Spring 2011

On low power test and low power compression techniques

Elham Khayat Moghaddam
University of Iowa

ON LOW POWER TEST AND LOW POWER COMPRESSION TECHNIQUES

by

Elham Khayat Moghaddam

An Abstract

Of a thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Sudhakar M. Reddy

ABSTRACT

With the ever increasing integration capability of semiconductor technology, today's large integrated circuits require an increasing amount of data to test them which increases test time and elevated requirements of tester memory.

At the same time, as VLSI design sizes and their operating frequencies continue to increase, timing-related defects are high proportion of the total chip defects and at-speed test is crucial. DFT techniques are widely used in order to improve the testability of a design. While DFT techniques facilitate generation and application of tests, they may cause the test vectors to contain non-functional states which result in higher switching activities compared to the functional mode of operation. Excessive switching activity causes higher power dissipation as well as higher peak supply currents. Excessive power dissipation may cause hot spots that could cause damage the circuit. Excessive peak supply currents may cause higher IR drops which increase signal propagation delays during test causing yield loss.

Several methods have been proposed to reduce the switching activity in the circuit under test during shift and capture cycles. While these methods reduce switching activity during test and eliminate the abnormal IR drop, circuits may now operate faster on the tester than they would in the actual system. For speed related and high resistance defect mechanisms, this type of undertesting means that the device could be rejected by the systems integrator or by the end consumer and thus increasing the DPPM of the devices. Therefore, it is critical to ensure that the peak switching activity generated during the two functional clock cycles of an at-speed test is as close as possible to the functional switching activity levels specified for the device.

The first part of this dissertation proposes a new method to generate test vectors that mimic functional operation from the switching activity point of view. It uses states obtained by applying a number of functional clock cycles starting from the scan-in state

of a test vector to fill unspecified scan cells in test cubes. Experimental results indicate that for industrial designs, the proposed techniques can reduce the peak capture switching on average by 49% while keeping the quality of test very close to conventional ATPG.

The second part of this dissertation addresses IR-drop and power minimization techniques in embedded deterministic test environment. The proposed technique employs a controller that allows a given scan chain to be driven by either the decompressor or pseudo functional background. Experimental results indicate an average of 36% reduction in peak switching activity during capture using the proposed technique.

In the last part of this dissertation, a new low power test data compression scheme using clock gater circuitry is proposed to simultaneously reduce test data volume and test power by enabling only a subset of the scan chains in each test phase. Since, most of the total power during test is typically in clock tree, by disabling significant portion of clock tree in each test phase, significant reduction in the test power in both combinational logic and clock distribution network are achieved. Using this technique, transitions in the scan chains during both loading of test stimuli and unloading of test responses decrease which will permit increased scan shift frequency and also increase in the number of cores that can be tested in parallel in multi-core designs. The proposed method has the ability of decreasing, in a power aware fashion, the test data volume. Experimental results presented for industrial designs demonstrate that on average reduction factors of 2 and 4 in test data volume and test power are achievable, respectively.

Abstract Approved: _____
Thesis Supervisor

_____
Title and Department

_____
Date

ON LOW POWER TEST AND LOW POWER COMPRESSION TECHNIQUES

by

Elham Khayat Moghaddam

A thesis submitted in partial fulfillment
of the requirements for the Doctor of
Philosophy degree in Electrical and Computer Engineering
in the Graduate College of
The University of Iowa

May 2011

Thesis Supervisor: Professor Sudhakar M. Reddy

Graduate College
The University of Iowa
Iowa City, Iowa

CERTIFICATE OF APPROVAL

_____

PH.D. THESIS

_____

This is to certify that the Ph.D. thesis of

Elham Khayat Moghaddam

has been approved by the Examining Committee
for the thesis requirement for Doctor of Philosophy
degree in Electrical and Computer Engineering at the May 2011 graduation.

Thesis Committee: _____
Sudhakar M. Reddy, Thesis Supervisor

_____
Janusz Rajski, Thesis Advisor

_____
Jon G. Kuhl

_____
David R. Andersen

_____
Xiaodong Wu

_____
Hantao Zhang

To my parent

ACKNOWLEDGMENTS

I welcome this opportunity to express my heartfelt gratitude to my advisor, Professor Sudhakar M. Reddy for the guidance, candor, encouragement, patience and support which he has provided me throughout the years needed to complete this work. I would also like to express my appreciation and sincere thanks to my supervisor, Dr. Janusz Rajski, who guided and encouraged me throughout my studies. His advice and research attitude have provided me with a model for my entire future career.

My sincere thanks are given to my advisory committee members, Professor Jon Kuhl, Professor David Andersen, Professor Xiaodong Wu and Professor Hantao Zhang for their comments and advices on this work.

I want to thank my friends at the Mentor Graphics Corp. and University of Iowa especially Xijiang Lin, Chen Wang, Santiago Remersaro, Mark Kassab and Nilanjan Mukherjee for their sharing and help.

Word cannot express my feelings of gratitude to my parent and my husband for their love, continual encouragement and support throughout this work.

Finally, I would like to dedicate this dissertation to the memory of my father. I am deeply indebted to him for his love, continued support and unwavering faith in me. He will be with me forever, in my heart and memories.

ABSTRACT

With the ever increasing integration capability of semiconductor technology, today's large integrated circuits require an increasing amount of data to test them which increases test time and elevated requirements of tester memory.

At the same time, as VLSI design sizes and their operating frequencies continue to increase, timing-related defects are high proportion of the total chip defects and at-speed test is crucial. DFT techniques are widely used in order to improve the testability of a design. While DFT techniques facilitate generation and application of tests, they may cause the test vectors to contain non-functional states which result in higher switching activities compared to the functional mode of operation. Excessive switching activity causes higher power dissipation as well as higher peak supply currents. Excessive power dissipation may cause hot spots that could cause damage the circuit. Excessive peak supply currents may cause higher IR drops which increase signal propagation delays during test causing yield loss.

Several methods have been proposed to reduce the switching activity in the circuit under test during shift and capture cycles. While these methods reduce switching activity during test and eliminate the abnormal IR drop, circuits may now operate faster on the tester than they would in the actual system. For speed related and high resistance defect mechanisms, this type of undertesting means that the device could be rejected by the systems integrator or by the end consumer and thus increasing the DPPM of the devices. Therefore, it is critical to ensure that the peak switching activity generated during the two functional clock cycles of an at-speed test is as close as possible to the functional switching activity levels specified for the device.

The first part of this dissertation proposes a new method to generate test vectors that mimic functional operation from the switching activity point of view. It uses states obtained by applying a number of functional clock cycles starting from the scan-in state

of a test vector to fill unspecified scan cells in test cubes. Experimental results indicate that for industrial designs, the proposed techniques can reduce the peak capture switching on average by 49% while keeping the quality of test very close to conventional ATPG.

The second part of this dissertation addresses IR-drop and power minimization techniques in embedded deterministic test environment. The proposed technique employs a controller that allows a given scan chain to be driven by either the decompressor or pseudo functional background. Experimental results indicate an average of 36% reduction in peak switching activity during capture using the proposed technique.

In the last part of this dissertation, a new low power test data compression scheme using clock gater circuitry is proposed to simultaneously reduce test data volume and test power by enabling only a subset of the scan chains in each test phase. Since, most of the total power during test is typically in clock tree, by disabling significant portion of clock tree in each test phase, significant reduction in the test power in both combinational logic and clock distribution network are achieved. Using this technique, transitions in the scan chains during both loading of test stimuli and unloading of test responses decrease which will permit increased scan shift frequency and also increase in the number of cores that can be tested in parallel in multi-core designs. The proposed method has the ability of decreasing, in a power aware fashion, the test data volume. Experimental results presented for industrial designs demonstrate that on average reduction factors of 2 and 4 in test data volume and test power are achievable, respectively.

TABLE OF CONTENTS

LIST OF TABLES

Table

LIST OF FIGURES

Figure

CHAPTER 1

INTRODUCTION

According to Moore's law, the number of transistors integrated per square inch on a die has doubled every year and half since the integrated circuit was invented. Also, every few years the size of the transistors employed is shrunken and the frequency of circuits increases. As these trends continue, several new challenges become relevant in the testing of very-large-scale-integrated (VLSI) circuits. With the advance of semiconductor manufacturing technology, the requirements of digital VLSI circuits have led to many challenges during manufacturing test. This is because of the large and complex chips which require a huge amount of test data and dissipate a substantial amount of power during test, resulting in considerable increases in the test cost.

This study addresses the problem of reducing power consumption during test and the problem of keeping the test data volume and test application time moderate. The main objective of this study is to introduce novel techniques that improve the power consumption and test data volume during at-speed test in scan designs and test data compression environments.

This chapter introduces some important concepts in testing of digital VLSI circuits and the importance of minimizing power consumption during test.

1.1 DFT Methods

Deploying reliable integrated circuits depends strongly on testing to eliminate defective circuits caused by the manufacturing process. Manufacturing test is performed after a circuit comes out of the manufacturing line to screen defective parts. Figure 1.1 shows the basic principle of manufacturing testing with its three basic components: circuit under test (CUT), automatic test equipment (ATE), and ATE memory to store test patterns or test vectors and expected responses obtained by automatic test pattern generation (ATPG) tools. As shown Figure 1.1, to test a digital circuit several test

vectors are applied to its inputs. Then, CUT response is analyzed. If the CUT responses match the fault-free responses, then the circuit is considered to be functioning properly. The input test vectors and their responses are stored in an Automatic Test Equipment (ATE) which applies the tests to the CUT and analyze its responses.

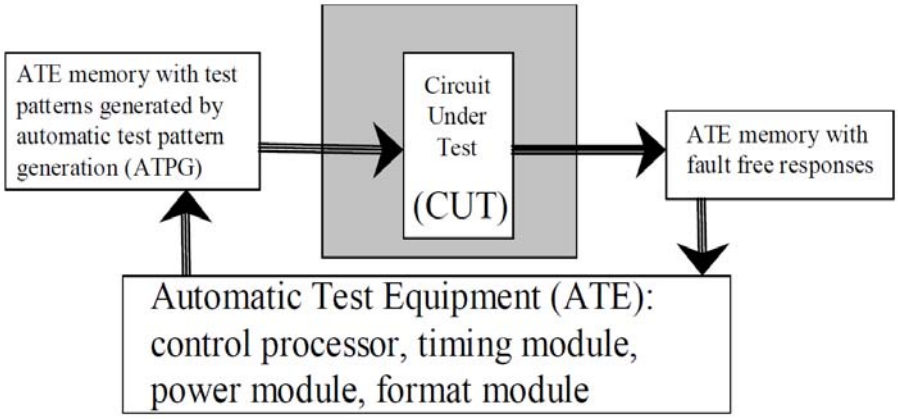


Figure 1.1 Manufacturing test of a circuit [1]

The manufacturing test of a circuit composed of only combinational logic is a relatively easy task. The primary inputs can be set to the desired values and the primary outputs can be observed. However, the test of circuits containing sequential elements such as flip-flops or latches is a more complicated task. Sequential elements in the circuit need to be set to the desired values. In this case, the ATPG needs to create test sequences over many clock cycles to justify desired assignments to circuit inputs that increases run times and complexity of the test generation. This method has been found to be impractical for large circuits.

Design for testability (DFT) refers to design techniques that make products easier to test. DFT techniques improve testability, in increasing controllability and observability of sequential elements by adding test hardware to the CUT. The most popular DFT

techniques for testing VLSI circuits include scan design, Built-In Self-Test (BIST) and test data compression. In this sub-section, we briefly describe each of these techniques.

1.1.1 Scan Design

The most common DFT methodology is scan design [2] where sequential elements are modified to scan cells to obtain controllability and observability for flip-flops. This is performed by adding a test mode to the circuit such that when the circuit is in this mode, all flip-flops functionally form one or more shift registers which is called scan chains. The inputs and outputs of these shift registers are made into primary inputs and primary outputs respectively. Thus using the test mode, all flip-flops can be set to any desired states by shifting appropriate logic values into the shift register. Similarly, the states of flip-flops are observed by shifting out the contents of the scan chains. All flip-flops can be set or observed in a time (in terms of clock periods) that equals the number of flip-flops in the longest scan chain.

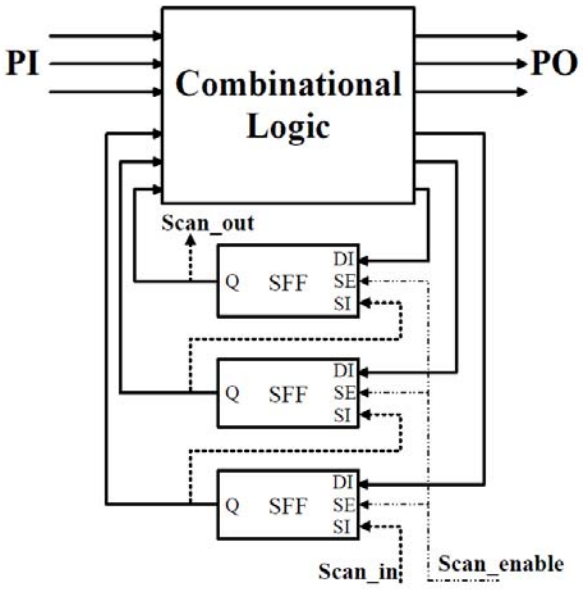


Figure 1.2: Scan based circuit

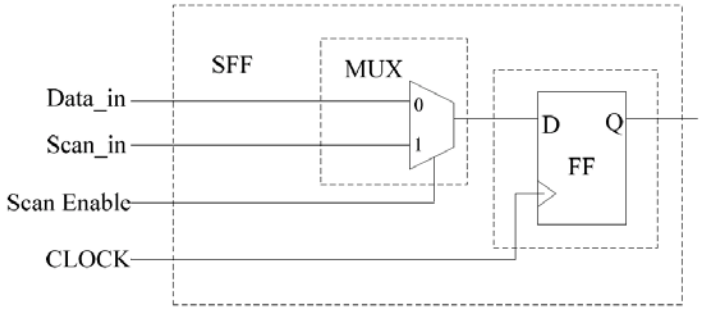


Figure 1.3: Multiplexer based scan cell

Scan design can be further divided into full scan and partial scan designs. The main advantage of full scan (Figure 1.2) is that by modifying all the sequential elements to scan cells it reduces the sequential TPG to combinational TPG. On the other hand, partial scan modifies only a small subset of sequential elements leading to lower test area overhead at the expense of more complex TPG.

There are more than one possible implementations of a scan cell; the most common is shown in Figure 1.3. This scan cell is composed of a D flip-flop and a multiplexer. The multiplexer uses a Scan_enable input to select between the Data_in and the Scan_in. In scan based test, the circuit sequential elements can be set to arbitrary combinations of values by shifting through Scan_in the desired combinations. Also the state of the circuit sequential elements can be observed by shifting out the values stored in scan cells through Scan_out. This enables the test of previously un-testable faults but it may set the circuit into non-functional states.

1.1.2 Built-In Self-Test (BIST)

Built-In Self-Test (BIST) [3, 68] is a technique of designing additional hardware and software features into integrated circuits to allow performing self-testing. BIST is a DFT technique which employs on chip test pattern generator (TPG) and signature analyzer (SA). Figure 1.4 shows a CUT with BIST. When the circuit is in test mode, a test pattern generator (TPG) generates patterns that are loaded into the CUT and a

signature analyzer (SA) examines the CUT response to the test patterns. The signature analyzer has an output to indicate if the circuit has passed or failed the test. In most BIST architectures, linear feedback shift registers (LFSRs) is usually used as a TPG because LFSR can generate sequence of good random property with little area overhead [3]. The typical components of an LFSR are memory elements (latches or flip flops) and exclusive OR (XOR) gates. The signature analyzers (SAs) are commonly constructed from multiple-input signature registers (MISRs). The MISR is basically an LFSR that uses and extra XOR gate at the input of each LFSR stage for compacting the output responses of the CUT into the LFSR during each shift operation.

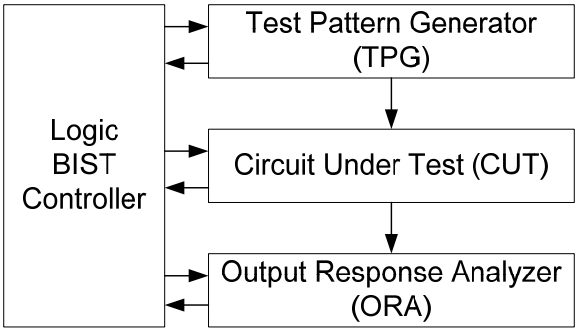


Figure 1.4: High level view of the BIST scheme

BIST is a good solution for testing of critical circuits that have no direct connections to external pins, such as embedded memories used internally by the devices. In BIST, typically up to 95% coverage of stuck-at faults can be achieved provided that test points are employed to address random pattern resistance. Other types of fault models, such as transition or path-delay faults, are not handled efficiently by pseudorandom patterns. In BIST, all test responses have to be known. Unknown values corrupt the signature and, therefore, have to be bounded by additional test logic. Also, deterministic tests are almost always needed to target the remaining random pattern resistance faults.

1.1.3 Test Compression

As devices grew in gate count, scan test data volume and application time grew as well. Test compression techniques have been developed to reduce test data volume and test application time. Test compression techniques are easy to adopt in industry because they are based on scan. Test compression is achieved by adding some additional on-chip hardware before the scan chains to decompress the test stimuli coming from the tester and after the scan chins to compact the response going to the tester (Figure 1.5).

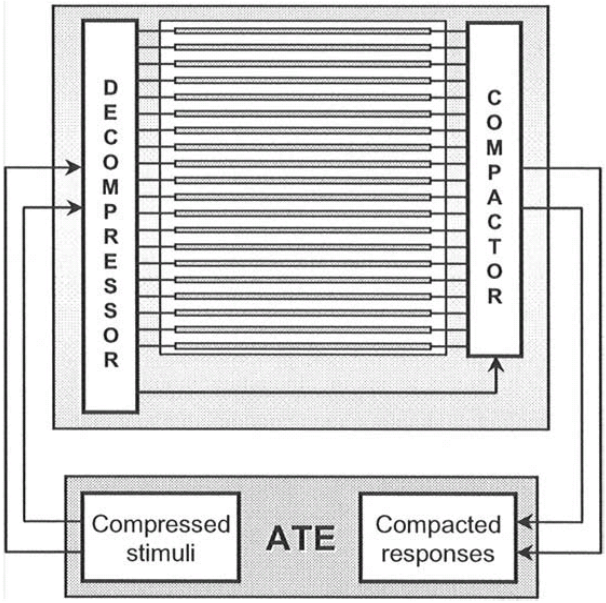


Figure 1.5: Architecture for test compression [4]

Here a technique called Embedded Deterministic Test (EDT) [4] is described. EDT is based on adding a data decompressor at the inputs and response compactor at the outputs of the circuit. The data decompressor is implemented by a ring generator (optimized LFSR) and a phase shifter (Figure 1.6). The phase shifter is necessary to drive a large number of scan chains and to reduce linear dependencies between sequences

entering the scan chains. In addition, the phase shifter's design guarantees balanced use of all memory elements in the ring generator.

The circuits scan chains are divided evenly, if possible, into several shorter chains. For the purpose of producing the desired output, the ring generator is continuously seeded with data. The ratio between the inputs of the ring generator and the outputs of the phase shifter determines the maximal compression possible. Figure 1.6 shows the internal schematic of an on-chip decompressor and Figure 1.7 shows an implementation of a four-output 8-bit decompressor.

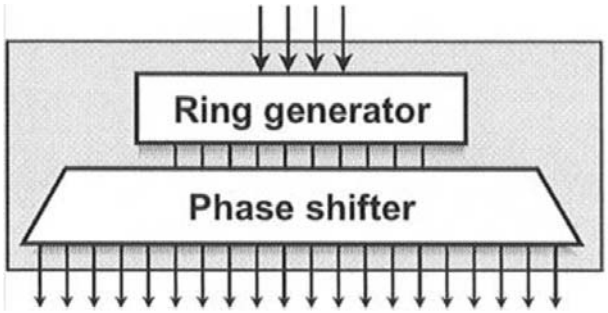


Figure 1.6: On-chip decompressor [4]

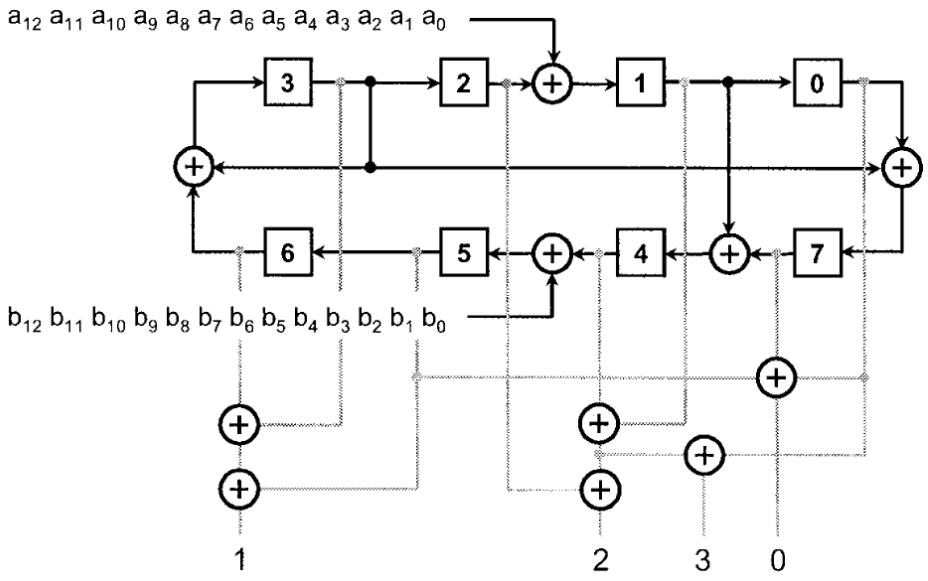


Figure 1.7: Example of four-output 8-bit decompressor [4]

In EDT the compactor consists of an XOR tree and masking logic. Since XORs always propagate fault effects (when no unknown values exist), every scan chain can be observed at the same time using a reduced number of outputs which effectively reduce the response data. Since unknown values can be present in the CUT response to a test, AND-gates are placed at the outputs of every scan chain to selectively block these unknown values.

1.2 Fault Models

In this sub-section some most popular fault models, the stuck-at fault model, the transition fault model and the path delay fault model, will be reviewed.

1.2.1 Stuck-at Fault Model

The stuck-at fault model is the earliest fault model, and still the most common. A stuck-at fault [5] happens when a line in the circuit is stuck at a fixed logic value. To test for stuck-at faults, two steps are involved: one to generate a test vector that excites the fault and the other to propagate the faulty effect to a primary output or a scan flip-flop. Research has shown that stuck-at fault model covers a large percentage of physical defects. However, with the continuously shrinking sizes of the transistors employed in modern designs, increasing clock speed and decreasing power supply voltage, other types of defects not covered by the tests for stuck-at faults are beginning to appear in the CUTs. For this reason, tests for other types of faults are being applied, such as the transition fault model, which is presented next.

1.2.2 Transition Fault Model

Certain types of defects in the manufacturing of the transistors that comprise the circuit gates may cause the gate to have a higher than normal delay. This abnormal delay causes the gate to switch at a lower than normal speed when its inputs change. When this delay is large enough the defect is modeled as a transition delay fault [6].

The transitional delay fault model is the first delay fault model to be developed and is also the simplest. A transition delay fault [6] occurs when the time required for switching outputs from 0 (1) to 1 (0) in the gate, due to a change in the gate's inputs, takes longer than its normal time. If the delay introduced is large enough so that its effects can be seen at least at one of the circuit primary outputs (POs) or captured in a scan cell, the circuit cannot operate at its intended clock speed without having a faulty behavior.

According to the transition-fault model, there are two types of faults possible: a slow-to-rise fault and a slow-to-fall fault. A slow-to-rise fault at any node means that the effect of any transition from 0 to 1 (or 1 to 0 for slow-to-fall) will not reach a primary output or scan flip-flop within the stipulated time. To perform a transition fault test, two test vectors (V_1, V_2) are required. Firs vector initialize the fault and second vector launch a transition and propagate and capture the fault effect in an observation point. Depending on how the transition is launched and captured, there are three transition fault pattern generation methods: launch-off-shift or skewed load test method [7], launch-off-capture or broadside test method [8] and enhanced scan [9] which are briefly explained below.

1.2.2.1 Launch-off-Shift Method (LOS)

In launch-off-shift (LOS) approach [7], the transition at the gate output is launched in the last shift cycle during the shift operation. Figure 1.8 shows the launch-off-shift method waveform. The launch clock is a part of the shift operation and is immediately followed by a fast capture pulse. The scan enable (SEN) is high during the last shift and must go low to enable response capture at the capture clock edge. Since the capture clock is applied at the full system clock speed after the launch clock, the scan enable signal, which typically drives all scan flip-flops in the CUT, should also switch in the full system clock cycle. This requires the scan enable signal to be driven by a

sophisticated buffer tree or strong clock buffer. Such a design requirement is often too costly to meet.

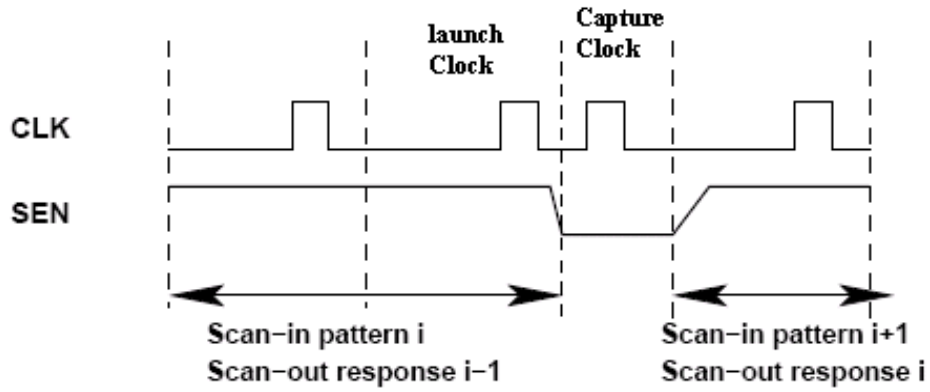


Figure 1.8: Waveform for Launch-off-Shift delay test

1.2.2.2 Launch-off-Capture Method (LOC)

In the launch-off-capture approach [8], the launch cycle is separated from the shift operation. Figure 1.9 shows the waveforms of the launch-off-capture (LOC) method. At the end of scan-in (shift mode), pattern V_1 is applied and CUT is set to an initialized state. A pair of at-speed clock pulses is applied to launch and capture the transition at the target gate terminal. This relaxes the at-speed constraint on the scan enable (SEN) signal and dead cycles are added after the last shift to provide enough time for the SEN signal to settle low. The launch-off-shift method is more preferable based on the ATPG complexity and pattern count compared to LOC method. The LOC technique is based on a sequential ATPG algorithm, while the LOS method uses a combinational ATPG algorithm. This will increase the test pattern generation time in case of LOC, and also, a high fault coverage cannot be guaranteed due to the correlation between the two patterns, V_1 and V_2; note that V_2 is the functional response of pattern V_1. The main concern about the LOS is its requirement to at-speed SEN signal.

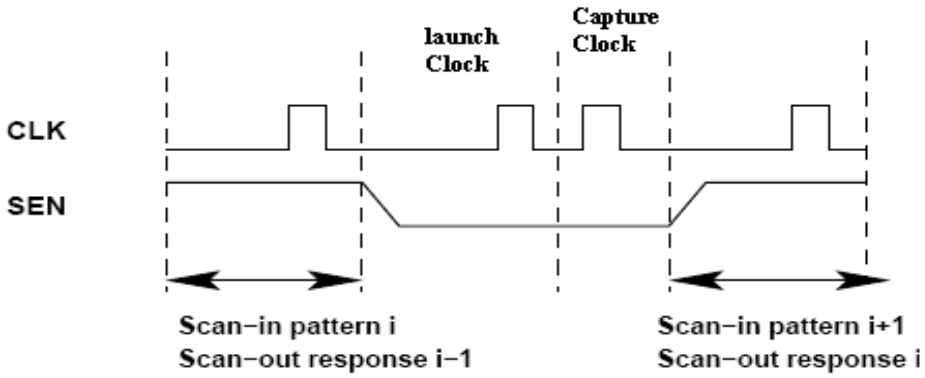


Figure 1.9: Waveform for Launch-off-Capture delay test

1.2.2.3 Enhanced Scan Method

In the enhanced scan approach [9], two vectors V_1 and V_2 are shifted into the scan flip–flops simultaneously in order to initialize and propagate the fault. The drawback on enhanced scan is that it needs hold-scan flip–flops which make it unattractive for application specific integrated circuit (ASIC) designs.

1.2.3 Path Delay Fault Model

The path delay fault model [10] focuses on the testing of a set of predefined structural paths in order to detect the accumulated delays along these paths. A path is defined as an ordered set of gates $\{g_o, g_1, \ldots g_n\}$, where g_o and g_n are primary input and primary output, respectively and gate g_i is an input to gate $g_{i+1\ (0 < i < n-1))}$. A delay defect on a path can be observed by propagating a transition through the path. The path delay fault model takes the sum of all delays along a path into accounts, while the transition fault model accounts for localized faults (delays) at the inputs and outputs of each gate. Test for path delay fault model can detect small distributed delay defects caused by statistical process variations. A major limitation of this fault model is that the number of paths in the circuit can be very large. Therefore testing all path delay fault in the circuit is not practical.

