UNIVERSITY OF BIRMINGHAM

LOW POWER HIGH FAULT COVERAGE TEST TECHNIQUES

FOR

DIGITAL VLSI CIRCUITS

By

Abdallatif S. Abuissa

A thesis submitted to

The University of Birmingham

for the Degree of

Doctor of Philosophy

School of Electronic, Electrical and Computer Engineering
College of Engineering and Physical Sciences
The University of Birmingham

September 2009

LOW POWER HIGH FAULT COVERAGE TEST TECHNIQUES FOR DIGITAL VLSI CIRCUITS

ABSTRACT

Testing of digital VLSI circuits entails many challenges as a consequence of rapid growth of semiconductor manufacturing technology and the unprecedented levels of design complexity and the gigahertz range of operating frequencies. These challenges include keeping the average and peak power dissipation and test application time within acceptable limits.

This dissertation proposes techniques to addresses these challenges during test. The first proposed technique, called bit-swapping LFSR (BS-LFSR), uses new observations concerning the output sequence of an LFSR to design a low-transition test-pattern-generator (TPG) for test-per-clock built-in self-test (BIST) to achieve reduction in the overall switching activity in the circuit-under-test (CUT). The obtained results show up to 28% power reduction for the proposed design, and up-to 63% when it is combined with another established technique. The proposed BS-LFSR is then extended for use in test-per-scan BIST. The results obtained while scanning in test vectors show up to 60% reduction in average power consumption. The BS-LFSR is then extended further to act as a multi-degree smoother for test patterns generated by conventional LFSRs before applying them to the CUT. Experimental results show up to 55% reduction in average power.

Another technique that aims to reduce peak power in scan-based BIST is presented. The new technique uses a two-phase scan-chain ordering algorithm to reduce average and peak power in scan and capture cycles. Experimental results show up to 65% and 55% reduction in average and peak power, respectively.

Finally, a technique that aims to significantly increase the fault coverage in test-per-scan BIST, while keeping the test-application time short, is proposed. The results obtained show a significant improvement in fault coverage and test application time compared with other techniques.

To my mother Zuhria and my wife Bayan
To all people who love and support me
Thank you very much for everything

ACKNOWLEDGMENTS

First of all, I would like to thank my supervisor Dr. Steven F. Quigley for his support and guidance throughout this PhD. I am grateful for his invaluable help during the preparation of this thesis and several conference and journal papers. Dr. Quigley was not only a supervisor, but also a friend who gave me advice, help, counsel, and constant support during my stay in the UK.

I wish to thank Mrs Mary Winkles, the EECE postgraduate secretary, for her support and help in many occasions during the last 5 years I spent in the School of Electronic and Computer Engineering as an MSc and PhD Student.

During the period of my PhD, I met many people, either in the university or in the house. I would like to thank all of them for making life in Birmingham more enjoyable.

Last but certainly not least, I would like to thank my mother for her patience and constant prayers and my wife for her support. Also, I would like to thank my cousin Mohammed Abu-Issa, the main sponsor for my undergraduate studies, and Hani Qaddoumi Scholarship Foundation (HQSF), the main sponsor for my MSc studies. Special thanks to my cousin and friend Ayed Abu-Issa and all of my brothers, sisters, nephews, nieces and other relatives in my village Al-Jalamah – Jenin, Palestine and in Jordan.

This work was funded by Dorothy Hodgkin's Postgraduate Award (DHPA) and the School of Electronic, Electrical and Computer Engineering at the University of Birmingham, for which I am very grateful.

The viva for this thesis was held on 09/09/09. Dr Gordon Russell (University of Newcastle) was the external examiner and Mr Phil Atkins was the internal examiner. I would like to thank both of them for their constructive comments and suggestions to improve this thesis

Contents

List of Figures

List of Tables

LIST OF ABBREVIATIONS

ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generator
BIST	Built-In Self-Test
BS-LFSR	Bit-Swapping LFSR
CMOS	Complementary Metal Oxide Semiconductor
CUT	Circuit-Under-Test
DF	Detected Faults
DS-LFSR	Dual-Speed LFSR
EFC	Effective Fault Coverage
FC	Fault Coverage
FF	Flip-Flop
IC	Integrated Circuit
ISCAS	International Symposium of Circuits And Systems
LFSR	Linear Feedback Shift Register
LT-RTPG	Low Transition Random Test Pattern Generator
MISR	Multiple-Input Signature Register
MO-LFSR	Multi-Output LFSR
MT-Filling	Minimum Transition Filling
MUX	Multiplexer
PI	Primary Inputs
PO	Primary Outputs
PODEM	Path Oriented DEcision Making
RAM	Random Access Memory
RF	Redundant Faults
RLFSR	Rotational LFSR
ROM	Read Only Memory
RPR	Random Pattern Resistant
s-a-0	Stuck-At-0
s-a-1	Stuck-At-1
SA	Signature Analyser
SC	Scan Chain

SOC	System-On-Chip
TF	Total Faults
T-FF	Toggle FF
TL	Test Length
TPG	Test Pattern Generator
VLSI	Very Large Scale Integration
WSA	Weighted Switching Activity
WT	Weighted Transition

CHAPTER 1

Introduction

In recent years, with the advance of semiconductor manufacturing technology, the requirements of digital very-large-scale-integrated (VLSI) circuits, which are composed of tens to hundreds of millions of gates, have led to many challenges during manufacturing test. Moreover, the unprecedented levels of design complexity and the gigahertz range of operating frequencies make the testing of nanometre system-on-chip (SOC) designs a most demanding challenge. This is because the large and complex chips require a huge amount of test data and dissipate a substantial amount of power during test, which greatly increases the system cost. There are many test parameters that should be improved in order to reduce the test cost. These parameters include the test power, test length (test application time), test fault coverage, and test hardware area overhead.

This thesis addresses the problem of reducing power consumption during off-line test (i.e. when the circuit is switched to test mode and stopped from carrying out its normal operation) and the problem of keeping the test application time moderate without incurring unacceptably low fault coverage. The main objectives of this dissertation are

1. To introduce novel techniques that improve the power consumption during test with minimum effect in test length, fault coverage, or hardware area overhead.
2. To introduce novel techniques to achieve high fault coverage in the circuit under test (CUT) with an acceptable test length and small hardware area overhead.

3. To combine these new techniques with already existing techniques in order to obtain further reduction in power consumption.

This chapter introduces some important concepts in testing of digital VLSI circuits and the importance of minimising power consumption during test. It then provides the motivation for this study and briefly summarises the organisation and contribution of this dissertation.

1.1 Automatic Test Equipment (ATE)

Automatic test equipment (ATE) is instrumentation that is used in external testing to apply test patterns to the CUT, to analyse the responses from the CUT, and to mark the CUT as good or bad according to the analysed responses [1, 2]. Fig. 1.1 shows a basic diagram for external testing using ATE with its three basic components:

1. The CUT: this is the integrated circuit (IC) part which is tested for manufacturing defects.
2. The ATE control unit: this unit includes the control processor, the timing module, and the power module.
3. The ATE memory: this memory contains test patterns that will be supplied to the CUT and the expected fault free responses which are compared with the actual responses obtained from the CUT to determine whether the CUT is faulty or not.

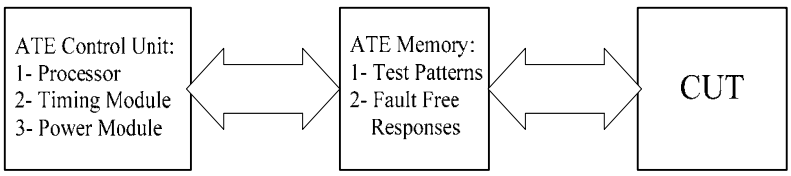

Fig. 1.1 External testing using ATE.

External testing using ATE has a serious disadvantage since the ATE (control unit and memory) is extremely expensive and its cost is expected to grow in the future as the number of chip pins increases [2].

1.2 Built-In Self-Test (BIST)

As the complexity of modern chips increases, external testing with ATE becomes extremely expensive. Instead, built-in self-test (BIST) [3-6] is becoming more common in the testing of digital VLSI circuits since it overcomes the problems of external testing using ATE. BIST test patterns are not generated externally as in case of ATE; instead they are generated internally using some parts of the circuit, also the responses are analysed using other parts of the circuit. When the circuit is in test mode, test patterns generators (TPGs) generate patterns that are applied to the CUT, while the signature analyser (SA) evaluates the CUT responses. One of the most common TPGs for exhaustive, pseudo-exhaustive, and pseudorandom TPG is the linear feedback shift register (LFSR) [4, 7]. LFSRs are used as TPGs for BIST circuits because, with little overhead in hardware area, a normal register can be configured to work as a test generator, and with an appropriate choice of the location of the XOR gates, the LFSR can generate all possible output test vectors (with the exception of the 0s-vector, since this will lock the LFSR). The pseudorandom properties of LFSRs lead to high fault coverage when a set of test vectors is applied to the CUT compared with the fault coverage obtained using normal counters as TPGs. Also LFSRs can be configured to act as signature analysers for the responses obtained from the CUT. Despite their simple appearance, LFSRs are based on complex mathematical theory that helps explain their behaviour as TPGs or SAs [4, 7].

The characteristic polynomial of an LFSR determines which flip-flop locations of the LFSR feed the inputs of the XOR gates in the feedback path[1] [4, 8]. If the characteristic polynomial of an LFSR is primitive, then the LFSR will generate the maximum length non-repeating sequence, which is called an m-sequence. LFSRs can be divided into two main categories: external-XOR LFSR (simply external LFSR) and internal-XOR LFSR (simply internal LFSR). These are distinguished by the way in which XOR gates are inserted into the system. In an external LFSR the XORs appear only in the feedback, while in the internal LFSR the XORs appear between flip-flops[4]. As a simple example, the characteristic polynomial $p(x) = x^3 + x + 1$ is a primitive polynomial of degree 3 (i.e. 3 flip-flops are needed for implementation).

[1] See appendix A for more details

Fig. 1.2(a) shows the external LFSR implementation of this polynomial, while Fig. 1.2(b) shows the internal LFSR implementation.

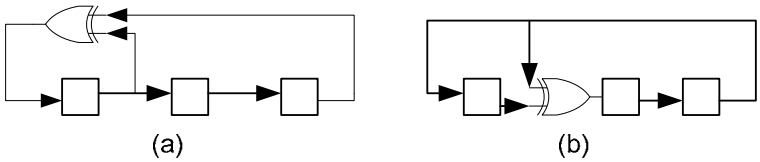

Fig. 1.2 External (a) and Internal (b) LFSRs that implement characteristic polynomial $p(x) = x^3 + x + 1$

1.3 Test-per-Clock and Test-per-Scan BIST

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 [5, 7]. BIST schemes can be divided into two main types according to the way in which test patterns are applied to the CUT [1, 5]. The first scheme is test-per-clock, in which the outputs of a TPG directly feed the inputs of the CUT, and the outputs of the CUT are directly connected to an SA. In this scheme a test vector is applied to the CUT, and a response is captured from the CUT on each clock cycle. Fig. 1.3 shows a test-per-clock configuration.

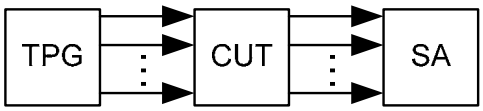

Fig. 1.3 Test-per-clock configuration.

The second scheme is test-per-scan, in which a scan path is used to shift test patterns into a CUT. A full scan cycle requires *m+1* clock cycles, where *m* is the number of flip-flops in the scan chain. The response to an applied test pattern is captured into a scan chain and scanned out in the next scan cycle in parallel with scanning in another test pattern. The main advantage of this scheme over the former one is its lower hardware area overhead while the main disadvantage is in the test application time. Fig. 1.4 shows one of many possible configurations for test-per-scan schemes.

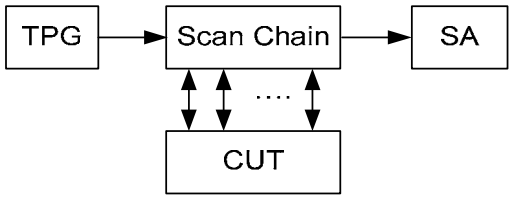

Fig. 1.4 Test-per-scan configuration.

1.4 Fault Modelling and Fault Coverage

A fault model can be defined as a description of the behaviour of, and assumptions about, how components (nodes, gates…etc.) in a faulty circuit behave. In this way, a high percentage of the faults that may occur in a circuit can be modelled. One of the most popular and common fault models at the logic level of abstraction is the stuck-at-fault model (single and multiple stuck at faults). It makes the assumption that a node under consideration is permanently connected with ground, called stuck-at-0 (s-a-0), or permanently connected with V_{dd}, called stuck-at-1 (s-a-1). This fault model is considered to be the most common model in logic circuits [9]. This fault model is the target fault model used throughout this thesis due to its popularity.

In addition to the stuck-at fault model, there are other fault models which include stuck-at-open, transition delay, path delay, and bridging fault models [9, 10].

A commonly used metric to represent the percentage of faults detected using a fault model is the fault coverage (FC). The FC can be represented as in equation (1.1)

$$FC = \frac{DF}{TF} \tag{1.1}$$

Where DF represents the number of detected faults, TF represents the total number of faults in the CUT. However, most CUTs contain redundant faults (RF) that are not detectable due to the presence of redundant hardware in the circuit, hence another way to represent the effective fault coverage (EFC) is given by equation (1.2)

$$EFC = \frac{DF}{TF - RF} \tag{1.2}$$

1.5 Fault Simulation

In order to determine the fault coverage for a specified set of test vectors applied to a CUT, fault simulation is carried out. For each fault expected in the CUT (excluding redundant faults), the output produced when a test vector is applied to a faulty circuit differs from the output produced in a fault-free circuit. Thus, fault simulation produces a list of detected faults for each test vector. There are many fault simulators that can be used for this purpose, some commercial and others academic. The fault simulator that is predominantly used within this thesis is an academic tool called FSIM [11] which is based on parallel pattern single fault propagation for stuck-at faults defects.

1.6 Automatic Test Pattern Generator (ATPG)

The automatic test pattern generator (ATPG) is software dedicated to the generation of test vectors that are used to detect the modelled faults in a CUT. Since in many cases the generated vectors do not achieve 100% fault coverage, the ATPG gives statistics about the FC achieved, the percentage of redundant faults, and the aborted faults (which will therefore not be detected) for these test vectors.

ATPG tools can be divided into two types: combinational ATPG and sequential ATPG. The combinational ATPG is dedicated to generating test sets for combinational circuits, or scan-based sequential circuits where all of the state elements can be accessed directly through the scan-chain. This ATPG, if it is well-designed, can generate test vectors that achieve high fault coverage. Most of the combinational ATPGs depends on random and deterministic phases in the generation of test vectors [1, 3]. In the random phase, the ATPG applies pseudo-random vectors to inputs of the CUT and then performs fault simulation to check the fault coverage and the faults remaining undetected. Normally, most of the faults are detected in this phase. In the deterministic phase, the ATPG generates test vectors for specific faults (that are hard to detect by pseudorandom means) and normally uses algorithms such as the path sensitisation method for this purpose.

The sequential ATPG, which is dedicated to the generation of test vectors for sequential circuits, is more complicated as a result of the timing signals and memory elements present in the circuit. In general, two test vectors are needed to test a fault (or group of faults). The first test vector is used to initialise the memory elements to a specified state, and then the next is used to detect the presence of the fault(s). One of the aims of design for testability techniques is to reduce the complexity of test generation for sequential circuits. One common technique to achieve this is to change the sequential circuit to a scan-based circuit.

