using ISCAS89 and ITC99 benchmark shows both significant scan-in power reduction rate (the original rate of 50% is reduced to 7~8%) and capture power reduction rate (the original rate of 20% is reduced to 6~7%). With a new scan-out controlling method, the scan-out power can be reduced further (from 17.2% to 8.4%, 51% reduction rate), which is not achieved by the conventional methods.

Moreover, in order to control the test power to the specified rate to accommodate the various test power requirements of design. A scan-in power controlling scheme is also presented. In the current stage, it has the capability to control the toggle rate of scan-in patterns to a specified rate between 6.7% and 50%. In addition, with the proposed scan-out power controlling method, the scan-shift power can be accurately controlled that is remained in the future works.

5. Summary

Advances in semiconductor process technology have resulted in various aging issues in field operation of the Very Large Scale Integration (VLSI) circuit. The effects of aging can cause serious reliability problems. BIST-based field test is a promising way in guaranteeing the reliability of the circuit through detecting the aging-induced faults during circuit operation. However, the limitation of test application time and the high power dissipation during test are the big challenges for the high test quality. In this dissertation, we proposed a new field test application and a new low power BIST scheme to tackle these two problems.

In chapter 3, a new field test application was presented in which a set of pre-generated test patterns used to detect the aging induced faults is partitioned into several subsets, and apply each test subset in a test chance by rotation in the field. The number of test patterns of each subset will not exceed the pattern limit derived from the upper bound of test application time. As the number of test patterns applied to each test is decreased, it would cause test quality loss. In order to maximize the test quality, we proposed the test partitioning algorithms refer to two items: First one, aiming at maximizing fault coverage of each subset obtained by partitioning, the proposed partition method consists of two phases, test pattern partitioning and test pattern replacement. Experimental results showed the effectiveness for the high quality rotating field test. Second one, we focused on shorting the detection time interval of all faults in rotating test to avoid the system failure. A novel method is proposed to estimate the Failure Appearance Probability (FAP) for rotating test, and also proposed a test partitioning approach to minimize the FAP by distributing patterns repeatedly. Experimental results demonstrated the effectiveness of the proposed partitioning approach.

In chapter 4, a new low power logic-BIST scheme which can control scan-in power, scan-out power and capture power while keeping test coverage at high level was proposed. In this scheme, a new circuit called pseudo low-pass filter (PLPF) is developed for scan-in power control, and multi-cycle capture test technique is employed to reduce the capture

power. In order to control scan-out power dissipated by test responses, we proposed a novel method which selected some FFs of scan chains at logic design phase, and filled the selected FFs with proper values before starting scan-shift operation so as to reduce the switching activity associated with scan-out. The experimental results using ISCAS89 and ITC99 benchmark shows significant scan-in power reduction rate (the original rate of 50% is reduced to 7~8%) and capture power reduction rate (the original rate of 20% is reduced to 6~7%). And also, with the scan-out controlling method, the scan-out power can be reduced further (from 17.2% to 8.4%, 51% reduction rate), which is not achieved by the conventional methods. Moreover, in order to control the test power to the specified rate to accommodate the various test power requirements of design. A scan-shift power controlling scheme which has the capability to control the scan-shift toggle rate between 6.7% and 50% is also presented. As various test applications or designs have different power constraints, controlling the test power to a specified rate accommodate to the requirement of design is beneficial not only for test quality but also test cost. Then, this low power Logic-BIST method will be contributive.

REFERENCES

[1] ML.Bushnell,VD.Agrawal, “Essentials of Electronic Testing for Digital, Memory & Mi xd-Signal VLSI Circuits,” Kluwer Academic Publishers, 2000.

[2] Laung-Terng Wang, Cheng-Wen Wu , Xiaoqing Wen, “VLSI Test Principles and Architectures: Design for Testability (Systems on Silicon),” Morgan Kaufmann Publishers Inc., San Francisco, CA, 2006

[3] J. A. Waicukauski, E. Lindbloom, B. K. Rosen and V. S. Iyengar, “Transition Fault Simulation", *IEEE Design and Test of Computers*, Vol. 4, Issue 2, pp. 32-38, 1987.

[4] J. Savir, “Skewed-Load Transition Test: Part I, Calculus", Proc. of ITC, pp. 705-713, September 1992.