1.3 Test Power Issues

With the development of portable devices and wireless communication systems, design for low power VLSI circuits has become an important issue. Minimizing power dissipation in VLSI circuits increases battery lifetime and the reliability of the circuit [11-12]. The power dissipation of complementary metal oxide semiconductors (CMOS) circuits can be generally divided into two main categories: static power and dynamic power [11].

- The static power dissipation: the power is dissipated by a gate when it is inactive. A significant fraction of static power is caused by the reduced threshold voltage used in modern the CMOS technology that prevents the gate from completely turning off, but causing sources to drain leakage. All components of static power dissipation have a minor contribution to the total power dissipation.
- The dynamic power dissipation: the power dissipation is predominantly caused by the current required for charging/discharging the load capacitance through the pull-up/ pull-down networks (Figure 1.10).

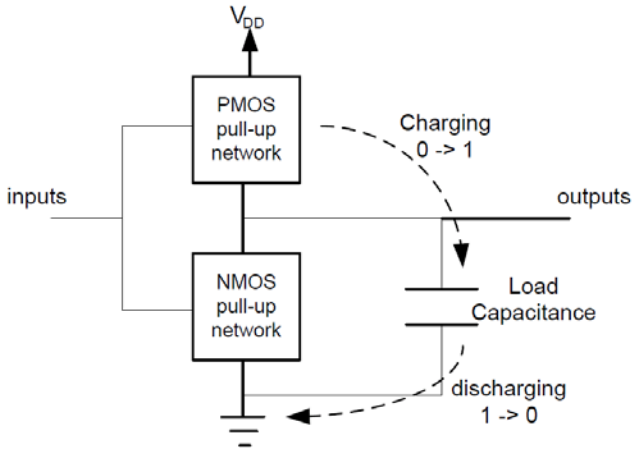


Figure 1.10: Power dissipation in CMOS circuits [12]

Dynamic power dissipation is the dominant source of power dissipation in CMOS circuits. The energy consumed from the source for changing the output from 0 to 1 during the time interval [0, T] is

$$E_{0\to1} = V_{DD}\int_0^T i(t)dt = V_{DD}\int_0^{V_{DD}} C_L dV = C_L {V_{DD}}^2$$

where C_L is the load capacitance. Only half of this energy is stored in the capacitor, while the other half is converted into heat [12]. In the same manner, when the output switches from 1 to 0 the capacitor discharges through the pull down network and the same amount of energy is dissipated as a heat. Therefore, the rate at which the outputs change their value determines the average dynamic power dissipation.

Previous studies also confirm considerably higher switching activity during testing compared to that during functional operation which may result in higher dynamic power dissipation during the test. This can decrease the reliability of the circuit under test due to excessive temperature and current density which cannot be tolerated by circuits designed using power minimization techniques. On the other hand, high switching activity during test application 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, which tend to increase signal propagation delays of affected gates. Increased gate delays during test can erroneously change the logic state of circuit lines leading to incorrect operation of circuit gates which may cause some good dies to fail the test and consequently yield loss [13]. Therefore, addressing the problems associated with testing low power VLSI circuits has become an important issue.

In the meantime, as feature size shrinks into the deep-submicron (DSM) scale and circuit speed increases, due to the timing-related defects a higher number of chips failure may be observed in the circuit. As a result, at-speed scan testing, which captures the test response of the scan design at the rated clock-speed, is becoming mandatory to ensure high product quality [14, 15]. Despite the importance of at-speed scan testing, it is being

severely challenged by excessive power supply current during test and test-induced yield loss. It has been reported that a 10% drop in power supply voltage may increase path delay by 30% [16]. Consequently, increased gate delays during test may cause good chips to fail at-speed tests resulting in test-induced yield loss [17-19]. Therefore, it is important to reduce IR-drop during at-speed scan testing, in order to avoid IR-drop-induced yield loss.

1.4 Organization and Contributions of This Work

This research presents new techniques for reducing power supply noise and IR-drop during at-speed delay test in VLSI circuits. Also it presents a new low power compression technique using clock gating circuitry to achieve better test data volume and test power. The rest of the dissertation is organized as follows. Motivation for low power testing and a review of previously reported approaches for minimizing power dissipation and overtesting during test application is provided in Chapter 2.

Chapter 3 introduces a new technique [20] for IR-drop and overtesting during test application in scan sequential circuits with no penalty in area overhead, test efficiency, performance, or volume of test data when compared to standard scan method described in Section 1.2. It uses states obtained by applying a number of functional clock cycles starting from the scan-in state of a test vector to fill unspecified scan cells in test cubes. Since the number of specified values in test cubes is small, switching activity caused by using functional backgrounds is very close to the switching activity caused by functional operation.

Chapter 4 introduces a new low power techniques [23] in embedded deterministic test environment [4] and shows how with low overhead in test area and volume of test data, and no penalty in test efficiency, significant reduction in power dissipation and ovetesting during test application in large sequential circuits is achieved.

Chapter 5 introduces a new low power test data compression technique [47] using systematically clock gater circuitry to simultaneously reduce test data volume and test power. Using this technique, transitions in the scan chains during both loading test stimuli and unloading test responses decrease which results in the acceleration of the speed of shifting and an increase in the number of cores that can be tested in parallel.

Chapter 6 outlines a summary of the methods proposed in this study. Recommendations for future research into this subject are also provided.

CHAPTER 2

MOTIVATION AND PREVIOUS WORKS

With the development of wireless communication technology and high-performance portable computing devices, design for low power has become a major objective in system design. Power dissipation is not only a critical parameter in the design procedure, but also during manufacturing test as the system may consume much more power during test than during normal operation [14-19]. Thus, low power test of digital VLSI systems has become a major issue of research in recent years. On the other hand, as was discussed in the previous chapter, at-speed tests are more important in order to maintain sufficient level of outgoing quality [14, 73-74]. However, generating at-speed test patterns using functional patterns is very complex and time consuming. DFT techniques are widely used in order to improve the testability of a design. While DFT techniques increase the number of testable faults, they may cause the test vectors contain non-functional states which result in higher switching activities compared to the normal mode of operation.

In this chapter, the need for low power testing to preserve high circuit yield and reliability will be discussed and justified in Section 2.1. In Section 2.2 and Section 2.3, an overview of the solutions proposed in the literature to reduce power dissipations and IR-drop during test application in scan design and test data compression techniques is provided, respectively.

2.1 Motivation for Low Power Testing

Considerable research on low power design and testability of VLSI circuits have been shown that the power consumed in test mode of operation is often much higher than the power consumed in normal mode of operation due to the high switching activity in the nodes of the circuit under test [16-19, 21 and 22]. The main reasons for high switching activity in test mode [17, 19] are as follows:

- Scan based tests eliminate various functional constrain imposed on the sequential circuits by changing sequential circuits to combinational circuits. Given a circuit with k scan cells, there are 2^k possible states that some of which are not functional or legal states. For example in a BCD counter with 4 scan cells, only ten (0-9) of sixteen states are legal states. Non-functional characteristics of test stimuli and test responses in scan based test causes higher switching activity at circuit nodes during test. (We use the words legal states, reachable states, and functional states interchangeably to mean the states of the sequential circuit under test that can be reached during normal/functional operation of the circuit. By unreachable or illegal or non-functional states, we mean the states that cannot occur during normal functional operation.)
- Modern ATPG tools tend to generate test patterns with a high toggle rate in order to reduce pattern count which may lead to a shorter test application time. Thus, the node switching activity in the CUT in the test mode is much higher than that in normal mode.
- In the test mode, parallel testing is often used to reduce the test application time. This parallelism inevitably increases switching activity during the test.
- The design for testability circuitry inserted in the circuit will probably be idle during normal mode but may be used intensively during test mode which may result in a considerable increase in switching activity during the test.
- The correlation between the successive functional input vectors applied to a given circuit during normal operation is generally very high. For instance, in a speech signal processing circuit, the input vectors behave in a predictable manner, with the least significant bits more likely to change than the most significant bits. In contrast, there is no definite correlation between successive test patterns generated

by an ATPG tool during scan testing. This will increase the switching activity in the circuit during test.

The excessive switching activity causes many problems in circuit-under-test. Higher switching activity causes higher power dissipation as well as higher peak supply currents. Excessive power dissipation may cause hot spot that could cause a device malfunction, shorter product lifetime, or even permanent damage of the circuit. Excessive peak supply currents may cause higher IR drops, which tend to increase signal propagation delays of affected gates. Increased gate delays during test may cause good chips to fail tests causing yield loss [13, 19]. Therefore, in order to avoid IR-drop-induced yield loss, it is important to reduce IR-drop during test.

Several methods have been proposed to reduce the peak switching activity during shift and capture cycles in scan design, logic BIST and test data compression techniques. Following, earlier methods to reduce switching activity during shift and capture cycles will be discussed.

2.2 Low Power Testing in ATPGs

Several methods have been proposed to reduce the switching activity in the circuit on the test (CUT) during shift and capture cycle. These techniques can be classified into three main categories: ATPG techniques, DFT modification techniques and functional/pseudo functional techniques. Following, a brief review of some of these techniques are presented.

2.2.1 ATPG Techniques

In conventional ATPG, each don't care bit (X) in a test cube is filled with 0 or 1 randomly; the resulting fully specified test cube (called test pattern) is then fault simulated to confirm the detection of all targeted faults and additional faults. While the state-of-the-art dynamic and static test pattern compaction techniques have been

extensively used to reduce pattern count, the number of unspecified bits in a generated test cube in ATPG remains high. This provides a great opportunity that can be exploited for power minimization during scan testing.

In [24], a low-power ATPG method was proposed to minimize difference between before-capture and after-capture output values of scan cells. This is achieved by introducing a capture conflict (C-conflict) in addition to the conventional detection conflict (D-conflict). In conventional PODEM-based ATPG, backtrack of decisions occurs when a detection conflict occurs, i.e. when there are no paths containing unspecified values between the gates on the D-frontier and any PO (primary output) or PPO (pseudo-primary output). Here a new backtrack condition is introduced; PODEM will also backtrack when the PPO and its correspondent scan cell input, PPI (pseudo-primary input), have different values, which is called a C-conflict. However, backtracking for C-conflict may make fault detection impossible. In this case, the backtracking for C-conflict is preserved, and the transition at the scan cell is tolerated because the primary goal is fault detection. This technique has the advantage that no modifications to the CUT are necessary (as the following software techniques) and that the pattern count is only slightly increased, while it has the disadvantages that it is necessary to modify the existing ATPG and relatively high run times, which makes it non-scalable to industrial circuits.

Another technique to reduce test power is the use of power-aware X-filling heuristics. These heuristics do not modify the overall ATPG process but assign appropriate values to don't care bits of deterministic test cube generated by ATPG algorithm to minimize the number of transitions in the scan cells. By reducing the number of transitions in the scan cells during scan shifting, the overall switching activity in the CUT is reduced, thus the power consumption during test is minimized.

The adjacent fill (AF) [25] method is one of technique developed based on X-filling solution in which a set of unspecified bits in each test is identified by the AF and

then, filled in the following way: whenever there is an unspecified bit, AF fills it with the previous specified logic value, and if there are unspecified bits at the beginning of the test cube, the AF will fill them with the first specified value. For example, if XX1XX100 is a test cube for a given set of faults, adjacent fill will create the test vector 11111100 which has only one transition in it. This method has the advantage that no modification on ATPG algorithms is needed. Results on benchmark circuits have shown that both average and peak power consumption during test can be efficiently minimized with the adjacent fill technique [26].

Another X-filling technique that is widely used to reduce the switching activity is zero-fill [13]. Zero-fill increases 0′s in a test vector, which helps reduce switching activity especially during scan shift, but tends to increase pattern counts significantly.

In the context of at-speed scan testing, some X-filling solutions have also been described to reduce power during the test cycle and thus avoid IR-drop-induced yield loss [27, 28 and 33].

In [31], a Progressive Match Filling (PMF) technique was proposed to reduce the peak current and power dissipation during the fast capture cycle in broadside delay fault testing. Figure 2.1 gives the flow of the progressive match filling procedure [31]. In this method, the unspecified values in the generated initialization vector are filled such that the resulting launch vector (V_2) will be in its minimal Hamming distance from the initialization vector (V_1). If a primary input is assigned a specified value *a* in V_1 (V_2) and there is an X value in the corresponding position in V_2 (V_1), then, it assigns the value *a* to this primary input in V_2 (V_1). If a primary input is unspecified in both V_1 and V_2, then, it assigns the same randomly chosen binary value to this primary input in both V_1 and V_2. Then, it searches for corresponding state inputs which have unspecified values in pseudo primary input (PPI) of V_1 and specified values in PPI of V_2. If PPI of V_2 has a specified value and the corresponding PPI in the V_1 has an unspecified value X, then it fills the X in V_1 with the specified value in V_2. Then, it performs logic simulation using the updated

initialization vector V_1 to find more newly specified values in the next state outputs which can be used to fill the unspecified values in V_1.

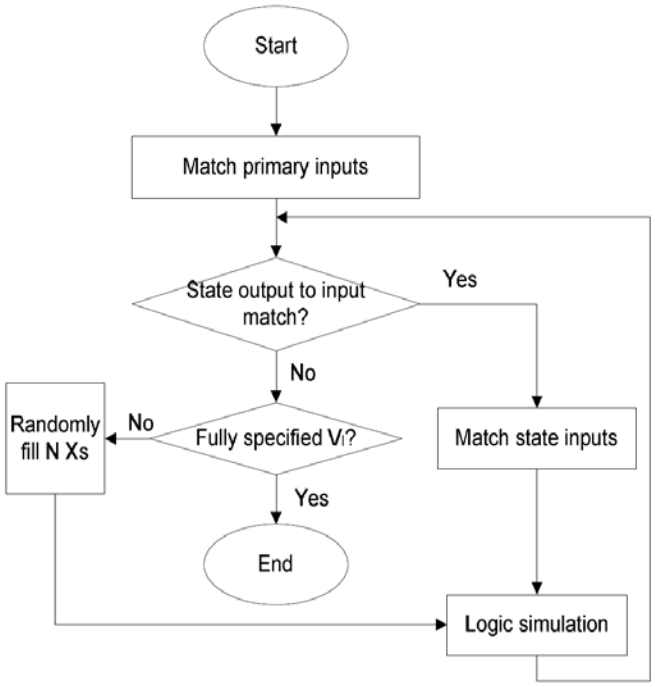


Figure 2.1: Flow of Progressive Match Filling [31]

Another technique for reducing capture power in transition faults is presented in [27]. This technique, called low-capture-power (LCP), attempts to reduce the hamming distance between the shifted in test vector and the CUT response to it by assigning 0s and 1s to unspecified bits in a test cube. LCP fill divides filling each scan cell into four cases depending on if the pseudo primary input and/or pseudo primary output is specified or not. If in a scan cell PPI and the PPO bits are both specified nothing can be done. If in a scan cell PPI bit is unspecified and the corresponding PPO bit is specified it assigns the PPO value to the PPI bit. This can be done because the PPI values are shifted into the CUT. If the PPI bit is specified and the PPO bit is unspecified it tries to justify in the PPO the value of PPI value. If both are unspecified, LCP tries to assign a value to the PPI that

will produce the same value in the PPO. When the filling is done, i.e. no more unspecified bits remain in the test cube, the resulting test pattern is fault simulated for newly detected faults. Then a new set of faults is targeted for detection and so on. The LCP technique leads to considerable reduction in peak and average WSA of LOC tests compared to random fill of the unspecified values in the test cubes. This method has the advantage of not requiring any modifications to the CUT and existing ATPGs. However the run time of this procedure could be potentially high. The reasons for this are the repeated simulations of incrementally updated test cubes and use of implications and line justification steps as part of the procedure so it is non-scalable to industrial designs. LCP fill increases pattern count considerably because the filling procedure detects fewer faults than the normal random fill method.

The preferred fill (PF) technique proposed in [28], attempts to reduce the Hamming distance between the initialized and captured patterns by using a procedure based on signal probabilities. PF fills all unspecified bits in the cube at once with predetermined values (called preferred values) that will produce less WSA than random fill. To determine these values, PF calculates the signal probabilities of each circuit PPO. Figure 2.2 shows an example of signal probability calculation for a simple circuit.

It uses simple procedures to compute signal probabilities that ignore correlation between gate inputs together. So it assigns the probability of 0.5 to logic value 0 and 1 for each PI and PPI, and calculates every signal probability until the PPOs are reached. If the PPO has larger signal probability of being at 1 the preferred value for the correspondent PPI is 1. The same happens at 0, and if they have equal probabilities the preferred value is unspecified and the position is filled at random. The method can be applied to both stuck-at and transition faults. This method increases pattern counts significantly because a constant pre-determined value is used to fill unspecified bits in test cubes and so fortuitous detection reduces.

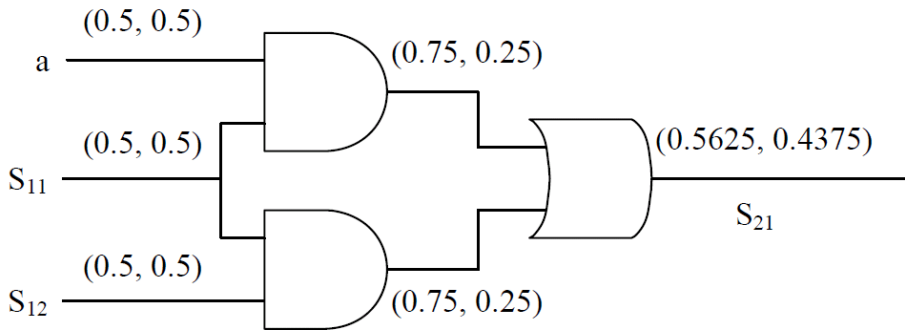


Figure 2.2: Signal probability calculations [28]

In order not to increase test pattern counts, a given set of specified test patterns are relaxed to obtain tests with unspecified values and then the unspecified values are filled using preferred fill [27] or LCP [28] values. In this method [29], after generating a test set in the conventional way the post processing applies a procedure called relaxation. Relaxation consists of identifying don't care positions on each test pattern and unspecifying them. After relaxing every pattern the filling method in [27] or [28] or combination of both methods is used to obtain a low capture power test set. The advantage over the previous methods is that the pattern count remains the same (or lower) when compared to the original test set, at the cost of increased run times, because relaxation is a time consuming procedure.

In [30], authors proposed a hybrid technique employing two methods; justification based method proposed in [27] and probability based method proposed in [28]. This technique, called JP-filling, is both effective and scalable in minimizing launch cycle power supply noise. JP-filling tries to reduce the Hamming distance between the pattern itself and its output response. The result is reduced flip-flop switching activity in the launch cycle, which indirectly brings down the switching activity during launch cycle. The first operation is conducted for PPI-PPO bit-pairs of the form <logic value, X>, but not for any PPI-PPO bit-pair of the form <X, X> for which time-consuming, multiple passes of justification are needed. This is to achieve higher scalability. The second

operation is conducted for PPI-PPO bit-pairs of the form <X, X>, but in multiple passes. That is, the logic value for a PPI X-bit is determined only if its corresponding PPO X-bit has significantly different 0 and 1 probabilities; otherwise, signal probabilities are re-calculated in the next pass. This is to achieve higher effectiveness through improved accuracy in logic value determination. Figure 2.3 gives a JP-filling example. The circled PPO's are the ones that become specified after event-driven simulation.

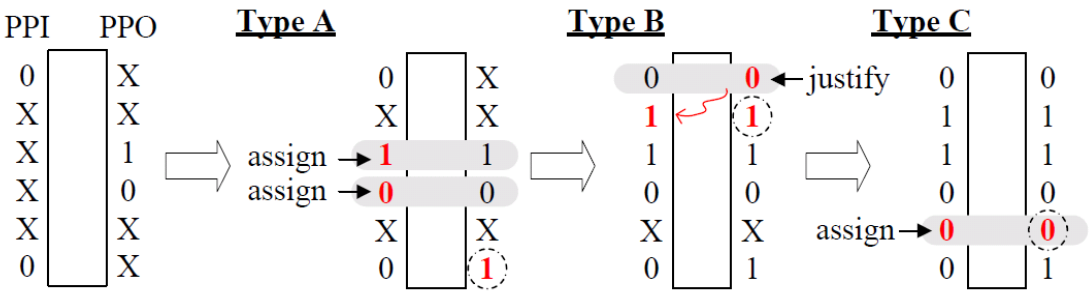


Figure 2.3: A JP-filling example

2.2.2 DFT Modification Techniques

In [32], the authors keep the peak power below a specified limit by inserting a few test points at selected scan cell outputs. Given a set of test patterns, logic simulation is carried out to identify the cycles in which peak power violations occur. Those cycles are called violating cycles. By using integer linear programming (ILP) techniques, the optimization problem is solved to select as few test points as possible such that all violating cycles can be eliminated. The disadvantages of this method are inserting test points that are test set dependent. Therefore, violating cycles may not be eliminated when the test set is changed. In addition, solving an ILP problem with a constraint matrix in the size of $vc \times 2sc$ is not applicable to large industrial circuits, where vc is the total number of violating cycles, and sc is the total number of scan cells.

In [33] authors used multiplexer at the output of the scan cells which hold the previous states of the scan register during shifting and, thus, prevent activity in combinational logic. As shown in Figure 2.4 the output of the scan cell is connected to one of the data inputs of the MUX, while the output of the MUX has a feedback connection to the other data input and also feeds the combinational gates that the scan cell will usually feed. In this way when SE is at 1 the MUX holds its previous value and when SE is at 0 the circuit assumes normal functioning. Another method for reduction in combinational power using blocking is to use a scan-hold circuit as a sequential element.

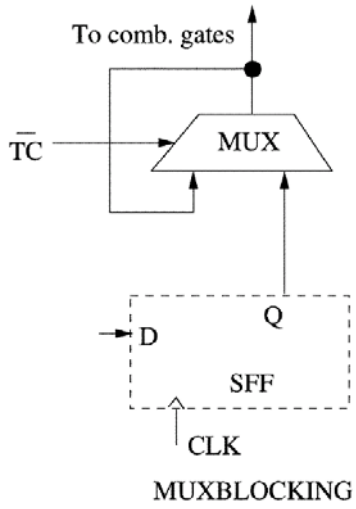


Figure 2.4: Mux as a blocking logic

In [35], authors proposed to use the existing clock gating logic in the design to hold the clock controlling the flops that are not required to switch within a given test pattern (Figure 2.5). During scan shift the scan enable signal overrides the clock gate and allows all flops to clock concurrently. During the capture cycle the scan enable is inactive and the clock gate is controlled from its functional enable. If the functional enable can be selectively disabled in conjunction with the ATPG algorithm then capture power can be reduced without impacting the ability to capture fault effects.

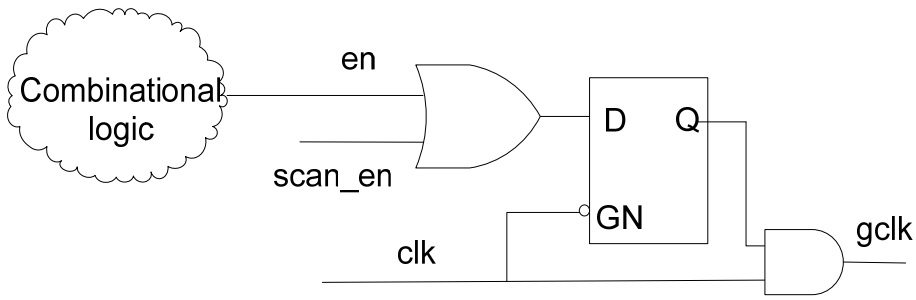


Figure 2.5: Clock gate cell operation during test

A scan cells reordering technique is proposed in [36] to reduce switching activity, and hence power dissipation in scan design by changing the order of scan cells in each scan chains. In deterministic test patterns, the best order of cells is the one that gives the best compromise between reducing the transitions in the scan cells during both scanning in test patterns and scanning out captured responses. As a simple example to demonstrate how cell ordering algorithms reduce the number of transitions in the scan-chain, assume a scan chain with 3 flip-flops A, B, and C with initial values of 111 as shown in Figure 2.6. Assume that the test vector 010 is to be scanned into the scan chain. The total number of transition generated in the scan chains by the loading of vector 010 will be equal to 6. However, by changing the order of scan cells B and C, the number of transitions is reduced from 6 to 2. The modified scan cell order may conflict with other objectives during scan chain optimization, such as wire length minimization and timing closure. To address this problem, the approach from [37] shows that, by clustering scan cells from the same physical region of the chip and chaining them in a power-aware fashion, it is possible to efficiently trade off test power reduction and wire length minimization. Although these algorithms reduce average and peak power consumption during the scanning cycles, but they have no effect on capture power reduction.

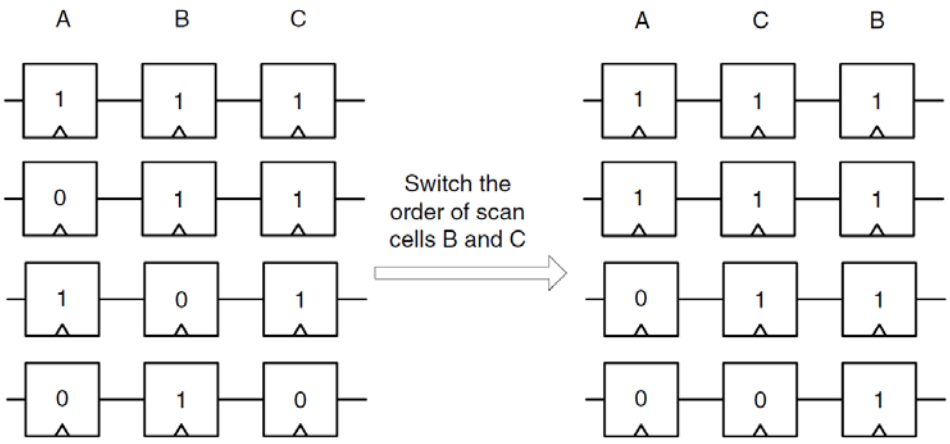


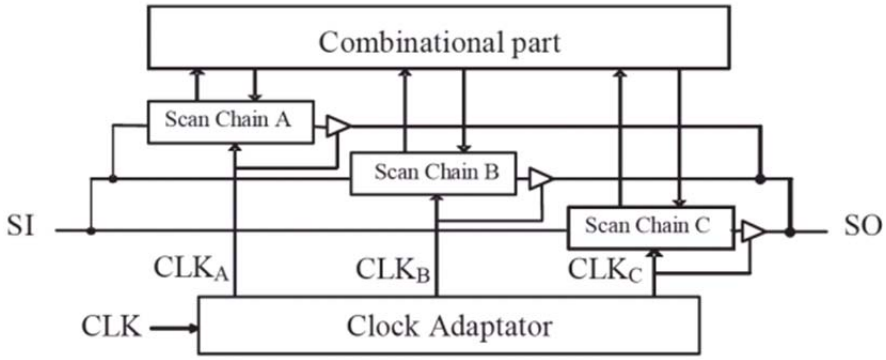
Figure 2.6: Scan chain reordering

A scan chain architecture using scan chain partitioning and disabling is proposed in [38] to reduce both the average and peak power consumption. The scan chain is split into several length-balanced segments and only one segment is enabled in each test clock during both shift and capture cycles. In this method, only a subset of scan cells is loaded with test stimulus and captured with test responses by freezing the remaining scan cells according to the spectrum of unspecified bits in the test cubes. Figure 2.7 shows a circuit whose scan chains have been segmented. Although this method reduces peak and average capture power during test, they increase test time and test pattern counts, in addition to consume additional chip area to enable independently clocking scan segments.

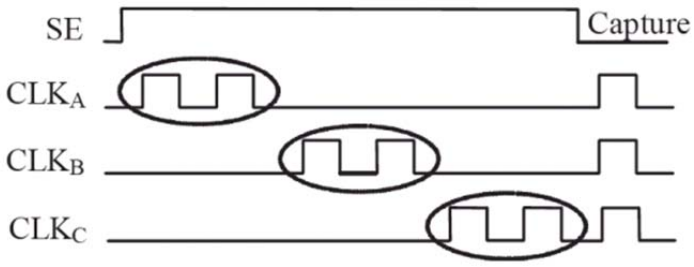
In [39] authors utilized a segmented scan chain environment to see the effect of partial clocking on transition faults and power reduction when segmented scan chains are clocked selectively in the capture mode. The benefit of partial clocking increases as the resolution of a gated clock (the number of registers with a gated clock) increases. By dividing a circuit into multiple scan segments, authors also improved transition fault coverage using launch-off-capture test method.

Staggering the clock [40] during shift or capture achieves power savings, without significantly affecting test application time. Staggering can be achieved by ensuring that

the clocks to different scan flip flops (or chains) have different duty cycles or different phases, thereby, reducing the number of simultaneous transitions. The biggest challenge to this technique is its implications on the clock generation, which is a sensitive aspect of chip design.