The aim of this thesis is to reduce the test power in combinational circuits and scan-based sequential circuits where a scan-path is present and memory elements are accessible through this path. Thus, a combinational ATPG was used through this thesis. The ATPG that is mostly used in this thesis is an academic tool called ATALANTA [12] which is used to generate test vectors for stuck-at faults in combinational and scan-based sequential circuits.

1.7 Test Vector Generation in Scan-Based Circuits

Internal scan design is one of the most efficient design-for-testability techniques to increase controllability and observability in sequential circuits. In scan design, the D flip-flops in the circuit are modified as in Fig. 1.5(a) to act as a scan D flip-flop. A group of scan D flip-flops are connected in such a way that in addition to their normal operation in normal mode, in test mode each flip-flop output is connected with the input of the successive one to form a full scan-chain as shown in Fig. 1.5(b). The first flip-flop input is connected with an external input to feed the scan-chain with patterns, and the last flip-flop output is connected to a signature analyser to check the response. In the multiple scan-chains, the flip-flops are divided into groups; each group forms its own scan-chain.

When the LFSR is used to generate test patterns for full scan-chain sequential circuits, one of its flip-flop outputs is connected with the scan-chain input. In this case the LFSR will be considered as a one-dimensional TPG. The main problem of this configuration is the long time needed to scan-in a test vector which is equivalent to the number of flip-flops in the scan-chain.

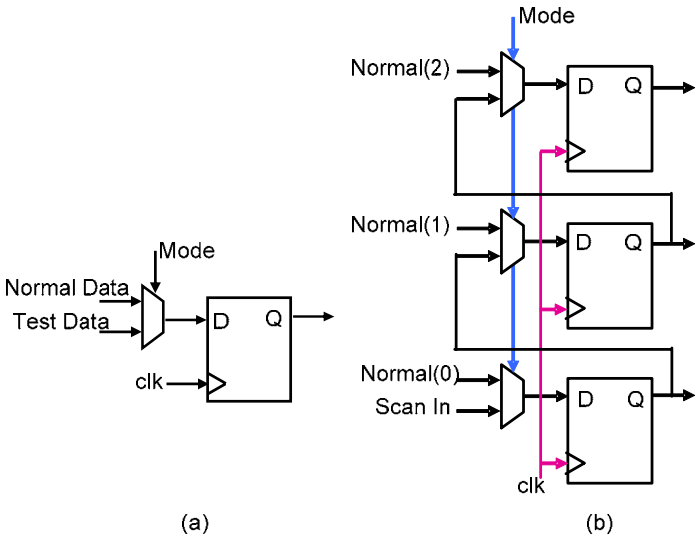

Fig. 1.5 (a) Scan D flip-flop (b) Scan-chain connection

In order to speed-up the scanning of test vectors (i.e. reducing test application time), the flip flops in the circuit can be divided into groups, and each group forms a separate scan-chain. This approach is called multiple scan-chains. In this case a two-dimensional TPG should be used to scan-in test vectors in the multiple scan-chains in parallel. The LFSR can be used for this purpose, where different flip-flops outputs can be connected with the different scan-chain inputs and the outputs of the scan-chains are connected with a multiple input signature register (MISR) as shown in Fig. 1.6 [13].

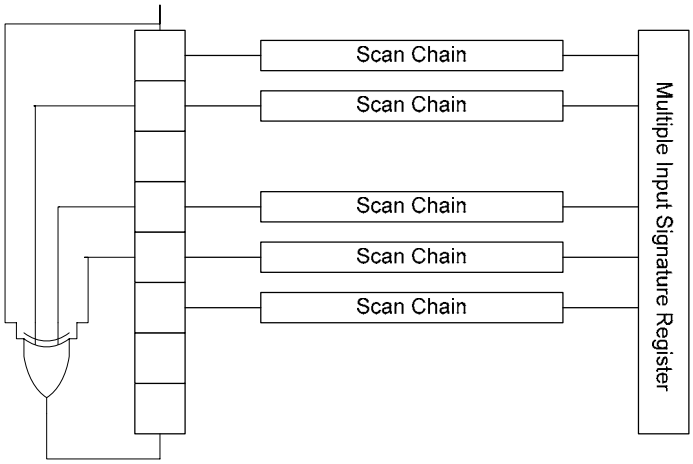

Fig. 1.6 Multiple scan-chain design with the LFSR as a two-dimensional TPG.

1.8 Phase Shifters

In the case of multiple scan-chains with the LFSR as a two-dimensional TPG as shown in Fig. 1.6, the fault coverage in such an environment is often unsatisfactory due to structural dependencies introduced by the test generator [7, 8]. Furthermore, if the scan-chains are fed directly from adjacent cells of the LFSR, then this will cause the neighbouring scan-chains to contain test patterns which are exactly the same with one clock shift. Thus, the test vectors seen by the CUT will no longer be pseudorandom patterns, which can adversely affect the fault coverage. In order to overcome this problem, while still using a short LFSR to feed many scan-chains in the CUT, extra logic is inserted between the LFSR and the scan inputs of the scan-chains. This logic is called a phase shifter [14].

A typical phase shifter consists of a network of XOR gates as shown in Fig. 1.7. The presence of the XOR gates breaks the structural dependencies and generates test sequences with the desired separation.

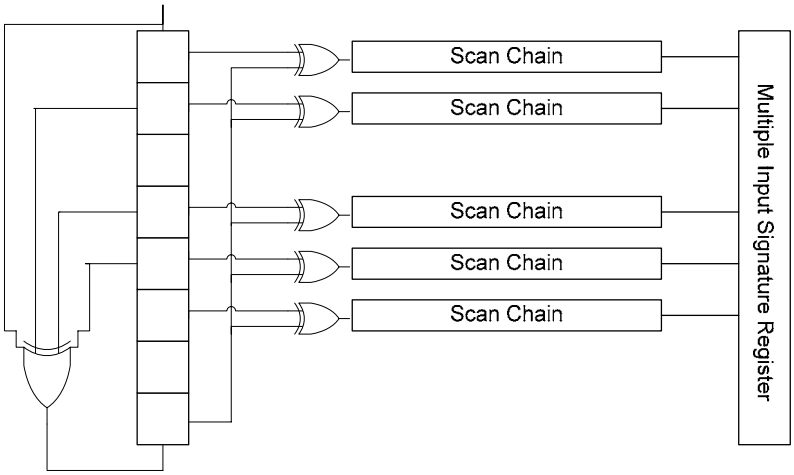

Fig. 1.7 Multiple scan-chains with the presence of phase shifter to remove correlation.

1.9 Power Dissipation in Digital VLSI Circuits

With the development of portable devices and wireless communication systems, design for low power has become an important issue. Minimising power dissipation in VLSI circuits increases battery lifetime and the reliability of the circuit [15, 16]. In general, the power dissipation of complementary metal oxide semiconductors

(CMOS) circuits can be divided into two main categories: static power and dynamic power [16].

Static power is the power dissipated by a gate when it is inactive, i.e. when it is not switching. A significant fraction of static power is caused by the reduced threshold voltage used in modern CMOS technology that prevents the gate from completely turning off, thus causing source to drain leakage. All the components of static power dissipation have a minor contribution to the total power dissipation, and can be minimised for well-designed circuits.

On the other hand, dynamic power dissipation, which is the dominant source of power dissipation in CMOS circuits, occurs while the circuit is switching [15]. The circuit is active when the applied voltage to an input of a cell changes, resulting in a logic transition in one or more of the outputs of the circuit at the transistor level. Hence, charging/discharging of the load capacitances of transistors is the main source of dynamic power dissipation.

The energy that is consumed over a given time T can be expressed as given by equation (1.3) [17]

$$\int_0^T P(t)dt = \int_0^T i(t)V_{DD}dt \tag{1.3}$$

where P(t) is the instantaneous power, V_{DD} is the supply voltage and i(t) is the current drawn from the voltage source. As the dynamic power is predominantly caused by the current required for charging/discharging the load capacitance through the pull-up and pull down networks as shown in Fig. 1.8, the energy consumed from the source for charging the output from 0 to 1 is given by equation (1.4) [17]

$$E_{0\to1} = V_{DD}\int_0^T i(t)dt = V_{DD}\int_0^{V_{DD}} C_i dV = C_i {V_{DD}}^2 \tag{1.4}$$

where C_i is the load capacitance. Only half of this energy is stored in the capacitor, while the other half is converted into heat [17, 18]. 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.

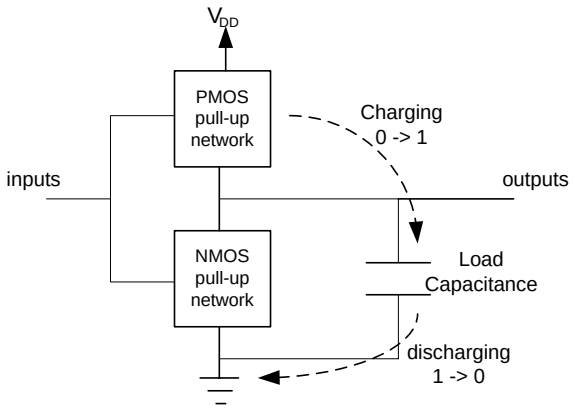

Fig. 1.8 Dynamic power dissipation in CMOS circuits [17].

Therefore, the rate at which the outputs change their value determines the average dynamic power dissipation. This is mainly dependant on the circuit activity, which can be particularly problematic during test for reasons that will be explained in section 1.13.

1.10 Terminology Relating to Energy and Power

This section defines some terms related to power consumption measures in low power testing (as defined in [19, 20]). These terms are used throughout this thesis:

- Energy: represents the total switching activity generated during the application of the complete test sequence. The increase of energy consumption during test has a direct effect on the battery lifetime in portable devices.
- Average Power: equals the total energy consumed during test divided by the test time in order to represent the average rate of energy consumption. If the average power is high this will cause a temperature increase in the CUT.
- Instantaneous Power: corresponds to the power consumed at any given instant during testing. This normally happens after applying a rising (or falling) edge of the system clock.
- Peak Power: corresponds to the highest value of instantaneous power measured during testing. The peak power generally determines the thermal and electrical limits of the circuit and the system package requirements.

1.11 WSA and WT Modelling

The energy dissipated at node i per switching event is given by equation (1.5) [19]

$$E_d = \frac{1}{2} \times C_i \times V_{DD}^2 \quad (1.5)$$

where C_i is the load capacitance and V_{DD} is the supply voltage. Thus, the total energy consumed in a period T is given by equation (1.6) [19]

$$E_T = \frac{1}{2} \times C_i \times V_{DD}^2 \times S_i \quad (1.6)$$

where S_i is the total number of switching events at node i for the period T [19]. Capacitance C_i is assumed to be proportional to the fan-out F_i of node i [21]. Thus an estimate of consumed energy E_i at node i can be given by equation (1.7).

$$E_i = \frac{1}{2} \times c_o \times F_i \times V_{DD}^2 \times S_i \quad (1.7)$$

where c_o is the output load capacitance for a fan-out of one. In equation (1.7) c_o and V_{DD} are constants for all nodes in the circuit, while F_i and S_i vary between nodes. The product "Si×Fi" is named the weighted switching activity (WSA) of node i and is used as a metric for the energy consumption at that node [22]. Finally, the total WSA produced by a CUT after applying all required test vectors is the summation of WSA for each node for each applied test vector. Thus, E_{total} equals $\frac{1}{2} \times c_o \times V_{DD}^2 \times WSA_{total}$ and the average power equals E_{total}/T_{cycles}, where T_{cycles} is the time needed to apply the test vectors.

Alternatively, 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 (WT) metric [23]. The weighted transition metric states that the power consumed in scan-based testing depends not only on the number of transitions in the scanned-in vector (or scanned out response), but also on the positions of the transitions. For example, for a test vector $V_1 = b_1b_2b_3b_4b_5 = 10000$, where b_5 is scanned into the scan-chain first, then vector V_1 (which has one transition between b_2 and b_1) will cause one transition in a scan-chain of length 5 (assuming that the scan-chain initial value is 00000). By contrast, $V_2 = 00001$ (which has one transition between b_5 and b_4) will cause 4 transitions in the scan-chain. The WT calculation is given by equation (1.8) [23]

$$WT=\sum [(Size\ of\ Scan\text{-}Chain) - (Position\ of\ Transition)] \quad (1.8)$$

It is important to note that the position of a transition in equation (1.8) is counted from right to left in scanned-in vectors, and from left to right in the scanned-out response (e.g. in $V_1 = b_1b_2b_3b_4b_5 = 10000$, the transition is in position 4, hence WT = 1. But if the response is $R1 = b_1b_2b_3b_4b_5 = 10000$, then the transition is in position 1, hence WT = 4).

1.12 Average and Peak Power

The power consumption in VLSI circuits can be analysed from two different perspectives: average power and peak power. The average power consumption is used to refer to the average power consumed in the circuit during its period of operation or during a large number of clock cycles. The instantaneous power consumption refers to the power consumed in an instant of time after a rising (or falling) edge of clock. The maximum instantaneous power during the whole period of operation is called the peak power.

Excessive average and peak power consumption during test can lead to many serious problems [24, 25]. High average power consumption (which means high power consumption sustained for long period of time) will shorten the battery lifetime in portable devices. Also, the high temperature during test and the heat dissipation produced in CMOS circuits is proportional to the average power consumption [21, 26, 27]; hence a circuit may malfunction if the temperature is too high or it can be permanently damaged as a result of excessive heat dissipation [28, 29]. Furthermore, high average power consumption speeds-up electro-migration and increases the circuit temperature, which can lead to reliability problems [16, 30].

On the other hand, a high value of peak power also cause a high rate of current flowing in the power and ground lines leading to excessive noise in these lines. This noise can erroneously change the logic state of circuit lines leading to incorrect operation of circuit gates causing some good dies to fail the test. Also, high power can lead to a drop in power supply voltage, called voltage drop or IR drop. IR drop reduces noise margins of cells and increases the probability of failure due to crosstalk noise [31].

1.13 Motivation for Low Power consumption and High Fault Coverage Testing

In recent years, with the fast growth of personal mobile communication and portable computing systems, design for low power has become one of the greatest challenges in high performance VLSI design. As a consequence, many techniques have been introduced to minimise the power consumption of new VLSI systems. However, most of these methods focus on the power consumption during normal mode operation (functional operation), whilst test mode operation has not normally been a predominant concern. However, it has been found that the power consumed during test mode operation is often much higher than normal mode operation because of the high switching activity in the nodes of the CUT during test [24, 29, 32-34].

The main motivation for considering low power testing is that a circuit consumes much more power during test than during normal mode operation. In [24] it has been shown that the power consumed in test mode can be more than twice the power consumed in normal mode. The main reasons for this increase in test power [19, 20] are as follows:

- Modern ATPG tools tend to generate test patterns with a high toggle rate in order to reduce pattern count which leads to a shorter test application time. Thus, the node switching activity in the CUT in test mode is much higher than normal operation mode.
- In normal operation mode, if the system contains several blocks, then it is likely that only one or few of the blocks will be active at the same time, hence reducing the power consumption. By contrast, in test mode parallel testing is often used to reduce test application time. This parallelism inevitably increases power dissipation during testing.
- The design for testability circuitry inserted in the circuit will probably be idle during normal mode but may be used intensively during test mode, hence increasing the power consumption.
- The correlation between the successive functional input vectors during normal operation is considered to be high compared with the correlation of test vectors in the test mode. For example, in the circuits that process digital and video signals, the inputs to most modules change relatively slowly, hence, successive inputs are

highly correlated. However, for the test vectors generated by a TPG such as LFSR, there is no definite correlation; this will increase the switching activity in the circuit.