[5] J. Savir and S. Patil, “Broad-Side Delay Test", *IEEE Transactions on CAD of Integrated Circuits and Systems*, Vol. 13, Issue 8, pp. 1057-1064, August 1994.

[6] S. Sesuh and D. N. Freeman, On improved diagnosis program, *IEEE Trans. Electron. Comput.*, EC-14(1), 76–79, 1965.

[7] D.B. Armstrong, “A deductive method for simulating faults in logic circuits,” *IEEE Trans. on Computer*, vol.C-21, No. 5, pp. 464-471, 1972.

[8] E. G. Ulrich, T. Baker, “The concurrent simulation of nearly identical digital networks,” *Proc. of Design Automation Workshop,* vol. 6, pp. 145-150, 1973.

[9] W. Wang, et al., “Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” *IEEE Trans. Device and Materials Reliability,* vol.7, no.4, pp.509-517, Dec. 2007.

[10] V. Reddy et al. “Impact of Negative Bias Temperature Instability on Product Parametric Drift,” *International Test Conf.*, pp146-155, 2004.

[11] S. Bhardwaj, et al., “Predictive Modeling of the NBTI Effect for Reliable Design,” *Custom Integrated Circuits Conference*, pp. 189-192, 2006.

[12] E.R. Hnatek. "Integrated circuit quality and reliability," 2nd edition. Marcell Dekker Inc, 1995.

[13] R.-P. Vollertsen, “Burn-In”, *IEEE Intl. Integrated Reliability Workshop*, pp.167-173, 1999.

[14] T. Barrette, V. Bhide, K. De, M. Stover, E. Sugasawara, “Evaluation of Early Failure Screening Methods,” *IEEE Intl. Workshop on IDDQ Testing*, pp.14-17, 1996.

[15] V.Chandra, “Early Life Failures and Burn-in Testing,” *IEEE Workshop on Design for Reliability and Variability*, Panel, pp.1-10, 2008.

[16] M. Nicolaidis, Y. Zorian, “On-line testing for VLSI – A Compendium of Approaches,” *Journal of Electronic Testing: Theory and Applications*, vol.12, pp. 7-20, 1998.

[17] P. D. Hyde, G. Russell, "A Comparative Study of the Design of Synchronous and Asynchronous Self-checking RISC Processors," *IEEE Intl. On-Line Testing Symposium*, pp.89-94, 2004.

[18] M. Benabdenbi, F. Pecheux, E. Faure, "On-line Test and Monitoring of Multi-processor SoCs: A Software-based Approach," *10th Latin American Test Workshop*, pp.1-6, 2009.

[19] Y.-Y. Chen, K.-F. Chen, "Incorporating Signature-monitoring Technique in VLIM Processors," *IEEE Intl. Symposium on Defect and Fault Tolerance in VLSI Systems*, pp.395-402, 2004.

[20] Y. Li, S. Makar, S. Mitra, "CASP: Concurrent Autonomous Chip Self-Test Using Stored Test Patterns," *Design, Automation and Test in Europe,* pp.885-890, 2008.

[21] H. Inoue, Y. Li, S. Mitra, "VAST: Virtualization-Assisted Concurrent Autonomous Self-Test," *IEEE Intl. Conference,* pp.1-10, 2008.

[22] Y. Li, O. Mutlu, D.S. Gardner, S. Mitra, "Concurrent Autonomous Self-test for Uncore Components in System-on-Chips," 28th *IEEE VLSI Test Symposium*, pp.232-237, 2010.

[23] E. Bohl, Th. Lindenkreuz, R. Stephan, "The Fail-stop Controller AE11," *Int'l Test Conf*, pp.1567-1578, 1997.

[24] S. Dikic, L.-J Fritz, D.Dell'Aquia, "BIST and Fault Insertion Re-use in Telecom Systems," *Int'l Test Conf*, pp.1011-1016, 2001.

[25] J. Braden, Q. Lin, B. Smith, "Use of BIST in FIRE™ Servers," *Int'l Test Conf*, pp.1017-1022, 2001.

[26] Y. Sato, Seiji Kajihara, Y. Miura, T. Yoneda, S. Ohtake, M. Inoue, H. Fujiwara, "A Circuit Failure Prediction Mechanism (DART) for High Field Reliability," *8th IEEE Int'l Conf. on ASIC*, pp. 581-584, Oct. 2009.