(a)



(b)

Figure 2.7: A circuit with segmented scan chains [38]

2.2.3 Functional / Pseudo Functional Scan Test Techniques

As we discussed in Chapter 1, design-for-testability (DFT) for synchronous sequential circuits allows the generation and application of tests that rely on non-functional operation of the circuit. This can result in unnecessary yield loss due to the detection of faults that do not affect normal circuit operation.

The test generation methods proposed in [41, 42] restrict the scanned in states to the set of reachable states to insure that the (CUT) operates only in the functional mode during capture cycles. Thus, such tests not only avoid abnormal switching activity during capture cycles but also avoid detection of certain faults that do not affect normal functional operation. Identifying reachable states has high complexity for large designs. This is addressed in [41, 42] by simulation based procedures that enumerate reachable states only as necessary for detecting targeted faults. No modifications are required to the test generation procedure and no sequential test generation is needed in these techniques. However for large designs determining that the state of the initialization pattern is a reachable state may be difficult.

In [43], authors used a sequential boolean satisfiability (SAT) solver to extract the functional constraints in the system. While theoretically SAT solver is able to find almost all the unreachable states in a circuit, its computational complexity is extremely high and hence cannot be applied to a large circuit.

In [44] an implication-based technique for illegal state identification was proposed. The method starts from any gate, say gate A, and finds the implications when its output value is logic '1' and logic '0', respectively. Suppose B and C are internal flip-flops in the circuit, and there exist two implications: [A(0)→B(1)] (i.e., A = 0 implies B = 1) and [A(1) →C(0)], we then have [B(0) → A(1)] and [C(1)→A(0)] according to contrapositive law. Consequently, we can conclude {B(0),C(1)} is an illegal state cube. Technique proposed in [44], considered the implications from impossible input-output combinations (e.g., for a 2-input AND gate, when an input is logic '0' while the output is logic '1') in their approach. To keep the computational complexity manageable, technique proposed in [43] implies values based on a single node in one time-frame only. This, however, significantly restricts the number of identified illegal states.

A data mining based illegal state identification strategy was proposed in [45]. In this technique, the circuit is expanded to multiple time-frames and simulated with a

number of random patterns. By analyzing the obtained data, some suspicious functional constraints are extracted. Then it uses a SAT solver to verify whether the obtained data are actually functionally-unreachable. While this dynamic learning method accelerates the search procedure, due to the large amount of data, it can only check pair-wise and three-node relations within small groups of state elements and hence cannot find many illegal states in the system.

On the other hand, finding a sufficiently large set of illegal states in large designs and especially those with multiple clock domains may not be practical. Thus, pseudo-functional tests do not guarantee avoiding non-functional operation. Functional tests and pseudo functional test may also reduce achievable fault coverage since they avoid detection of faults that require non-functional scan-in states.

In order to detect faults not detectable by functional tests partially-functional tests were introduced in [46]. Partially-functional tests use scan-in states that are at a minimal Hamming distance from a reachable state. By keeping the Hamming distance of scan-in states from reachable states to a small value the switching activity caused by the tests is kept close to or the same as that caused by reachable states.

2.3 Low Power Test Data Compression and BIST

Test data compression is an effective solution to the problem of increasing test data volume. They have an on-chip decompressor, which decompress the data as it is fed into the scan chains during test application. Typically, on-chip decompressors fill don't care bits with random values, and therefore the amount of flip-flop toggling during test may result in a power droop condition that would not occur in the chip's mission mode. Excessive switching activity in scan chains and other parts of the circuit results in overheating or supply voltage noise, both causing a device malfunction, and thus loss of yield, reliability degradation, shorter product lifetime, or even permanent damage of a circuit. To address this issue, several this issue several techniques has been proposed to

simultaneously reduce test data volume and test power in test data compression and BIST.

Dual-speed LFSR (DS-LFSR) architecture is presented in [48] to address the problem of abnormal switching activity during BIST. In DS-LFSR the use of a regular LFSR as a TPG is replaced by two LFSRs. One of them is clocked at regular speed while the second is clocked at a lower speed. Figure 2.8 shows this scheme.

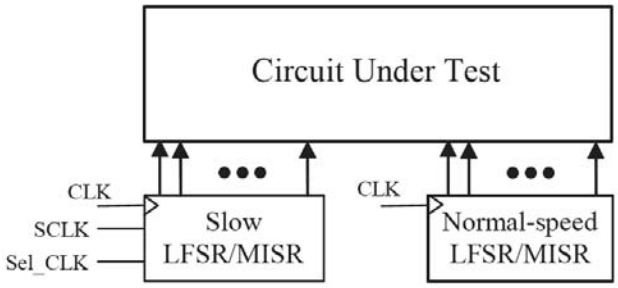


Figure 2.8: Low-power BIST with DS-LFSRs [48]

The slow LFSR has k inputs and two clocks and the normal speed LFSR m-k inputs and the normal clock, where m is the number of inputs to the CUT. The second clock in the slow LFSR is a slow clock whose period is 2^{m-k} times that of the normal clock. The authors prove that this architecture can produce the same, but rearranged, patterns as a normal LFSR, provided that the LFSRs are modified to produce also the all-zero output. Now some inputs to the circuit will have a slower transition rate because they are fed by a slower LFSR. The inputs of the CUT with the most fanouts are selected to be fed by the slow LFSR. This is like rearranging the vectors produced by an ATPG. The rearrangement of patterns that this architecture produces reduces the switching activity during test of the circuit.

Methods to automatically disable some of the scan chains during certain times have been proposed for deterministic test [49] and built-in self-test [51]. In [49], tests in a

given test set T are divided into groups and only changing portions of consecutive tests are shifted into the scan chains. However, peak power consumption is not necessarily reduced in this scheme since all of the chains are activated when a test from the next group in T is shifted in to the scan chains. To reduce the peak power consumption, this technique should be modified such that the simultaneous activation of all scan chains is prevented at test group boundaries.

In [50] to reduce the peak power consumption a scan chain partitioning scheme is proposed. Previous method considered loading each scan chain segment at a time by disabling the clock signals. In this method, a modified scan element shown in Figure 2.9 is used to load only one scan chain at the time. The modified scan cell has the same properties as the scan cells shown in Figure 1.3 except that when Scan Enable is at 1 and Group Enable is at 0 the flip-flop maintains its previous value.

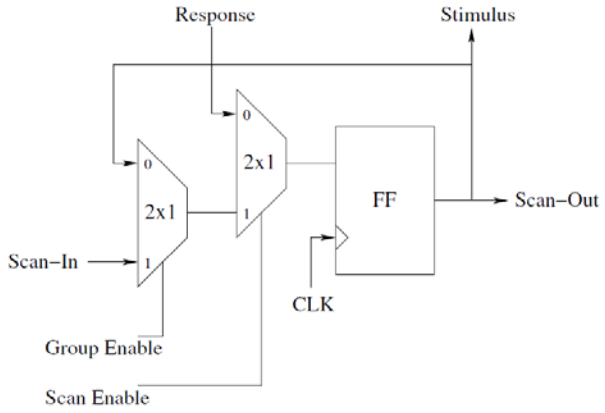


Figure 2.9: Modified scan cell [50]

As can be seen in Figure 2.10, each scan chain is divided into N scan groups, each one with its correspondent Group Enable. With this modification, only the active group will capture data and have data shifts, thus the switching activity in the logic will be confined to the fanout cone of the active groups and therefore reduced. Since disabling subsets of the scan chains results

in some of the faults not being activated, propagated or observed because some circuit inputs won't be assigned new values, it is necessary to run more test cycles (composed of a scan cycle and functional cycle).

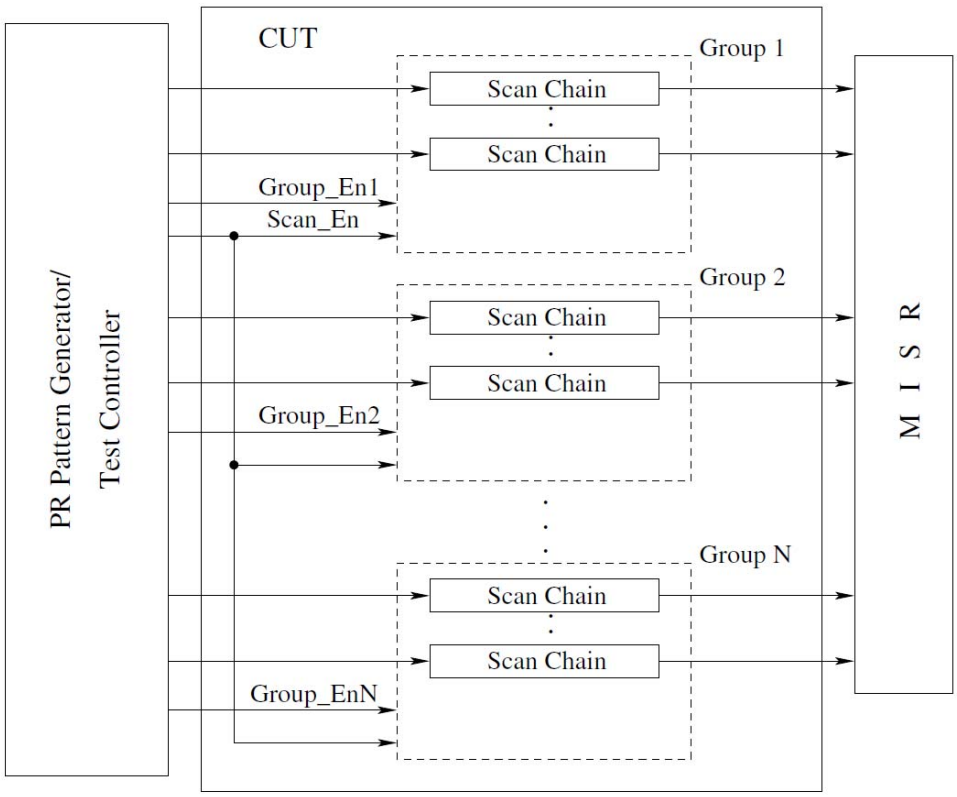


Figure 2.10: BIST scheme proposed in [50]

The minimum transition fill is deployed in [52] to reassign new values to unspecified positions whose locations are determined by means of bit stripping. This operation is performed on successive bits of a given test pattern to verify whether turning a bit into an unspecified one will affect the fault coverage. If so, then the bit is returned to its original value; otherwise, it is kept as a don't care bit to reduce the transition count.

Low power test data encoding schemes [53-55] adopt conventional LFSR reseeding techniques to reduce the scan-in transition probability. In particular, the method

of [55] uses two LFSRs to produce actual test cubes and the corresponding mask bits. Outputs of both LFSRs are AND-ed or OR-ed to decrease the amount of switching. The use of extra seeds may compromise compression ratios. To overcome this problem, the scheme of [53] divides test cubes into blocks and only uses reseeding to encode blocks that contain transitions. For the blocks that do not contain transitions, the logic value fed into the scan chain is simply held constant. This approach reduces the number of transitions in the scan chain and hence reduces test power.

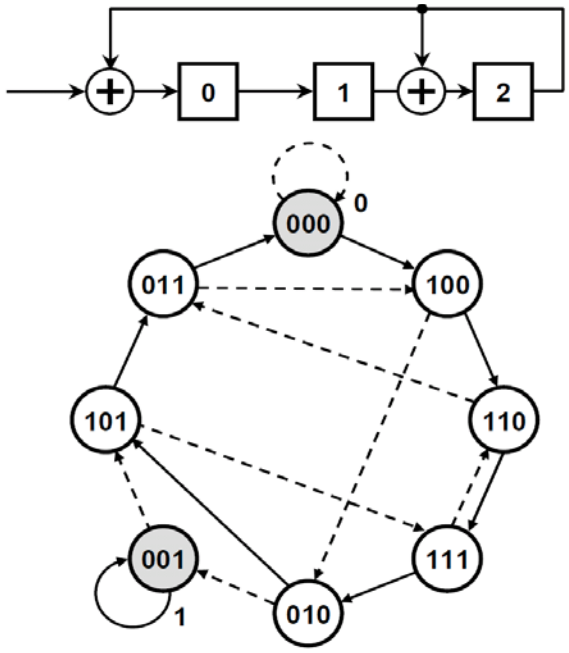
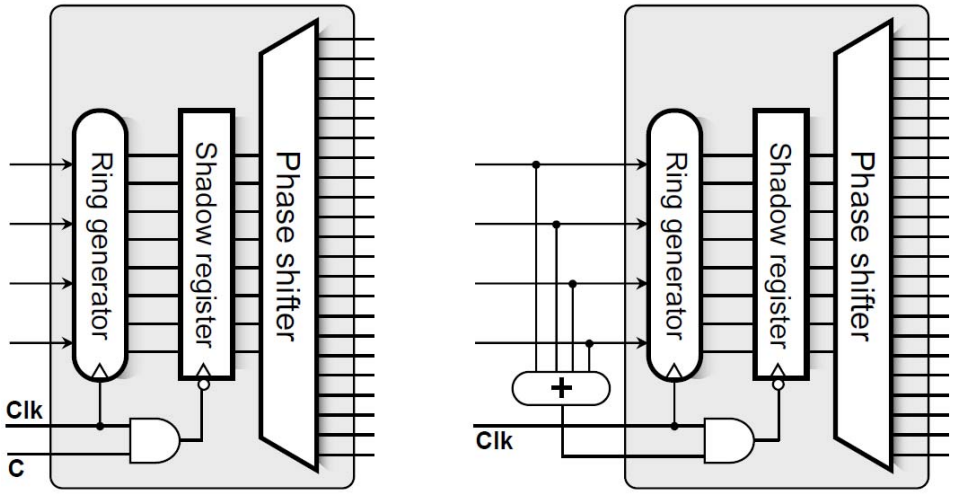


Figure 2.11: LFSR with one input [56]

Technique proposed in [56] identifies the self-loop (SL) states to apply identical data to the scan chains for a number of shift cycles, thereby reducing the total number of transitions. A self-loop (SL) state of an LFSR is one such that for a given input the next LFSR state is the same as the present. Figure 2.11 shows the implementation of a one input LFSR and its state diagram, the SL states are marked in gray. The authors prove

that the number of SL states in a ring generator with c inputs, with every pair of them separated by at least one flip-flop, is equal to 2^c. In EDT, a given test cube to be fed into the CUT must be split into the number of chains the circuit has. Then all chains are loaded at the same time by the ring generator and the phase shifter; each load is called a slice. The proposed technique tries to assign to each slice of the test that has a specified bit a SL state. Then the remaining slices with no specified positions inherit SL states from their most adjacent neighbors. The result is a test vector in which many neighboring slices are identical. This reduces power consumed in the circuit during shift by reducing the number of transitions in the output of the scan cells.

In [57] authors propose a technique that inserts a shadow register between the ring generator and the phase shifter in order to reduce power consumption during shift in EDT schemes. Figure 2.12 shows two different schematics of the new architecture. The shadow register is a negative edge-triggered device clocked by the ring generator clock and the XOR of all the inputs of the ring generator. Like conventional EDT the test cube to be encoded is divided into time slices in which each position of the time slice corresponds to an input of a scan chain. Then equations to solve the specified bits in the first slice are generated, and once the ring generator is in the desired state an extra equation is added to clock the shadow register. Then, while the set of equations can be solved and there is no conflict in the specified bits positions, new equations are added to solve the following time slices indicating that there is no need to update the shadow register. When there is a conflict in a certain position or the solver fails to encode a given slice, it marks the beginning of a new cluster with the shadow register reloaded. In this way the CUT chains are fed with the same values for many slices. This ensures low toggling at the outputs of the scan cells with little area overhead. Experimental results on industrial designs with this method showed that shifting power can be reduced up to 40 times.



(a) (b)

Figure 2.12: Low power decompressor proposed in [57]

Techniques proposed in [56, 57] just reduce power during shift mode. In [58] a technique is proposed to reduce power during both shift and capture. It employs a controller that either allows a given scan chain to be driven by a power-aware EDT decompressor, or by a constant value for the entire scan load. The controller's configuration is held constant for all cycles of a given scan load but is reloaded for every pattern with minimal control data. The same controller may be used to decide which scan chains remain in a shift mode during the capture cycle by judiciously disabling scan clocks. This is affective at reducing toggling during capture and unloading, especially when the scan chains kept in the shift mode loaded constant values.

Figure 2.13 shows a schematic for architecture proposed in [58]. The control block comprises a control register and combinational XOR logic driven by data stored, in a compressed form, in the register. For further reducing the total number of transition, the outputs of a decompressor are sustained for more than a single clock cycle, while allowing the decompressor to change its internal state to ensure successful encoding of next specified bits. Scheme for sustaining the outputs of a decompressor is shown in

Figure 2.14. Authors integrated scheme presented in Figure 2.13 with the on-chip test data decompressor presented in Figure 2.12. So the same data are provided to the scan chains for a number of shift cycles through a shadow register placed between the ring generator and the phase shifter. It captures and saves, for a number of cycles, a desired state of the ring generator, while the generator itself keeps advancing to the next state needed to encode another group of specified bits.

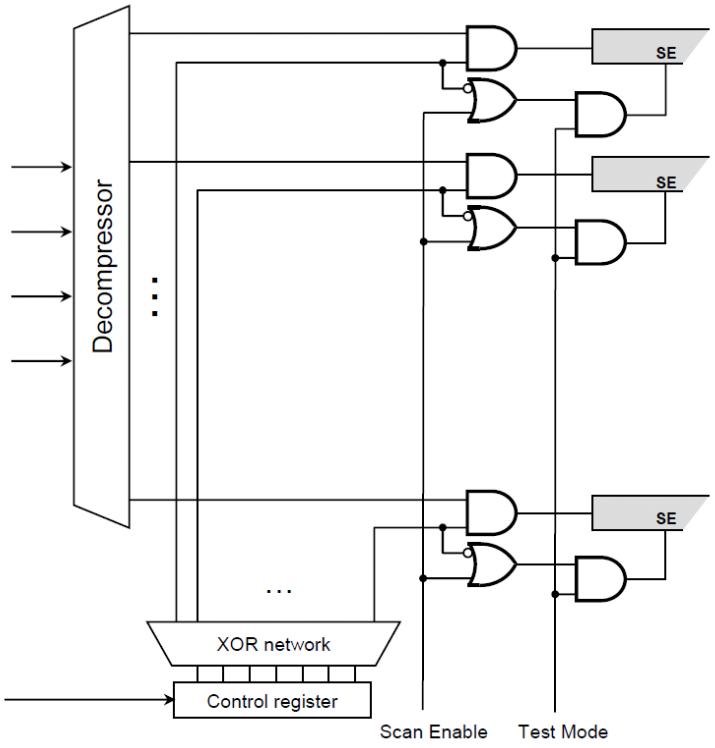


Figure 2.13: Low power scheme proposed in [58]

Compression compatible filling methods for linear decompressors were proposed in [60] and [59]. The first technique, after power-aware fixing of a given unspecified bit, determines (by implication) bits linearly dependent on it to ensure that a test cube remains compressible. The second scheme proposes to arrange a low power filling for free input variables (as opposed to pivot variables) supplied to a linear decompressor, as

they can be assigned any logic value. By evaluating the impact of free variables on test power dissipation and changing their order during the filling process, this approach claims a visible power reduction during all scan test phases. Another form of filling [61] handles unspecified bits of test cubes with the aim of reducing capture power in a code-based test compression environment.

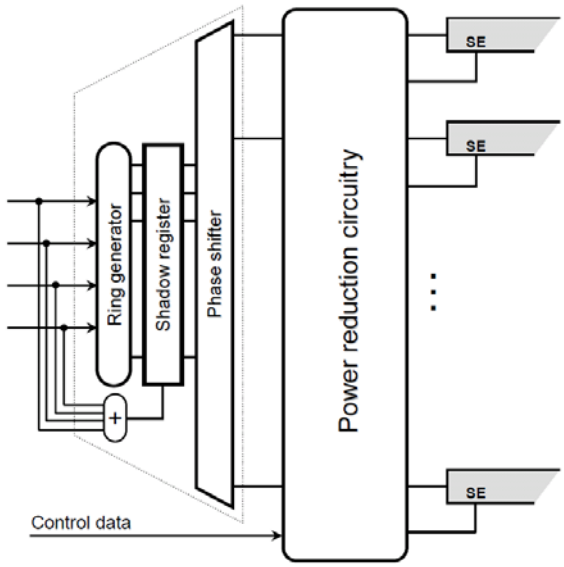


Figure 2.14: Shadow register in decompressor proposed in [58]

CHAPTER 3

LOW CAPTURE POWER AT-SPEED TEST

The scan-based DFT method makes sequential elements (latches or flip flops) controllable and observable by chaining them into a shift register (scan chain). However, due to violation of various functional constraints imposed by the functional logic on the sequential circuits, scan-based tests usually increase the switching activity of the circuit which results in abnormal power dissipation and supply current demand. This chapter presents an effective method to generate test patterns such that the switching activity during capture cycles of the tests is low. The proposed method uses a functional background or background states obtained by applying a number of functional clock cycles with scan in states to fill unspecified values in test cubes to keep the switching activity of test patterns low.

The rest of this chapter is organized in the following manner. Section 3.1 describes motivation for low power testing using functional background. Section 3.2 describes the power metric used for estimating the switching activity and power dissipation in this study. Section 3.3 describes method of generating near to functional background. Section 3.4 describes the proposed low power test method that generate near to functional test vectors. In Section 3.5, the experimental results for industrial designs are given. Finally, concluding remarks are given in Section 3.6.

3.1 Motivation

Several different methods have been proposed in literature to reduce switching activity in scan based test. A concern with low power test generation methods is that they may generate tests that cause switching activity far below that caused by functional tests [62], thus leading to under testing. In [62], it was observed that low power tests cause as much as 40% less switching activity than functional broadside tests. Therefore, it is desirable to find methods to keep switching activity of each pattern very close to that in

functional mode of operation without substantially increasing test cost (pattern count, ATPG run times, additional DFT etc.).

In this work, we investigate a new procedure to fill unspecified values in test cubes to cause low peak switching activity during test capture cycles which is close to what is believed to be functional operation. Since we don't identify reachable states to generate tests we cannot insure that the switching activity by the proposed tests is the same or determine how close it is to the functional switching activity. Our observations regarding the switching activity caused by the proposed tests is based on the fact that it is close to the minimum switching activity measured when the circuit under test is operated in functional mode for several clock cycles starting from the scan-in states of the tests in which the don't care bits are randomly filled. We discuss this in detail in the next section. Clocking the circuit in functional mode was proposed in [63] to address over testing. As shown in [42], this does not guarantee that the circuit will enter a reachable state starting from arbitrary scan-in state.

3.2 WTM and WSA Modeling

To evaluate the power dissipation effectively, several metrics have been proposed. In scan-based testing, a good way to estimate the power dissipated during scan-in of test vectors or scan-out of captured responses is the weighted transition metric (WTM) [65]. The weighted transition metric that the power consumed in scan-based testing depends not only on the number of induced transitions in successive scan cells but also their relative positions. Let m be the length of a scan chain, and $T = b_1 b_2 \ldots b_m$ represent a test pattern with bit b_k scanned in before b_{k+1}. The normalized form of the metric is then defined as follows:

$$S = 2[m(m+1)]^{-1} \sum_{i=1}^{m-1} (m-i)(b_i \oplus b_{i+1})$$

The average scan power dissipated during test application is obtained by summing up results provided by the above formula over all scan chains and all test patterns.

The capture power is more related to the total number transition at each gate in the circuit [64]. The metric for the capture power used in this dissertation is defined as follows:

$$W_P = C^{-1} \sum_{i=2}^{f+1} \sum_{k=1}^{g} t_k F_k$$

where $t_k = 1$ if gate k toggles in frame i, and $t_k = 0$ otherwise, F_k is the fan-out of gate k, g is the total number of gates, f is the number of frames in test pattern p (f + 1 represents the post capture frame), and C is the number of cycles of test pattern p.

3.3 Switching Activity Caused by LOC tests

Here, the Weighted Switching Activity (WSA) defined in previous section is used as a measure of power and supply current demand.

The WSA caused by different launch-off-capture (LOC) tests for transition delay faults (TDFs) is computed as follows. Recall that LOC tests for TDFs scan in a state, with the scan enable line at 1 (active) to set up the initialization vector of a two pattern test followed by two capture cycles during which the scan enable line is 0 (inactive). The two capture cycles are called launch and capture cycles. We simulated LOC tests by applying the launch and capture cycles followed by several additional (capture) cycles with scan enable at 0 and determined WSA caused by each capture cycle. This was based on the intuition that a longer sequence of primary input vectors applied after a scan operation is more likely to cause the circuit to enter its functional operation mode. The timing diagram for this simulation is shown in Figure 3.1.

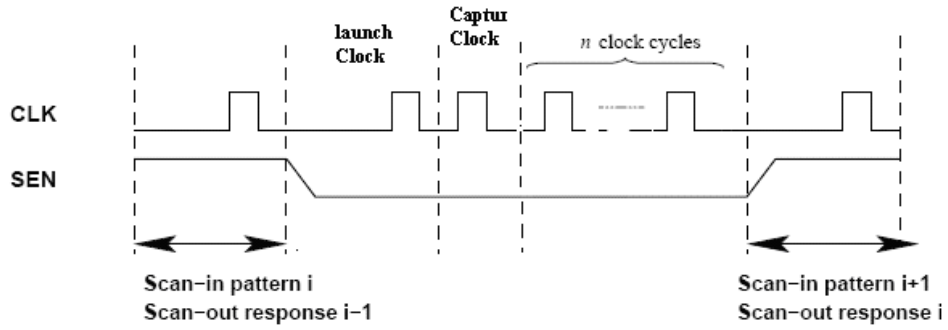


Figure 3.1: The timing diagram for generating functional background in LOC test

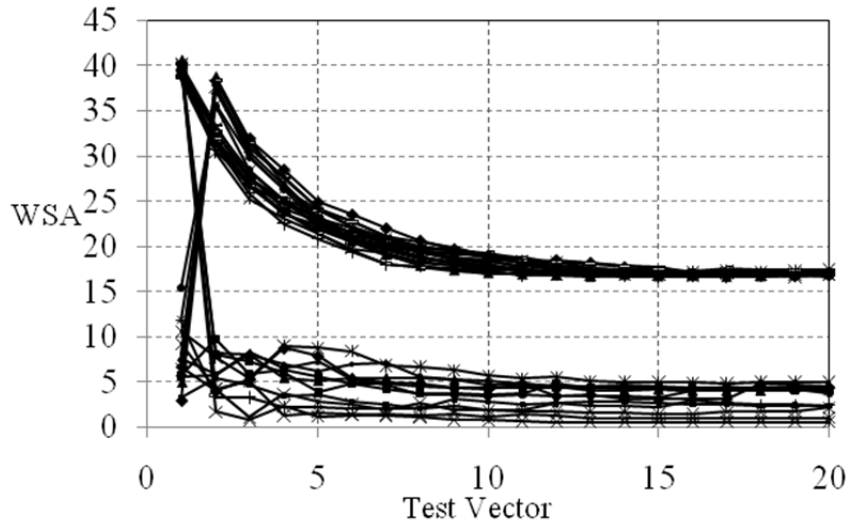


Figure 3.2: Applying 20 clock cycles to the test vectors generated for LOC test using random fill

In Figure 3.2 we show the WSA of up to a total of twenty capture cycles for 32 LOC tests obtained by random fill of the unspecified values in test cubes. The primary inputs were held at the values used in the test during the first two capture cycles and then were randomly set for the next 18 capture cycles. In Figure 3.2 we give WSA as a percentage of the maximum possible value of WSA which occurs if all the gates in the

circuit under test (CUT) switch state. From Figure 3.2 we can make the following observations;

(i) The WSA reaches a steady state value which is typically much lower than the WSA during the first two capture cycles,

(ii) The steady state WSA value reached depends on the scanned in state, and

(iii) Even when the WSA caused by the first (launch) capture cycle is low the WSA of the second capture cycle could be much higher.

We observed similar behavior for LOC tests obtained using preferred fill [28] and also using zero fill. This is illustrated in Figures 3.3 and 3.4. Additionally if the primary input values are held constant at their values during the second capture cycle the WSA during the last 18 cycles behaved similar to what is shown in Figures 3.2, 3.3 and 3.4.