As the excessive switching activity causes many problems (as mentioned in section 1.12), low power testing has become a very important issue to be considered in order to avoid reliability problems and manufacturing yield loss due to high power dissipation during test in VLSI circuits.

On the other hand, in order to make the test cost acceptable, the test should run for a short period of time. However, with conventional TPGs, running a test for a short period of time will lead to loss of fault coverage. Furthermore, running the test for a longer time, will not only increase the test cost, but also will increase the total energy consumed during test, hence decreasing the battery lifetime. Thus, it is important to develop techniques that significantly improve the fault coverage within a restricted test length and with an acceptable hardware area overhead.

1.14 Thesis Organisation and Contribution

This thesis presents new techniques for low power testing of combinational and scan-based sequential circuits. Also it presents a new technique to achieve high fault coverage in scan-based sequential circuits with an acceptable test application time.

Chapter 2 of this dissertation summarises the literature on previous work on low power testing and high fault coverage. Also, it summarises – in greater detail than this chapter – the contribution of this thesis.

Chapter 3 introduces a new technique for low power testing of combinational circuits. The technique presented is based on some new observations about the output sequences of a conventional LFSR. It modifies the conventional LFSR by using a simple bit swapping technique. In chapter 4, the technique presented in chapter 3 is extended and applied to scan-based sequential circuits. In chapter 5, the proposed

design is generalised from a two-bit swapping technique to a rotational technique for a group of bits.

A new technique for implementing a cell ordering algorithm is presented in chapter 6. The algorithm aims to reduce both average and peak power in scan-based circuits through a two-phase scan-chain ordering algorithm.

Chapter 7 presents a new technique for attaining high fault coverage in scan based sequential circuits. The proposed technique is based on using more than one cell of the LFSR to feed the same scan-chain.

Finally, the conclusions and suggestions for future work are presented in chapter 8.

The algorithms presented in chapters 3 to 8 have resulted in original work published in [35-38]. In addition to that, another two papers are under preparation for submission.

CHAPTER 2

Previous Work

The organization of this chapter is almost similar to a previously published survey about the techniques of low power testing [20]. This chapters approximately uses the same way of categorizing the published techniques but with more references than [20]. Also this chapter concentrates more on the previous techniques fall in the same category as the techniques proposed on this thesis in order to provide comparison with them in the next chapters.

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 and power-on test, as the system may consume much more power during test than during normal (functional) operation [24, 28, 29, 32-34]. Thus, low power test of digital VLSI systems has become a major issue of research in recent years. One direct technique to reduce power consumption during test is testing with a reduced operating frequency [39, 40]. This solution needs no extra hardware. However, although it reduces the average power, it has no effect in reducing the energy since the test time will increase. Also this technique cannot reduce the peak power. Furthermore, timing faults may not be detected using this technique. Another direct technique is by partitioning the CUT into blocks with appropriate test planning [39]. This technique, although it reduces the power consumption, increases test application time because it reduces test concurrency.

This chapter presents a short survey of the techniques used in previous work in the field of low power testing of digital VLSI circuits (section 2.1). Then it briefly describes some techniques that exist in the literature in order to attain high fault coverage with short test application time (section 2.2). Finally, section 2.3 summarises the new techniques presented in this thesis for low power testing and high fault-coverage, and their contributions to the field of digital VLSI testing.

2.1 Previous Work in Low Power Testing

Several techniques have been proposed to reduce power consumption during test. These techniques can be categorised according to their method of application, such as external testing techniques or BIST techniques (test-per-clock techniques and test-per-scan techniques). Also they can be categorised according to their purpose, such as average power reduction techniques, and peak power reduction techniques (scan peak power and capture peak power in scan testing). Finally, they can be categorised according to the general idea and algorithm on which these techniques are based as will be done in the following subsections where many of the fundamental algorithms for low power techniques will be briefly described. Fig. 2.1 shows a taxonomy diagram that summarises the categories of the published low power testing techniques.

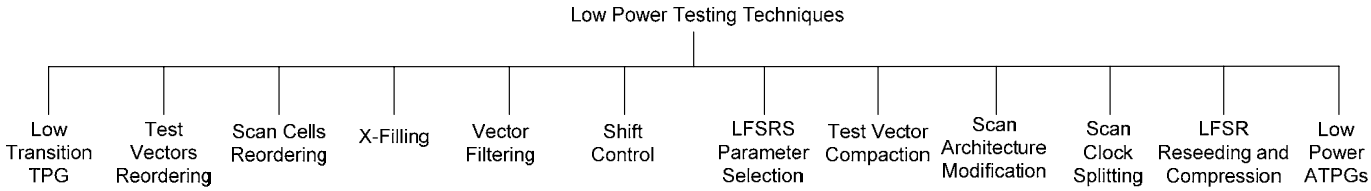

Fig 2.1 Taxonomy of the techniques used for low power testing.

2.1.1 Low Transition TPGs

One common technique to reduce test power consumption is the design of low transition TPGs [26, 41-50]. Most of these techniques modify the design of the LFSR (or other forms of TPGs such as cellular automata) in such a way as to reduce the transitions in the primary inputs of the CUT for test-per-clock BIST or inside the scan-chain for scan-based BIST.

An example of the low transition TPG for test-per-clock schemes is the approach presented in [26]. This approach, called Dual-Speed LFSR (DS-LFSR), is based on the use of two LFSRs working at two different speeds (a normal speed LFSR and a slow LFSR, see Fig. 2.2). The average power can be reduced by connecting the slow LFSR with the CUT inputs that cause high transition densities in the CUT. The remaining, inputs of the CUT are connected to the normal speed LFSR. The main drawbacks of this technique is that it is applicable only for test-per-clock BIST, and it needs a long sequence of test vectors to get an acceptable fault coverage. The approach presented in [46] is mostly similar to one presented in [26] but the power dissipation is reduced not only in the CUT but also in the clock tree feeding the CUT. The method presented in [47] uses weighted random TPG to reduce power consumption while keeping high fault coverage. This is done by inserting extra logic between the LFSR and the CUT.

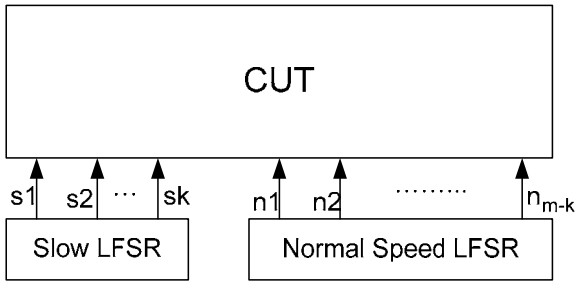

Fig. 2.2 DS-LFSR proposed in [26]

On the other hand, a TPG for low power consumption in scan-based BIST (test-per-scan BIST) is presented in [42]. The proposed design, called low transition random test pattern generator (LT-RTPG), is composed from an LFSR, a k-input AND gate, and a toggle flip-flop T-FF (see Fig 2.3). Some cells of the LFSR are connected with the inputs of the k-input AND gate, the output of the AND gate is connected with the input of the T-FF, and the output of the T-FF is connected with the scan-chain input (S_{in}). Since the output of the AND gate (input of the T-FF) is 0 in most of the cases, the T-FF output will not toggle in most of the clock cycles, and hence, neighbouring cells will have the same value in most cases. Thus the transition probability in the CUT will decrease. The main drawback of this system is that it reduces the average power while scanning-in a test vector not while scanning out the captured response. Also, in order to get a high fault-coverage, a long test sequence is needed.

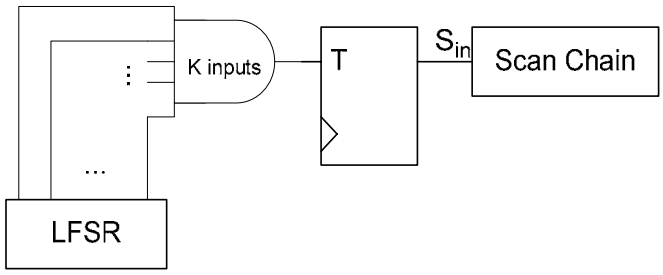

Fig. 2.3 LT-RTPG as proposed in [42]

2.1.2 Test Vectors Reordering

The test vectors reordering techniques aim to reduce the switching activity by modifying the order in which the tester applies the test vectors to the CUT. Normally these techniques can be used in external testing and deterministic BIST. In general, if the number of transitions between two consecutive vectors is reduced (i.e. the Hamming distance between two consecutive vectors is minimum), then the WSA will be reduced in the whole CUT [51]. The techniques presented in [51-57] aim to reorder the test vectors in such a way to reduce the number of transitions between the consecutive vectors before applying them to the CUT's primary inputs.

As a simple example to show how test vector ordering can reduce the number of transitions in the CUT, assume that we have a deterministic test patterns to test a CUT with 4 primary inputs in a test-per-clock scheme. These vectors are V1 = 0000, V2 = 1111, V3 = 0101, and V4 = 1010. If these vectors are applied to the CUT in the order V1, V2, V3, and V4, then they will cause a total number of transitions in the primary inputs of the CUT equal to 10. By contrast, if these vectors are applied in the order V1, V3, V2, and V4 then they will cause a total number of transitions in CUT's primary input equal to 6. Reducing the number of transitions in the primary inputs will reduce the overall switching activity in the CUT. This example is for illustration only, since in real cases more complex and advanced algorithms are proposed to order the test vectors to reach the optimal case of power reduction.

2.1.3 Scan Cells Reordering Techniques

Another category of techniques used to reduce the power consumption in scan-based BIST is the use of scan-chain cells ordering techniques [57-66]. Changing the order of the scan cells in each scan-chain can reduce the switching activity, and hence power dissipation, in scan designs. In the case of a deterministic set of test patterns, the best order of cells is the one that gives the best compromise between reducing the transitions in the scan cells both while scanning in test patterns and while scanning out captured responses. Scan-chain ordering is one of the most popular algorithms to reduce the power consumption in scan-based BIST because it has many advantages. For example, it does not require additional hardware, it has no effect on fault coverage or test application time, and its impact on design flow is low [20]. However, its main drawback is routing congestion problems during scan routing. Also, most of the techniques of scan-chain ordering aim to reduce the average power consumption during scanning in of test vectors and scanning out of captured responses by arranging the scan cells which cause more internal circuit transitions to the positions with low transition weights in the scan chain.

As a simple example to demonstrate how cell ordering algorithms reduce the number of transitions in the scan-chain, assume that there is a scan-chain with 3 flip-flops: A, B, and C with initial values of “000” as shown in Fig. 2.4. Assume that the deterministic test vector 101 is to be scanned into the scan chain (i.e. ABC = 101). Fig. 2.4(a) shows the scan shift with each clock. Since there are 3 flip-flops then 3 clock cycles are needed to scan in the test vector. The arrows in Fig. 2.4 indicate that a transition has occurred in a particular cell in between two adjacent clock cycles. In Fig. 2.4(a) it is clear that scanning in this vector produces 6 transitions in the scan-chain. Fig. 2.4(b) shows the scan-chain with the new order ACB (i.e. the test pattern will be ACB = 110), this will reduce the number of transitions in the scan-chain to 2 instead of 6 as shown in Fig. 2.4(b).

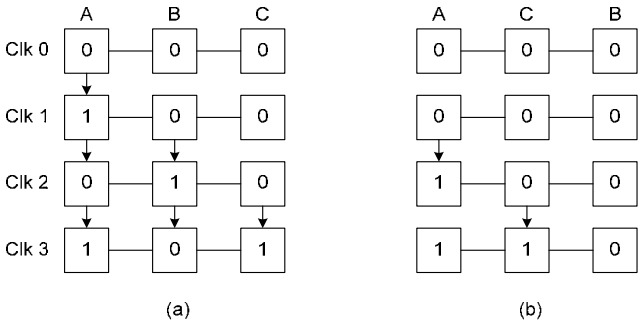

Fig. 2.4 Effect of cell ordering in the number of transitions. (a) Original order and (b) Cells after reordering

Although these algorithms mainly aim to reduce average power consumption, they can reduce the peak power that may occur in the CUT during the scanning cycles, but not the capture power that may result during the test cycle (i.e. between launch and capture). However, some of the ordering algorithms aim to order the scan-chain cells in order to minimise the peak power that may arise during the test cycle (capture power) [64]. But these algorithms for reducing capture power are orthogonal with the algorithms that aim to reduce the scan power. None of the techniques in the surveyed literature tried to find a compromise between these algorithms or to merge the two techniques together in order to minimise the overall peak power.

2.1.4 X-Filling Techniques of Test Cubes

Test cubes are test vectors where the values of some bits are left unspecified as X's (don't care bits); for example 1X00X1 is a test cube. In scan-based BIST, a test vector that detects many targeted faults may contain many don't care bits (X). In conventional scan ATPG, each X bit is filled with a 0 or 1 randomly since this will not affect the fault coverage. In fact, the number of X bits in each test cube is typically high [17, 67, 68]. In order to reduce the power consumption while scanning in a test vector, a power-aware X-filling technique is used to properly assign logic values to X-bits in a test cube. The main aim of such techniques is to assign a value to the X bits in the test cube so that the number of the transitions in the scan cells is minimised, hence reducing the transitions in the scan-cells, which leads to a reduction in the overall switching activity in the CUT during shift cycles [69-73]. The non-random X-filling techniques can be divided into 3 approaches in order to reduce shift power:

- Minimum transition filling technique (MT-filling), also called adjacent filling, where the most recent non-X bit is used to fill successive X values until a non-X bit is reached.
- 0-Filling. Set 0 to all X-bits.
- 1-Filling. Set 1 to all X-bits.

As a simple example, Consider the test cube [0XX1X0X1XX1X0X]. By applying the above filling techniques the resulting vectors and the number of transitions will be

- MT-Filling: 00011001111100 (4 transitions)
- 0-Filling: 00010001001000 (6 transitions)
- 1-Filling: 01111011111101 (5 transitions)

There are also a few X-filling techniques that aim to reduce the capture peak power by minimising the distance between a test vector and its response [74-78] .

2.1.5 Vector Filtering Techniques

The test vectors that are generated by TPGs such as LFSRs are pseudorandom vectors. The fault detection capability of these vectors quickly reaches diminishing returns. Hence, after running a sequence of test vectors and detecting many faults, then only a few of the subsequent test vectors can still detect new faults. The vectors that do not detect new faults, but do consume power when applied to the CUT, can be filtered or inhibited from being applied to the CUT [22, 79-81]. These algorithms, in general, use extra logic (e.g. decoder circuitry). Using prior knowledge of the sequences of test vectors generated by TPGs such as the LFSR, they can prevent some sequences from being transmitted to the CUT by knowing the first and last vectors in this sequence. Thus they reduce the power consumption in the CUT.

2.1.6 Shift Control Techniques

In scan-based BIST, during scan shifting of test vectors, the combinational part of the CUT will have many transitions although the test vector is not yet applied. There are many techniques that try to eliminate the switching activity in the combinational part of the CUT during scan shifting cycles [28, 82-84]. This can be achieved by identifying a test vector, called input control or shift control, which is applied to the

CUT's primary inputs during scan, thereby minimising or eliminating switching activity in the combinational part of the CUT.

As a simple example, consider the case of a 2-input AND gate in the CUT, with one of its inputs directly accessible through a primary input, and the second input coming from the output of D flip-flop in the scan chain. If the primary input is set to logic 0, then the scanning of a test vector will not affect the output of this gate; the output of the AND gate is an input to other parts of the CUT, and hence, a reduction to the switching activity in the CUT is achieved.