[27] S. Wang, S. Kajihara, Y. Sato, Xiaoxin Fan, S.M. Reddy, "A Pattern Partitioning Algorithm for Field Test," Proc. 2nd Int'l Workshop on Reliability Aware System Design and Test (RASDAT'11), pp.31-36, Jan. 2011.

[28] J. –S. Chang and C. –S. Lin, "Test Set Compaction for Combinational Circuits," IEEE Trans. on Computer-Aided Design of ICs & Systems.,vol. 14, no. 11, pp. 1370-1378, 1995.

[29] X. Fan, S.M Reddy, S. Wang, S. Kajihara, Y. Sato, "Genetic algorithm based approach for segmented testing," *IEEE/IFIP Int'l Conf. on DSNW*, pp.85-90, 2011.

[30] International Technology Roadmap for Semiconductors, 2011 Edition, http://www.itrs.net/, 2011.

[31] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, Digital Integrated Circuits: A Design Perspective, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2003.

[32] J. Saxena, K. M. Butler, V. B. Jayaram, “A Case Study of IR-drop in Structured At-Speed Testing”, Proc ITC 2003, pp. 1098 – 1104

[33] P. Girard, N. Nicolici and X. Wen, “Power-Aware Testing and Test Strategies for Low Power Devices,” Springer, ISBN 978-1-4419-0927-5, 2010.

[34] S. Gerstendorfer and H.-J. Wunderlich, “Minimized Power Consumption for Scan-Based BIST,” Proc. Int'l Test Conf., pp. 77-84, 1999.

[35] A. Hertwig and H.-J. Wunderlich, “Low Power Serial Built-In Self-Test,” Proc. European Test Workshop, pp. 49-53, 1998.

[36] L. Whetsel, “Adapting Scan Architecture for Low Power Operation,” Proc. Int'l Test Conf., pp. 863-872, 2000.

[37] F. Corno, M. Rebaudengo, M. S. Reorda and M. Violante, “A new BIST architecture for low power circuits,” Proc. European Test Workshop, pp. 160-164, 1999.

[38] P. Girard, L. Guiller, C. Landrault and S. Pravossoudovitch, “A Test Vector Inhibiting Technique for Low Energy BIST Design,” Proc. VLSI Test Symp., pp. 407-412, 1999.

[39] C. Zoellin, H.-J. Wunderlich, N. Maeding and J. Leenstra, “BIST Power Reduction Using Scan-Chain Disable in the Cell Processor,” Proc. Int'l Test Conf., paper 32.3, 2006.

[40] S. Wang and S. K. Gupta, “LT-RTPG: A New Test-Per-Scan BISTTPG for Low Heat Dissipation,” Int. Test Conf., pp. 85-94, 1999.

[41] X. Lin and J. Rajski, “Adaptive Low Shift Power Test Pattern Generator for Logic BIST,” Asian. Test Symp., pp. 355-360, 2010.

[42] M. Filipek, Y. Fukui, H. Iwata, G. Mrugalski, J. Rajiski, M. Takakura and J. Tyszer, “Low Power Decompressor and PRPG with Constant Value Broadcast,” Asian. Test Symp., pp. 84-89, 2011.

[43] J. Rajski, J. Tyszer, G. Mrugalski and B. N.-Dostie, “Test Generator with Preselected Toggling for Low Power Built-In Self-Test,” Proc. VLSI Test Symp., pp.1-6, April. 2012

[44] Y. Sato, S. Wang, T. Kato, K. Miyase and S. Kajihara, “Low Power BIST for Scan-Shift and Capture Power,” Asian. Test Symp., pp.173-178, Nov. 2012.

[45] E. K. Moghaddam, J. Rajski, S. M. Reddy and M. Kassab, “At-Speed Scan Test with Low Switching Activity,” Proc. VLSI Test Symp., pp.177-182, 2010.

[46] Y. Sato, H. Yamaguchi, M. Matsuzono and S. Kajihara, “Multi-Cycle Test with Partial Observation on Scan-Based BIST Structure,” Asian Test Symp., pp. 54-59, 2011.