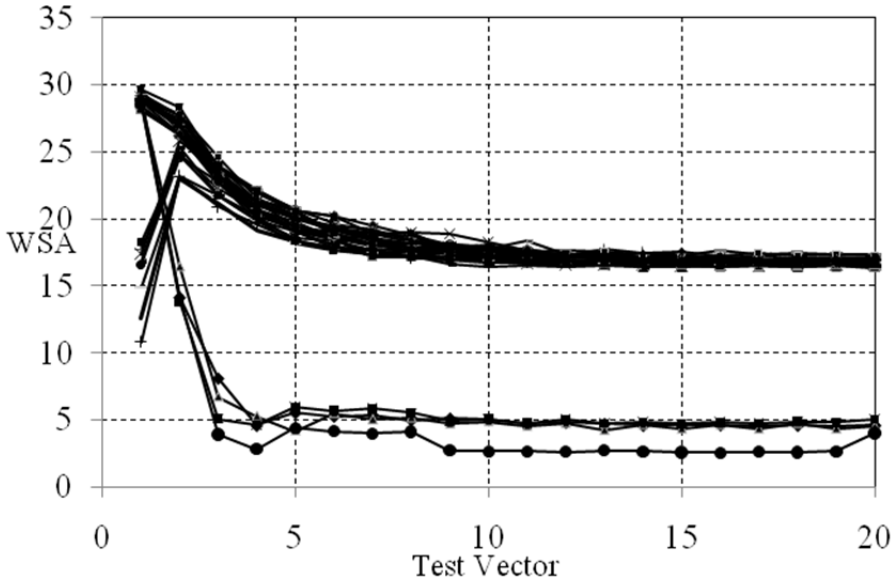


Figure 3.3: Applying 20 clock cycles to the test vectors generated for LOC test using zero

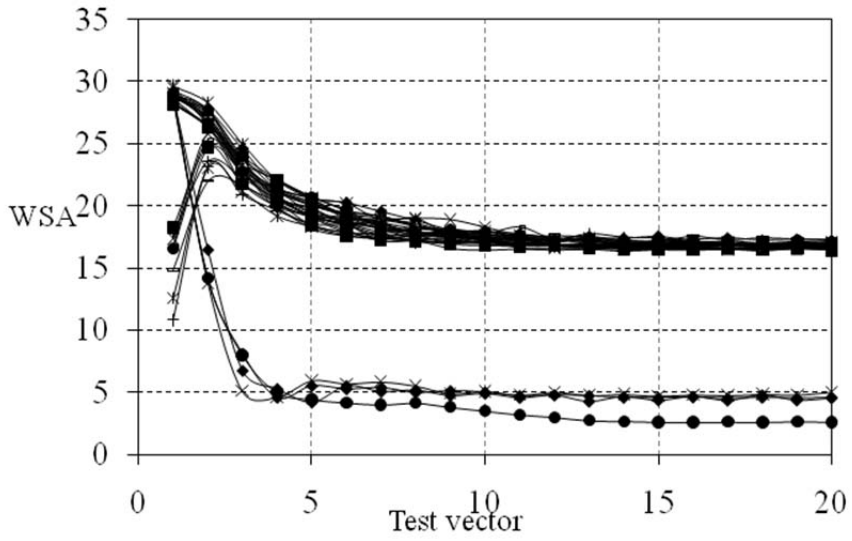


Figure 3.4: Applying 20 clock cycles to the test vectors generated for LOC test using preferred fill

3.4 Low Power Test Vectors with Functional Profile

The proposed method, called ACF-scan to generate tests with low WSA during capture cycles is based on the observations made in the last section.

The observations made above suggest that if one were to fill the unspecified values in test cubes by the values in the scan cells found by simulating the test cube for some number of cycles with random values for primary inputs and scan enable at 0, after random filling the unspecified values in test cubes, the WSA of the resulting tests can be expected to be low and close to the steady state value determined during simulation. The reason for this expectation is that the number of specified values in test cubes is small and hence the state obtained after filling the unspecified values with the same as those of the state, say S, after simulation, differs from S only in some positions where the test cube had specified values.

We illustrate the proposed test generation using an example. Consider a test cube with T1= (0, X, 1, X, X, 0, X). After filling the Xs randomly with zeros and ones and

simulating the resulting fully specified test for several extra cycles assume that the state obtained is T2 = (0, 1, 0, 0, 1, 1, 0). We fill the Xs in T1 by the corresponding values in T2 to obtain the test T = (0, 1, 1, 0, 1, 0, 0). Note that the test T obtained differs from the state after simulation of T1 in only 2 positions even though three cells contained specified values in T1.

After extensive experiments using many tests we chose to use five cycles of simulating test cubes as described above and used the states of the scan cells obtained to fill the unspecified values in the test cubes of the LOC tests. Table 3.1 shows reduction in peak WSA during the first capture cycle for tests in two industrial circuits, C2020 and C2225 when the unspecified values in the test cubes are filled from the state obtained after different cycles of simulation. (The profiles of the industrial circuits we used in our experiments are given in Table 3.2 in the next section.) It can be noticed that after five cycles we obtain most of the reduction in WSA.

It is important to note that the proposed method to fill unspecified values does not require changes to the ATPG procedures. It only changes the way the unspecified values in test cubes are filled which is done after the test cubes are generated.

Table 3.1: Reduction in peak WSA after different numbers of cycles of simulation

	% Peak WSA Reduction				
Circuit	**#cycle:1**	**#cycle:3**	**#cycle:5**	**#cycle:10**	**#cycle:20**
C2020	25.18	30.05	31.28	32.00	32.45
C2225	66.84	70.95	71.8	73.52	74.58

The pseudo-code for the test generation procedure using the method to fill unspecified values as described above is shown in Figure 3.5.

while (not all faulst targeted)
    Select a fault and generate a test cube using dynamic compaction
    Fill unspecified values randomly
    Apply 5-10 clock cycles to obtain a pattern that mimic WSA of functional patterns
    Use the obtained pattern to fill the unspecified values of the test cube
    Fault simulate the test and drop the detected faults
end while

Figure 3.5: The proposed low-power test generation procedure

As noted above, since only a small percentage of the scan cells have specified values in a test cube, when the unspecified values are filled using the state obtained after simulating as proposed, the state part of the test obtained will be close to the state obtained after simulation. Thus we can expect that the WSA of the capture cycles for the tests obtained as proposed here will be close to the steady state value illustrated in Figure 3.2. In order to verify whether this conclusion is valid we computed the WSA for the two capture cycles for all the LOC tests of an industrial circuit C2020 and the results are given in Figures 3.6 and 3.7. In Figures 3.6 and 3.7 we report the percentage increase in WSA of the two test capture cycles compared to the WSA caused by the background state obtained after five cycles of simulation. In Figures 3.6 and 3.7 X-axis gives the test number. It can be seen that the percentage increase in the WSA for the capture cycles of tests is less than 30% for all the tests and for majority of the tests less than 10%. The higher increase is only for the first few tests since in these tests typically higher percentage of test cube inputs are specified.

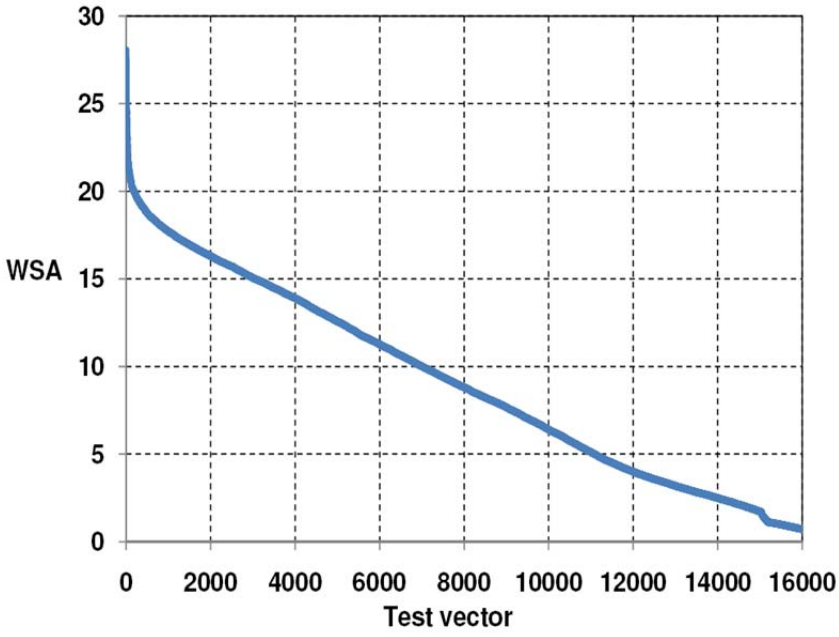


Figure 3.6: Percentage difference in WSA in the first capture cycle of the proposed tests relative to the WSA when the test cubes were simulated for 5 cycles after random fill

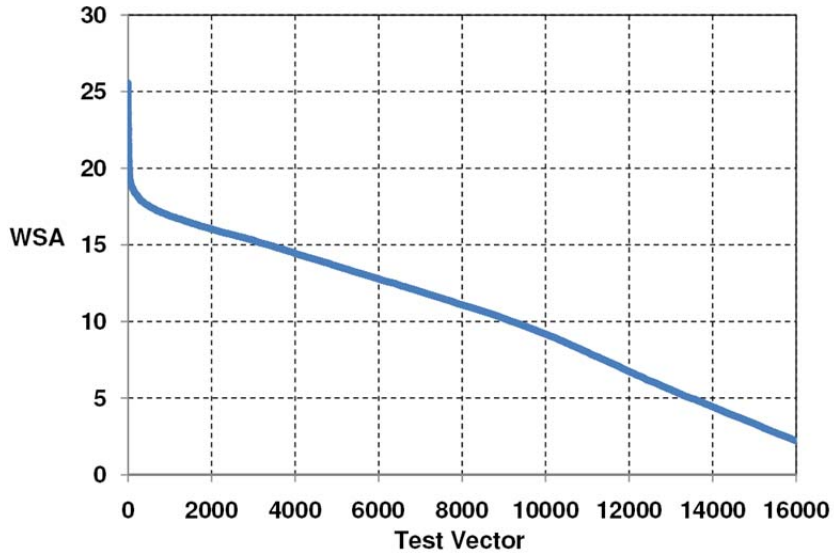


Figure 3.7: Percentage difference in WSA in the second capture cycle of the proposed tests relative to the WSA when the test cubes were simulated for 5 cycles after random fill

Figures 3.8 and 3.9 show WSA of vector pairs in LOC test during the first capture and the second capture for circuit C2225 by using random fill, and our proposed method respectively. The vector pairs are sorted according to the WSA of the first capture in descending order. It can be seen from these figures that the capture power during the first cycle and the second cycle for random filling are completely different. However, WSA using functional background for the second capture cycle is almost the same as the WSA for the first capture cycle.

In the proposed test generation procedure the unspecified values in test cubes were filled to match contents of state elements after simulating randomly filled test cube for several cycles. Instead, one can fill the unspecified values to match any known functionally reachable state. One can also choose a functionally reachable state which minimally differs in the positions where the test cube has specified values. If this is done we will obtain what have been called partially-functional tests [46]. This approach will require computing a sufficient number of reachable states.

Our primary goal in this work is to reduce capture power in order to avoid IR-drop-induced yield loss. Experimental results in Section 4 show that we can achieve our goal with this method. But higher than normal average switching activity caused by scan shifts also could cause excessive heat due to increased power dissipation. We can reduce switching activity during both scan shift and during capture cycles by combining preferred fill method with ACF-scan. Instead of filling all unspecified scan cells after generating test cubes randomly, one can fill a percentage of the unspecified scan cells with preferred values and the remaining ones with random values (ACF/PF). Then apply 5-10 clock cycles to obtain the state used to fill the unspecified values in the test cube. We used 50% fill with preferred and 50% with random values.

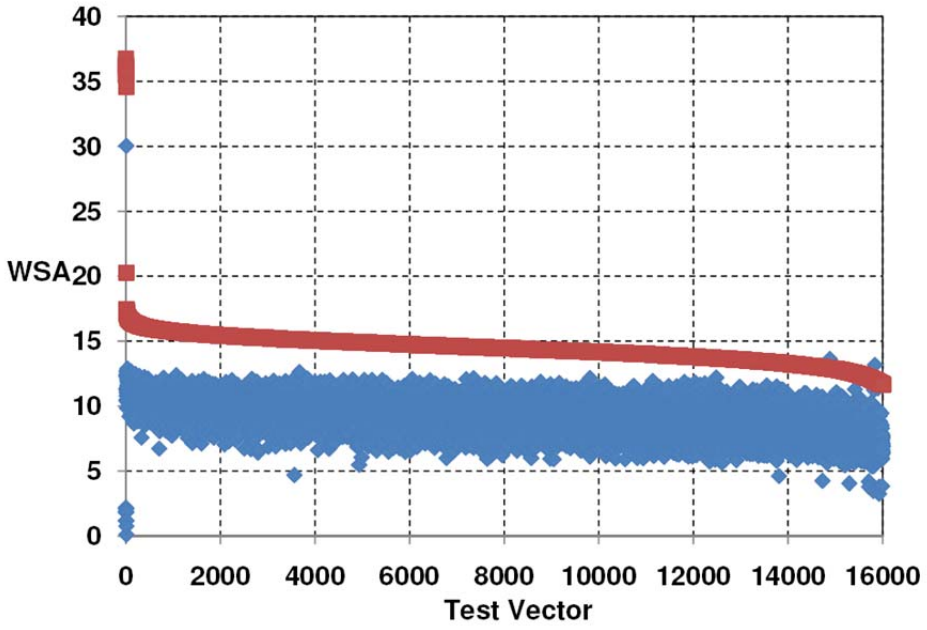


Figure 3.8: WSA of 1st and 2nd capture using random fill for Circuit C2225

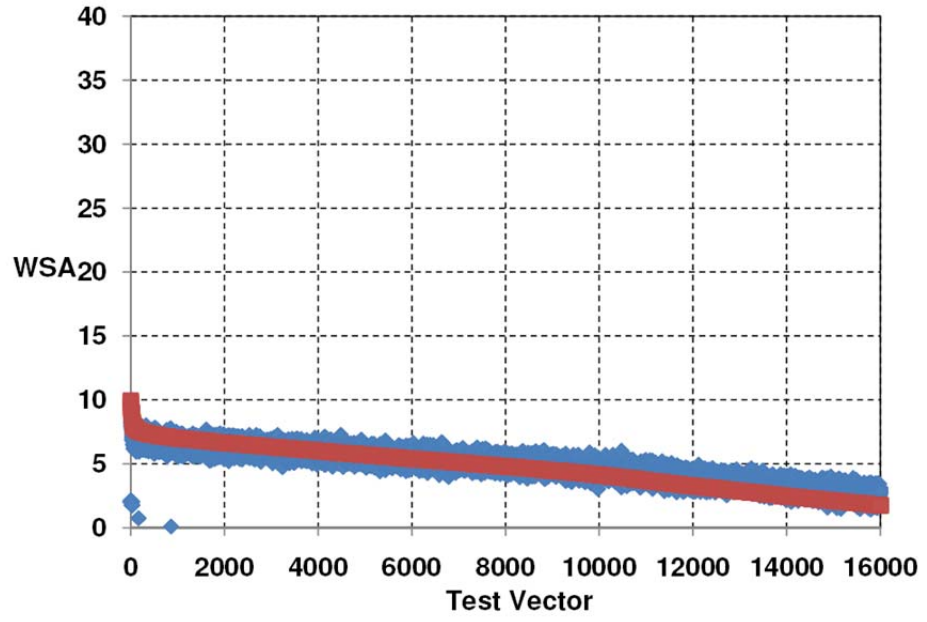


Figure 3.9: WSA of 1st and 2nd capture using proposed method for Circuit C2225

3.5 Experimental Results

The proposed method was implemented using a state of the art commercial ATPG for launch-off-capture (LOC) tests. As discussed earlier we used five cycles of simulation of test cubes to obtain the background state used to fill the unspecified values in test cubes. The test cubes were generated using dynamic compaction procedures to detect all detectable transition faults (TDFs) using LOC tests. The dynamic compaction procedure was allowed to use all unspecified values in the test cube generated for the primary target fault to target detection of additional faults values. The unspecified values remaining after dynamic compaction were filled using the proposed method (both ACF-scan and ACF/PF), preferred fill [28] and zero fill methods, resulting in four test sets for the four methods of fill. All test sets were generated to achieve the same maximum coverage of TDFs. On average the test generation times using the proposed method were approximately 15% higher than that for random fill.

Table 3.2 Circuit characteristics

Design	Gates	Faults	Scan cells
C260	260 K	390 K	14 K
C305	300 K	500 K	21 K
C419	410 K	900 K	19 K
C496	490 K	1.3 M	45 K
C844	800 K	1.2 M	79 K
C1498	1.4 M	3.6 M	45 K
C2020	2.0 M	3.3 M	130 K
C2225	2.2 M	3.3 M	140 K
C2511	2.5 M	4 M	160 K

The proposed low power schemes were tested on several industrial designs, ranging in size from 260K to 2.5M gates. The basic data regarding the designs, including the number of gates, number of transition faults and the number of scan cells are listed in Table 3.2.

We also computed the Bridge Coverage Estimate (BCE) of the three test sets. The Bridge Coverage Estimate (BCE) metric of a test set was proposed in [66] and is often used to estimate the bridging fault coverage of test sets. The BCE of a test set is determined from the information on the number of times each stuck-at fault is detected by the test set, and is calculated as follows:

$$BCE = \sum_{i=1}^{N} (1 - 2^i) \frac{f_i}{|F|}$$

where $|F|$ is the total number of stuck-at faults, f_i is the number of faults detected i times, and N is the maximum number of times any fault is detected.

Table 3.3 shows percentage reduction in peak WSA caused in the first and the second capture cycles using the proposed test generation method (ACF-scan), preferred fill method, zero fill method and ACF/PF method which is the combination of ACF-scan and PF method that fills 50% of unspecified scan cells with PF as described earlier. All reduction percentages are relative to the case when the unspecified values are filled randomly.

From Table 3.3, it can be seen that if proposed method (ACF-scan) is used, peak WSA is reduced, on average, by 49% in the first capture cycle and by 28% in the second capture cycle. These reductions are higher than WSA reduction for the preferred fill and zero fill methods. Peak WSA reduction of ACF/PF is very similar to ACF-scan Method.

Table 3.4 shows the percentage reduction in peak state element transition (SET), which is the maximum number of scan cells whose states change during the capture mode

in first and second capture cycles. It can be seen that ACF-scan and ACF/PF reduce peak state element transition (SET) more than preferred fill and zero fill.

As shown in Table 3.4, ACF-scan and ACF/PF methods reduce peak SET on average by 71% and 72% respectively while PF and 0 fill methods reduces peak SET on average by 56% and 59%.

Table 3.3: Percentage reduction in Peak WSA during 1st and 2nd capture cycles

Design	1st Capture				2nd Capture			
	ACF-scan	PF	0 Fill	ACF/PF	ACF-scan	PF	0 Fill	ACF/PF
C260	50.88	49.42	47.29	53.21	7.06	0	-4.12	9.42
C305	50.45	48.37	42.61	46.98	7.43	15.11	10.67	8.38
C419	41.97	32.43	49.31	46.66	18.76	22.92	25.34	26.44
C496	33.91	21.34	10.17	32.13	21.45	21.29	17.26	18.91
C844	73.71	72.87	72.54	72.59	8.36	15.16	12.87	12.47
C1498	31.89	37.39	44.45	40.48	21.11	29.98	35.14	23.35
C2020	30.94	32.48	32.58	31.56	31.88	31.22	32.57	28.44
C2225	74.45	60.76	70.12	68.59	74.31	61.12	74.61	72.89
C2511	51.89	42.65	4.21	52.71	64.47	38.93	5.14	60.83
Ave.	**49.01**	**44.19**	**41.48**	**49.43**	**28.31**	**26.19**	**23.28**	**29.01**

Another point to be noted from Tables 3.3 and 3.4 is that peak WSA and SET for circuit C2511 using zero fill is not much reduced compared to the conventional tests that

use random fill of unspecified values. Consequently we conclude that zero fill can reduce overtesting in some circuits but not in all circuits.

Table 3.4: Percentage reduction in Peak SET during 1st and 2nd capture cycles

Design	1st Capture				2nd Capture			
	ACF-scan	PF	0 Fill	ACF/PF	ACF-scan	PF	0 Fill	ACF/PF
C260	88.2	86.78	86.52	91.05	86.33	38.43	23.11	55.43
C305	61.09	64.45	60.34	58.27	65.25	17.28	22.34	15.28
C419	57.34	40.32	69.43	61.65	57.73	30.39	12.29	33.47
C496	61.26	22.74	15.71	55.44	67.64	48.65	15.77	51.39
C844	94.92	92.18	91.03	95.83	95.39	56.74	75.31	68.64
C1498	55.32	56.34	59.13	57.67	69.28	36.81	41.67	30.85
C2020	60.44	5.73	57.28	59.42	33.46	54.75	53.83	64.62
C2225	81.27	75.38	89.63	85.38	93.56	61.32	61.47	87.41
C2511	85.15	67.42	3.77	86.41	67.27	89.19	69.03	92.52
Ave.	**71.67**	**56.82**	**59.20**	**72.35**	**70.66**	**48.17**	**41.65**	**55.51**

Table 3.5 shows the average power reduction during shift mode. It can be seen that ACF-scan only reduces the average WSA during shift by 38% and is not as effective as preferred fill and zero fill. It is because our main goal is to reduce IR-drop and overtesting during at-speed scan test. However it can be seen that by combing the other low power method attempt to reduce shift power with our proposal method, it is possible to reduce switching activity during both shift and capture cycle. As shown in Table 3.5,

ACF/PF which combines PF method with the proposed method achieves better shift power reduction compare to ACF-scan.

Table 3.5: Comparing Reduction in average power during shift

Design	ACF-scan	PF	0 Fill	ACF/PF
C260	7.62	91.70	92.81	63.84
C305	20.24	19.61	87.74	34.91
C419	69.22	88.75	92.92	83.55
C496	13.05	89.94	93.78	39.23
C844	39.89	99.18	99.33	63.72
C1498	37.06	87.13	87.85	69.66
C2020	56.36	84.92	95.93	70.74
C2225	39.41	90.95	96.70	81.71
C2511	63.54	63.72	92.71	65.90
Ave.	**38.49**	**79.54**	**93.31**	**63.70**

Impact on pattern count: Application of tests in the low power mode reduces the degree of randomness observed in the scan chains. In Tables 3.6, under column ΔPC the test pattern difference between conventional ATPG and low power techniques (ACF, PF, 0 fill and ACF/PF) are represented. One can notice that using ACF-scan and ACF/PF the test pattern counts increase, on average, by 14% and 19% respectively instead of 38% and 34% when PF and 0fill is used. Since the proposed method uses different near to

functional background instead of constant value to fill unspecified bits in test cubes, pattern count increase is not significant.

Table 3.6: Comparing pattern counts and bridging coverage estimate

Design	ΔPC				ΔBCE			
	ACF-scan	PF	0 Fill	ACF/PF	ACF-scan	PF	0 Fill	ACF/PF
C260	7.01	10.72	9.19	6.22	0.61	1.01	0.88	0.11
C305	23.21	42.43	41.53	24.43	0.22	1.45	1.23	0.32
C419	21.31	49.28	38.28	23.15	0.10	2.08	3.52	1.31
C496	5.65	10.54	5.64	6.43	0.23	0.82	0.71	0.30
C844	15.95	88.18	76.92	38.23	1.92	16.79	17.12	7.36
C1498	13.32	47.76	51.27	26.82	0.49	2.22	2.85	1.32
C2020	31.43	50.23	55.54	32.42	3.87	6.20	6.03	4.48
C2225	4.18	15.54	15.48	1.27	1.46	4.66	5.53	2.71
C2511	7.32	23.29	15.34	14.93	1.51	5.52	3.01	5.73
Average	**14.38**	**37.55**	**34.35**	**19.32**	**1.16**	**4.53**	**4.54**	**2.63**

Impact on bridging coverage estimate: Low power test methods usually reduce the degree of randomness of data in the scan chains, so the probability of detecting fortuitously the more faults is reduced and quality of test decrease. Here, we use the bridging coverage estimate (BCE) [66] to assess the impact of conventional ATPG test patterns and test patterns generated by different low power ATPGs on quality of test. As shown in Table 3.6, under column ΔBCE, BCE is reduce, on average, by 1.16%, 2.63%,

4.53% and 4.54% using ACF-scan, ACF/PF, PF and 0 Fill. Comparing our proposed methods (ACF-scan and ACF/PF) and PF and 0 Fill in terms of BCE, it can be seen that our methods always achieved better test quality. It is because in PF and 0 Fill methods, unspecified scan chains always fill with constant predefined value, therefore fortuitous detection is much less than when using recycling data.

Impact on ATPG run time: The increase in ATPG run time on average is typically less than 15% for each individual method. This is when the pattern count is relatively unchanged. Of course, if more patterns are generated, the run time increases proportionally.

3.6 Conclusion

Power-aware at-speed tests have become essential for ensuring the quality of submicron designs. In this chapter we present a new technique called ACF-scan for generating test vectors for at-speed delay tests that reduce switching activity during test. ACF-scan uses background states obtained by applying a number of clock cycles to test vectors to fill unspecified values in test cubes. The time to obtain background states depends on the number of additional clock cycles simulated. Experimental results show that after few clock cycles the WSA of background states become stable and applying more clock cycles has no significant effect on reducing WSA.

ACF-scan is shown to achieve substantial reductions in peak WSA of capture cycles of launch-off-capture tests in industrial circuits with no lose in test coverage and minimal increase in test pattern counts. Experimental result shows test pattern generated using ACF-scan method keep the good quality of test.

CHAPTER 4

LOW CAPTURE POWER AT-SPEED TEST IN EMBEDED DETERMINISTIC TEST ENVIRONMENT

Chapter 3 has shown how power overtesting and IR-drop during scan based test in ATPG tools can be minimized by using near to functional backgrounds. However this method cannot directly be used in test compression environments.

This Chapter presents a new scheme to generate pseudo functional test patterns with switching activity close to the switching activity of functional patterns in embedded deterministic test environment [4]. The proposed solution employs extra hardware to allow a given scan chains to be driven directly by EDT decompressor or by near to functional backgrounds. For generating pseudo functional test vector, we first initiate the circuit with a pseudo functional pattern and then apply test cube to the circuit. The proposed technique has a substantial effect on peak power reduction with negligible effect on pattern count or test application time.

The rest of this chapter is organized in the following manner. Section 4.1 describes motivation for low power EDT testing using functional background. In Section 4.2, we describe the proposed low capture power EDT methods. Section 4.3 describes the encoding algorithm. Experimental results for industrial circuits are shown in Section 4.4. The chapter concludes in Section 4.5.

4.1 Motivation

As we describe in chapter 1, Embedded Deterministic Test [4] consists of an n-bit ring generator and an associated phase shifter driving scan chains. Compressed test patterns are delivered to the decompressor through c external channels in a continuous manner, i.e., a new c-bit vector is injected into the ring generator every scan shift cycle moving effectively the decompressor from one of its states to another.

The EDT pattern generation is deterministic. For a given testable fault, as with conventional ATPG, a pattern is generated to satisfy the ATPG constraints and to avoid bus contentions. Test cubes are inherently highly compressible because typically only 1% to 5% of the bits in a test pattern generated by an ATPG tool have specified values. Additionally scan cells in many scan chains may not have any specified values. Consider, as an example, results of experiments done in [58] to analyze the number of scan chains populated by specified bits, and its impact on pattern count. Authors attempted to limit the number of scan chains that feature specified cell contents to 20% and analyzed its impact on pattern count.

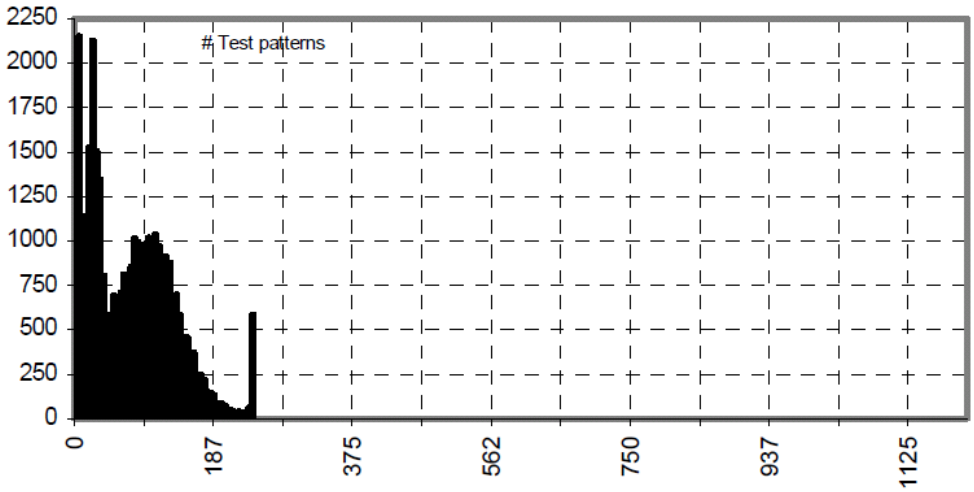


Figure 4.1: Scan chains with specified bits [58]

Figure 4.1 shows the number of test patterns after dynamic compaction on y-axis, which has their specified locations confined to the number of scan chains shown on x-axis. The circuit used in this experiment had 2.5M gates and 143K scan cells forming 1200 scan chains. As can be seen, the vast majority of test cubes feature specified bits in less than 20% of the scan chains. At the same time, limiting the number of scan chains with specified bits cause pattern count increase of 3.4% compared to a test generation

scheme that has no restrictions on scan chains hosting specified bits. It is important to note that there is no fault coverage loss reported due to this technique.

In [20] switching activity caused by different launch of capture (LOC) tests for transition delay faults was analyzed. Recall that LOC tests for TDFs scan in a state, with the scan enable line at 1 (active) to set up the initialization vector of a two pattern test followed by two capture cycles during which the scan enable line is 0 (inactive). In [20] tests generated by several different test generation methods were simulated for several additional (capture) cycles with scan enable at 0 and switching activity was recorded. Figure 4.2 shows WSA as a percentage of the maximum possible value of WSA which occurs if all the gates in the circuit under test (CUT) switch state.