2.1.7 Selection of LFSR Parameters

Some parameters of the LFSR may affect the power consumption of the CUT, namely the seed and the characteristic polynomial of the LFSR. In [85] it is found that the WSA obtained for a given primitive polynomial of the LFSR strongly depends on the seed (the difference in the WSA count between the best and worst seeds is significant). On the other hand, WSA count is almost independent of the primitive polynomial selection for the LFSR [85, 86]. Hence, it is more critical to select the best seed than to select the best characteristic polynomial. Furthermore, since the characteristic polynomial is not critical regarding the power consumption in the CUT, it is recommended in [86] to use the LFSR that needs the minimum number of XOR gates in its feedback since this will reduce area overhead and reduce the power consumption in the LFSR itself.

2.1.8 Low Power Test Vector Compaction

In scan-based circuits, in order to reduce the test data volume, compacting techniques are introduced to merge several test cubes. However, compacting test vectors greatly increases the power dissipation (it could be several times higher). Thus, low power test vector compaction techniques have been introduced to minimise the number of test cubes generated by the ATPG tool by merging test cubes that are compatibles in all bit positions under a power constraint [23, 87, 88]. In [23], it has been shown that by carefully merging the test cubes in a specific manner, the number of transitions in the scan-chain can be minimised. Thus, a greedy heuristic procedure was used to

merge test cubes in a way that minimises the number of transitions. Significant reductions in average and peak power consumption can be obtained by using this approach.

As a simple example to demonstrate how compaction can significantly affect the power consumption, assume that there are three test cubes in a system: C_1 = 0X0X0X, C_2 = X1X1X1, and C_3 = 0XXX11. By using the MT-filling technique described previously in section 2.1.4, the number of transitions in C_1 is 0 (no transitions), in C_2 is 0, and in C_3 is 1. However, using this method, all the test vectors after MT-filling must be applied to the scan-based CUT one by one, which is time-consuming. Instead, merging test cubes together is a good way to reduce test data and test application time. Thus, C_1 can be merged with C_2; since they are compatibles in all bit positions, they will produce the test cube C_4 = 010101. However, C_4 cannot be merged with C_3 since they are not compatible.

Now only two test vectors need to be applied to the CUT: C_3 and C_4. This is good from the point of view of test application time. However, C_4 has 5 transitions which significantly increases the average and peak power in the CUT, so this merging of C_1 and C_2 is risky from the point of view of power reduction, and is not recommended.
On the other hand, from the original test cubes: C_1, C_2, and C_3, the test cubes C_2 and C_3 are compatible with each other (they have no conflict in any bit position). Thus they can be merged together to produce test cube C_5 = 01X111. However C_5 cannot be merged with C_1 since they are not compatible. So, we still have 2 test cubes (C_1 and C_5); C_5 has one transition which is good regarding power consumption compared with the previous scenario of compaction.

2.1.9 Scan Architecture Modification

This technique involves modifying the scan architecture by inserting new elements and partitioning the scan-chain into segments. In [89] the scan-chain is partitioned into N segments where only one segment is active at a time. This technique reduces the average power consumption in the CUT, but it will not affect the power consumption in the clock tree feeding the circuit. This problem has been addressed in [40], where separate clock trees are used for each scan segment. The scan segments

will be enabled by using the gated clock trees instead of scan enable signals as was used in the previous technique.

A similar approach can be used to reduce both peak and average power consumption as described in [90] where mutually exclusive scan segment activation signals are used in the scan architecture to get high reductions in peak power consumption. Other techniques based on scan architecture modifications are described in [91-96].

2.1.10 Scan Clock Splitting

This technique modifies the scan clock in order to reduce the power consumption during scan testing [97-99]. In [97] a technique based on scan clock splitting is presented. It involves reducing the operating frequency of the scan cells during scan shifting without modifying the total test time. In this technique, two clocks have been used; each of them is at half the frequency of the system clock, and is used to activate half of the scan cells in the scan-chain. The use of such a system reduces the average and peak power consumption without increasing test time. Another technique that uses a staggered clock scheme to reduce the peak power consumption during scan testing is presented in [98].

2.1.11 LFSR Reseeding and Test Data Compression Techniques

Considering together the problems of test data compression and low power test is very important. A long sequence of test vectors needs more time to be applied to the CUT, which also means high energy consumption since the total energy is a function of time. Hence, compression of test data leads to reduction to the total energy. Furthermore, using clever techniques for the compression will also reduce the average and peak power consumption.

The low power test data compression techniques can be divided into three main categories [100]: coding-based techniques, linear-decompression-based techniques (LFSR reseeding techniques) and broadcast-scan-based techniques.

Examples of coding-based techniques of low power test data compression can be found in [101-105]. In [101], the test cubes generated by ATPG are encoded using Golomb codes which are developed from run-length codes. All don't care (X) bits are mapped to 0 and the Golomb code is used to encode runs of 0s. This technique is efficient to compress the runs of 0s but inefficient to encode the runs of 1s and the test storage may increase if there are many runs of 1s in the test cubes. The technique presented in [102] based on alternating run-length coding overcomes this problem.

On the other hand, LFSR reseeding techniques for test data compression are considered to be an efficient approach to compression since only 1% to 5% of the bits in a test vector are considered to be specified (all remaining bits are don't care). The basic idea of LFSR reseeding is to generate deterministic test cubes by expanding seeds. A seed is an initial state of the LFSR that is expanded to a test vector by running the LFSR. Given a deterministic test cube and characteristic polynomial of the LFSR, the seed can be computed by solving a set of linear equations. The main advantage of this is that the seed is normally much shorter than the test cube [106].

In [107] a method that uses dual LFSR reseeding technique is used to reduce total power by masking some bits in the test vector by using AND and OR gates, thus reducing the transition probability. However, the method needs a hardware area overhead and increases test storage since there are two LFSRs used in the technique. In [108] the authors present a scheme that uses a LFSR reseeding technique to reduce the number of transitions in the scan-chain with an acceptable area overhead and good test power reduction.

Finally, the broadcast-scan-based-schemes of low power test data compression are based on broadcasting the same value to multiple scan chains when the scan chain is split into multiple segments, thus allowing the same data to be loaded into all compatible segments [109, 110] .

2.1.12 Low Power ATPG Algorithms

Many new ATPGs have been proposed in order to reduce the power consumption while still achieving the main objectives of classical ATPGs such as fault coverage

and test length [111-114]. For example, in [111] the authors propose a new version of the path oriented decision making (PODEM) algorithm where the clever assignment of don't care bits minimises the number of transitions at the CUT's primary input. This algorithm is extended to scan-based circuits in [114]. The algorithm described in [113] is dedicated to reducing the capture power during scan testing by generating test cubes that will reduce the power in the capture cycle.

2.1.13 Summary of Previous Low Power Testing Techniques

Each technique of low power testing described in the previous sections has some advantages and some disadvantages. Also, some of them are applicable to test-per-clock circuits, test-per-scan-circuits, or both of them. Table 2.1 summarises these techniques with their advantages and disadvantages. It is important to note that the techniques for low power testing – including the proposed techniques on this thesis – are circuit dependent (i.e. while a technique can significantly reduce the power consumption in one CUT, it may slightly reduce that for another CUT. This point is investigated in the experimental results of each proposed technique).

2.2 Testing for High Fault Coverage with Short Test Time (Test Energy Minimisation)

Testing for high fault coverage has a direct and important relationship with low energy testing; the conventional approach to testing, in order to increase the reliability of the system, the test should run till a high-fault coverage is achieved. This normally needs very long sequence of test vectors for most circuits. Hence, a long time is needed. Since the total energy consumed in the CUT during test is a function of time, conventional approach to testing, without improvement techniques, consumes high energy, which in turn decreases the battery lifetime. The main reason why the conventional way of testing requires a long test length is that the fault coverage is limited by the presence of random-pattern resistant (RPR) faults when a pseudorandom TPG is used.

TABLE 2.1 SUMMARY OF THE LOW POWER TESTING TECHNIQUES WITH THEIR ADVANTAGES AND DISADVANTAGES

Technique	Application	Advantages	Disadvantages
Low Transition TPGs	Test-per-Clock and Test-per-Scan BIST	Easy to implement with low hardware area overhead	Need a longer sequence of test vectors to get a high fault coverage
Test Vectors Reordering	External Testing	No extra cost or modification in the CUT and can be easily combined with other techniques	For large set of test vectors, the CPU time needed to reorder vectors is a bit high.
Scan Cells Reordering	Test-per-Scan BIST	Easy to implement and to be combined with other techniques. Can reduce the power significantly for many circuits	Routing congestion problems during scan routing
X-Filling	Test-per-Scan BIST	No area overhead is required and don't require any modification to the CUT	They require more test patterns to achieve a target fault coverage.
Vector Filtering	Test-per-Scan and Test-per-Clock BIST	Significant reduction to power and energy consumption during test	Significant area overhead which has a negative impact in the circuit performance.
Shift Control Techniques	Test-per-Scan BIST	Easy to implement and effective in reducing the power consumption	Significant area overhead is required, which may degrade the circuit performance as well.
LFSR Parameter Selection	Test-per-Clock and Test-per-Scan BIST	Reduce the power in significant amount without incurring a degradation to fault coverage or area overhead	Needs huge CPU time to find the best parameters (e.g. the best seed).
Test Vector Compaction	Test-per-Clock and Test-per-Scan BIST	Efficient in reducing the number of transitions without needing a hardware area overhead or modification to the CUT	In some cases more test patterns are needed to get a target fault coverage
Scan Architecture Modification	Test-per-Scan BIST	Reducing the power consumption without degrading the fault coverage.	Hardware area overhead is significantly increased
Scan Clock Splitting	Test-per-Scan BIST	Reducing the power consumption without affecting fault coverage.	It may not work properly with other fault models such as path delay faults
LFSR Reseeding and Compression Techniques	Test-per-Scan BIST	Efficient in compromising between reducing test data and reducing power consumption	Needs a moderate hardware area overhead
Low Power ATPG	External Testing	Reducing the power consumption without affecting fault coverage and hardware area overhead	Needs more CPU time than conventional ATPG

There are many techniques in the literature that suggest solutions to compromise between area overhead, fault coverage, and test application time. The first category of these techniques uses an LFSR (or other pseudorandom TPG) to detect easily-detectable faults (most of the faults) with an acceptable test length, and then generates an additional number of deterministic test patterns to detect RPR faults. The simplest approach for generating deterministic patterns on-chip is to store them in a read-only-memory (ROM) The problem with this approach is the expensive cost of the ROM needed to store the test patterns. Although there are many techniques that can be used to reduce the size of the ROM, the result is still not so efficient [115-121]. Another technique that can be used to achieve high fault coverage while reducing the size of the ROM is the LFSR reseeding technique. This technique uses the ROM to store the seeds that generate the deterministic test patterns instead of storing the patterns themselves since the seed is much shorter. The seeds can be computed with linear algebra as described in [106] . The main difference between the various reseeding methods is in the method used to select, order, and compute seeds so as to reduce the size of the ROM and increase the fault coverage [122-133].

Another category of techniques to increase the fault coverage is the weighted random TPG [134-141]. These techniques normally employ an LFSR and combinational logic and use analytical calculations of fault detection probabilities and some heuristics to generate test patterns with a non-uniform distribution of 0s and 1s for some inputs. Multiple weights are used in these techniques to detect more faults. The main drawback of these techniques is the area overhead required for additional logic, and that in many version of this technique, the number of test patterns required can still be excessively large.

Another category of techniques uses additional logic between the TPG and the CUT in order to map some useless test vectors to new vectors that can detect hard faults [142-145]. These techniques include algorithms such as bit-flipping and bit-fixing that transform a useless pattern to a deterministic one that detects RPR faults.

Another category of techniques to increase the fault coverage uses the test insertion point method to increase the controllability and observability of the internal nodes in the CUT. But adding new points to the CUT will add hardware area and slow the

performance of the circuit. Thus, it is an important issue to find the best location of the test points to be inserted in order to maximise the fault coverage and minimise the number of test points required. Several approximation techniques for placement of test points have been introduced using either fault simulation or testability measures to find the optimal locations of test points [146-148].

2.3 Contribution of the Work Presented in this Thesis

This thesis presents several new techniques for low power consumption, and also a high fault coverage testing technique. These techniques improve many test parameters such as

- reduction of average power consumption in test-per-clock and test-per-scan BIST,
- reduction of peak power consumption in scan-based BIST (both during scanning in/out of vectors/responses and during the capture cycle too), and
- reduction of the test application time while achieving high fault coverage in scan-based BIST (This will also reduce the energy consumption in the system).

All of these techniques increase the reliability of the system and reduce the test cost.

The presented techniques can be divided into three main categories. The first category includes the techniques that are based on the design of a low transition TPG for test-per-clock and test-per-scan BIST. The second category includes the techniques of scan-chain cell reordering algorithms that aim to reduce both average power and peak power in scan-based BIST. Finally, the last category includes the technique that increases fault coverage with a short test application time. This thesis has several contributions in theoretical concepts and experimental results in the field of digital VLSI circuits testing.

Firstly, chapter 3 presents a new low transition LFSR for test-per-clock BIST. The new design is based on a swapping property between every pair of adjacent cells in a conventional LFSR. The presented design, called Bit-Swapping LFSR (BS-LFSR), is mathematically proved to save 25% of the number of transitions produced by a conventional LFSR while still having the same properties to act as a pseudorandom TPG. The presented design passes many randomness tests and it had been

demonstrated that the BS-LFSR has negligible effect on fault coverage. Although the presented design (as a standalone design) has moderate results on average power savings during test, its main contributions come from two main points

- Firstly, the BS-LFSR can easily replace the conventional LFSR in many other low power techniques in order to achieve superior results in power savings. An example of this is presented in the experimental results.
- The second contribution is in the theory, since the presented design is systematic and the new properties are proved to work for any LFSR. This allows further investigation into the use of the BS-LFSR in other applications where the LFSR is used.

In chapter 4, the algorithm presented in chapter 3 is extended to be used with scan-based BIST. This gives the BS-LFSR the advantage of being universal as a TPG (i.e. can be used to generate test vectors for both test-per-clock and test-per-scan BIST). This is in contrast to many low transition TPGs found in the literature, which are designed either for test-per-clock BIST applications or test-per-scan BIST. The general design for the BS-LFSR for scan-based BIST can also save 25% of the transitions that occur in the scan-chain cells. Furthermore, chapter 4 introduces some special configurations of the BS-LFSR where the design can save 50% of the transition that may occur in the scan-chain. These special cases give the BS-LFSR the ability to substantially reduce the average power consumption, and the part of the peak power that occurs while scanning in a test vector. Furthermore, as there are many techniques for power minimisation proposed for scan-based BIST where the conventional LFSR is used as a TPG, many of these techniques are still applicable when the conventional LFSR is replaced by the proposed BS-LFSR. This will significantly increase the saving in power consumption.

In chapter 5, a new low transition LFSR is introduced for test-per-scan BIST. The new design, called Rotational LFSR (RLFSR), is based on a rotational technique between a group of cells in a conventional LFSR. The number of cells included in the group determines the smoothing degree of the generated test patterns. The more cells in the group, the smoother test patterns, but the greater the hardware area overhead. If the rotation group contains only two cells, then the RLFSR will be identical to BS-

LFSR. The chapter then shows how to extend the RLFSR to generate test patterns for test-per-clock BIST. It also shows how to use the RLFSR in multiple scan-chain BIST, where the RLFSR has an advantage over the conventional LFSR regarding the need for phase shifting. In the RLFSR it is shown that the correlation between the vectors in the different chains of a multiple scan-chain CUT is close to 0, whereas the correlation in a conventional LFSR is 1. Hence, there is no need for a phase-shifter in a system where the RLFSR is used, whilst it is needed with a conventional LFSR.