[47] S. Wang, Y. Sato, K. Miyase and S. Kajihara, "A Scan-Out Power Reduction Method for Multi-cycle BIST," Asian. Test Symp., pp.272-277, Nov. 2012.

[48] Senling WANG, Yasuo SATO, Seiji KAJIHARA, Kohei MIYASE, “Scan-Out Power Reduction for Logic BIST,” IEICE Transactions on Information and Systems, VOL.E96-D, pp.2012-2020, NO.9, 2013. DOI: 10.1587/transinf.E96.D.2012

[49] J. Abraham, U. Goel, and A. kumar, ”Multi-Cycle Sensitizable Transition Delay Faults,” IEEE VLSI Testing Symp., pp. 306-311, April 2006.

[50] K. Yoon, “A Reconciliation among Discrete Compromise Situation,” Journal of Operational” Research Society, Vol.38, No. 2, pp.277-286, 1987.

[51] W. Wang, V. Reddy, A. T. Krishnan, R. Vattikonda, S. Krishnan and Y. Cao, “Compact Modeling and Simulation of Circuit Reliability for 65-nm CMOS Technology,” IEEE Trans. on Device and Material Reliability, Vol.7, No.4, pp.509-507, 2007.

LIST OF PUBLICATIONS

Journal Publications

1. S. Wang, Y. Sato, S. Kajihara, K. Miyase, "Scan-Out Power Reduction for Logic BIST," IEICE Transactions on Information and Systems, VOL.E96-D, pp.2012-2020, NO.9, 2013. DOI: 10.1587/transinf.E96.D.2012

International Conference Publications

1. S. Wang, Y. Sato, K. Miyase, S. Kajihara, "A Scan-Out Power Reduction Method f or Multi-cycle BIST," 2012 IEEE 21st Asian Test Symposium, pp.272-277, Nov. 2 012. DOI: 10.1109/ATS.2012.50

2. Y. Sato, S. Wang T. Kato, K. Miyase, S. Kajihara, "Low Power BIST for Scan-Shift and Capture Power," 2012 IEEE 21st Asian Test Symposium, pp.173-178, Nov. 2012. DOI: 10.1109/ATS.2012.27

International Conference Presentations

3. Xiaoxin Fan, S.M. Reddy, S. Wang, S. Kajihara, Y. Sato, "Genetic Algorithm Based Approach for Segmented Testing," 2011 IEEE/IFIP 41st International Conference on Dependable Systems andNetworks Workshops (DSNW), pp.85-90, June 2011.
10.1109/DSNW.2011.5958841

4. S. Wang, S. Kajihara, Y. Sato, Xiaoxin Fan, S.M. Reddy, "A Pattern Partitioning Algorithm for Field Test," Proc. 2nd Int'l Workshop on Reliability Aware System Design and Test (RASDAT'11), pp.31-36, Jan. 2011.

National Conferences

1. 王　森レイ, 佐藤 康夫, 梶原 誠司, 宮瀬 紘平, “マルチサイクル BIST におけるスキャン出力の電力低減手法,” 電子情報通信学会技術研究報告. 信学技報 112(321), 249-254, 2012-11-26. NAID:40019523514

2. 佐藤 康夫, 王 森レイ, 加藤 隆明, 宮瀬 紘平, 梶原 誠司, “論理 BIST の電力低減手法と評価,” 電子情報通信学会技術研究報告. 信学技報 112(102), 33-38, 2012-06-22. NAID: 110009588442

3. 加藤 隆明, 王 森レイ, 宮瀬 紘平, 佐藤 康夫, 梶原 誠司, “低電力 BIST におけるシフトトグル率低減手法について,” 電子情報通信学会技術研究報告. DC, ディペンダブルコンピューティング 111(435), 25-29, 2012-02-06. NAID: 110009481745

4. S. Wang, Seiji Kajihara, Yasuo Sato, Kouhei Miyase, Xiaoqing Wen, “New Test Partition Approach for Rotating Test with Lower Failure Rate,” FTC 研究会, Jan. 2012.

5. S. Wang, Seiji Kajihara, Yasuo Sato, Kouhei Miyase, Xiaoqing Wen, “Rotating Test and Pattern Partitioning for Field Test,” 情報処理学会研究報告， Vol.2010-SLDM-147 No.8, 2010-11-27.