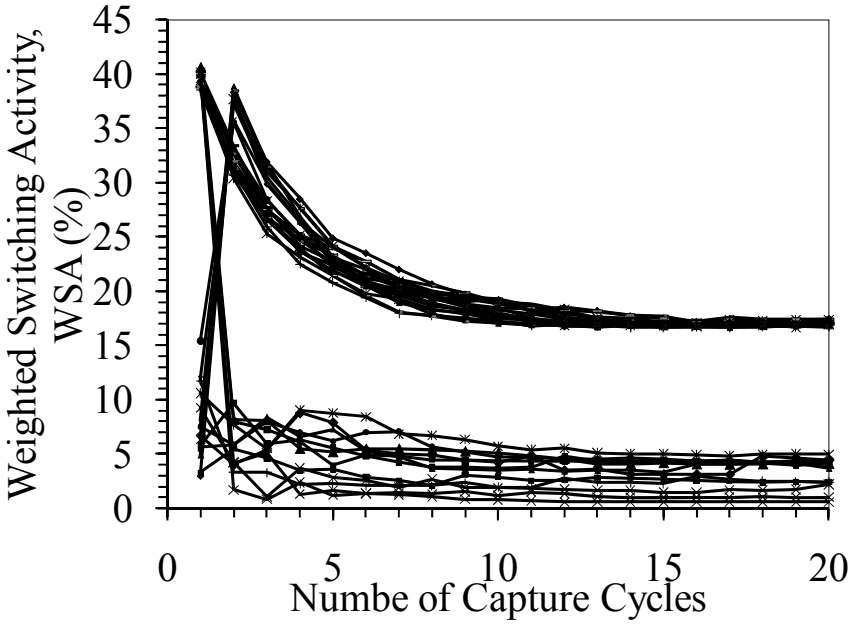


Figure 4.2: Applying 20 capture cycles to LOC test vectors generated using random fill

As seen from Figure 4.2, the WSA after some clock cycles reaches a steady state value which is typically much lower than the WSA during the first two capture cycles.

After several capture cycles the circuit can be expected to enter functional operation and WSA reaches a steady state value. The above motivates the work studied in this Chapter.

4.2 Low Capture Power EDT Architecture

Our proposed method is based on the idea behind the method proposed in previous Chapter [20], now extended to EDT environment. In section 4.1, we show that if one were to fill the unspecified values in the test cubes by pseudo functional background, WSA of the resulting tests can be expected to be low and close to the WSA of corresponding background. The reason for this expectation is that the number of specified values in test cubes is small and hence the test vector obtained after merging test cube with the pseudo functional pattern differs from original pseudo functional pattern only in some positions where the test cube has specified values.

In scan based test, we have complete freedom to assign a desired logic value to each scan cells. But in EDT environment we cannot apply specific value to each scan cells because of encoding capacity. So we need to come up with a new DFT scheme to have both compression and low test power.

As discussed in Chapter 3, if we first initiate the circuit with a pseudo functional pattern and then apply test cube to the circuit, switching activity of the resulting test vectors is close to the switching activity of initial pseudo functional pattern. Since the test responses in LOC tests are obtained through functional mode, we can use the response of test vector as a pseudo functional pattern for filling the unspecified scan chains of the next test cube.

The other observations made in the section 4.1 clearly indicate that very often locations of specified bits can be confined to a few scan chains. Consequently, we can feed the scan chains requiring specified values from a test data decompressor while recycling the pseudo functional response of previous test vector to fill unspecified scan chains of new test cube.

Our low power solution to implement the above idea is shown in Figure 4.3. Input channels provide compressed test patterns to the decompressor. As can be seen, the 2-input multiplexors at the input of scan chains select the stimuli either from the decompressor or from the scan chain outputs. When re-circulating the scan chain outputs, in essence the captured response from the previous vector is used as a background for the current test vector. The decompressor provides the incremental scan data corresponding to the specified bits that are needed for targeted faults. The input channels to the decompressor provide the test data in a compressed form along with the control data for the select input of the multiplexors.

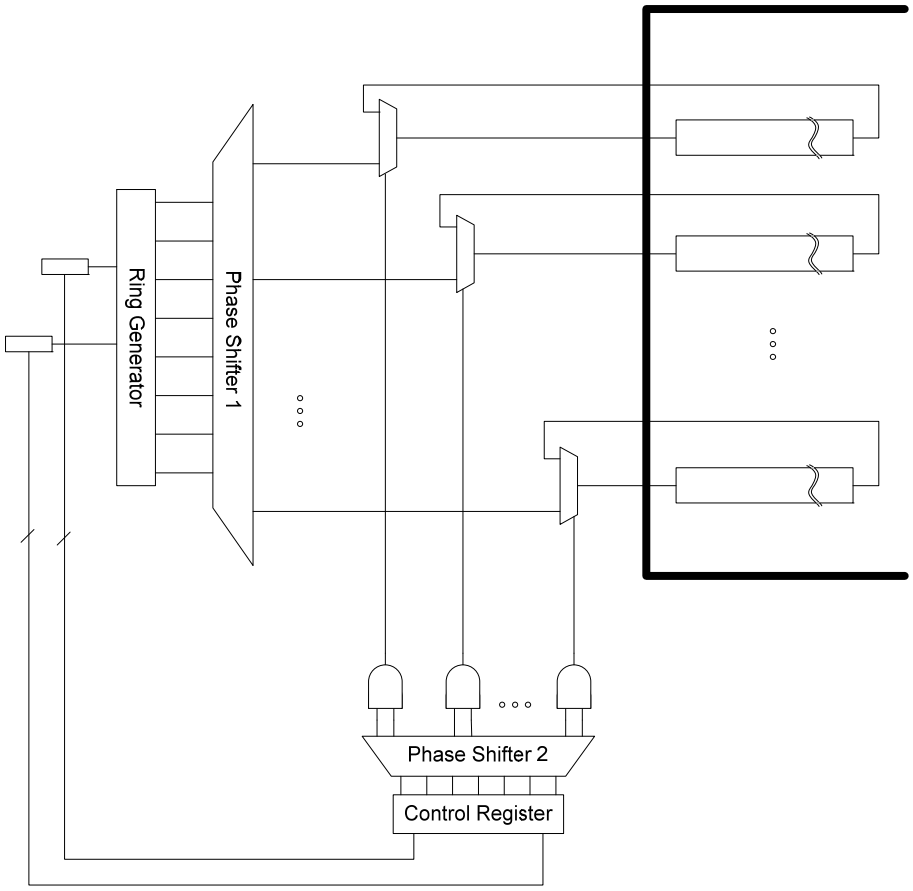


Figure 4.3: Proposed low power EDT architecture

The control block comprises of a control register, combinational XOR network, and some form of biasing logic. The control register stores the control data in a compressed form and when decompressed, it assures that a subset of chains with specified bits receive the stimuli from the decompressor while the remaining scan chains gets the re-circulated scan chain contents. The fraction of scan chains which are driven directly by the decompressor can be changed by adding a biasing circuit, i.e., a group of AND gates driven by the XOR network. For example, if 2-input AND gates (Figure 4.3) are used, then approximately 25% of scan chains are driven by the decompressor (approximately 25% of scan chains have specified bits), while the remaining are re-circulated. This percentage can be reduced even further by adding more inputs to the AND gates. For example, the third input reduces the percentage of scan chains driven by the decompressor down to 12.5%, while the fraction of scan chains getting the scan chains output value increases accordingly. It must be noted that the control data in Figure 4.3 stay constant throughout the entire test pattern.

Below we describe the low capture power test generation flow proposed in Figure 4.3 for EDT environment.

1. *Put limitation on the number of scan chains with specified value (e.g., 25% of scan chains)*
2. *Initiate scan chains with functional patterns*
   a. *Reset circuit, or loading the initial state from an external source*

   Or

   b. *Generate and compress test cube then load test cube to the decompressor, apply some number of functional (capture) cycles to obtain a pseudo functional state.*
3. *Repeat steps 4 - 6*
4. *Generate and compress a test cube for an yet undetected fault.*
5. *Load test cube to the decompressor*

6. *The multiplexors at the input of scan chains select the stimuli either from the decompressor or from the scan chain outputs.*

Although the scheme presented in Figure 4.3 shows that the scan chain are all balanced, the proposed scheme can work for designs that don't have balanced scan chains also. For a design with unbalanced scan chains, chains with similar sizes can be grouped together to form a set. Apart from the longest scan chains, each one of the smaller sets of scan chains require a clock gating circuitry for the shift clocks to allow re-circulation of the captured responses to their respective inputs.

A typical implementation of a clock gating module is shown in Figure 4.4. It consists of a down counter that is loaded with an initial value, which is dependent on the offset with respect to the longest scan chains. When the down counter counts down, the clock gater blocks the clock thereby shutting down the scan chains during those cycles. Once the counter gets to zero, the clock gater opens and the scan chains start loading data either from the decompressor or from re-circulated scan responses.

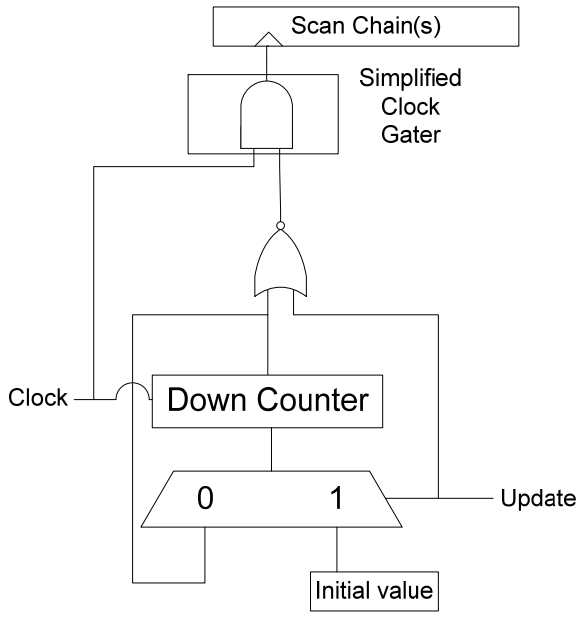


Figure 4.4: Clock gating for unbalanced scan chains

We can provide the ability to encode scan chains driven directly from the decompressor on a per-cycle basis. Instead of having a register to store the compressed control data, we can use a continuous flow decompressor to deliver the control inputs throughout the loading of a test vector. One or more channels can be used to drive the ring generator that is part of the control logic. Since a new set of variables are pumped into the additional Ring Generator in every cycle, this has the ability to encode a large number of scan chain control values on a per cycle basis. In this case we can feed all unspecified scan cells with the value of scan chains output. But it needs more hardware and more control data.

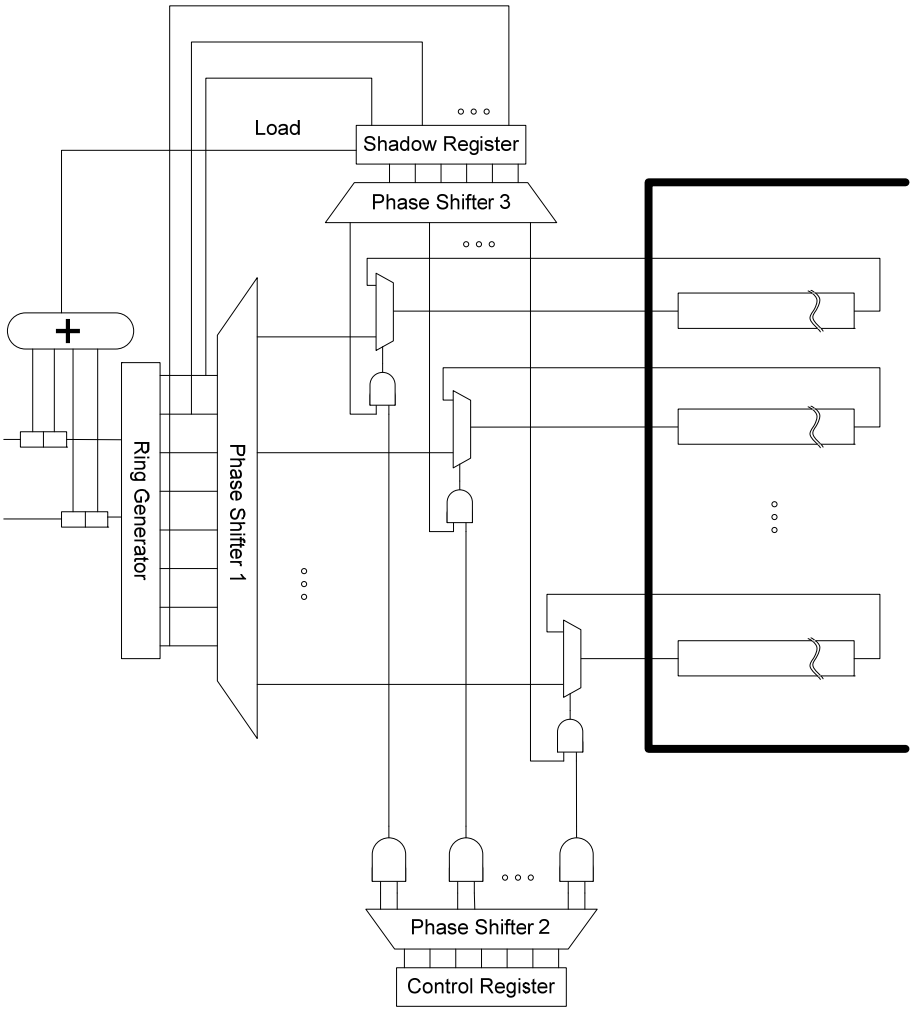


Figure 4.5: Proposed low power EDT architecture using shadow register

Instead we can use a shadow register and a phase shifter in order to keep the difference between background and test cube filled with the background as small as possible (Figure 4.5).

As an example, consider the following test cube shown in Figure 4.6 (It shows only scan chains with specified bits). As can be seen, almost more than half of the scan cells in each scan chain are don't care. Therefore, at every scan shift cycle, we can use the associated control pattern to decide whether a given scan chain receives data from the decompressor, or from the scan chains output.

x x x **0** *x x x x x x x x x x x x x x x x x x x x x x x x* **1 1**
x x x x x x x x x x x **1** *x x x x x x x x* **0** *x x x x x x x x x x*
x x x x x x x x **0** *x x x x x x x x x x x x x* **1** *x x x x x x* **1 1**
x x x x x **1** *x x x x x x x x x x x x x x x x x x x x x x x x x*
x **1** *x x x x x x x x x* **1** *x x x x x* **0** *x x x x x x x* **1** *x x x*
x x x x x x x x x **0** *x x x* **0** *x x x* **1** *x x x x x x x x x x*
0 *x x x x x x x x x x x x x* **1** *x x x x x x x x x x x x x x*
x x x x x **1** *x x x x x x x x x x x x x x x x x x x x x x x x* **1**

Figure 4.6: An example of test cube

The control pattern assumes the value of 1 every time the test cube features a specified value. So corresponding control pattern of Figure 4.6 is of the form below:

x x x **1** *x x x x x x x x x x x x x x x x x x x x x x x x* **1 1**
x x x x x x x x x x x **1** *x x x x x x x x* **1** *x x x x x x x x x x*
x x x x x x x x **1** *x x x x x x x x x x x x x* **1** *x x x x x x* **1 1**
x x x x x **1** *x x x x x x x x x x x x x x x x x x x x x x x x x*
x **1** *x x x x x x x x x* **1** *x x x x x* **1** *x x x x x x x* **1** *x x x*
x x x x x x x x x **1** *x x x* **1** *x x x* **1** *x x x x x x x x x x*
1 *x x x x x x x x x x x x x* **1** *x x x x x x x x x x x x x x*
x x x x x **1** *x x x x x x x x x x x x x x x x x x x x x x x x* **1**

Figure 4.7: Corresponding Control pattern of test cube of Figure 3.14

Control pattern of Figure 4.7 can be efficiently encoded by assuming that the decompressor outputs are sustained for more than a single clock cycle to deliver the identical test data to the AND gate of the multiplexer select inputs for a number of shift cycles. As a result, in addition to scan chains loading data from the scan chains output, one can bring some scan cells in those chains with specified to re-circulating mode, thus reducing the total switching activity even further.

In order to implement the above idea, we use a shadow register to keep the decompressor outputs unchanged (Figure 4.5). It is placed between the ring generator and the phase shifter 3. The shadow register captures and saves, for a number of cycles, a desired state of the ring generator, while the generator itself keeps advancing to the next states in order to encode incoming specified bits. Since in most cases single output of the XOR network is producing 0 with probability 0.5, thus approximately half of the scan chains with specified value get the value of the functional background in every cycle. Note that we reuse the ring generator as its encoding capabilities are sufficient to handle low control pattern fill rates.

There are different ways to encode the information required for updating the shadow register. Here, small buffers placed in parallel with the decompressor inputs drive an XOR tree which computes a parity of input variables, including not only data currently entering the ring generator, but also those used in previous cycles. If the parity of the inputs is odd, then the shadow register is reloaded with the current content of ring generator before new seed variables enter the generator (and it reaches its next state). Otherwise, the contents of the shadow register remain unchanged. Reusing the same control data across multiple clock cycles reduces the stress on the encoding capacity needs for the ring generator, as well as, reduces the control data volume needed to be stored on an ATE.

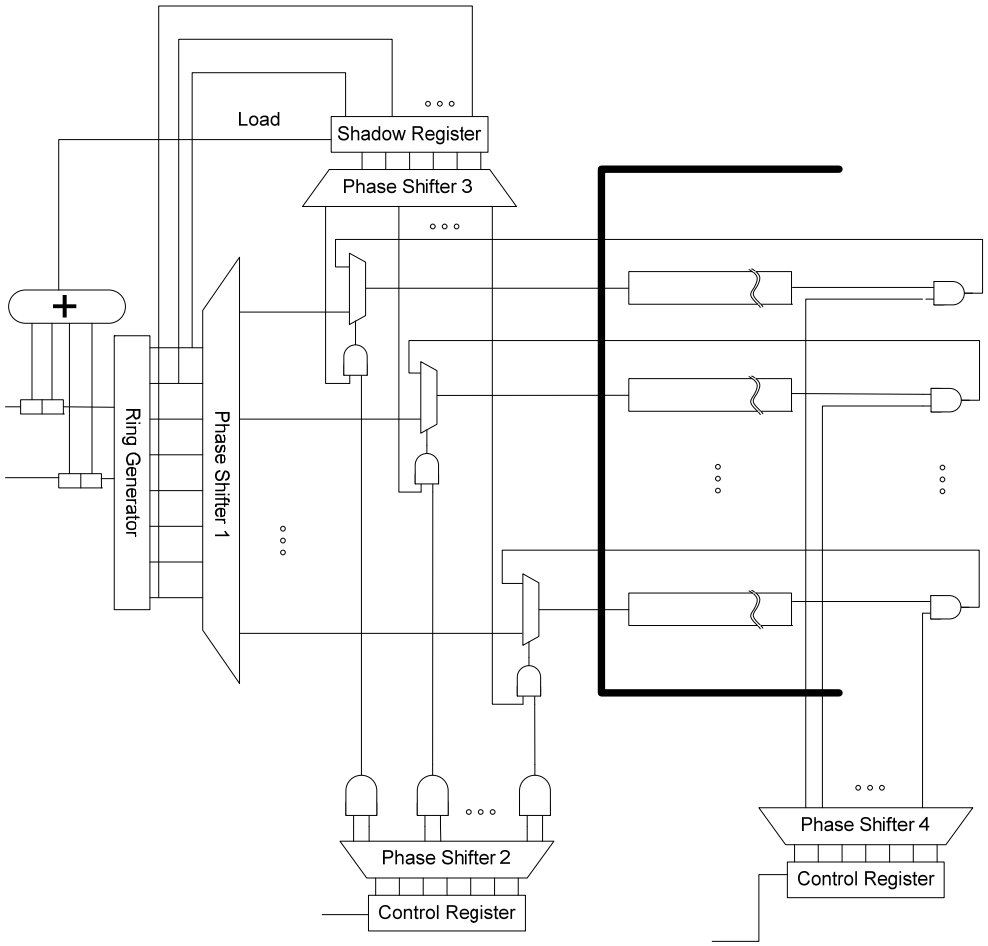


Figure 4.8: Low power EDT Architecture with separate X-masking scheme

When using the captured values in the scan chains as the background for subsequent patterns, one of the requirements is to ensure that the unknowns (X-states) are masked and known values drive the inputs of scan chains. In the scheme presented in Figure 4.3, the scan chain controls can be encoded along with the seed for the specified bit locations, but by also taking into account the need to mask X's when re-circulating the scan data as part of the test stimuli. Since the number of scan chains with specified bits is usually small, the decompressor is able to encode the scan chains that need X-masking as well. One can have a separate signal for the multiplexor inputs (not shown in the figure) such that if the decompressor fails to block a particular cycle of test responses with Xs

due to encoding capacity issues, the control signal can behave as an over-ride forcing the decompressor to drive all scan chains.

Figure 4.8 shows another scheme of the proposed low power EDT, where the X-masking logic is separated from the control logic that encodes the scan chains to be driven directly from the decompressor. As shown in Figure 4.8, control register blocks any Xs that are captured in the test responses. This would serve two purposes – (a) it blocks any Xs from propagating to the output of the compactor thereby enabling the usage of a temporal compactor such as a MISR at the output, and (b) the re-circulated values from the scan chains will have no Xs and therefore can be used to drive the scan chains directly.

4.3 Encoding Algorithm

Similar to the conventional EDT, the low capture power compression of test cubes represent all specified bits by linear functions of variables injected into the decompressor. A compressed pattern is then determined by solving the system of linear equations in the Galois field modulo 2. Here, we divide encoding algorithm into two phases: Encoding procedure for control register, and encoding procedure for the shadow register.

Encoding procedure for control register: As we discussed in the section 4.2, the control register is used to select a subset of chains with specified bits. It must be noted that the actual number of scan chains that can be driven by the decompressor depends on the encoding capabilities of the XOR network and the control register. Since the encoding process is equivalent to solving a set of linear equations, allowing the decompressor to drive a given scan chain with specified bits requires solving two equations (one for each input of the AND gate) per scan chain. If one decides to use 3 input AND gates instead of 2 input AND gate in control block, then three equations are required to determine the status of each scan chain.

We can use the 3-input XOR network to load the control register with all-1 pattern in order to drive all scan chains by the decompressor. This feature can be used to turn off the low capture power decompression circuitry. Figure 4.9 shows the test pattern generation flow for one test pattern that has limited number of specified scan chains.

```
while (not all faults targeted)
    generate and compress test cube for one fault;
    give solver list of chains with specified bits
    solve ignoring power constraints if necessary;
    while (abort and scan chains' limits not reached) {
        give low power solver list of chains driven by decompressor;
        generate and merge test for another fault;
        if (low power solver succeeded) {
            compress new combined test cube;
            if (decompressor solver fails) backtrack solver;
        } else backtrack;
    end while
end while
```

Figure 4.9: The proposed low-power test generation procedure

Encoding procedure for the shadow register: In order to further reduce switching activity in the scan chains with specified bits, a shadow register is placed between the ring generator and phase shifter 3. As mentioned earlier, the control pattern is comprised of only 1-bits and don't care bits. This property is a key factor in reducing the volume of test data as one may deliver the identical data to select input of the multiplexers for a number of scan shift cycles. Figure 4.5 allows the decompressor to drive approximately 12.5% of scan chains with specified value in each time frame, while the remaining ones get the value from scan chains output (functional background). Indeed, a single output of the shift register 2 is producing 0 with probability 0.5 and 1 with probability 0.5, and thus

probability of 0 on select input of the multiplexors in scan chain with specified value is 50%. In overall, probability of 1 on select input of the multiplexers can compute as follow:

- Probability of 1 in one input of AND gate connected to control clock is 25% (section 3).
- Probability of 1 in the other input of AND gate connected to shadow register is 50%.

Therefore, probability of 1 on the output of AND gate (select input of Multiplexer) is 50% * 25% = 12.5%

One can decrease probability of driving a scan chain from the decompressor in every time frame even more by adding some biasing circuitry. The encoding algorithm for shadow register, partitions a given control pattern into several blocks. This allows one to repeat a given combination many times in succession by using the shadow register storing a state that the ring generator entered at the beginning of a block. Such a technique gives the ring generator enough time to compensate for fading encoding ability by collecting new input variables. They will facilitate successful compression during next steps once the shadow register is reloaded.

The ability of a decompressor to decode data within boundaries of the block determines its size. The algorithm uses the following rules for encoding the control patterns:

- It begins with a block and the corresponding state of a ring generator which should be applied first, and it gradually moves towards the end of a control pattern.
- Whenever specified bits are repeated many times in succession within the same block and the same scan chain, there is no need to encode all of them but the first one.

- For each scan shift cycle there is an associated equation representing a request to store, if needed, the content of the ring generator in the shadow register before new variables change the state of the generator during the next cycle.
- An equation representing the first specified bit in the current block and in a given scan chain is expressed in terms of variables injected until this block.

The last rule is needed as a ring generator state which is to cover a given bit has to be completely determined before it is moved to the shadow register during the first cycle of a next block. This is equivalent to conceptually moving this specified bit to the beginning of the block. As long as such bits can be encoded, the algorithm works by repeatedly increasing the size of the block, adding new equations, and invoking the solver again. Clearly, at some point a solution may not exist anymore. This particular time frame is then assigned a new block, and the procedure continues. As a result, we arrive with virtually the smallest number of blocks that cover the entire control pattern.

We will illustrate the basic steps of the encoding algorithm for control pattern by using the control pattern shown in Figure 4.7. It consists of 32 slices (columns) corresponding to successive scan shift cycles, and it is to be loaded into some scan chains that 8 of them have specified bits. Consider a 2-input decompressor in Figure 4.3 and shadow register 2 that controlled by 4-input XOR gate whose inputs comprise the last two variables injected through each input of the ring generator. The input variables a0, b0, a1, b1 ..., are provided in pairs.

Figure 4.10a illustrates a hypothetical partition of the control pattern into blocks with those specified bits that need to be encoded highlighted. By relocating these specified bits to the border lines of the blocks, we get locations corresponding to linear equations that will be used to perform actual encoding (Figure 4.10b).

x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	**1**
x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	**1**	**1**
x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	**1**	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	**1**	x	x	x
x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	**1**	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x
1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**

(a)

x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**
x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x
x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	**1**
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x
x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x
x	x	x	x	x	x	x	x	**1**	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	**1**

(b)

Figure 4.10: Compression of control pattern

Note that a new block is formed (with the shadow register reloaded) anytime a given scan frame has a conflict with the previous slice. Furthermore, a set of linear equations used by the compression procedure must include one additional equation per every scan slice (time frame) of the following form:

$$ak + bk + ak–1 + bk–1 = r,$$

where r is equal to 1, if the shadow register is to be reloaded during the k-th scan shift cycle, and it is equal to 0, otherwise. In general, the number of variables used in the above equation depends on the number of decompressor external channels and the size of input buffers, as shown in Figure 4.3.

4.4 Experimental Results

The proposed low capture power decompression scheme was tested on several industrial designs. The proposed low power schemes were tested on several industrial cores. This section reports results for some of them, ranging in size from 520K to 2.7M

gates. The basic data regarding the designs, including the number of gates and the scan configuration are listed in the left part of Table 4.1. The peak and average switching activity during capture and shift in standard EDT which result in random fill are also given in the right part of Table 4.1. The test cubes were generated using standard EDT to detect all detectable transition faults using launch-off-capture test method.

Table 4.1 Circuit characteristics

Design	Gates	Scan chains (no × size)	WSA (%)		
			Shift	1st capture	2nd capture
D1	260K	300 × 114	49.26	39.34	37.18
D2	470K	500 × 121	49.63	36.49	32.39
D3	460K	832 × 158	48.90	43.27	39.76
D5	1.4 M	1200× 115	49.46	32.09	28.51

We compare the percentage reduction in peak WSA, relative to conventional EDT, caused in the first and the second capture cycles using the methods proposed in this chapter and the method in [58].

Results shown in Table I provide the following information for each test case:

- low-power decompressor method(s):
  - PFB1 used scheme proposed in Figure 4.3
  - PFB2 used scheme proposed in Figure 4.5
  - CL-CG used method proposed in [58], it loads approximately 75% of scan chains with constant value and uses clock gaters to prevent a relatively large number of flip-flops from toggling during capture mode.
- The peak WSA (peak SET) rates for first and the second capture cycles.

- Peak switching activity reduction factor, i.e., the ratio of the standard EDT switching (presented earlier, for each design, in Table 4.1) to the low power switching.
- Peak state element transition reduction factor which is the maximum number of scan cells whose states change during the capture cycles of the tests.