In chapter 6, a two-phase scan-chain ordering algorithm is presented. Much of the previously existing work that aims to reduce the peak power in scan-based BIST concerns the scan-shift power, while the capture power in the test cycle is ignored. Furthermore, the previous work that aims to reduce peak power in a capture cycle ignores the peak power in the scan shift cycles. Thus, the chapter starts with an experimental study about the sources of peak power violations in scan-based BIST. It concludes that the peak power violations come from many sources; these sources have relatively comparable amounts. So in order to substantially reduce peak power in a CUT, the peak power that may arise from scanning a new test vector should be reduced, the peak power that arises while scanning out a response should be reduced, and the peak power that may arise in the test cycle (capture power) should be reduced as well. The chapter concentrates on solving these problems jointly, and finding a compromise between them since reducing the peak power of one source and ignoring the peak power of the other source will not reduce the overall peak power by a significant amount. Firstly, the technique uses the BS-LFSR as a TPG; in this way it is guaranteed that the peak power that may arise while scanning in a test vector will be reduced. Secondly, it uses an existing scan-chain ordering algorithm with some modifications in order to eliminate the peak power that may arise while scanning out a response captured in the scan-chain. Finally, it presents a new scan-chain ordering algorithm that aims to reduce the peak power in the test cycle by reducing the Hamming distance between the input test vector and the captured response.

Finally, chapter 7 presents a novel technique for high fault coverage in full scan-based BIST. In most circuits there are faults known as hard to detect faults or random resistant patterns (RPR) faults that can not be detected by a random TPG till a very long sequence of test vectors has been generated; some circuits need hours or days of

test time to detect these faults. In conventional scan-based BIST the output of one LFSR cell is connected with the scan-input (S_{in}) of a single scan-chain. The novel technique presented in chapter 7, called Multi-Output LFSR (MO-LFSR), uses more than one LFSR cell's output to feed the S_{in} of the scan-chain, one every scan cycle. By cleverly selecting which cell to feed the S_{in} at the beginning of a scan cycle, most of the hard to detect fault can be detected with a short or moderate test length. Chapter 7 explains in detail the theoretical background of this technique and many of the LFSR properties which make this technique useful.

The work presented in chapters 3 to 7 has produced original work that has been published in [35-38]. These publications are:

1. A.S. Abu-Issa and S.F. Quigley, "Bit-Swapping LFSR and Scan-Chain Ordering: A Novel Technique for Peak and Average Power Reduction in Scan-Based BIST", IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, 28(5), May 2009, pp. 755-759.
2. A.S. Abu-Issa and S.F. Quigley, "Bit-Swapping LFSR for Low Power BIST", Electronics Letters, 44(6), March 2008, pp. 401- 403.
3. A.S. Abu-Issa and S.F. Quigley, "A Multi-Output Technique for High Fault Coverage in Test-per-Scan BIST", 3rd International Conference on Design and Technology of Integrated Circuits and Systems in Nanoscale Era, March 2008, pp. 1-6.
4. A.S. Abu-Issa and S.F. Quigley, "LT-PRPG: Power Minimisation Technique for Test-per-Scan BIST", 3rd International Conference on Design and Technology of Integrated Circuits and Systems in Nanoscale Era, March 2008, pp. 1-5.

Furthermore, another 2 papers are under preparation for a journal and a conference submission.

CHAPTER 3

BS-LFSR as a Low Transition TPG for Test-per-Clock BIST

This chapter presents a new TPG that aims to change the order of consecutive test patterns at the outputs of an LFSR in order to reduce the number of transitions between 0 and 1 that occur in the output stream of the LFSR by 25%. The proposed TPG, called Bit Swapping LFSR (BS-LFSR), can be used to reduce the average power consumed by the CUT in test-per-clock BIST during the application of test vectors by reducing the internal switching activity. The BS-LFSR can be implemented with a modest hardware overhead and can simply replace the normal LFSR used in other low power BIST designs to achieve better results in power saving. Experimental results have been obtained on international symposium on circuits and systems (ISCAS'85 and ISCAS'89) benchmark circuits [149, 150]. The results show reductions of internal node transitions of up-to 28% with an average of 22% during test vector application. Another set of experimental results show that significant reduction in transitions occurs when the BS-LFSR replaces the conventional LFSR in another low power test technique.

This chapter is organised as follows: Section 3.1 introduces some new observations about the output stream of the LFSR with their proofs and demonstration examples. Section 3.2 describes the design and the general appearance of the BS-LFSR. Section 3.3 discusses the randomness of the outputs produced by the new design. Section 3.4

shows how the BS-LFSR can replace the conventional LFSR in other low power techniques to get further reductions in average power. Section 3.5 shows the experimental results obtained on the ISCAS'85 and ISCAS'89 benchmark circuits. Finally, section 6 presents the conclusions and suggestions for future work.

3.1 The Proposed Approach to reducing the Transition Count

This section introduces two observations about the output of a maximal length LFSR. Both observations are related to the number of transitions produced by LFSR outputs which feed the CUT inputs (assuming a test-per-clock configuration).

Observation 3.1: For any n-bit maximal-length LFSR that starts with any seed and runs for 2^n clock cycles until it returns to the starting seed value, then the total number of transitions T_{total} that occurs is given by the formula in equation (3.1):

$$T_{total} = n \times 2^{(n-1)} \quad (3.1)$$

Proof

The sequence of 1s and 0s that is followed by one bit position of a maximal length LFSR is commonly referred to as an m-sequence. Each bit within the LFSR will follow the same m-sequence with a one time step delay. The m-sequence generated by an LFSR of length n has periodicity 2^n-1.

It is a well-known standard property of an m-sequence of length n that the total number of runs of consecutive occurrences of the same binary digit is 2^{n-1}. Furthermore exactly half of the runs are of length 1, a quarter are of length 2, an eighth of length 3 and so on up to a single run of zeroes of length n-1 and a single run of ones of length n [7, 151].

The beginning of each run is marked by a transition between 0 and 1, so the total number of transitions for each stage of the LFSR is 2^{n-1}. Since the LFSR has n stages, each following an m-sequence, the total number of transitions must be as given in equation (3.1). Furthermore, the total number of transitions is equally distributed

between the outputs of the LFSR, which means that each bit of the n-bit LFSR has a total transition count of 2^{n-1}. This observation can also be proved using the toggle property of the XOR gates used in the feedback of the LFSR [35]. Also, it can be proved in using a different approach as will be shown in chapter 5.

Observation 3.2: Consider any *n*-bit maximal length LFSR (*n*>2). Suppose we modify the LFSR by considering one of its outputs (say bit *n* - the last bit) to be a selection line that will swap two neighbouring bits elsewhere in the LFSR when the selection line has a specific value (say 0) (i.e. if *n* is odd and bit *n* = 0, then bit 1 will be swapped with bit 2, bit 3 with bit 4, …, bit *n*-2 with bit *n*-1. If *n* is even and bit *n* = 0, then bit 1 will be swapped with bit 2, bit 3 with bit 4,…, bit *n*-3 with bit *n*-2, in all cases the selection line, bit n in this case, is excluded from the swapping operation. If bit *n* = 1, then no swapping is performed). In this case:

1- The exhaustive set of test vectors generated by this modified LFSR will be the same as the test vectors generated by the corresponding unmodified LFSR, but their order will be different.
2- Each pair of neighbouring bits that are swapped will save a number of transitions equal to $T_{saved} = 2^{(n-2)}$. Since these two bits originally produced $2 \times 2^{(n-1)}$ transitions as shown in the previous observation, the bit swap will therefore save $T_{saved\%} = 2^{(n-2)} / (2 \times 2^{(n-1)}) = 25\%$.

Proof

Part 1 of observation 3.2 is true because all possible vectors (except for the all-zero vector) will be produced at the output of an LFSR exactly once during a complete run of 2^n-1 clock cycles. For an arbitrary output vector v, its swapped counterpart v1 will occur once in the output stream. The swapping operation is symmetric, mapping v to v1 and v1 to v. Both v and v1 will have an identical value for the bit which acts as the selection line (since this bit is excluded from the swapping operation). As a result, if the value at the selection line dictates that swapping should take place, then v will replace v1 and v1 will replace v. So the set of output vectors is unchanged, but they will appear in a different order.

The simplest way to prove point 2 is by exhaustive enumeration of all possible initial states of a pair of bits together with all possible subsequent states. Suppose we have

an n-bit LFSR with bits 1, 2, 3… *m*, *m*+1,… *n*. If bit *n* (the selection bit) has a value of 0, then bit 1 will be swapped with bit 2, and bit 3 with 4,… and bit *m* will be swapped with bit *m*+1, and so on. If bit *n* has a value of 1, then no swapping will be performed. If we consider all possible values of current state for bits *m*, *m*+1, and *n* we have 8 possible combinations. If we then consider all possible values of the following state, we have 4 possible combinations (not 8, because the value at location *m*+1 in the next state is determined by the value at location *m* in the present state). Table 3.1 exhaustively summarises all possible combinations of present state and successor states for both the original vector and the swapped vector.

TABLE 3.1 EXHAUSTIVE LIST OF POSSIBLE TRANSITIONS FOR A PAIR OF BITS IN A STANDARD LFSR AND IN THE PROPOSED BS-LFSR

	Original Vector ***m,m+1,*** ***,n***				Swapped Vector ***m,m+1,*** ***,n***			
Possible State # 1	**00** 0				**00** 0			
Possible Next States	**00** 0	**00** 1	**10** 0	**10** 1	**00** 0	**00** 1	**01** 0	**10** 1
No. of Transitions (m,m_{+1})	0	0	1	1	0	0	1	1
Possible State # 2	**00** 1				**00** 1			
Possible Next States	**00** 0	**00** 1	**10** 0	**10** 1	**00** 0	**00** 1	**01** 0	**10** 1
No. of Transitions (m,m_{+1})	0	0	1	1	0	0	1	1
Possible State # 3	**01** 0				**10** 0			
Possible Next States	**00** 0	**00** 1	**10** 0	**10** 1	**00** 0	**00** 1	**01** 0	**10** 1
No. of Transitions (m,m_{+1})	1	1	2	**2**	1	1	2	**0**
Possible State # 4	**01** 1				**01** 1			
Possible Next States	**00** 0	**00** 1	**10** 0	**10** 1	**00** 0	**00** 1	**01** 0	**10** 1
No. of Transitions (m,m_{+1})	1	1	**2**	2	1	1	**0**	2
Possible State # 5	**10** 0				**01** 0			
Possible Next States	**01** 0	**01** 1	**11** 0	**11** 1	**10** 0	**01** 1	**11** 0	**11** 1
No. of Transitions (m,m_{+1})	2	**2**	1	1	2	**0**	1	1
Possible State # 6	**10** 1				**10** 1			
Possible Next States	**01** 0	**01** 1	**11** 0	**11** 1	**10** 0	**01** 1	**11** 0	**11** 1
No. of Transitions (m,m_{+1})	**2**	2	1	1	**0**	2	1	1
Possible State # 7	**11** 0				**11** 0			
Possible Next States	**01** 0	**01** 1	**11** 0	**11** 1	**10** 0	**01** 1	**11** 0	**11** 1
No. of Transitions (m,m_{+1})	1	1	0	0	1	1	0	0
Possible State # 8	11 1				11 1			
Possible Next States	**01** 0	**01** 1	**11** 0	**11** 1	**10** 0	**01** 1	**11** 0	**11** 1
No. of Transitions (m,m_{+1})	1	1	0	0	1	1	0	0
Total if all states occur once	**32**				**24**			
%Total	**100%**				**75%**			

Table 3.1 shows that for any case of swapping bits *m* and *m*+1, either the swapped bits will have the same number of transitions as in the un-swapped case, or 2 transitions will be saved. In no case will swapping increase the number of transitions. In most cases, the number of transitions is unchanged. Savings occur in four of the

cases: moving from state (0,1,0) to (1,0,1), from (0,1,1) to (1,0,0), from (1,0,0) to (0,1,1), or from (1,0,1) to (0,1,0). In each of these cases 2 transitions are saved.

All of the 8 possible starting states are equally probable, so the probability being in any particular starting state is ⅛. All of the four successor states are equally probable, so the probability of transitioning to any particular successor state (given a value of current state) is ¼. The probability of making a saving P_{save} is therefore given by equation (3.2)

$$P_{save} = P_{010\rightarrow101} + P_{011\rightarrow100} + P_{100\rightarrow011} + P_{101\rightarrow010} \quad (3.2)$$
$$= P(010)\times P(101|010) + P(011)\times P(100|011) + P(100)\times P(011|100) + P(101)\times P(010|101)$$
$$= ⅛ \times ¼ + ⅛ \times ¼ + ⅛ \times ¼ + ⅛ \times ¼ = ⅛$$

If the LFSR is allowed to move through a complete cycle of 2^n states, then observation 3.1 shows that the number of transitions expected to occur in the pair of bits under consideration is $2\times2^{n-1}$. Using the swapping approach, in ⅛ of the cases, a saving of 2 transitions will occur, giving a total saving of $⅛\times2\times2^n$. Dividing one figure by the other, we see that the total number of transitions saved by the swapping method applied to these two bit positions is 25%.

Example 3.1: Fig. 3.1 shows a 5-bit maximal-length LFSR, table 3.2 shows the test vectors generated by the LFSR (using a seed value of 11111) and the corresponding swapped vectors if bit ‘a’ swapped with bit ‘b’, and bit ‘c’ with bit ‘d’ when bit ‘e’ = 0. This example shows that each bit in the 5-bit LFSR generates 16 transitions as stated in observation 3.1, and a swap of every 2 neighbouring bits (the selection bit is excluded) will save 8 transitions as stated in observation 3.2 (In this example, the number of saved transition for bit ‘a’ is exactly the same as for bit ‘b’; similarly the saving for bit ‘c’ is the same as for bit ‘d’. This is not always the case as will be discussed in the following chapters. However, the overall total of saved transitions is always 25% for each pair of swapped bits).

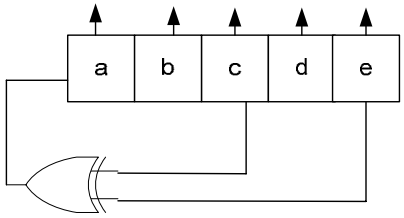

Fig. 3.1 5-Bit Maximal Length LFSR

TABLE 3.2 OUTPUT SEQUENCE GENERATED BY A 5-BIT MAXIMAL LENGTH LFSR AND THE CORRESPONDING SWAPPED SEQUENCE WHEN BIT E = 0.

Clock Cycle	Original Vectors					Vectors after swap				
	a	b	c	d	e	a	b	c	d	e
1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	1	0	1	1	1	1
3	0	0	1	1	1	0	0	1	1	1
4	0	0	0	1	1	0	0	0	1	1
5	1	0	0	0	1	1	0	0	0	1
6	1	1	0	0	0	1	1	0	0	0
7	0	1	1	0	0	1	0	0	1	0
8	1	0	1	1	0	0	1	1	1	0
9	1	1	0	1	1	1	1	0	1	1
10	1	1	1	0	1	1	1	1	0	1
11	0	1	1	1	0	1	0	1	1	0
12	1	0	1	1	1	1	0	1	1	1
13	0	1	0	1	1	0	1	0	1	1
14	1	0	1	0	1	1	0	1	0	1
15	0	1	0	1	0	1	0	1	0	0
16	0	0	1	0	1	0	0	1	0	1
17	0	0	0	1	0	0	0	1	0	0
18	0	0	0	0	1	0	0	0	0	1
19	1	0	0	0	0	0	1	0	0	0
20	0	1	0	0	0	1	0	0	0	0
21	0	0	1	0	0	0	0	0	1	0
22	1	0	0	1	0	0	1	1	0	0
23	0	1	0	0	1	0	1	0	0	1
24	1	0	1	0	0	0	1	0	1	0
25	1	1	0	1	0	1	1	1	0	0
26	0	1	1	0	1	0	1	1	0	1
27	0	0	1	1	0	0	0	1	1	0
28	1	0	0	1	1	1	0	0	1	1
29	1	1	0	0	1	1	1	0	0	1
30	1	1	1	0	0	1	1	0	1	0
31	1	1	1	1	0	1	1	1	1	0
32	1	1	1	1	1	1	1	1	1	1
Transitions per bit	16	16	16	16	16	12	12	12	12	16
Transitions per pair of neighbouring bits	32		32			24		24		

The method works because it exploits the correlation between bit *m* on one clock cycle and bit *m*+1 on the successor clock cycle that will inevitably occur within an LFSR. Savings can be achieved when transitioning a pair of bits from the unswapped state into the swapped state, or from swapped state into the unswapped state. If the bits remained permanently swapped (or permanently unswapped) then no savings would be made. In this work we have chosen to use the last bit position (bit *n*) within the LFSR as the controlling input that dictates whether or not the bits should be swapped. One could envisage other possible choices, but they would need to satisfy the following conditions:

- It must make frequent transitions between 0 and 1
- It must ensure that 2^n-1 distinct outputs can be produced by the LFSR in 2^n-1 clock cycles
- It must be possible to implement in simple hardware
- It should not destroy the randomness of the distribution of outputs

Using the last bit position (or any other bit position) within the LFSR is by far the simplest way to achieve these requirements.

It is also possible to envisage a version of the method that is applied to larger groups of bits (using 4, 8, 16 bits rather than 2) and applying rotations to the group. This can give larger savings as will be explained in the following chapters.

3.2 Design of the BS-LFSR

Based on the observations given above, the BS-LFSR can be designed using the original LFSR and a group of 2-input multiplexers. For an n-bit LFSR the n^{th} bit is considered to be the selection line. Each pair of neighbouring bits is multiplexed (i.e. bit 1 is multiplexed with bit 2, bit 3 with bit 4, and so on). If n is an odd number then all the bits except the n^{th} will be multiplexed and (n-1) 2-input multiplexers will be used. If n is even, then bit n-1 is not multiplexed as well as bit n (if bit n and n-1 were multiplexed together, we would lose property 1 of observation 2). So for n even, (n–2) 2-input multiplexers are required. Fig. 3.2 shows the appearance of the BS-LFSR.

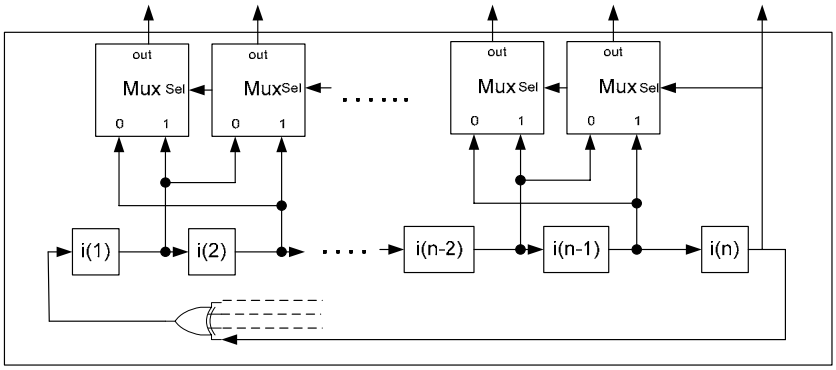

Fig. 3.2 General appearance of the proposed BS-LFSR

3.3 Randomness of the New Set of Test Vectors

One important requirement for the test vectors output by the BS-LFSR is that they should be randomly distributed. If this is not the case then there could be groups of inputs for the CUT that are repeatedly assigned to the same value. If this occurs, then it could cause the BS-LFSR to require a much longer stream of test vectors before an acceptable fault coverage is obtained than would be the case with a conventional LFSR. The randomness properties of the BS-LFSR were compared to those of a conventional LFSR in two ways.

Firstly, the output vectors were subjected to an equidistribution (frequency) test [26]. This essentially treats each of the n-bit test vectors as an n-bit binary representation of a natural number and checks the uniformity of the distribution of these numbers across the entire number range $[0,2^{n-1}]$. This is done by dividing the number range into a set of L bins each of size $I = \frac{2^n}{L}$. If we then generate a sequence of L test vectors, a uniform distribution would correspond to one generated number falling in each of the bins. The formula for the equidistribution test is then given by equation (3.3)

$$V = \sum_{i=1}^{L} \frac{\left(f_e^{\,i} - f_o^{\,i}\right)^2}{f_e^{\,i}} \tag{3.3}$$

where $f_e^{\,i}$ is the number of patterns expected to fall in each bin i and $f_o^{\,i}$ is the number of generated test vectors observed to fall in the bin. The smaller the value of V, the more uniform is the distribution of test vectors. A zero value of V indicates the ideal

uniform distribution. Table 3.3 shows a comparison between the values of V for test vector sequences generated by the BS-LFSR and also by a conventional LFSR. The test has been carried out for values of the test vector width n varying from 15 to 31. For each value of n, three different values have been used for L, the number of bins (which is also equivalent to the test length). The exact result for V varies slightly with the choice of starting seed used for the generation of the test vector sequence. Results are shown for three different randomly chosen seeds.

Table 3.3 compares the equidistribution properties of the BS-LFSR with those of the conventional LFSR. Perfect agreement between the two would correspond to a value of V2/V1 equal to 1. Where the ratio exceeds 1, this means that the BS-LFSR has inferior equidistribution properties to the conventional LFSR; where the ratio is less than 1, this means that the BS-LFSR is superior. The mean value of the ratio is very close to 1 and the values are equally distributed around this mean value with a small standard deviation.

The second test used to evaluate the randomness of the test vectors produced by the BS-LFSR was to apply the test vectors to a variety of benchmark circuits and to record how long a test sequence was required in order to achieve a target value of required fault coverage. This was done both for the BS-LFSR and the conventional LFSR using a variety of circuits from the ISCAS'85 and ISCAS'89 test sets. Table 3.4 shows the results which are based on the average of three randomly selected seeds. The results show that the results for the BS-LFSR are comparable to those obtained with a conventional LFSR.

3.4 Combining the BS-LFSR with other Low Power Test Techniques

One of the main advantages of the proposed BS-LFSR is the simplicity with which it can be used to replace the conventional LFSR used in other low power test techniques to obtain superior results in terms of power consumption. For example, Fig. 3.3 shows the Dual Speed LFSR (DS-LFSR) suggested in [26] to reduce power consumption during test and described in more details in section 2.1.1. If the slow and normal

speed LFSRs shown in Fig. 3.3 are replaced by slow and normal speed BS-LFSRs then superior power consumption can be achieved as will be shown in the experimental results presented in section 3.5. Also, many other low power techniques can be investigated to determine whether it is possible to replace the LFSR in these techniques by the BS-LFSR in order to achieve further reduction in power consumption.

TABLE 3.3 RESULTS OF THE EQUIDISTRIBUTION TEST. A VALUE OF V2/V1 OF 1 INDICATES THAT THE EQUIDISTRIBUTION PROPERTIES OF THE BS-LFSR MATCH THOSE OF THE CONVENTIONAL LFSR.

n	*L*	*V*								
		Seed1			Seed2			Seed3		
		LFSR (V1)	BS-LFSR (V2)	(V2/V1)	LFSR (V1)	BS-LFSR (V2)	(V2/V1)	LFSR (V1)	BS-LFSR (V2)	(V2/V1)
15	256	394	358	0.909	242	240	0.992	244	228	0.934
	512	474	482	1.017	518	508	0.981	372	374	1.005
	1024	1138	1162	1.021	918	916	0.998	1370	1352	0.987
17	1024	961	991	1.031	950	980	1.032	848	936	1.104
	2048	1811	1777	0.981	1762	1802	1.023	1854	1858	1.002
	4096	3781	3711	0.981	3764	3840	1.020	3717	3755	1.010
19	1024	1010	1022	1.012	988	968	0.980	902	898	0.996
	2048	2432	2446	1.006	1955	1935	0.990	2012	2008	0.998
	4096	4162	4126	0.991	4014	4016	1.000	3890	3854	0.991
23	2048	2186	2190	1.002	1845	1837	0.996	2000	2004	1.002
	4096	4336	4270	0.985	3893	3975	1.021	4058	4076	1.004
	8192	7592	7594	1.000	7870	7866	0.999	8184	8192	1.001
24	2048	1882	1880	0.999	1987	1965	0.989	2048	1972	0.963
	4096	3988	3980	0.998	4490	4498	1.002	4023	4039	1.004
	8192	7956	7884	0.991	8791	8661	0.985	8110	8100	0.999
25	2048	2044	2032	0.994	1941	1933	0.996	2078	2072	0.997
	8192	9158	9152	0.999	8230	8218	0.999	8578	8598	1.002
	16384	17236	17120	0.993	16765	16661	0.994	15343	15361	1.001
	32768	33383	33407	1.001	32873	32821	0.998	32027	32031	1.000
31	8192	12622	12642	1.002	8335	8359	1.003	8106	8116	1.001
	16384	21994	21614	0.983	16521	16441	0.995	17621	17549	0.996
	32768	40790	40672	0.997	36641	36647	1.000	33834	33734	0.997

TABLE 3.4 LENGTH OF TEST VECTOR SEQUENCE REQUIRED IN ORDER ACHIEVING A GIVEN LEVEL OF FAULT COVERAGE FOR THE CONVENTIONAL LFSR AND THE PROPOSED BS-LFSR.

Circuit	Fault Coverage (%)	Vectors Needed using LFSR	Vectors Needed using BS-LFSR	circuit	Fault Coverage (%)	Vectors Needed using LFSR	Vectors Needed using BS-LFSR
C6288	99.5	104	107	S641	98	4,470	5,150
C1908	99	4,450	4,243	S838	87	51,450	45,455
C432	99	733	800	S1423	99	25,013	15,910
C1355	99	1,525	1,640	S5378	99	51,900	45,760
C499	98.5	530	512	S9234	86	25,760	26,277
C3540	96	31,050	39,700	S13207	95	51,050	45,210
C880	99.5	2,760	3,025	S15850	91.5	35,665	33,740
C5315	98.5	790	825	S35932	89.8	4,915	5,380
C7552	94	11,900	10,425	S38417	87.5	53,930	57,645
C2670	85	62,450	45,950	S38584	94	43,400	40,660

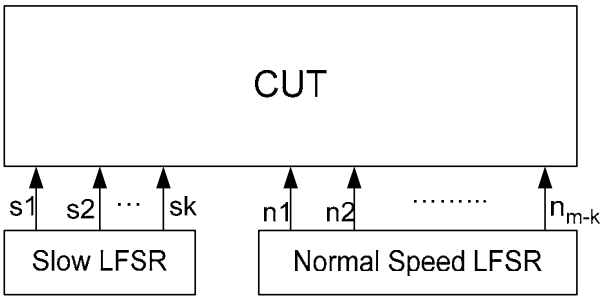

Fig. 3.3 DS-LFSR as Proposed in [26].

3.5 Experimental Results

A set of experiments were performed in order to evaluate the effect of the proposed BS-LFSR in power saving and fault coverage. Firstly, a standalone BS-LFSR is evaluated and the results of transition saving and fault coverage achieved is compared with a conventional LFSR. The test vectors generated both by the LFSR and the BS-LFSR are applied to ISCAS'85 and ISCAS'89 Benchmark circuits. The tap sequences for maximal-length LFSRs was used for all LFSRs of length less than or equal to 786 bits and obtained from [7, 152, 153]. For all longer LFSRs, the polynomial $x^n + x + 1$ was used. In general, this gives a very large number of unrepeated test vectors which was sufficient for the purpose of our experiment to get high fault coverage.

For each of the benchmark circuits with n<100 (where n is the number of primary inputs for ISCAS'85 circuits and the sum of primary and state inputs for ISCAS'89) , a set of 100, 500, and 5000 test vectors generated by both the LFSR and the BS-LFSR were applied to CUT in order to compare the fault coverage in each case. For the benchmark circuits with n>100 a set of 500, 5000, and 20000 test vectors were used since in these circuits, in general, more test vectors are needed to get an acceptable fault coverage. Application of these test vectors and recording of the fault coverage was carried out using the FSIM tool [11].

The above procedure was repeated for three randomly selected seeds to check the effect of the seed value on the fault coverage and the total transition count. The average value for all seeds in each of the above cases is tabulated in table 3.5(a) and table 3.5(b). In all cases, good savings in switching activity were made by the BS-LFSR, with negligible difference in fault coverage.

It is important to note that the total number of transitions in table 3.5 is the transitions occurred in the CUT. However, the extra multiplexers which are added to the LFSR will increase the number of transitions in the whole system. An estimation for the transitions that will be caused by the multiplexers shows that 5% to 10% transition overhead will occur in the system.

In order to provide comparison with other existing techniques, and to show how the BS-LFSR can replace conventional LFSR in other low power techniques to obtain extra savings in power consumption, table 3.6 shows a comparison of the reductions in switching activity for the BS-LFSR proposed in this chapter with those obtained in [26]. The first column shows the benchmark circuits used in both designs. The second column shows the results obtained in this chapter, while the third column shows the results obtained in [26]. The last column shows the new results obtained by replacing the LFSRs used in [26] with the BS-LFSR proposed in this chapter.