Table 4.2: Comparing capture power reduction in first and second capture cycles

Design	Method	Peak WSA %		Peak SET %	
		1st Cap	2nd Cap	1st Cap	2nd Cap
D1	PFB1	46.47	45.97	43.53	45.87
	PFB2	48.98	49.49	46.24	49.22
	CL-CG	38.92	39.27	47.68	51.23
D2	PFB1	25.33	26.26	27.19	19.06
	PFB2	28.90	26.94	30.92	23.83
	CL-CG	16.74	12.54	19.32	17.71
D3	PFB1	24.06	27.57	23.58	43.63
	PFB2	28.53	29.33	21.76	47.98
	CL-CG	28.54	23.77	39.98	45.06
D4	PFB1	42.76	36.01	18.2	17.05
	PFB2	44.46	37.89	29.32	23.25
	CL-CG	37.1	33.74	18.32	16.5

In all three methods, the threshold was set such that when possible, at least 75% of the scan chains will be loaded with constant values (zeros) in CG or with recycling value of previous pattern response in PFB1 and PFB2.

In Table 4.2 under column "Peak WSA", we show percentage reduction in peak WSA caused in the first and the second capture cycles using PFB1, PFB2 and CL-CG methods. As shown in Table 4.2, peak WSA in the first capture is reduced on average by 35%, 38% and 30% using PFB1, PFB2 and CL-CG methods respectively. It must be noted that with the smaller length of scan chains the better capture power will be achieved. Table 4.2, under column "Peak SET", shows the percentage reduction in peak state element transition (SET), which is the maximum number of scan cells whose states change during the capture mode in first and second capture cycles. It can be seen that peak state element transition (SET) is reduced on average by 35%, 38% and 30% using PFB1, PFB2 and CL-CG methods respectively.

Table 4.3: Comparing shift power reduction

Design	PFB1	PFB2	CL-CG
D1	63.41	63.81	66.52
D2	6.87	6.43	58.48
D3	62.08	67.23	67.28
D4	48.63	51.92	66.83
Average	45.25	47.35	64.78

Table 4.3 shows the average switching reduction percentage during shift mode. As shown in Table 4.3, PFB1 and PFB2 methods reduce the shift power on average by

45% and 47% respectively while CL-CG method reduces shift power on average by 65%.

Impact on pattern count: Application of tests in the low power mode reduces the degree of randomness observed in the scan chains. Additionally, restricting the number of scan chains with specified value may restrict dynamic compaction and therefore usually leads to higher pattern counts. In Tables 4.4, under column "ΔPC", the test pattern difference between unconstrained EDT and EDT with the low power schemes is represented. One can notice that using PFB1 and PFB2 the test pattern counts increase, on average, by 21% and 26% respectively instead of 42% when CL-CG is used. The pattern count increase varies per design and reflects the extent to which the low power scheme constrains dynamic compaction

Table 4.4: Comparing pattern counts and bridging coverage estimate

Design	ΔPC			ΔBCE		
	PFB1	**PFB2**	**CL-CG**	**PFB1**	**PFB2**	**CL-CG**
D1	15.54	17.23	31.65	0.36	0.47	2.24
D2	24.35	30.30	47.96	0.39	0.71	1.34
D3	24.79	33.44	42.34	0.86	0.91	1.19
D4	20.65	22.87	47.18	0.57	0.65	1.98
Average	21.33	25.96	42.28	0.55	0.69	1.69

Impact on bridging coverage estimate: Low power test methods usually reduce the degree of randomness of data in the scan chains, so the probability of detecting fortuitously the more faults is reduced and quality of test decrease. As shown in Table

4.4, under column "ΔBCE", BCE is reduce, on average, by 0.55%, 0.69% and 1.69% using PFB1, PFB2 and CL-CG. Comparing our proposed methods (PFB1, PFB2) and CL-CG in terms of BCE, it can be seen that our methods always achieved better test quality. It is because in CL-CG method, unspecified scan chains always fill with constant value "0", therefore fortuitous detection is much less than when using recycling data.

Impact on ATPG run time: The increase in ATPG run time on average is typically less than 15% for each individual method. This is when the pattern count is relatively unchanged. Of course, if more patterns are generated, the run time increases proportionally.

4.5 Conclusion

This Chapter presents a novel low capture power test compression scheme in EDT environment that is able to generate test vectors that mimic functional operation from switching activity point of view.

The proposed solution employs a controller that either allows a given scan chain to be driven by the EDT decompressor or recycle the response of previous pattern to fill unspecified bits of test cubes. We try to generate the test vector responses that are close to pseudo functional patterns. For filling unspecified scan chains, we recycle the response data for the previous pattern. Since we initiate the circuit with a pseudo functional pattern and only change a few percentages of bits during test pattern generation, the resulting test vector is a pseudo functional pattern.

Experimental results show that for industrial designs, our scheme in Figure 4.3 reduces the peak capture switching on average by 36% while keep the BCE very close to BCE obtained by conventional EDT.

CHAPTER 5

LOW POWER COMPRESSION UTILIZING CLOCK-GATING

As devices grow in gate count, scan test data volume and application time grows as well. For larger designs, the growing test data volume has significantly increased the test cost because of excessively long test time and elevated requirements of tester memory and external test channels. Most of the low power techniques proposed so far for scan design or test data compression techniques increase pattern count. Even thought our proposed methods have better pattern counts compared to the previous works, but still we have higher pattern counts compare to conventional ATPG and EDT. As both large test data volume and high test power are major concerns for the industry today, it is essential to develop a solution that takes both problems into account.

In this Chapter, a new low power compression scheme utilizing clock gater circuitry to minimize the number of scan chains that are loaded and observed in each test phase is presented. Using this technique, transitions in the scan chains during both loading test stimuli and unloading test responses decreases which results in the acceleration of the speed of shifting and an increase in the number of cores that can be tested in parallel. On the other hand, by maximizing the compatibility between test cubes and freezing the compatible scan chains content, test data volume and fill rate is reduced and therefore, it is applicable in any existing test data compression techniques.

The rest of this Chapter is organized as follows. Section 5.1 gives preliminaries and motivation of this work. Section 5.2 describes the proposed compression and test generation algorithm. In Section 5.3, the proposed low power compression architecture is discussed. Experimental results for stuck-at faults test on industrial circuits are shown in Section 5.4. Section 5.5 gives motivation of low power compression for transition faults. In Section 5.6, an enhanced launch-off-capture test method using clock gater circuitry is proposed to reduce the test data volume and test power of transition faults

simultaneously. Section 5.7 describes the test cube merging algorithm for transition faults. Experimental results for transition faults on industrial circuits are shown in Section 5.8. Section 5.9 concludes this paper.

5.1 Basic Concepts and Motivation

Test cubes tend to be correlated. Faults that are structurally related require similar input bits in order to be sensitized to an output. Thus, many test cubes in the test set will have many similar specified positions and a few conflicting positions.

Typically, ATPG tools or test data compression techniques uses cube merging to reduce pattern counts. A test pattern is gradually developed by merging compatible test cubes with appropriate values assigned to unspecified positions. Two cubes are compatible if in every position where one of the cubes has a value of 0 or 1, the other one either features the same value or a don't care.

In [67], a test data compression scheme was proposed which exploits the occurrence of similar vectors in test stimuli to increase the compression ratio and reduce test data volume. The method merges test cubes with many similar specified bits despite conflicts on certain positions. It partitions test cubes to different clusters each contains one so-called parent pattern and a number of its derivatives obtained by imposing some extra bits on the parent pattern. For loading the test cubes of each cluster to the scan chains, it repeatedly applies the same parent pattern which is compatible with all the other incremental test cubes in the cluster, every time using a different incremental test cube. However, this scheme may limit the number of cores that are tested in parallel due to high power consumption. This is attributed to the amount of flip-flop toggling where the switching activity typically goes well beyond that of the mission mode.

Instead of shifting the parent pattern every time for all the test cubes in the cluster, scan chains including difference bits can be reloaded every time while the remaining scan

chains can hold their previous content. This can reduce the amount of switching activity during shift and test data volume simultaneously.

Definition 5.1: Two chains are compatible if they do not specify complementary values in any bit position.

Definition 5.2: Two chains are conflicting if they specify with complementary values at one or more bit positions.

Definition 5.3: Two test cubes are compatible if in every position where one of the test cubes has a value of 0 or 1, the other one either features the same value or don't care.

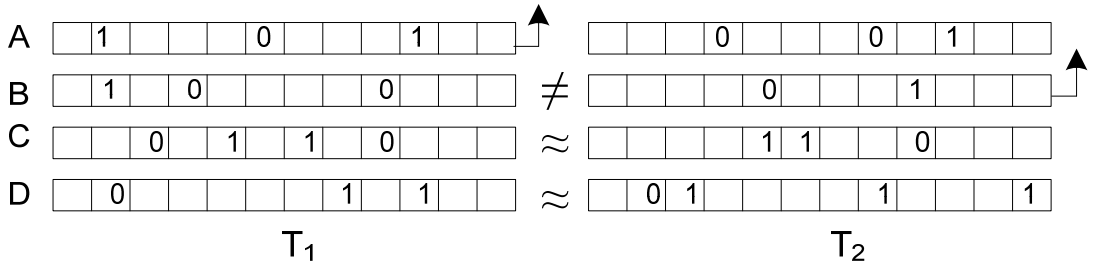
Example 5.1: Two incompatible test cubes T1 and T2 given in Figure 5.1a. Assume that circuit has 4 scan chains A, B, C and D each of which has an independent clock control that can be enabled/ disabled separately.

The observation points for detecting faults targeted with test cube T_1 and T_2 are located in scan chain A and B, respectively (shown with an upward arrow in Figure 5.1a). Therefore for detecting targeted faults, it is sufficient to only capture and observe the test results on the observed scan chains and hold the value of the remaining scan chains.

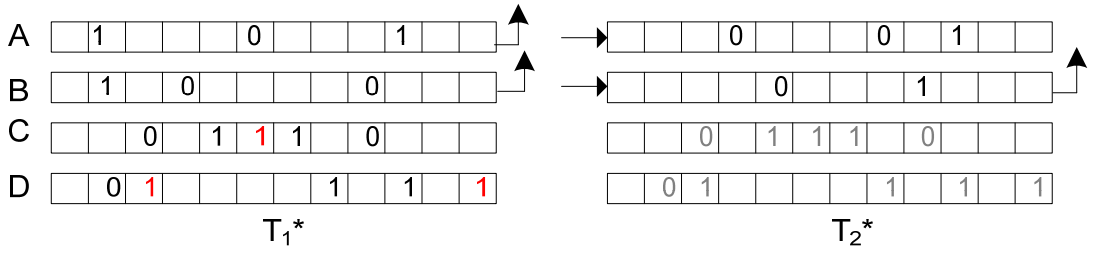
Comparing test cube T_2 to T_1, some conflicting positions in scan chain B can be observed and the remaining scan chains are all compatible. Therefore, we can merge the values of scan chains A, C, D of test cube T_2 with test cube T1. However, for observing the fault effect of test cube T_1, scan chain B should capture the test results and it cannot hold the same value as it is loaded with. Therefore, we can only merge compatible scan chains that are disabled during capture cycles and hold their values. In this example, we can only merge the scan chains C and D of test cube T_2 with the corresponding scan chains in test cube T_1.

Figure 5.1b shows new test cube T_1 (T_1*) after merging with compatible part of test cube T_2. As can be seen, instead of encoding all 7 specified bits in scan chains C and D of test cube T_2, we only need to encode 3 more positions in test cube T_1 and then retain

the value of scan chains C and D while the next test cube T_2 is loaded to the scan chains. This will reduce the total amount of test data as well as test power.



(a)



(b)

Figure 5.1: Two consecutive test vector

Since the unload of test results of T_1 happens at the same time as the load of test pattern T_2, in addition to the conflicting scan chain B, the observed scan chains of test cube T_1 (A) is also required to be reloaded with new values. Therefore, the scan chains A and B are reloaded with new test data as their captured response is shifted out and the remaining scan chains retain their previous states.

The observation presented above motivates the low power test architecture described in the following sections.

5.2 Cube Merging and Compression for stuck-at faults

In the conventional scan-based test method, each test pattern shifts values into all scan chains which results in an increase in switching activity in the circuits and consequently, higher power consumption during the shift phase.

In this section, to reduce the test data volume and power consumption in the circuit, a new low power compression scheme is presented which is capable of generating and ordering test cubes in such a way that compatible scan chains between consecutive test patterns are loaded into the scan chains only once while conflicting scan chains are reloaded during applying each test pattern.

The effectiveness of the proposed scheme is determined by the number of compatible scan chains which can hold their values between each load. To maximize the number of compatible scan chains to be shared with different test patterns, a dedicated merging algorithm is required.

In algorithm describe below, stuck-at faults model is used to generate test cube. In section 5.5, a modified algorithm for generation low power compression scheme for transition faults will be described. In this algorithm, a test cube consists of a set of necessary scan cells with necessary or specified assignment to them, and the location of observed scan cells. A test vector is generated by merging some number of compatible test cubes capable of detecting both faults targeted by the merged test cubes and additional faults fortuitously due to random fill of unspecified values in the test vector.

In the proposed algorithm, during the generation of each test vector, the scan chains to be loaded and unloaded must be selected based on the current and previous test vector. The number of scan chains selected for loading and unloading cannot exceed the given threshold Δ and δ, respectively. Since the unload of the test response to vector T_{i-1} happens at the same time as the load of vector T_i, content of scan chains that are unloaded to detect the faults targeted by pattern T_{i-1} will be loaded with new data during shifting pattern T_i. Therefore, in order to have more freedom for generating the next test vector T_i,

δ is set to be less than Δ. This means the algorithm considers up to Δ scan chains for loading with new values in each pattern. Similarly, the algorithm can only consider δ scan chain to be unloaded even though the final pattern will unload the same number of scan chains as will be loaded.

The pseudo-code for the compression algorithm is shown in Figure 5.2.

The proposed compression algorithm is explained through an example shown in Figure 5.3 which represents the process of generating five different consecutive test vectors. In this example, the hashed scan chains are loaded with new data while the blank scan chains hold previous pattern values. Upward arrows indicate the scan chains required to be unloaded to observe the fault effects and arrows with dot at one side indicates the scan chains with conflicting values with the previous test vector. Also buffer B is considered to consist of test cubes targeting only one primary stuck-at fault, C_{load} and C_{unload} are considered to include the scan chains that should be loaded and unloaded for each test vector and the thresholds Δ and δ are assumed to be 2 and 1, respectively.

During the process, the test vector T_1 is initially generated by merging all compatible test cubes in the buffer such that the number of scan chains including observation points (C_{unload}) does not exceed the given threshold δ and compression does not fail.

Scan chain 1 is the observed scan chain for test vector T_1. For the first test vector, all the scan chains must be loaded with test stimuli to initiate the scan cells with a known value. In order to avoid compression failure and high peak shift power during loading first vector, loading test stimuli of the first vector will be done in multiple steps. It must be noted that the random fill process of unspecified bits of test vector T_1 is postponed in order to maximize the compatibility between test vectors and to be able to add more deterministic bits to the scan chains that hold their value during unloading the test result of T_1.

for (every fault in fault list)
Generate a test cube and Put test cube in buffer.
end for
Set C_{load} and C_{unload} to be the list of scan chains need to be loaded and unloaded for each test vector.
Set P to be an all-X pattern
while ($C_{unload} < \delta$)
Select one test cube from buffer and merge with P.
Add observation scan chains to C_{unload}
end while
Add P to test set T.
while (buffer is not empty)
$C_{load} = C_{unload}$, $C_{unload} = \emptyset$
Set P to be an all-X pattern
while($C_{unload} < \delta$ and $C_{load} < \Delta$ **)**
Sort test cube in buffer based on the highest degree of similarity with previous test cube.
Pick first compatible test cube with P from buffer.
if (no compatible test cube in buffer)
exit while loop
Add new load and unload chains to C_{load} and C_{unload}.
Compress the specified bits on C_{load}.
Move the specified bits of remaining chains to the test vector in which the scan chains are reloaded.
Try to compress new combined test cube.
if (compression fails)
exit while loop
end while
If ($C_{load} < \Delta$**)**
Randomly select (Δ - C_{load}) unselected chains and add to C_{load}.
for (each test vector P of set T)
if (all the scan chains have been loaded at least one in the successor test vector P){
decompress compress test vector P
perform fault simulation
}
end for
end while

Figure 5.2: Generating of low power compressed pattern

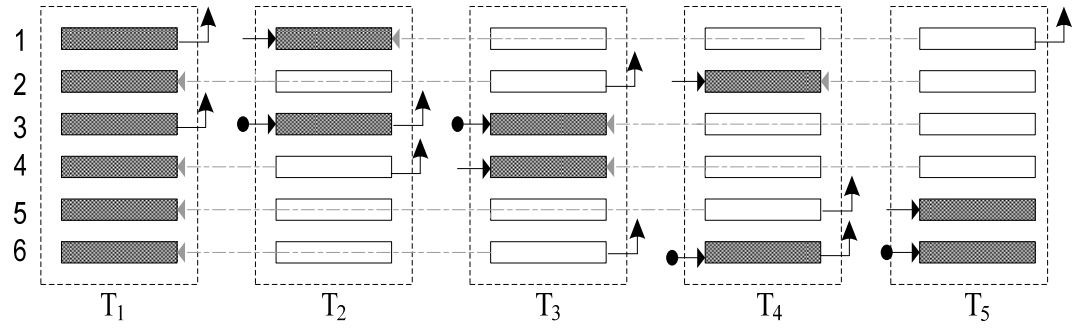


Figure 5.3: An example of merging and generating test cubes

Now, the next test vector T_2 is generated based on the test vector T_1. In this step, since C_{unload} of the test vector T_1 should be loaded with new value during shifting test vector T_2 into scan chains, C_{load} of T_2 is set to include the scan chains in C_{unlod} of T_1. Also T2 is set to be an all-X vector.

For every remaining test cube in the buffer, the number of specified bits that each cube has in common with test vector T_1 is determined. It must be noted that, only the values of scan chains that are not included in C_{load} are compared with test cubes in buffer. Based on this fact, since the scan chain 1 is included in C_{load} of T_2 (scan chain 1 is the observed scan chain for T_1), only the values of scan chains 2 to 6 are compared with the cubes in the buffer.

The algorithm also checks each test cube to determine if its number of conflicting scan chains with T_1 plus C_{load} is greater than Δ. If it is, that test cube will be deleted from the list of cubes that could be merged with T_2.

Subsequently, the entire list of candidate test cubes are scanned and the one with the highest degree of similarity with T_1 (the largest number of common specified bits) is picked to create T_2.

Then, all statistics regarding the remaining test cubes (number of common specified bits with current value of scan cells) are updated. The merging algorithm for generating test vector T_2 continues by greedily adding first compatible test cube from the

sorted buffer to T_2 until no test cubes can be chosen due to incompatibility constrains imposed by factors δ and Δ or compression failure.

If the total number of scan chains that should be loaded for T_2 (C_{load}) is less than Δ, (Δ - C_{load}) scan chains are randomly selected and placed in C_{load}. In this regard, those scan chains that have not been loaded with a new value for a long time are tried to be recognized and selected.

In the test vector T_2, only the specified bits that are included in C_{load} are needed to be encoded and the specified bits in the remaining scan chains 2, 4 and 5 are moved and merged to the test cube T_1. The scan chains whose specified bits are moved to the previous vectors are shown with dashed backward arrows. One has to make sure, however, that the additional specified bits added to T_1 can be still encoded.

After finishing with the test vector T_2, the generation of the next test vector T_3 starts. Scan chains in C_{unload} of T_2 (scan chain 4) are placed in C_{load} of T_3. Then the same procedure as mentioned above is executed to generate test vector T_3.

For test vector T_3, only the specified positions are included in C_{load} are encoded while the remaining specified bits are moved to the nearest test vector in which the scan chains are reloaded with the new value. For example, all specified bits in the scan chain 1 of T_3 are moved to the test vector T_2 in which scan chains 1 is loaded with new values. All the specified bits in the remaining scan chains 2, 5 and 6 are moved to test cube T1 in which they are loaded with new values.

As can be observed in Figure 5.3, by the time test vector T_5 is generated all the scan chains are reloaded at least once. Therefore, none of specified bits of successor test cubes can be moved to T_1. Therefore, all don't care bits in the scan chains included in C_{load} of T_1 can now be filled with random values and fault simulation can be done on T_1. The same procedure is followed until no test cubes remain in the buffer.

5.3 Low Power Test Architecture

Figure 5.4 shows the general structure of a new low power test data decompressor unit employing the approach presented in Section 5.2. In addition to decompressor and compactor, a programmable control block is used to independently enable or disable the clock of each scan chains.

Input channels provide compressed test patterns to the decompressor. The same channels can be used to deliver information to a control block. The control block comprises of a control register, combinational XOR network, and some form of biasing logic. The control block outputs provide gating signals for controlling the clock of each scan chain so that when gating signal is 0, the corresponding scan chain will not receive any clock transitions either in shift mode or in capture mode and no clock power will be dissipated for those scan chains. When gating signal is 1, the corresponding scan chain will be driven directly by decompressor.

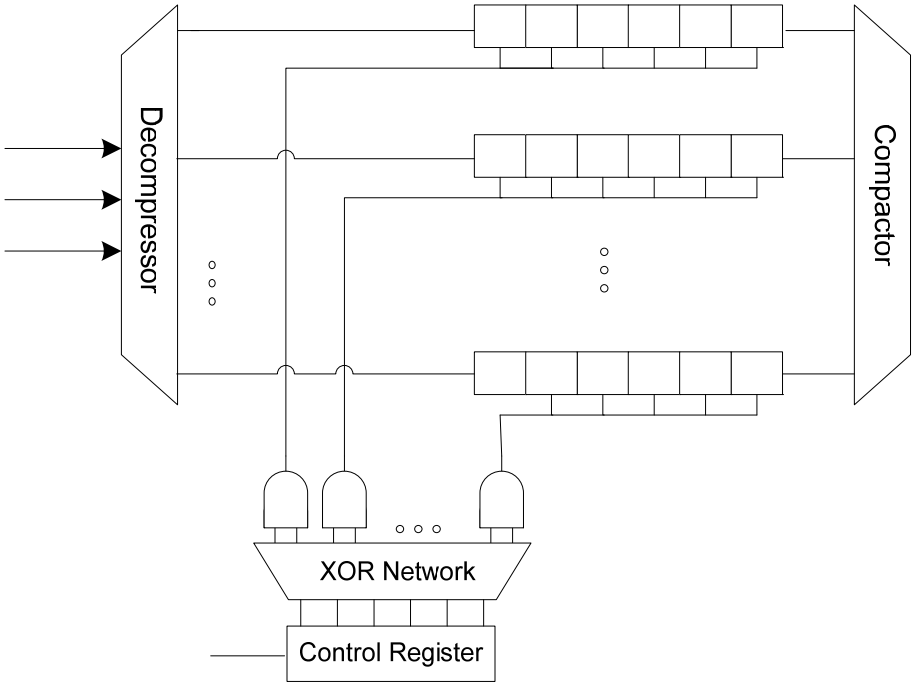


Figure 5.4: Low power test architecture

The fraction of scan chains which are driven directly by the decompressor can be changed by adding a biasing circuit, i.e., a group of AND gates driven by the XOR network. For example, if 2-input AND gates (Figure 5.4) are used, then approximately 25% of scan chains are driven by the decompressor while the remaining ones retain their values. This percentage can be reduced even further by adding more inputs to the AND gates. For example, 3-input AND gates reduce the percentage of scan chains driven by the decompressor down to 12.5%, while the fraction of scan chains holding their states increases accordingly. The procedure we used for encoding the control data is the same as the procedure used in [58 and 23].

Encoding procedure for control register: It is worth noting that the actual number of scan chains that can be driven by the decompressor depends on the encoding capabilities of the XOR network and the control register. Since the encoding process is equivalent to solving a set of linear equations [4], setting a gating signal to a pre-specified value requires one or two equations. For instance, allowing the decompressor to drive a given scan chain requires solving two equations xi=1 (one for each input of the AND gate). For scan chains that need to hold their value, one equation xi = 0 is enough to set the output of AND gate to 0 (any input of AND gate can be chosen to set to 0).

5.4 Experimental Results for Stuck-at Faults

The two primary objectives of the experiments reported in this section are to measure the amount of test data volume and toggling and compare the switching activity and test data volume when applying test patterns produced by a standard compression result in random fill such as EDT [4] and by using the proposed low power compression, respectively.

The proposed low power schemes were tested on several industrial cores. This section reports results for some of them, ranging in size from 260K to 1.4M gates. The basic data regarding the designs, including the number of gates and the number of scan

cells are listed in the left part of Table 5.1. The results of the experiments for stuck-at faults using a standard compression which result in random fill are also given in the right part of Table 5.1. Consequently, the next four entries to the Table 5.1 report the resultant number of test patterns, the total number of specified bits needed to be encoded for all pattern, the peak and average switching activity during shift.

Table 5.1 Circuit characteristics

Design	Gates	Scan cells	Test Pattern	Specified bits	Swathing Activity (%)	
					Capture	Shift
D1	260 K	14 K	3168	1,197,366	19.36	50.1
D2	470 K	21 K	10273	2128345	23.09	48.6
D3	460 K	27 K	918	1,145,765	28.9	46.6
D4	890 K	79 K	8346	1,845,607	21.6	49.9
D5	1.4 M	60 K	5789	10,431,849	19.64	49.5

Results of the experiments for stuck-at faults using low power compression are summarized in Table 5.2 and Table 5.3. A biasing circuitry of Figure 5.4 is deployed to help maintaining the amount of enabled scan chains at a desired level (25% of scan chains). The first part of the Table 5.2 consists of two columns specifying the scan configuration and the control register size. The size of control register depends on the number of scan chains and it is chosen in such a way that high encoding efficiency is achieved [58].

The remaining parts of Table 5.2 provide the following information for each test case:

- Total number of specified bits needed to be encoded for all patterns (control data is not included) using low power compression.
- The fill rate reduction of the low power scheme, i.e., the ratio of the total number of specified bits as presented (for each design) in Table 5.1 and the total number of specified bits that have to be encoded for low power compression scheme; this quantity includes the total number of specified bits across all test patterns reported in table 5.2 under column "Specified bits" as well as the total number of control bits - the latter value is equal to the number of test vectors multiplying by the size of control register.
- Impact on pattern count
- Impact on bridging coverage estimate.

As shown in Table 5.2, under column "Reduction of fill rate", the average fill rate reduction computed across examined industrial designs is equal to 1.98 with the maximal reduction elevated to the value of 2.5. In fact, low power compression has achieved on average 1.98X reduction of test data volume with the switching activity at the level of 10% to 16% only. For each design, two different scan configurations are used. As can be seen, by reducing the number of scan chains and the size of control register, the control data volume is reduced and better test data volume is achieved. However, for design D5, the fill rate reduction of second configuration which has more scan chains is slightly better as the amount of control data comparing to the specified bits needed to detect the faults is negligible.

In presented experimental results, each test pattern has an individual control data. However, the amount of control data can be reduced by sharing the same control data between different patterns such that the control data for multiple test patterns is loaded into the programmable controller only once and a unique control data is stored in the external tester.

Table 5.2 Result for test data volume reduction and pattern count

Design	Scan chains (no × size)	Control register	Specified bits	Reduction of fill rate	ΔPC (%)	ΔBCE (%)
D1	50 × 280	18	456395	2.31	9.97	-1.63
	100 × 140	35	403858	2.28	8.35	-2.09
D2	50 × 420	18	670200	2.5	-2.4	-1.34
	100 × 210	35	601217	2.24	-3.53	-1.89
D3	100 × 270	35	7175645	1.78	10.7	-0.08
	200 × 135	56	6044815	1.61	8.6	-0.1
D4	150 × 527	42	731253	1.72	-2.26	-1.28
	300 × 264	84	604078	1.44	-3.73	-1.36
D5	200 × 300	56	5012108	1.94	10.5	-0.1
	400 × 150	110	4500311	2.01	9.1	-0.12
Average				1.98	4.53	-1.00

In Table 5.2, the test pattern count difference between standard compression and the low power compression is represented by ΔPC which varies per design. In some examined test cases (designs D1, D3 and D5), the low power compression results in slightly higher pattern count. It is because disabling some scan chains during test may reduce fortuitous detection of non-targeted faults and lead to higher pattern counts. However, for designs D2 and D4, despite disabling over 75% of scan chains during test pattern generation, because of choosing the best test cube for merging by the compression algorithm, a better pattern count is achieved.

Furthermore, the bridging coverage estimate [66] has been used to assess the impact of standard compression test patterns and low power test patterns on bridging defects. Data collected in the column ΔBCE of Tables 5.2 indicates that the negative impact of low power compression on the bridging coverage is negligible. In summary, results confirm that the presented low power compression does not compromise the quality of test.

Table 5.3 Result for test power reduction

Design	**Scan chains (no × size)**	**Shift power reduction (%)**		**Capture power reduction (%)**	
		Peak	**Average**	**Peak**	**Average**
D1	50 × 280	77.46	79.84	40.60	49.57
	100 × 140	80.66	82.24	44.57	50.81
D2	50 × 420	79.98	80.1	23.15	48.44
	100 × 210	81.64	81.9	26.52	49.85
D3	100 × 270	67.20	68.63	40.60	35.10
	200 × 135	69.06	70.59	41.92	40.29
D4	150 × 527	74.97	78.70	32.42	31.47
	300 × 264	79.11	80.50	34.70	32.79
D5	200 × 300	83.80	85.26	39.84	45.36
	400 × 150	84.21	87.50	40.94	45.99
Average		**77.81**	**79.53**	**36.53**	**42.97**

Table 5.3, under column “Shift power reduction”, shows the peak and average shift switching reduction factor across all test patterns, i.e., the ratio of the standard compression switching to the low power switching. Also, the peak and average capture switching reduction factor across all test patterns are shown under column “Capture power reduction” in Table 5.3.