TABLE 3.5A EVALUATION OF FAULT COVERAGE AND SAVINGS IN TRANSITIONS FOR THE BS-LFSR COMPARED TO A CONVENTIONAL LFSR

circuit	n*	100 Test Vectors					500 Test Vectors					5000 Test Vectors					Avg. save %
		LFSR		BS-LFSR		Saving %	LFSR		BS-LFSR		Saving %	LFSR		BS-LFSR		Saving %	
		#transitions	FC%	#transitions	FC%		#transitions	FC%	#transitions	FC%		#transitions	FC%	#transitions	FC%		
C6288	32	88820	99.48	80164	99.47	9.7	461573	99.56	405139	99.56	12.2	4621272	99.56	4017846	99.56	13.1	11.7
C1908	33	38554	76.24	30061	77.09	22.0	200540	91.49	160524	91.09	20.0	1959311	99.29	1597096	99.28	18.5	20.2
C432	36	7219	91.13	5844	91.13	19.0	36501	98.09	30259	98.01	17.1	393334	99.24	308617	99.24	21.5	19.2
C1355	41	22007	88.63	16790	88.66	23.7	117674	95.05	89721	96.19	23.8	1151353	99.49	903185	99.49	21.6	23.0
C499	41	9879	93.47	7985	93.47	19.2	50664	98.36	41563	98.42	18.0	496663	98.95	419297	98.95	15.6	17.6
C3540	50	54997	80.84	48798	80.31	11.3	280270	92.23	240068	91.95	14.3	2905006	95.89	2457729	95.86	15.4	13.7
C880	60	15287	91.09	13003	91.51	14.9	74482	96.50	66054	96.77	11.3	735406	99.84	646312	99.58	12.1	12.8
S641	54	15560	88.88	11833	87.05	24.0	76978	94.06	63114	93.09	18.0	771971	98.06	617024	97.95	20.1	20.7
S838	66	13817	73.63	9549	70.71	30.9	71553	82.79	51929	82.21	27.4	693557	85.41	522801	85.41	24.6	27.6
S1423	91	31940	90.17	24330	88.98	23.8	160234	95.39	118118	94.75	26.3	1588963	98.29	1189413	98.58	25.1	25.1

TABLE 3.5B EVALUATION OF FAULT COVERAGE AND SAVINGS IN TRANSITIONS FOR THE BS-LFSR COMPARED TO A CONVENTIONAL LFSR

circuit	n*	500 Test Vectors					5000 Test Vectors					20000 Test Vectors					Avg. save %
		LFSR		BS-LFSR		Saving %	LFSR		BS-LFSR		Saving %	LFSR		BS-LFSR		Saving %	
		#transitions	FC%	#transitions	FC%		#transitions	FC%	#transitions	FC%		#transitions	FC%	#transitions	FC%		
C5315	178	497472	98.13	406624	98.07	18.3	4885155	98.88	4023167	98.88	17.6	19883413	98.90	16292485	98.90	18.1	18.0
C7552	207	779125	91.49	653585	91.51	16.1	7700663	93.27	6490393	93.48	15.7	31001959	94.65	25010099	94.66	19.3	17.0
C2670	233	309516	83.48	238876	83.42	22.8	3103850	84.57	2406376	84.53	22.5	12992933	84.68	9912907	84.66	23.7	23.0
S5378	214	554527	91.31	421805	91.03	23.9	5603595	96.89	4299109	96.93	23.3	22479531	97.30	16264210	98.19	27.6	24.9
S9234	247	1207849	67.73	864051	66.37	28.5	12136753	80.17	8687127	79.56	28.4	45440830	85.44	34066100	85.16	25.0	27.3
S13207	700	1719619	74.73	1270977	74.74	26.1	17000543	82.46	12588819	84.25	26.0	66401120	89.09	50172832	89.95	24.4	25.5
S15850	611	1985477	81.38	1527389	81.19	23.1	21399492	87.38	16198875	87.54	24.3	83183413	90.72	64537449	90.87	22.4	23.3
S35932	1763	4213060	89.57	3124328	89.63	25.8	42042237	89.80	31797609	89.79	24.4	178287901	89.81	132455865	89.81	25.7	25.3
S38417	1664	2305660	54.56	1845130	55.12	20.0	22691012	56.26	17063044	57.05	24.8	127681406	75.29	94916603	76.08	25.7	23.5
S38584	1464	3744411	86.84	3004712	86.64	19.8	38375391	90.58	30832567	90.43	19.7	156028529	92.84	124923691	92.91	19.9	19.8

* n is the number of primary inputs for ISCAS'85 and the sum of primary and state inputs for ISCAS'89

TABLE 3.6 COMPARISON OF THE SAVINGS IN NUMBER OF TRANSITIONS ACHIEVED BY THE PROPOSED BS-LFSR WITH THOSE IN [26] AND THE COMBINED SAVINGS

circuit	Saved T% BS-LFSR	Saved T% in [26]	Combined T% saving
C6288	11.7	13	20.6
C1908	20.2	18	29.7
C1355	23.0	14	31.2
C3540	13.7	30	40.1
C880	12.8	27	37.5
C5315	18.0	19	32.0
C7552	17.0	14	27.5
C2670	23.0	24	39.0
S641	20.7	51	62.7
S838	27.6	32	51.1
S1423	25.1	44	59.9
S5378	24.9	39	53.8
S9234	27.3	47	63.2
Average Saving in these CUTs	21.9	28.6	42.2

3.6 Conclusion

This chapter has presented a modification to the structure of the LFSR that re-orders the output test vectors in a manner that reduces the number of transitions between 0 and 1 within each bit position of the output test patterns. Experimental results show a good saving in switching activity during test (ranging from 11.7% to 27.6%), without any significant effect on the fault coverage. The suggested design can be combined with other techniques to achieve extra reduction in the power consumption. An example has been shown in the experimental results, leads to further reductions in the number of node transitions. The proposed BS-LFSR can be achieved with modest hardware area overhead (about 16% overhead to the LFSR size).

This work can be extended further by taking the principles underlying the proposed BS-LFSR and combining them with other approaches to low transition test generation that have already been reported in the literature. This has the potential to give excellent power savings for device testing. Also the proposed BS-LFSR can be extended to work for test-per-scan BIST as will be shown in the following chapters.

CHAPTER 4

BS-LFSR as a Low Transition TPG for Test-per-Scan BIST

In this chapter, the BS-LFSR proposed in the previous chapter will be extended to be used as a low transition TPG for test-per-scan BIST. The new BS-LFSR is composed of an LFSR and a 2x1 multiplexer (i.e. the hardware area overhead is negligible: only one 2x1 multiplexer).

In a single scan-chain test-per-scan BIST, one of the cells of the LFSR is chosen to feed a scan chain. It is easy to see from the result of the previous chapter that using a BS-LFSR to feed the scan chain can reduce the number of transitions that occur at the scan chain input during scan shift operation by 25% when compared to those patterns produced by a conventional LFSR. Furthermore, as will be shown in this chapter by lemmas 3 to 10, the BS-LFSR can be configured to increase the transition reduction to 50% (instead of 25%) in full scan-chain circuits. Hence it significantly reduces the overall switching activity in the CUT during test application.

The proposed technique has a substantial effect on average power reduction with negligible effect on fault coverage or test application time. Experimental results on the ISCAS'89 benchmark circuits show up to 60% reductions in average power consumption while scanning in the sequence of test vectors to the scan-chain.

Various properties of the proposed design and the methodology of the design are presented in this chapter. This chapter is organised as follows: section 4.1 states the

lemmas of the special configurations that achieve 50% transition reduction at the scan-input of the CUT with their proofs and some illustrative examples. Section 4.2 shows the general architecture of the BS-LFSR as a TPG for test-per-scan BIST. Then section 4.3 shows the experimental results of the proposed design. Finally, section 4.4 states the main conclusions of this chapter.

4.1 The Proposed Approach to Design the BS-LFSR for Scan-Based BIST

The proposed BS-LFSR for test-per-scan BIST is based upon some observations (lemmas) concerning the number of transitions produced at the output of an LFSR. This chapter starts from two observations that were presented in the previous chapter (but are summarised again in lemma 4.1 and 4.2 for the sake of completeness). It then moves on to identify the special cases (lemmas 4.3 to 4.10) that achieve 50% saving in transitions for one of the LFSR outputs that will be selected to feed the scan-chain input of a full scan circuit. Before stating the lemmas and their proofs, we start with a definition for the term “adjacent cells” which will be used frequently in the text.

***Definition 4.1*:** Two cells in an n-bit LFSR are considered to be adjacent if the output of one cell feeds the input of the second directly (i.e. without intervening XOR gate).

For external n-bit LFSR, each cell (say cell x) is adjacent to its subsequent cell (cell x+1), but cell n (last cell) is not adjacent to first cell since they are connected with each other using XOR gate. For an internal LFSR it is not necessary that each cell is adjacent to its subsequent cell since in some cases there is an intervening XOR gate between them, but cell n (the last cell) is adjacent to the first cell since the output of cell n feeds the input of cell 1 directly.

Lemma 4.1: Each cell in a maximal-length n-stage LFSR (internal or external) will produce a number of transitions equal to 2^{n-1} after going through a sequence of 2^n clock cycles.

***Proof**:* This lemma is the same as stated in observation 3.1 in the previous chapter. See the proof there.

Lemma 4.2: Consider a maximal-length n-stage internal or external LFSR (n>2). We choose one of the cells and swap its value with its adjacent cell if the current value of a third cell in the LFSR is 0 (or 1) and leave the cells unswapped if the third cell has a value of 1 (or 0). Fig. 4.1 shows this arrangement for external LFSR (the same is valid for an internal LFSR). In this arrangement, the output of the two cells will have its transition count reduced by $T_{saved} = 2^{(n-2)}$ transitions. Since the two cells originally produce $2 \times 2^{n-1}$, then the resulting percentage saving is $T_{saved\%} = 25\%$.

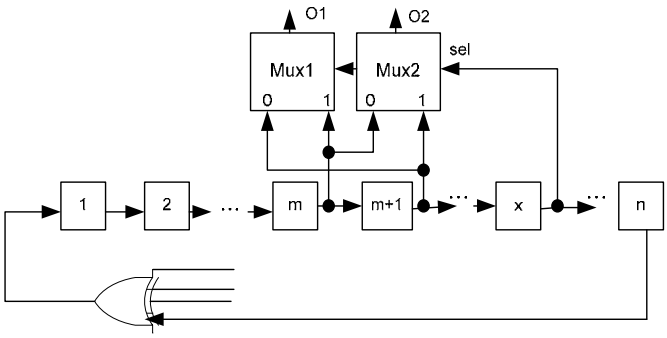

Fig. 4.1 Swapping arrangement for a LFSR.

Proof: This lemma is a repetition of observation 3.2 in the previous chapter, and the proof is identical. However, Table 3.2 is repeated in Table 4.1 in a different format in order to provide a starting point for the discussion that follows, where it is important to consider how evenly or unevenly the savings are distributed between the outputs O_1 and O_2. The general proof in chapter 3 showed that the total percentage of transition savings after swapping is 25%. In the case where cell x has no direct effect on cells m and m+1 (i.e. they are not connected by an XOR gate), each of the cells has the same share of savings, i.e. each cell (cells m and m+1) saves exactly 2^{n-3} transitions as shown in table 4.1.

The following lemmas introduce a series of special cases where the saving (of 25% per pair) is concentrated in one output (i.e. 50% saving in one output and none in the other). Using the output with a high saving as the input to the scan chain can then provide a low power approach to test-per-scan BIST as will be explained in section

4.2. Full proofs are presented for lemmas 4.3A and 4.3B. For the lemmas 4.4A to 10, the proofs follow in a similar manner and are omitted.

TABLE 4.1 EXHAUSTIVE ENUMERATION OF POSSIBLE STATES AND SUBSEQUENT STATES FOR BITS M, M+1, AND X (SEE FIG. 4.1).

LFSR outputs of m, m+1									Multiplexers outputs O_1, O_2						
States			Next states			Transitions			States		Next States		Transitions		
m	m+1	x	m	m+1	x	m	m+1	∑	O_1	O_2	O_1	O_2	O_1	O_2	∑
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			0	0	1	0	0	0			0	0	0	0	0
			1	0	0	1	0	1			0	1	0	1	1
			1	0	1	1	0	1			1	0	1	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
			0	0	1	0	0	0			0	0	0	0	0
			1	0	0	1	0	1			0	1	0	1	1
			1	0	1	1	0	1			1	0	1	0	1
0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1
			0	0	1	0	1	1			0	0	1	0	1
			1	0	0	1	1	2			0	1	1	1	2
			1	0	1	1	1	2			1	0	0	0	0
0	1	1	0	0	0	0	1	1	0	1	0	0	0	1	1
			0	0	1	0	1	1			0	0	0	1	1
			1	0	0	1	1	2			0	1	0	0	0
			1	0	1	1	1	2			1	0	1	1	2
1	0	0	0	1	0	1	1	2	0	1	1	0	1	1	2
			0	1	1	1	1	2			0	1	0	0	0
			1	1	0	0	1	1			1	1	1	0	1
			1	1	1	0	1	1			1	1	1	0	1
1	0	1	0	1	0	1	1	2	1	0	1	0	0	0	0
			0	1	1	1	1	2			0	1	1	1	2
			1	1	0	0	1	1			1	1	0	1	1
			1	1	1	0	1	1			1	1	0	1	1
1	1	0	0	1	0	1	0	1	1	1	1	0	0	1	1
			0	1	1	1	0	1			0	1	1	0	1
			1	1	0	0	0	0			1	1	0	0	0
			1	1	1	0	0	0			1	1	0	0	0
1	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1
			0	1	1	1	0	1			0	1	1	0	1
			1	1	0	0	0	0			1	1	0	0	0
			1	1	1	0	0	0			1	1	0	0	0
Total Transitions						**16**	**16**	**32**					**12**	**12**	**24**

Lemma 4.3A: For an external n-bit maximal length LFSR that implements the primitive polynomial[1] x^n+x+1, as shown in Fig. 4.2, if the first two cells (C_1 and C_2) have been chosen for swapping and cell n (C_n) is used as a selection line (as shown in Fig. 4.2), then O_2 (the output of MUX2) will produce a total transition savings of 2^{n-2} compared to the number of transitions produced by each LFSR cell, while O_1 has no

[1] It should be noted that this is not true of all values of n. Appendix A shows for which values of n x^n+x+1 is primitive polynomial.

saving (i.e. the savings in transitions is concentrated in one multiplexer output, which means that O_2 will save 50% of original transitions produced by each LFSR cell, while O_1 has no savings).

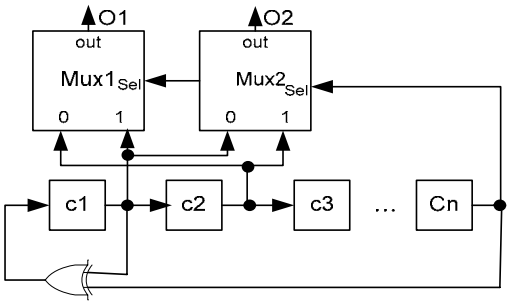

Fig. 4.2 External LFSR that implements prime polynomial xn+x+1 and the proposed swapping arrangement.

Proof: There are 8 possible combinations for the initial state of the cells C_1, C_2, and C_n. If we then consider all possible values of the following state, we have 2 possible combinations (not 8, because the value of C_2 in the next state is determined by the value of C_1 in the present state, also the value of C_1 in next state is determined by $C_1 \oplus C_n$ in the present state). Table 4.2 shows all possible states and subsequent states.

TABLE 4.2 POSSIBLE STATES AND SUBSEQUENT STATES FOR CELLS C1, C2, AND CN (SEE FIG. 4.2).

LFSR outputs of m, m+1									Multiplexers outputs O_1, O_2						
States			Next states			transition			states		Next States		transition		
C_1	C_2	C_n	C_1	C_2	C_n	C_1	C_2	∑	O_1	O_2	O_1	O_2	O_1	O_2	∑
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			0	0	1	0	0	0			0	0	0	0	0
0	0	1	1	0	0	1	0	1	0	0	0	1	0	1	1
			1	0	1	1	0	1			1	0	1	0	1
0	1	0	0	0	0	0	1	1	1	0	0	0	1	0	1
			0	0	1	0	1	1			0	0	1	0	1
0	1	1	1	0	0	1	1	2	0	1	0	1	0	0	0
			1	0	1	1	1	2			1	0	1	1	2
1	0	0	1	1	0	0	1	1	0	1	1	1	1	0	1
			1	1	1	0	1	1			1	1	1	0	1
1	0	1	0	1	0	1	1	2	1	0	1	0	0	0	0
			0	1	1	1	1	2			0	1	1	1	2
1	1	0	1	1	0	0	0	0	1	1	1	1	0	0	0
			1	1	1	0	0	0			1	1	0	0	0
1	1	1	0	1	0	1	0	1	1	1	1	0	0	1	1
			0	1	1	1	0	1			0	1	1	0	1
Total Transitions						**8**	**8**	**16**					**8**	**4**	**12**

It is important to note that the overall savings of 25% is not equally distributed between the outputs of the multiplexers as it was in lemma 4.2. This is because the value of C_1 in the present state will affect the value of C_2 and its own value in next state ($C_2^+ = C_1$ and $C_1^+ = C_1 \oplus C_n$). To see the effect of each cell in transition savings, table 4.2 shows that O_1 will save one transition when moving from state (0,0,1) to (1,0,0), from (0,1,1) to (1,0,0), from (1,0,1) to (0,1,0), or from (1,1,1) to (0,1,0). In the same time O_1 will increase one transition when moving from (0,1,0) to (0,0,0), from (0,1,0) to (0,0,1), from (1,0,0) to (1,1,0), or from (1,0,0) to (1,1,1). Since O_1 increases the transitions in four possible scenarios and save transitions in other four scenarios, then it has a neutral overall effect because all the scenarios have the same probabilities.