As indicated by data under columns “Shift power reduction” and “Capture power reduction” of Table 5.3, due to deactivate a significant portion of the clock tree in the proposed scheme, a substantial reduction in both peak (with averages of 77% and 36% during shift and capture, respectively) and average power (with averages of 79% and 43% during shift and capture, respectively) is achieved. However, for design D2, the peak switching activity during capture is reduced only by 23%. This is because of larger fan-outs in some scan chains which affects more gates and increases switching activity.

Table 5.4 shows the result of the same experiment using multiple-detect ATPG [66]. Multiple-detect ATPG was first defined in Random Excitation and Deterministic Observation (REDO) scheme [77]. In this scheme, in order to reduce the overall defective part level for a device, observability of each site was increased by targeting stuck-at faults multiple times. The ATPG algorithm used random decision order in fault targeting and fault simulation was modified to drop faults from the list only when they were targeted and detected a specified number of times. Multiple-detection ATPG usually increases the test data volume and test pattern count.

The results of the experiments using multiple-detect ATPG for our proposed low power compression including peak capture power reduction percentages, peak shift power reduction percentages, test data volume reduction, pattern count increase, and BCE reduction are listed in Table 5.4. As can be seen, the proposed low power compression technique reduces the shift and capture power significantly even by using multiple-detect ATPG.

Table 5.4 Result for multiple-detection ATPG

Design	Scan chains (no × size)	Peak capture power reduction %	Peak shift power reduction %	Reduction of fill rate	ΔPC (%)	ΔBCE (%)
D1	50 × 280	43.68	77.05	1.88	2.71	0
D2	50 × 420	20.8	78.15	2.11	-1.11	-0.02
D3	100 × 270	37.67	68.96	1.52	4.6	-0.02
D4	150 × 527	32.59	72.88	1.43	-3.08	-0.08
D5	200 × 300	36.19	88.26	1.71	-6.37	-0.01
Average		34.19	77.06	1.73	-0.65	-0.03

The average fill rate reduction computed across examined industrial designs is equal to 1.7 using multiple-detection ATPG with the proposed low power compression method. Data collected in the column ΔPC of Tables 5.4 indicates that the proposed low power compression technique using multiple-detect ATPG reduces pattern count on average by 0.6 % and it has no negative effect on the number of patterns. Data collected in the column ΔBCE of Tables 5.4 indicates that the negative impact of low power compression on the bridging coverage using multiple-detect ATPG is negligible (less than 0.03 %). Therefore the proposed low power compression method is able to significantly reduce test data volume and test power with no penalty in test efficiency, test pattern and test quality.

5.5 Motivation for Low Power Compression of Transition Faults

As VLSI design sizes and their operating frequencies continue to increase, the timing-related defects become more important. In this case, the stuck-at fault test cannot

ensure high quality level of chips and the role of at-speed delay test in maintaining the product quality level becomes more significant [14, 73-74].

The launch-off-shift (LOS) and launch-off-capture (LOC) techniques are commonly used to test transition faults in the circuits. The LOS technique launches the transition by shifting the initialization pattern one more bit [7]. The LOC technique launches a transition through functional clock instead of shift clock. In LOC, the second pattern is just the functional response of the circuit to the first pattern [8]. Typically, the LOC test method achieves lower fault coverage and generates larger test pattern set compared to the LOS test methods [14]. However, the LOC test method is often preferred due to the fact that it does not require at-speed switching of the scan enable line which is required for the LOS tests methods.

Several techniques [70-72] have been proposed to improve the LOC fault coverage. In [70 and 71], circuits are partitioned into two regions. One region is controlled by slow scan-enable signals to be tested by LOC test. The other region is controlled by locally generated fast scan-enable signals and tested by LOS test. These methods partition circuits into many regions by a controllability measure or a developed cost function. The quality of the partitioning determines the effectiveness of the methods. In some cases, the number of patterns generated by the hybrid method exceeds the LOC pattern count.

Another approach was to implement multiple scan-enable signals [72]. These scan-enable signals do not require fast switching capability, but are controlled separately. In other words, while some scan-enable signals are de-asserted to perform LOC test, other scan-enable signals are kept high, keeping flip-flops in the shift mode. Therefore, some flip-flops perform launch and capture while other flip-flops only launch transitions by shift operations.

Meanwhile, as also discussed in Chapter 1, because of the direct effect of peak power on timing, the delay testing is especially sensitive to excessive power

consumption. Hence, there is the concern of over-testing due to excessive power consumption during at-speed delay tests [13, 19]. Several different methods have been presented that reduces power during built-in self-test (BIST) and test data compression [17, 75]. The scan enabling techniques, which activate only a small number of scan chains at a time, have been proposed in [51, 52 and 76]. These scan enabling techniques assume a combinational circuit and fault model, and they are not directly applicable to multicycle delay testing. In next section, the low power compression technique proposed for stuck-at faults model to reduce the test data volume and power consumption in the previous section is extended for launch-off-capture test method. The proposed techniques uses systematically clock gater circuitry to reduce test data volume and test power by enabling only a subset of the scan chains with specified bits in each test phase. However, for transition faults, in addition to scan chains with specified bits, the scan chains with no specified bits that launch and capture the fault effect also need to be enabled during capture cycles which increase the number of active scan chains in each test phase. In the next section, a new sequential test generation method is proposed to reduce the number of active scan chains during capture cycle.

5.6 Proposed Launch-off-Capture Test Cube Generation

The hardware architecture of proposed low power compression method for transition faults is the same as the one used for stuck-at faults (Figure 5.4).

As discussed in Section 5.3, in the proposed low power compression method, in order to reduce both test power and test data volume, the clock gating circuitry is used for each scan chains to limit the number of flop allowed to toggle during shift and capture and freeze the compatible scan chains content between consecutive test pattern are utilized. In Figure 5.5, an example of clock gating circuitry (CGC) is presented. Different clock gater circuitry (CGC) with the similar functionality can also be used in the proposed method.

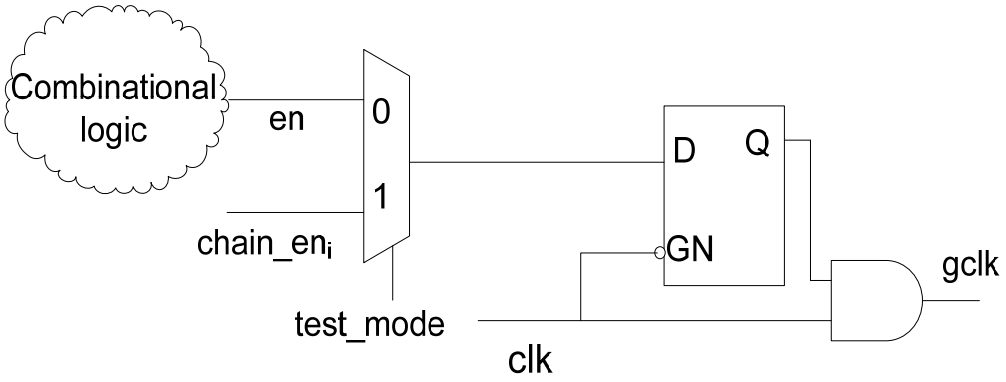


Figure 5.5: An example of Clock gater structure

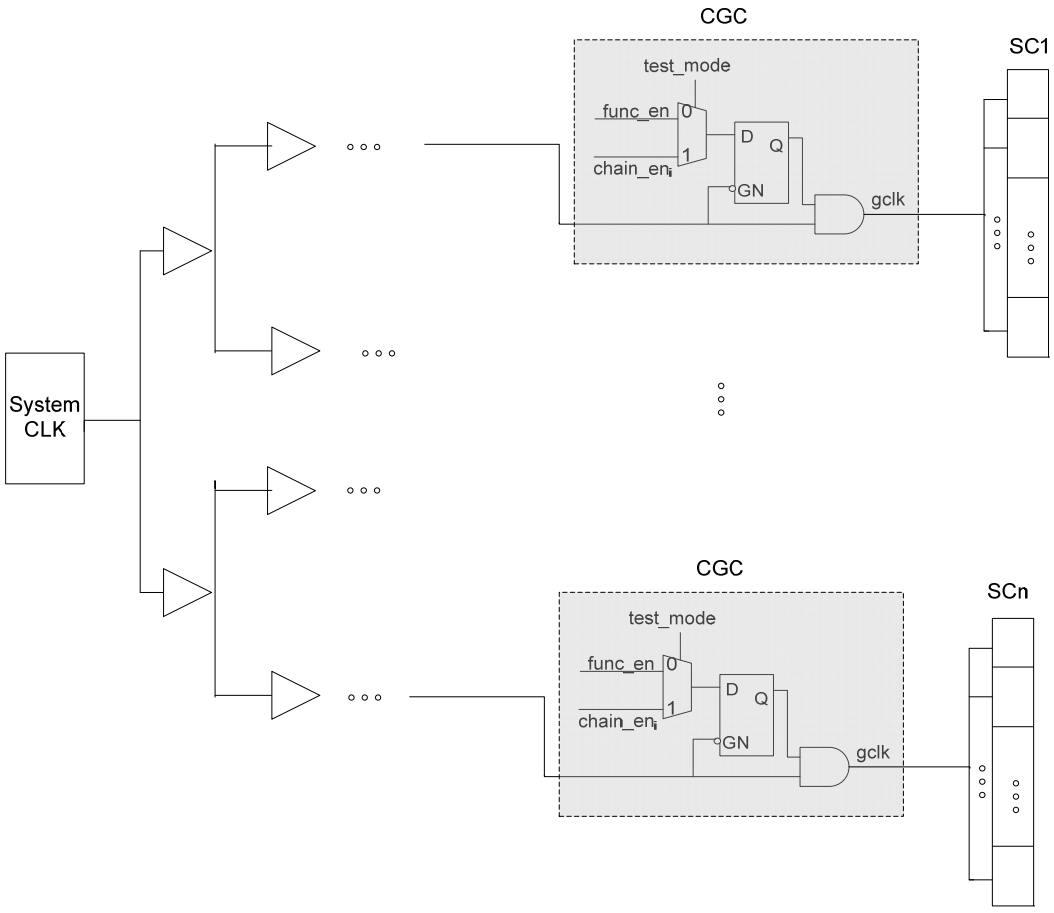


Figure 5.6: An example of clock tree with clock gaters

The test_mode signal controlling the multiplexer of clock gater (Figure 5.5) is present in all industrial design. Test_mode is high throughout the duration of the entire test and is low during the functional operation of the circuit. During test mode, clock gater is controlled by $chain\_en_i$ signal provided by the control block outputs (Figure 5.4) for each scan chains. During normal operation, the test mode is inactive and clock gater is controlled from its functional enable signal.

In Figure 5.6, the hardware structure of gated clock tree scheme is shown. A clock gater circuitry (CGC) of Figure 5.5 is used as a part of the clock tree synthesis with a CGC added at the last stage of the clock tree, where all flip-flops of a given scan chain are safely driven without strongly affecting skew and delay budgets. In this figure, the term gclk is the gating clock signal which is used to shut down the branch clock that goes to a given scan chain SC_i and can be directly controlled by $chain\_en_i$ signal during testing.

Similar to the stuck-at faults, if a scan chain i contains any conflict bits compared to the previous test cube or observation points, its corresponding $chain\_en_i$ signal is set to 1 to allow the scan chain i to be driven by decompressor. Otherwise, the $chain\_en_i$ signal is set to 0 and the scan chain i holds its previous states and receives clock transitions neither in shift mode nor in capture mode. However, for transition faults in addition to the scan chains including conflict bits and observation points, those scan chains that launch a transition and propagate fault effect to primary output need also to be enabled during capture cycle. Therefore, the number of active scan chains in each test phase will increase and the probability to reuse the previous pattern content will be reduced.

Example 5.2: Consider the two-timeframe circuit in Figure 5.7 with three scan chains, SC1, SC2 and SC3, each consisting of three scan cells. Consider a slow-to-rise transition fault at line f. Assume that scan cell sc_5 should be assigned a 0 at first time frame to initialize the faulty line f to 0, and scan cells sc_2, sc_5 and sc_8 should be assigned

to 0, 1 and 1 respectively at second time frame to launch a transition at line f and propagate fault effect to scan cell sc_6.

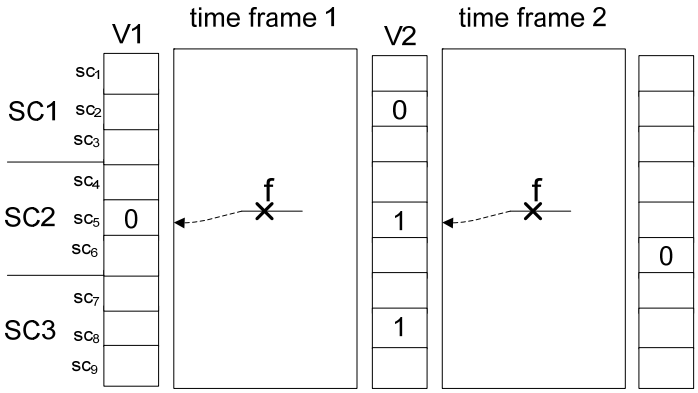


Figure 5.7: Two-timeframe Circuit for detecting slow-to-rise transition fault at line f

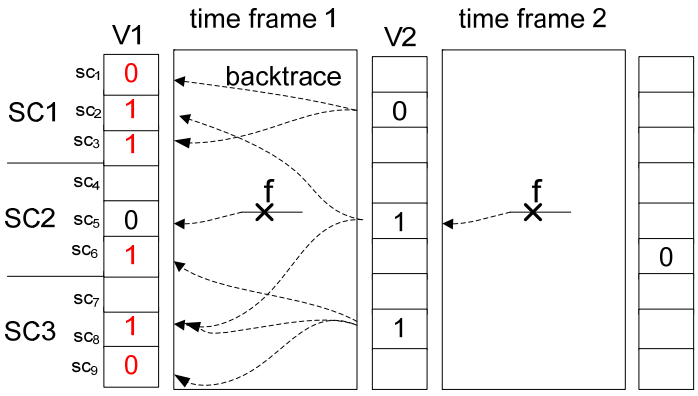


Figure 5.8: Example of backtrace operation in LOC test method

In LOC test method, the second vector is the functional response of the circuit to the first vector. Therefore, some other scan cells need to be specified in first time frame to set sc_2, sc_5 and sc_8 to a 0, 1 and 1 in second time frame. Figure 5.8 illustrates backtrace operation to set sc_2, sc_5 and sc_8 to desired value in time frame 2. Backtrace (justification) is a process to determine which inputs should be set to achieve a desire goal. As indicated from the results, assigning logic value 0 to sc_2 in time frame 2 is obtained by setting sc_1 and sc_3 to 0 and 1 in time frame 1, assigning logic value 1 to sc_5 in time frame 2 is obtained by setting sc_2 and sc_8 to 1 and 1 in time frame 1, and finally assigning logic

value 1 to sc_8 in time frame 2 is obtained by setting sc_6, sc_8 and sc_9 to 1, 1 and 0 in time frame 1.

On the other hand, since each scan chain includes at least one specified position, all of them should be enabled during capture cycle to launch a transition and propagate fault effect to primary output. However, as can be observed in Figure 5.7, the content of some scan chain in time frame 2 has no conflict with its content in time frame 1. So instead of using backtrace operation to justify the value of scan chains in second time frame, it is possible to set those scan chains to a desired value in the first time frame and then force them to hold their content during capture cycle by disabling their clocks.

In this section, a new LOC test method is proposed that utilizes clock gaters circuitry of low power compression scheme (Figure 5.4) to eliminate some backtrace operation during test pattern generation. Therefore, the number of specified bits and the number of enabled scan chains will be reduced which result in a better compression and lower test power.

Example 5.3: Consider the same circuit used in previous example. As can be observed in Figure 5.7, sc_5 has a different value in the first and second time frame. It means that the test requires the performance of backtrace operation only on scan cell sc_5 to assign logic value 1 to it in the time frame 2 which results in assigning logic value 1 to both scan cells sc_2 and sc_8 in the time frame 1. Now as also observed in the figure, the value of sc_2 in time frames 1 and 2 is different. This difference causes the need for sc_2 to be also justified and assigning 0 and 1 to sc_1 and sc_3, respectively. As can be seen, after these backtrace operations, there is no conflict in scan cell sc_8 in the time frames 1 and 2. Therefore, sc_8 can be set to 1 at time frame 1 frozen during capture cycle by disabling the clock of scan chain SC3. In Figure 5.9, the backtrace operation using proposed LOC test method with clock gater circuitry is observed.

As shown in Figure 5.8, the number of specified bits for generating slow to rise fault on line f using conventional LOC test method is equal to 7 and the number of

enabled scan chains during capture cycle in 3. However, by using the proposed LOC test method, the number of specified bits is reduced from 7 to 5 and the number of enabled scan chains during capture is reduced from 3 to 2. In this method, since the ATPG process to specify sc_8 in the launch time frame through circuit lines in the initialization time frame is no longer required, the ATPG run time can be also reduced. By reducing the functional dependency between two test vectors by eliminating some backtrace operation, the proposed method also increases the fault coverage of LOC test.

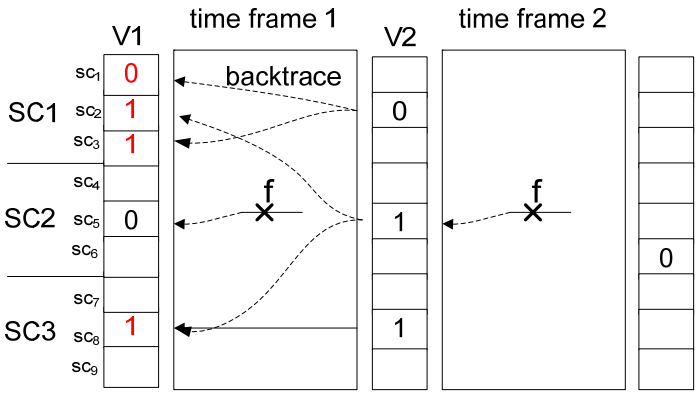


Figure 5.9: Proposed LOC using clock gater circuitry

The proposed test generation method for LOC test is explained below.

A transition fault needs to apply two consecutive test vectors (V_1, V_2) for its detection. Now consider a transition fault f on line l. The conditions of two consecutive test vectors (V_1, V_2) to detect f are represented as follows.

Condition of first vector: Vector V_1 sets 0 (1) on line l, if f is a rising (falling) transition fault.

Condition of second vector: Vector V_2 detects stuck-at 0 (1) on line l, if f is a rising (falling) transition.

In LOC test method, since the second vector is the functional response of the circuit to the first vector, after the generation of the second vector V_2 for detecting stuck-at fault at faulty line, a backtrace process (justification process) is called to justify every

specified scan cells in the second vector V_2 to find the corresponding bits in first vector V_1.

However, in proposed LOC test, the backtrace process is performed only on some scan cells of test vector V_2 and only the clock of scan chains including those scan cells with backtrace process are active during capture cycles and the remaining scan chains with no backtrace process are disable. Therefore, in addition to two test vectors (V_1, V_2), a clock test vector C is also required which is defined as follows:

$$C = c_1, c_2, \ldots, c_N$$

where N is the number of scan chains. Each bit is called a chain-enable bit. The K^{th} chain-enable bit disables (enables) the K^{th} scan chain clock. The test cube t_i for a transition fault with a gated clock is represented as $t_i = <V_1, V_2, C>$

Definition 5.4: A variable chain is a scan chain that has an observation point or a scan cell with opposite value in the first and second time frame which needs a backtrace (justification) process.

Definition 5.5: A constant chain is a chain that none of its specified bits in second time frame are different than corresponding bit in the first time frame and requires no backtrace operation.

The only condition to perform backtrace process on a specified scan cell S_i in test vector V_2 is that S_i is included in a variable chain.

The algorithm for performing backtrace process consists of several steps.

1. *Generate two consecutive test vectors (V1, V2) to detect f (see condition of first and second vector).*

2. *Find the variable chains (scan chains including observation point as well as scan chains including opposite value in first and second vector).*

3. *Perform backtrace process on all the specified bits included in variable chains*

4. *Check to see if any new variable chain is created after backtrace operation.*

5. *If there is new variable chains go to step 2.*

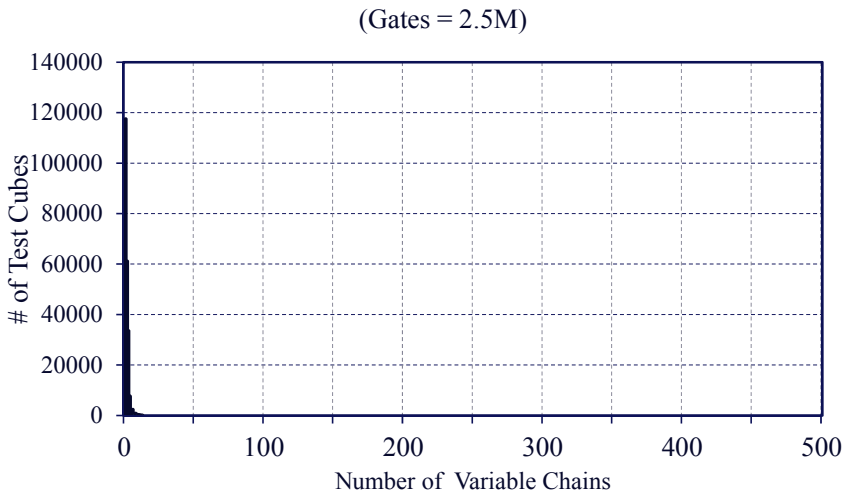
6. *Mark all the other scan chains with specified bits as a constant scan chains.*

7. *Assign the value of constant scan chains in test vector V_2 to test vector V_1 directly without backtrace process.*

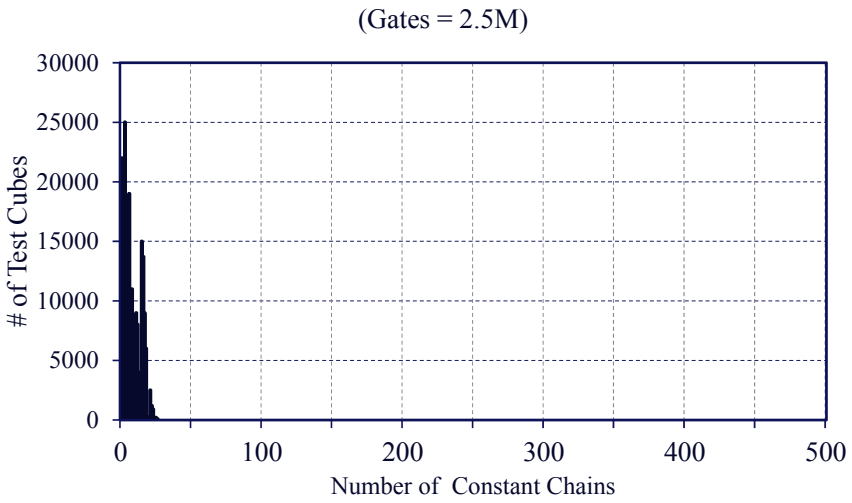
8. *In test vector C= (c_1, …, c_N), set bits corresponding to the variable chains to 1 and bits corresponding to the constant chains to 0. Reaming bits are all X.*

The effectiveness of the proposed LOC test is determined by the number of variable chains. A smaller number of variable chains cause a smaller number of backtrace process.

In this study, to determine the percentages of variable chains and constant chain for each test cube, a substantial amount of experiments are carried out on different industry designs. The experimental results indicate that very often number of variable chains is much less than number of constant chains. In Figures 5.10a and 5.10b, the number of variable chain and the number of constant chains for all primary target transition faults using new LOC method are presented, respectively. The circuit under this experiment has 143K scan cells and 500 scan chains. It can be seen that each test cube has less than 2% variable chains and less than 5% constant chain. However, the number of variable chains can be reduced even further by reordering scan cells and assigning adjacent scan cells in the same scan chains.



(a)



(b)

Figure 5.10: Distribution of variable chains and constant chains

5.7 Cube Merging for Proposed LOC Test Cubes

In this section present, a low power compression algorithm based on the algorithm proposed in section 5.2 for stuck-at faults to reduce the test data volume and power consumption in the circuit is presented.

As discussed in previous pattern, a gated clock LOC test is denoted by $<V_1, V_2, C>$ where V_1 initializes the fault site, V_2 invokes the signal transition at the fault site and

propagates the error signal to primary outputs or scan cells, and C indicates disabled and enabled scan chains during capture cycles. It must be noted that two gated clock LOC test $t_i = <V_{1i}, V_{2i}, C_i>$ and $t_j = <V_{1j}, V_{2j}, C_j>$ are compatible if and only if all the corresponding test vectors of t_i and t_j are compatible.

The overall compression algorithm for generating low power transition test vector is the same as the one proposed in section 5.2 for stuck-at faults with a small change in details. For stuck-at fault, the compression algorithm uses two chains list: one for the list of enabled scan chains during loading test data (C_{load}) and the other for the list of enabled scan chains during capturing and unloading test responses (C_{unload}). The later list includes scan chains that contains observation points. However, for transition faults, the scan chains which are required to be enabled during capture cycle are indicated by clock test vector C of t_i. The clock test vector C includes scan chains which launch a transition as well as those scan chains which observe the fault effect.

For the sake of simplicity, reducing the number of controlling data and increasing the fortuitous detection of untargeted faults, all the enabled scan chains during capture cycles will be unloaded as the next test pattern is loaded to the scan chains. Therefore, C_{load} of the pattern T_i is the clock test vector C of pattern T_{i-1}.

It should be noted that the conflicting chains of pattern T_i compared to pattern T_{i-1} can only be included in those chains with controlling value 1 or X in the clock test vector C of pattern T_{i-1}. Because scan chains with controlling value 0 in the clock test vector C of pattern T_{i-1} should hold theirs values during capture cycle as well as unloading test responses and cannot be loaded with new data.

The pseudo-code for the compression algorithm for transition faults is described below.

1. *For every TDF in fault list generate a gated clock LOC test $t_i = <v_1, v_2, c>$ and put test cube in buffer.*

2. *Set test cube t_1 to be an all-X pattern.*
3. *Repeat step 6-10 until buffer is not empty:*
4. *Set C_{load} to be the list of scan chains need to be loaded for each test vector.*
5. *Set P= (V1, V2, C) to be an all-X vector.*
6. *while(# of chains in C< δ and $C_{load} < \Delta$)*
    a. *Sort test cube in buffer based on the highest degree of similarity with previous states of scan cells.*
    b. *Pick first test cube from buffer and check if it is compatible with P.*
    c. *Add new conflict scan chains that should be reloaded to C_{load}.*
    d. *Move the specified bits of compatible chains to the test vector in which the scan chains are reloaded.*
    e. *Try to compress new combined test cube.*
    f. *if (compression fails)*
        i. *exit while loop*
7. *If ($C_{load} < \Delta$), randomly select (N-S) unspecified BECs in C and assign logic value 0 to disable the selected BECs.*
8. *for (each generate test vector) do:*
    a. *if (all the scan chains have been loaded at least one in the successor test vector){*
        i. *decompress compress pattern*
        ii. *perform fault simulation*

5.8 Experimental Results for Transition Faults

The algorithm described in the previous section was evaluated using a set of experiment performed on several industrial cores to detect transition delay faults. This section reports results for some of them, ranging in size from 470K to 2M gates. The basic data regarding the designs, including the number of gates and the number of scan

cells are listed in the left part of Table 5.5. The next five entries to the Table 5.5 report the resultant number of test patterns, the peak and average switching activity during shift and capture using a standard test data compression technique (EDT [4]).

Table 5.5 Circuit characteristics

Design	Gates	Scan cells	Test Pattern	Capture power (%)		Shift power (%)	
				Peak	Average	Peak	Average
D1	470K	21 K	35,607	42.24	19.23	51.85	49.72
D2	460K	27 K	9258	40.62	15.34	50.66	49.87
D3	1.4 M	60 K	19,252	19.67	15.92	50.73	49.30
D4	2. M	110 K	22,978	36.32	14.5	50.47	49.83

The proposed low power compression method is compared with the method proposed in [58]. The method in [58] employs a controller that allows a given scan chain to be driven either by EDT decompressor, or by constant value for the entire scan load to reduce switching activity during shift. Also it utilizes clock gaters circuitry already available in designs to prevent a relatively large number of flip-flops from toggling during capture mode.

Results of the experiments for transition faults using proposed low power compression and method in [58] are summarized in Table 5.6 - Table 5.9. The control circuitry of both methods is the same (Figure 5.4). In both methods, the threshold was set such that when possible, at least 75% of the scan chains will be loaded with constant values (zeros) in [58] or will be disabled in each test phase.

Results shown in Table 5.6 provide the following information for each test case:

- Scan configuration.

- Peak shift power reduction factor for the proposed low power compression scheme and method in [58], i.e., the ratio of the standard EDT peak shift power (presented earlier, for each design, in Table 5.5) to the low power scheme shift power.
- Average shift power reduction factor for the proposed low power compression scheme and method in [58].

Table 5.6 Result for peak and average power reduction during shift

Design	Scan chains (no × size)	Peak Shift Power		Average Shift Power	
		Proposed	[58]	Proposed	[58]
D1	50 × 420	74.12	55.85	83.36	64.46
	100 × 210	76.88	57.82	84.11	65.11
D2	100 × 270	75.98	55.23	82.39	64.77
	200 × 135	74.81	58.07	80.71	65.01
D3	150 × 400	79.89	62.98	82.96	78.19
	300 × 200	82.00	66.15	83.41	81.40
D4	200 × 588	87.81	63.46	90.06	67.97
	400 × 294	88.83	64.91	90.26	68.25
Average		80.04	60.56	84.66	69.40

As indicated by data under column “Peak Shift Power” of Table 5.6, the proposed method has better switching activity reduction compare to the method proposed in [58]. In the proposed scheme, due to deactivate a significant portion of the clock tree in the

proposed scheme, a substantial reduction in both peak and average switching activity during shift is achieved. Method in [58] is effective in switching activity reduction during shift in operation. However, power consumption during shift out operation and in the clock distribution network is not considered in [58].

Table 5.7 Result for peak and average power reduction during capture

Design	Scan chains (no × size)	Peak Capture Power		Average Capture Power	
		Proposed	[58]	Proposed	[58]
D1	50 × 280	41.90	27.34	35.82	15.00
	100 × 140	46.71	34.64	38.77	19.43
D2	50 × 420	61.35	60.98	40.48	32.33
	100 × 210	62.58	60.73	44.46	34.22
D3	100 × 270	51.01	24.56	62.43	66.32
	200 × 135	53.71	39.07	61.90	67.81
D4	150 × 527	53.25	47.60	68.38	68.75
	300 × 264	55.15	48.40	70.71	68.75
Average		53.21	42.92	52.87	46.58

Table 5.7 shows the percentage reduction in peak and Average switching activity during capture cycle for each test case using the proposed low power compression scheme and method in [58]. Both peak and average power reduction percentages are relative to the case when the conventional EDT is used which fill unspecified values randomly. As can be seen, the proposed method has a better capture power reduction

compare to method in [58]. The peak capture power is reduced, on average, by 53% and the average capture power is reduce by 52% using the proposed scheme.

Table 5.8 compares the two low power schemes in term of test data volume, pattern count and bridging coverage estimate. The first part of the Table 5.8 consists of two columns specifying the scan configuration and the control register size. The size of control register depends on the number of scan chains and it is chosen in such a way that high encoding efficiency is achieved [58].

The reaming parts of Table 5.8 provide the following information for each test case:

- The fill rate reduction of the low power scheme, i.e., the ratio of the total number of specified bits needed to be encoded for all pattern using conventional EDT and the total number of specified bits that have to be encoded for low power schemes; this quantity includes the total number of specified bits across all test patterns as well as the total number of control bits - the latter value is equal to the number of test vectors multiplying by the size of control register.
- Impact on pattern count.
- Impact on bridging coverage estimate (BCE)

As shown in Table 5.8, under column “Reduction of fill rate”, the average fill rate reduction using the proposed method computed across examined industrial designs is equal to 2.7 with the maximal reduction elevated to the value of 3.65. In fact, low power compression has achieved on average 2.7X reduction of test data volume with the switching activity at the level of 10% to 16% only. However, as observed in Table 5.8, the method in [58] increases the test data volume by factor of 1.2. For each design, two different scan configurations are used. As can be seen, by reducing the number of scan chains and the size of control register, the control data volume is reduced and better test data volume is achieved.

In Table 5.8, the test pattern count difference between standard compression and the low power compression is represented by ΔPC. As shown in Table 5.8, the proposed method increases the test pattern counts, on average, by 29% while method in [58] increases the pattern counts, on average, by 53%.

Table 5.8 Result for test data volume, pattern count and BCE

Design	Scan chains (no × size)	Control register	Reduction of fill rate		ΔPC (%)		ΔBCE (%)	
			Proposed	[58]	Proposed	[58]	Proposed	[58]
D1	50 × 280	18	2.34	0.93	26.00	28.93	2.12	2.01
	100 × 140	35	2.28	0.87	20.92	26.59	3.07	2.56
D2	50 × 420	18	2.14	0.92	28.51	35.28	1.01	0.98
	100 × 210	35	2.10	0.87	22.43	29.23	1.28	1.16
D3	100 × 270	35	2.86	0.70	38.95	97.00	3.12	1.45
	200 × 135	56	2.89	0.68	31.57	82.04	3.75	1.87
D4	150 × 527	42	3.65	0.85	36.70	65.25	2.01	1.70
	300 × 264	84	3.40	0.81	30.57	61.93	2.65	1.98
Average			2.71	0.83	29.46	53.28	2.38	1.71

Furthermore, the bridging coverage estimate [66] has been used to assess the impact of standard compression test patterns and low power test patterns on bridging defects. BCE is reducing, on average, by 2.38%, and 1.71% using the proposed method and the method in [58] respectively. In the proposed method, during each test phase, only 25% of scan chains content are observed to detect the faults. However, in [58], the

content of all scan chains are shifted out in order to observe the fault effects. Therefore, the bridging coverage estimate of [58] is slightly better than the proposed low power compression technique.

The transition fault model used in above experiments considers a gross delay at every gate terminal in the circuit and assumes that the additional delay at the fault site is large enough to cause a logic failure. However, the transition fault test generation ignores the actual delays through the fault activation and propagation paths, and is more likely to detect a fault through a shorter path. As a result, the generated test set may not be capable of detecting small delay defects. To achieve better testing of small delay defects, timing-aware ATPG [78] is used to generate test patterns that propagate the delay faults through longer paths. In timing-aware ATPG, timing information of the design is integrated into the test generation flow in order to guide the test pattern generation and to evaluate the quality of the test set.

Three industrial designs are used to evaluate our low power compression scheme when timing-aware ATPG is used. The characteristics of the designs including the number of gates and the scan chain configuration are listed in the left part of Table 5.9. The remaining parts of Table 5.9 provide the following information for each test case:

- The fill rate reduction of the low power scheme using timing-aware ATPG (this data has been calculated as explained in Table 5.8).
- Impact on pattern count.
- Peak and Average capture power reduction factor.
- Peak and Average shift power reduction factor.

As shown in Table 5.9, under column “Reduction of fill rate”, the average fill rate reduction of the proposed low power compression method using timing-aware ATPG across three industrial designs with timing information is equal to 1.9 with the maximal reduction elevated to the value of 2.

In Table 5.9, the test pattern count difference between standard compression and the low power compression using timing-aware ATPG is represented by ΔPC. As shown in Table 5.9, when timing-aware ATPG is used, the proposed method increases the test pattern counts, on average, by 30%.

Table 5.9 Result for timing-aware ATPG

Design	Gates	Scan chains (no × size)	Reduction of fill rate	ΔPC (%)	Capture Power Reduction (%)		Shift Power Reduction (%)	
					Peak	Average	Peak	Average
D1	240K	50 × 300	2.09	31.65	51.88	62.76	60.67	65.94
D2	590K	100 × 280	1.92	38.19	48.14	55.86	72.92	80.47
D3	1.06M	170 × 349	1.70	21.08	49.69	65.98	68.94	69.94

As indicated by data under columns “Capture power reduction” and “Shift power reduction” of Table 5.9, due to deactivate a significant portion of the clock tree in the proposed scheme, a substantial reduction in both peak power (with averages of 49% and 67% during capture and shift, respectively) and average power (with averages of 61% and 72% during capture and shift, respectively) is achieved.

5.9 Conclusion

In this paper, a new low power test data compression scheme is proposed using systematically clock gater circuitry to reduce test data volume and test power by enabling only a subset of the scan chains in each test phase. Since, significant portion of the total power during test is typically in clock tree, by disabling up to 75% of clock tree in each test phase in the proposed compression scheme, significant reduction in the test power in

both combinational logic and clock distribution network is achieved. Using this technique, transitions in the scan chains during both loading test stimuli and unloading test responses decrease which results in the acceleration of the speed of shifting and an increase in the number of cores that can be tested in parallel. The proposed method has the ability of decreasing, in a power aware fashion, the amount of test data and the fill rate, as well. Therefore, any existing test data compression techniques can take advantages of this method.

CHAPTER 6

CONCLUSIONS AND RECOMMENDATION

6.1 Conclusion

As devices grow in gate count, scan test data volume and application time grows as well, even for single-stuck-at faults with single-detection [68]. This poses a serious problem on manufacturing test because, as test data volume increases, it takes more tester memory to hold the complete test set, and longer to deliver the test set through limited test channels, both leading to higher test cost. In addition, considerable research on low power design and testability of VLSI circuits have been shown that the power consumed in test mode of operation is often much higher than the power consumed in normal mode of operation due to the high switching activity in the nodes of the circuit under test which may result in higher dynamic power dissipation or higher supply current demand. This can decrease the reliability of the circuit under test due to excessive temperature and current density which cannot be tolerated by circuits designed using power minimization techniques. Excessive power dissipation may cause hot spots that could damage the CUT. High supply current may cause excessive power supply droops leading to larger gate delays which may cause good chips to fail tests causing yield loss [11-15].

This dissertation focused on these two problems: low power test and compression.

Chapter 3 has proposed a new technique called ACF-scan for reducing IR-drop and overtesting for at-speed delay test in scan design DFT method. ACF-scan uses background states obtained by applying a number of clock cycles to test vectors to fill unspecified values in test cubes. The time to obtain background states depends on the number of additional clock cycles simulated. Experimental results show that after few clock cycles the WSA of background states become stable and applying more clock cycles has no significant effect on reducing WSA. ACF-scan is shown to achieve substantial reductions in peak WSA of capture cycles of launch-off-capture tests in

industrial circuits with no lose in test coverage and minimal increase in test pattern counts. Experimental result shows test pattern generated using ACF-scan method keep the good quality of test. Also, ACF-scan requires no additional hardware overhead as it is a software solution to low power test. It can also be combined with other existing techniques. It was shown that combining the described technique with the recently reported low power technique that fill unspecified bits with preferred value yields substantial reductions in shift power dissipation during test .

Chapter 4 has presented a new low power technique for reducing IR-drop and overtesting for at-speed delay test in Embedded Deterministic Test (EDT) environment based on technique proposed in Chapter 3. The proposed method employs a controller that either allows a given scan chain to be driven by the EDT decompressor or recycle the near to functional response of previous pattern to fill unspecified bits of test cubes. For generating near to functional background to initiate the scan cells, one can apply some number of clock cycles to a test vector to reach a steady state switching activity. Since the circuit is initiated with a pseudo functional pattern, the test responses have also switching activity near to functional pattern. Therefore, the response data for the previous test vector t_{i-1} can be used as a background to fill unspecified scan chains of test cube t_i. Experimental results has shown that with low overhead in test area and volume of test data, and with no penalty in test efficiency, or performance, our proposed method can achieve substantial reduction in the peak capture switching activity for at-speed delay test.

Chapter 5 a new low power test data compression scheme is proposed using systematically clock gater circuitry to reduce test data volume and test power by enabling only a subset of the scan chains in each test phase. Since, significant portion of the total power during test is typically in clock tree, by disabling up to 75% of clock tree in each test phase in the proposed compression scheme, significant reduction in the test power in both combinational logic and clock distribution network is achieved. Using this

technique, transitions in the scan chains during both loading test stimuli and unloading test responses decrease which results in the acceleration of the speed of shifting and an increase in the number of cores that can be tested in parallel. The proposed method has the ability of decreasing, in a power aware fashion, the amount of test data and the fill rate, as well. Therefore, any existing test data compression techniques can take advantages of this method.

6.2 Future research

In his study, a new procedure to fill unspecified values in test cubes to cause low peak switching activity during test capture cycles was proposed, which is close to the functional operation. Since in this study, we don't identify reachable states to generate tests we cannot insure that the switching activity by the proposed tests is the same or determine how close it is to the functional switching activity. Our observations regarding the switching activity caused by the proposed tests is based on the fact that the switching activity is close to its steady state value measured when the circuit under test is operated in functional mode for several clock cycles starting from the scan-in states of the tests in which the don't care bits are randomly filled. However, the steady states switching activity may be far less than the maximum possible switching activity caused by functional tests. Thus the proposed method may lead to undertesting or increase pattern count. Because a test with a higher switching activity is likely to detect more faults since it can create transitions on one or more lines which result it less pattern counts. In order to ensure that the low power patterns do not lead to undertesting or overtesting during test, the measurement of the peak power that may be dissipated during normal operation is necessary.

As the complexity of modern chips increases, external testing with ATE becomes extremely expensive. The BIST design methodology has been widely adopted in the design of VLSI circuits in order to enable the chip to test itself and to evaluate its response with an

acceptable cost [3, 68]. However, a major issue in logic BIST techniques is that power consumption during BIST can exceed the power rating of the chip or package. Increased average power can cause heating of the chip and increased peak power can produce noise-related failures [34]. It is anticipated that low power test compression technique proposed in chapter 5 and near to functional test methodologies proposed in Chapters 3 and 4 can successfully be compatibilzed with logic BIST.

Finally, despite the fact that there are many proposed techniques to minimize power consumption during test, there are only a few solutions that are dedicated to memories. During normal system operation, only one memory bank is accessed at any given time from the several banks included in the memory [69]. By contrast, during test, it is desired to test all banks concurrently in order to reduce the test time and to simplify the BIST control circuit. Unfortunately, this will lead to very high power consumption compared with normal operation. Thus reducing test power in memories is one of the hot directions for future research.

REFERENCES

[1] N. Nicolici, “Power minimisation techniques for testing low power VLSI circuits", Ph.D. Dissertation, University of Southampton, October 2000.

[2] E. B. Eichelberger and T. W. Williams, “A Logic Design Structure for LSI Testability", Proc. of DAC, pp. 462-468, 1977.

[3] V. D. Agrawal, C. R. Kime and K. K. Saluja, “A Tutorial on Buil-In Self Test, Part 1: Principles", IEEE Design and Test of Computers, Vol. 10, Issue 1, pp. 73-82, March 1993.

[4] J. Rajski, J. Tyszer, M. Kassab and N. Mukherjee, “Embedded Deterministic Test", IEEE Transactions on CAD of Integrated Circuits and Systems, Vol. 23, Issue 5, pp. 776-792, May 2004.

[5] R. D. Eldred, “Test Routines Based on Symbolic Logical Statements", Journal of the ACM, Vol. 6, pp. 33-36, 1959.

[6] 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.

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

[8] 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.

[9] B. Dervisoglu and G. Stong, “Design for Testability: Using Scanpath Techniques for Path-Delay Test and Measurement,” in Proc. Int. Test Conf. (ITC’91), pp. 365-374, 1991.

[10] G.L. Smith, “Model for Delay Faults Based Upon Paths,” Proc. ITC 1985, pp.342-349.

[11] M. Pedram, “Power minimization in IC design: Principles and applications”,ACM Transactions on Design Automation of Electronic Systems (TODAES), 1(1), pp. 3-56, January 1996.

[12] N. Nicolici and X. Wen, “Embedded tutorial on low power test”, IEEE European Test Symposium, pp. 202-210, May 2007.

[13] 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

[14] J. Saxena, K. M. Butler, J. Gatt, R. Raghuraman, S. P. Kumar, S. Basu, D. J. Campbell, J. Berech, “Scan-Based Transition Fault Testing – Implementation and Low Cost Test Challenges,” in Proc. International Test Conference (ITC’02), pp. 1120 - 1129, Oct. 2002.

[15] X. Lin, R. Press, J. Rajski, P. Reuter, T. Rinderknecht, B. Swanson and N. Tamarapalli, “High-Frequency, At-Speed Scan Testing,” IEEE Design & Test of Computers, pp. 17-25, Sep-Oct 2003.

[16] J. Wang, et al., “Power Supply Noise in Delay Testing,” Proc. Int’l Test Conf., Paper 17.3, 2006.

[17] P. Girad, “Survey of Low-Power Testing of VLSI Circuits,” IEEE Design & Test of Computers, vol. 19, no. 3, pp. 82-92, 2002.

[18] S. Ravi, “Power-Aware Test: Challenges and Solutions,” Proc. Int’l Test Conf., Lecture 2.2, 2007.

[19] P. Girard, X. Wen, and N. Touba, “Low power testing, in System On Chip Test Architectures”, L.-T. Wang, C.A. Stroud, and N. Touba, Editors, Morgan Kaufmann. pp. 307-350, 2008.

[20] Elham K. Moghaddam, Janusz Rajski, Sudhakar M. Reddy, Mark Kassab, “At-Speed Scan Test with Low Switching Activity,” in Proc. IEEE VTS Test Symp, pp. 177-182, April, 2010

[21] Y. Zorian. A distributed BIST control scheme for complex VLSI devices. In Proc. 11th IEEE VLSI Test Symposium, pages 4–9, 1993.

[22] S. Chakravarty, J. Monzel, V.D. Agrawal, R. Aitken, J. Braden, J. Figueras, S. Kumar, H.J. Wunderlich, and Y. Zorian, “Power dissipation during testing: Should we worry about it?”, In 15th IEEE VLSI Test Symposium (VTS), page 456, 1997

[23] Elham K. Moghaddam, Janusz Rajski, Sudhakar M. Reddy, Xijiang Lin, Nilanjan Mukherjee and Mark Kassab, “Low capture power at-speed test in EDT environment”, In Proc. ITC, Nov. 2010.

[24] Xiaoqing Wen; Kahijara, S.; Miyase, K.; Suzuki, T.; Saluja, K.K.; Laung-Terng Wang; Abdel-Hafez, K.S.; Kinoshita, K.; “A new ATPG method for efficient capture power reduction during scan testing”; VLSI Test Symposium, 2006. Proceedings. 24th IEEE; 30 April-4 May 2006 Page(s):6pp

[25] Kahijara, S.; Ishida, K.; Miyase, K.; “Test vector modification for power reduction during scan testing”; VLSI Test Symposium, 2002. (VTS 2002), Proceedings 20th IEEE; 28 April-2 May 2002 Page(s): 160-165.

[26] N. Badereddine, P. Girard, S. Pravossoudovitch, C. Landrault, A. Virazel, and H. J. Wunderlich, “Minimizing Peak Power Consumption during Scan Testing: Test Pattern Modification with X Filling Heuristics,” Proc. Int’l Conf. on Design & Test of Integrated Systems, pp. 259-264, September 2006

[27] X. Wen, Y. Yamashitam, S. Kajihara, L. T. Wang, K. K. Saluja and K. Kinoshita, “On Low-Capture-Power Test Generaion for Scan Testing”, Proc. VTS 2005, pp. 265 – 270.

[28] S. Remersaro, et al, “Preferred Fill: A Scalable Method to Reduce Capture Power for Scan Based Designs,” in Proc. of Intl. Test Conf., Oct. 2006, pp. 1–10.

[29] A. El-Maleh and A. Al-Suwaiyan, “An Efficient Test Relaxation Technique for Combinational & Full-Scan Sequential Circuits”, Proceedings of the 20th IEEE VLSI Test Symposium, pp. 53-59.

[30] X. Wen, Kohei Miyase, Seiji Kajihara, Tatsuya Suzuki, Yuta Yamato, Patrick Girard, Yuji Ohsumi, and Laung-Terng Wang, "A Novel Scheme to Reduce Power Supply Noise for High-Quality At-Speed Scan Testing", In Proc. International Test Conference, pages 25.1.1–25.1.10, 2007.

[31] Wei Li; Reddy, S.M.; Pomeranz, I.; "On reducing peak current and power during test"; VLSI, 2005. Proceedings. IEEE Computer Society Annual Symposium on; 11-12 May 2005 Page(s):15-161.

[32] X. Lin and Y. Huang, "Scan Shift Power Reduction by Freezing. Power Sensitive Scan Cells," Journal of. Electronic Testing, 2008.

[33] X. Zhang and K. Roy, "Power reduction in test-per-scan BIST," in Proc. Int. On-Line Testing Workshop, 2000, pp. 133–138.

[34] M. L. Bushnell and V. D. Agarwal, Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits. Boston, MA: Kluwer, 2000.

[35] Illman R, Keller B, Gallagher P, "ATPG power reduction using clock gate 'default' constraints," Proc. 1st International Workshop on the Implications of Low Power design on Test and Reliability, 2008, pp.45-46.

[36] Dabholkar, V.; Chakravarty, S.; Pomeranz, I.; Reddy, S.; "Techniques for minimizing power dissipation in scan and combinational circuits during test application"; Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on; Volume 17, Issue 12, Dec. 1998 Page(s):1325-1333.

[37] Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "Efficient Scan Chain Design for Power Minimization During Scan Testing Under Routing Constraint," Proc. Int'l Test Conf., pp. 488-493, October 2003.

[38] P. Rosinger, B.M. Al-Hashimi, N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift- and capture-power reduction", Computer-Aided Design of Integrated Circuits and Systems, IEEE Transactions on; Vol. 23, pp:1142-1153, July 2004.

[39] Z. Zhang, S. M. Reddy, I. Pomeranz, J. Rajski, B. M. Al-Hashimi, "Enhancing Delay Fault Coverage through Low Power Segmented Scan", Computers & Digital Techniques, Vol. 1, pp. 220 – 229, 2007.

[40] R. Sankaralingam and N. A. Touba, "Multi-Phase Shifting to Reducing Instantaneous Peak Power During Scan," Proc. Latin American Test Workshop, pp. 78-83, February 2003.

[41] I. Pomeranz, "On the Generation of Scan-based Test Sets with Reachable States for Testing under Functional Operation Conditions", Proc. DAC 2004, pp. 928 – 933.

[42] I. Pomeranz and S. M. Reddy, "Generation of Functional Broadside Tests for Transition Faults", IEEE TCAD, 2005.

[43] Y.-C. Lin, F. Lu, and K. Cheng, "Pseudofunctional Testing", IEEE Transactions on Computer-Aided Design, Vol. 25, pp: 1535–1546, August 2006.

[44] Z. Zhang, S. Reddy, and I. Pomeranz, “On Generating Pseudo-Functional Delay Fault Tests for Scan Design”, Proc. DFTS 2005, pp. 398-405.

[45] W. Wu, and M. S. Hsiao, “Mining Sequential Constraints for Pseudo-Functional Testing”, In Proceedings IEEE Asian Test Symposium (ATS), pp: 19–24, 2007.

[46] I. Pomeranze and S. M. Reddy, “Definition and Generation of Partially-functional Broadside Tests”, IET Computer and Digit. Jan. 2009, pp.1-13.

[47] Elham K. Moghaddam, J. Rajski, S. M. Reddy, “Low Power Compression Utilizing Clock-Gating”, to be submitted to ITC 2011.

[48] Seongmoon Wang; Gupta, S.K.; “DS-LFSR: a new BIST TPG for low heat dissipation”; Test Conference, 1997. Proceedings., International; 1-6 Nov. 1997 Page(s):848-857.

[49] R. Sankaralingam, N. A. Touba, and B. Pouya, “Reducing power dissipation during test using scan chain disable”, Proc. VTS, April 2001, pp. 319–325.

[50] N.Z. Basturkmen, S.M. Reddy and I. Pomeranz, “A Low power pseudo-random BIST technique,” Proc. ICCD 2002, pp. 468-473.

[51] C. Zoellin, H.-J. Wunderlich, N. Maeding, and J. Leenstra, “BIST power reduction using scan-chain disable in the Cell processor”, In Proc.ITC, 2006.

[52] R. Sankaralingam and N. A. Touba, “Controlling peak power during scan testing,” Proc. VTS, pp. 153-159, 2002.

[53] J. Lee and N. A. Touba, “Low power test data compression based on LFSR reseeding,” Proc. ICCD, pp. 180-185, 2004.

[54] J. Lee and N. A. Touba, “LFSR-reseeding scheme achieving low-power dissipation during test,” IEEE Trans. CAD, vol. 26, pp. 396-401, Feb. 2007.

[55] P. M. Rosinger, B. M. Al-Hashimi, and N. Nicolici, “Low power mixed-mode BIST based on mask pattern generation using dual LFSR re-seeding,” Proc. ICCD, pp. 474-479, 2002.

[56] D. Czysz, G. Mrugalski, J. Rajski, J. Tyszer, “Low power embedded deterministic test”, US patent application, 2006.

[57] D. Czysz, G. Mrugalski, J. Rajski, and J. Tyszer, “New test data decompressor for low power applications,” in Proc. DAC, 2007, pp. 539–544.

[58] D. Czysz, M. Kassab, X. Lin, G. Mrugalski, J. Rajski, and J. Tyszer, “Low Power Scan Shift and Capture in the EDT Environment,” In Proc. IEEE International Test Conference (ITC), paper 13.2, 2008.

[59] X. Liu and Q. Xu, “On simultaneous shift- and capturepower reduction in linear decompressor-based test compression environment,” Proc. ITC, paper 9.3, 2009.

[60] M. F. Wu, J. Lang, and X. Wen, “Power Supply Noise Reduction for At-Speed Scan Testing in Linear-Decompression Environment,” IEEE Trans. Computer Aided Design, vol. 28, pp 1767-1776, Nov. 2009.

[61] J. Li, X. Liu, Y. Zhang, Y. Hu, X. Li, and Q. Xu, “On capture power-aware test data compression for scan-based testing,” Proc. ICCAD, pp. 67–72, 2008.

[62] I. Pomeranz and S. M. Reddy “Functional Broadside Tests with Minimum and Maximum Switching Activity”, Low Power Electronic, Dec. 2008, pp.247 - 262.

[63] J. Rearick, “Too Much Delay Fault Coverage Is a Bad Thing”, Proc. ITC, 2001, pp. 624 – 633.

[64] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A Test vector inhibiting technique for low energy BIST design", Proceedings of the 17th VLSI Test Symposium, pp. 407-412, April 1999.

[65] R. Sankaralingam, R. Oruganti, and N. Touba, "Static compaction techniques to control scan vector power dissipation", IEEE VLSI Test Symposium, pp. 368-374, May 2000.

[66] B. Benware, C. Schuermyer, N. Tamarapalli, Kun-Han Tsai, S. Ranganthan, R. Madge, J. Rajski and P. Krishnamurthy, “Impact of multiple-detect test patterns on product quality”, Proc. ITC 2003, pp. 1031- 1040.

[67] G. Mrugalski, N. Mukherjee, J. Rajski, D. Czysz, J. Tyszer, “Compression Based on Deterministic Vector Clustering of Incompatible Test Cubes”, Proc. ITC, 2009, paper 9.2.

[68] M. Abramovici, M.A. Breuer, and A.D. Friedman, Digital Systems Testing and Testable Design, IEEE Press, 1990.

[69] H. Cheung and S. Gupta, "A BIST methodology for comprehensive testing of RAM with reduced heat dissipation", Proceedings of International Test Conference, pp. 22-32, October 1996.

[70] S. Wang, X. Liu, S.T. Chakradhar, “Hybrid Delay Scan: A Low Hardware Overhead Scan-Based Delay Test Technique for High Fault Coverage and Compact Test Sets,” in Proc. Design, Automation and Test in Europe (DATE’03), pp. 1296-1301, 2004.

[71] N. Ahmed, M. Tehranipoor, “ Improving Transition Delay Test Using a Hybrid Method,” IEEE Design & Test, vol. 23, issue 5, pp. 402-412, 2006.

[72] N. Devtaprasanna, A. Gunda, P. Krishnamurthy, S. M. Reddy and I. Pomeranz, “A Novel Method of Improving Transition Delay Fault Coverage Using Multiple Scan Enable Signals,” Proc. ICCD, pp. 471-474, 2005.

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

[74] C-W. Tseng and E. J. McCluskey, “Multipleoutput propagation transition fault test," in Proc. 2001 IEEE Int. Test Conf., Oct.-Nov. 2001, pp. 358-366.

[75] P. Girard, N. Nicolici, and X. Wen, Eds., Power-Aware Testing and Test Strategies for Low Power Devices. Springer-Verlag Berlin Heidelberg, 2009.

[76] M. E. Imhof, C. G. Zoellin, H.-J. Wunderlich, N. Mäding, and J. Leenstra, “Scan test planning for power reduction,” in Proceedings of the 44th Design Automation Conference, DAC 2007, San Diego, CA, USA, June 4-8, 2007, 2007, pp. 521–526.

[77] M. R. Grimaila, Sooryong Lee; J. Dworak, K. M. Butler, B. Stewart, H. Balachandran, B. Houchins, V. Mathur, Jaehong Park, L.-C. Wang, M. R. Mercer, “REDO-random excitation and deterministic observation-first commercial experiment”, Proc. VTS, pp. 268 – 274, 1999.

[78] X. Lin, H. Tsai, C. Wang, M. Kassab, J. Rajski, T. Kobayashi, R. Klingenberg, Y. Sato, S. Hamada, T. Aikyo, “Timing-Aware ATPG for High Quality At-speed Testing of Small Delay Defects”, In Proc. of IEEE ATS, pages 139–146, 2006.