For O_2, it saves one transition when moving from (0,1,0) to (0,0,0), from (0,1,0) to (0,0,1), from (0,1,1) to (1,0,0), from (1,0,0) to (1,1,0), from (1,0,0) to (1,1,1), or from (1,0,1) to (0,1,0). In the same time it increases one transition when moving from state (0,0,1) to (1,0,0) or from (1,1,1) to (0,1,0). This gives O_2 an overall saving of one transition in four possible scenarios where the initial states has a probability of ⅛, and final states of probability ½, hence P_{save} is given by equation (4.1).

$$P_{save} = \frac{1}{8} \times \frac{1}{2} + \frac{1}{8} \times \frac{1}{2} + \frac{1}{8} \times \frac{1}{2} + \frac{1}{8} \times \frac{1}{2} = \frac{1}{4} \quad (4.1)$$

If the LFSR is allowed to move through a complete cycle of 2^n states, then lemma 4.1 shows that the number of transitions expected to occur in the cell under consideration is 2^{n-1}. Using the swapping approach, in ¼ of the cases, a saving of one transition will occur, giving a total saving of $\frac{1}{4} \times 2^n = 2^{n-2}$. Dividing one figure by the other, we see that the total number of transitions saved at O_2 is 50%. In this case, O_1 saves 0% while O_2 saves 50%, which gives an overall saving of 25% (i.e. we still have an overall transitions savings of 25%, but this time they are concentrated on O_2 and not distributed between the two outputs as in lemma 4.2).

Lemma 4.3B: For the internal n-bit maximal length LFSR that implements prime polynomial x^n+x+1 as shown in Fig.4.3, if the first and last cells (C_1 and Cn) have been chosen for swapping while cell 2 (C_2) has been chosen as a selection line (see

Fig. 4.3), then O_2 (the output of MUX2) will produce a total transition savings of 2^{n-2} while O_1 (the output of MUX1) has no savings.

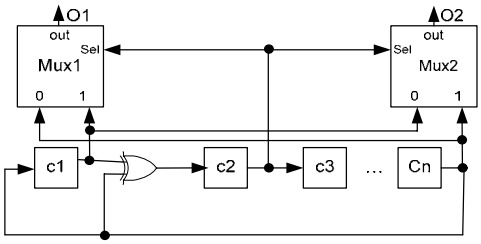

Fig. 4.3 Internal LFSR that implements prime polynomial x^n+x+1 and the proposed swapping arrangement.

Proof: It is important to notice that C_1 and C_n in the configuration shown in Fig. 4.3 are considered as adjacent cells since one of them feeds the other without intervening XOR gate while C_1 and C_2 are not adjacent cells because neither of them directly feeds the other. Table 4.3 shows all the current and subsequent states of C_1, C_n, and C_2. From table 4.3 it is straightforward procedure to prove this lemma following the same steps as for lemma 4.3A.

TABLE 4.3. POSSIBLE STATES AND SUBSEQUENT STATES FOR CELLS C_1, C_N, AND C_2 (SEE FIG. 4.3).

LFSR outputs of m, m+1									Multiplexers outputs O_1, O_2						
States			Next states			transition			states		Next States		transition		
C_1	C_n	C_2	C_1	C_n	C_2	C_1	C_n	$\sum$	O_1	O_2	O_1	O_2	O_1	O_2	$\sum$
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
			0	1	0	0	1	1			1	0	1	0	1
0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
			0	1	0	0	1	1			1	0	1	0	1
0	1	0	1	0	1	1	1	2	1	0	1	0	0	0	0
			1	1	1	1	0	1			1	1	0	1	1
0	1	1	1	0	1	1	1	2	0	1	1	0	1	1	2
			1	1	1	1	0	1			1	1	1	0	1
1	0	0	0	0	1	1	0	1	0	1	0	0	0	1	1
			0	1	1	1	1	2			0	1	0	0	0
1	0	1	0	0	1	1	0	1	1	0	0	0	1	0	1
			0	1	1	1	1	2			0	1	1	1	2
1	1	0	1	0	0	0	1	1	1	1	0	1	1	0	1
			1	1	0	0	0	0			1	1	0	0	0
1	1	1	1	0	0	0	1	1	1	1	0	1	1	0	1
			1	1	0	0	0	0			1	1	0	0	0
Total Transitions						**8**	**8**	**16**					**8**	**4**	**12**

Example 4.1: Table 4.4 shows the patterns generated by cells 'a', 'b', 'c' and'd' of external and internal 4-bit LFSRs that implements the prime polynomial $x^4 + x + 1$ (see Fig. 4.4). Table 4.4 shows O_1 and O_2 when cell 'a' is swapped with cell 'b'

according to the value of cell ‘d’ in the external LFSR. It also shows O_1 and O_2 when cell ‘a’ is swapped with cell‘d’ according to the value of cell ‘b’ in the internal LFSR. The saved transitions in this example are in exact conformity as with results of lemma 4.3A and lemma 4.3B where the transitions savings are concentrated in O_2, which saves 50% of the number of transitions originally produced by each LFSR cell.

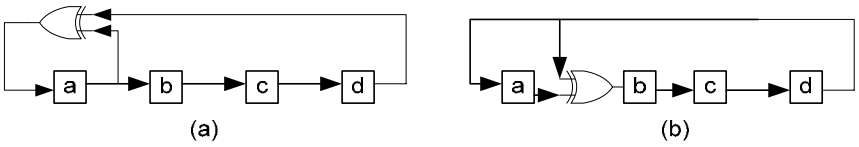

Fig. 4.4 (a) External and (b) Internal LFSRs that implement prime polynomial x^4+x+1

TABLE 4.4. DEMONSTRATION OF LEMMA 4.3A AND 4.3B FOR EXTERNAL AND INTERNAL LFSR

clk	External LFSR x^4+x+1 (Fig. 4.4(a))						Internal LFSR x^4+x+1 (Fig. 4.4(b))					
	Original Vectors				Swap ‘a’&’b’ when ‘d’ = 0		Original Vectors				Swap ‘a’&’d’ when ‘b’ = 0	
	a	b	c	d	O_1	O_2	a	b	c	d	O_1	O_2
1	1	1	1	1	1	1	1	1	1	1	1	1
2	0	1	1	1	0	1	1	0	1	1	1	1
3	1	0	1	1	1	0	1	0	0	1	1	1
4	0	1	0	1	0	1	1	0	0	0	0	1
5	1	0	1	0	0	1	0	1	0	0	0	0
6	1	1	0	1	1	1	0	0	1	0	0	0
7	0	1	1	0	1	0	0	0	0	1	1	0
8	0	0	1	1	0	0	1	1	0	0	1	0
9	1	0	0	1	1	0	0	1	1	0	0	0
10	0	1	0	0	1	0	0	0	1	1	1	0
11	0	0	1	0	0	0	1	1	0	1	1	1
12	0	0	0	1	0	0	1	0	1	0	0	1
13	1	0	0	0	0	1	0	1	0	1	0	1
14	1	1	0	0	1	1	1	1	1	0	1	0
15	1	1	1	0	1	1	0	1	1	1	0	1
16	1	1	1	1	1	1	1	1	1	1	1	1
Trans	**8**	**8**	**8**	**8**	**8**	**4**	**8**	**8**	**8**	**8**	**8**	**4**
Transition Savings%					**0%**	**50%**					**0%**	**50%**

Lemma 4.4A: Consider an external n-bit maximal length LFSR that implements prime polynomial $x^n+x^{n-1}+1$ as shown in Fig. 4.5. If the last two cells (C_{n-1} and C_n) have been chosen for swapping and the first cell (C_1) as selection line, then O_1 of MUX1 in Fig. 4.5 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

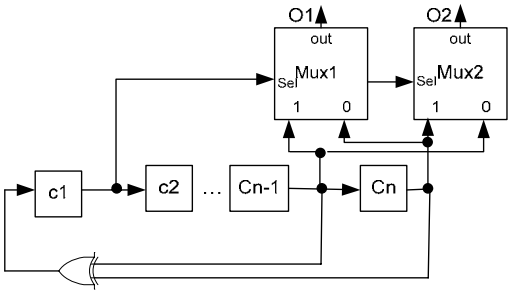

Fig. 4.5 External LFSR that implements prime polynomial $x^n+x^{n-1}+1$ and the proposed swapping arrangement

Lemma 4.4B: Consider an internal n-bit maximal length LFSR that implements prime polynomial $x^n+x^{n-1}+1$ as shown in Fig. 4.6. If the first and last cells (C_1 and C_n) have been chosen for swapping and cell C_{n-1} as selection line, then O_1 of MUX1 in Fig. 4.6 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

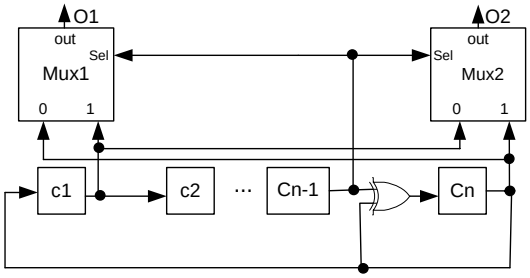

Fig. 4.6 Internal LFSR that implements prime polynomial $x^n+x^{n-1}+1$ and the proposed swapping arrangement

Lemma 4.5: Consider an external n-bit maximal length LFSR that implements prime polynomial x^n+x^2+1 as shown in Fig. 4.7. If the first two cells (C_1 and C_2) have been chosen for swapping and cell n (C_n) as selection line (see Fig. 4.6), then O_1 of MUX1 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

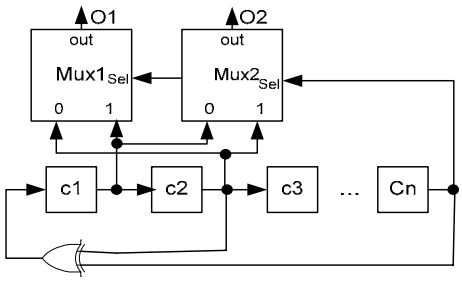

Fig. 4.7 External LFSR that implements prime polynomial x^n+x^2+1 and the proposed swapping arrangement

Lemma 4.6: Consider an internal n-bit maximal length LFSR that implements prime polynomial $x^n+x^{n-2}+1$ as shown in Fig. 4.8. If the last two cells (C_{n-1} and C_n) have been chosen for swapping and C_{n-2} as selection line (see Fig. 4.8), then O_1 of MUX1 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

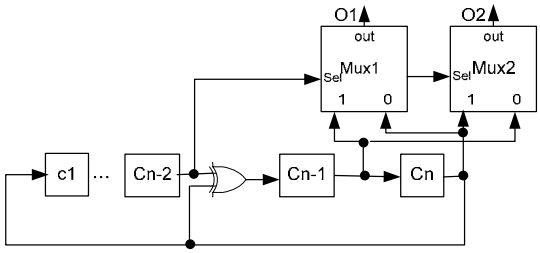

Fig. 4.8 Internal LFSR that implements prime polynomial $x^n+x^{n-2}+1$ and the proposed swapping arrangement.

Lemma 4.7: Consider the external n-bit maximal length LFSR that implements the prime polynomial $x^n + x^{ym} + \ldots + x^{y2} + x^{y1} + x + 1$ as shown in Fig. 4.9. Where y_1 to y_m is any run of ascending numbers that satisfy $y_1>1$ and $y_m<n$. If the first two cells (C_1 and C_2) have been chosen for swapping and the selection line is connected with signal S where S = "$C_n \oplus C_{ym} \oplus \ldots \oplus C_{y2} \oplus C_{y1}$" (see Fig. 4.9), then O_1 of MUX1 in Fig. 4.9 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

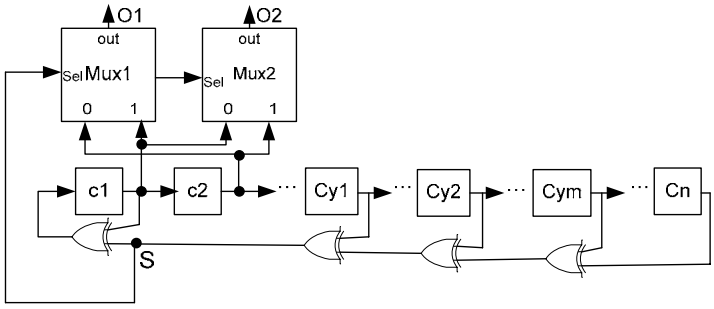

Fig. 4.9 External LFSR that implements prime polynomial $x^n + x^{ym} + \ldots + x^{y2} + x^{y1} + x + 1$ and the proposed swapping arrangement.

Lemma 4.8: For external n-bit maximal length LFSR that implements prime polynomial $x^n + x^{ym} + \ldots + x^{y2} + x^{y1} + x^2 + 1$. Where y_1 to y_m is any run of ascending numbers that satisfy $y_1>2$ and $y_m<n$. If the first two cells (C_1 and C_2) have been chosen for swapping and the selection line is connected with signal S where S = "$C_n \oplus C_{ym} \oplus \ldots \oplus C_{y2} \oplus C_{y1}$", then O_1 of MUX1 in this configuration will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

Lemma 4.9: For internal n-bit maximal length LFSR that implements prime polynomial $x^n + x^{n-1} + x^{ym} + \ldots + x^{y2} + x^{y1} + 1$. Where y_1 to y_m is any run of ascending numbers that satisfy $y_1>1$ and $y_m<n-1$. If the first and last cells (C_1 and C_n) have been chosen for swapping while cell C_{n-1} is chosen as selection line, then O_1 of MUX1 will produce a total transition savings of 2^{n-2} while O_2 of MUX2 has no savings.

Lemma 4.10: For internal n-bit maximal length LFSR that implements prime polynomial $x^n + x^{n-2} + x^{ym} + \ldots + x^{y2} + x^{y1} + 1$. Where y_1 to y_m is any run of ascending numbers that satisfy $y_1>1$ and $y_m<n-2$. If the last two cells (C_{n-1} and C_n) has been chosen for swapping while the selection line as cell C_{n-2}, then the output of MUX1 will produce a total transition savings of 2^{n-2} while MUX2 output has no savings

4.2 Architecture of the Proposed TPG for Scan-Based BIST

In conventional test-per-scan BIST, one of the LFSR cells is used to feed the single scan-chain input of the CUT. If an additional 2x1 multiplexer is added to the system such that its input and selection lines are connected with some cells of the LFSR that corresponds to one of the configurations mentioned in lemmas 4.3 to 4.10 in order to concentrate the transition savings at the output of this multiplexer, then the output of the multiplexer will save 50% of the number of transitions produced by any LFSR cell that may feed the scan-chain in a scan-based system. Hence, if this output is used to feed the scan-chain input, it will reduce the number of transitions at the inputs of the scan-chain by 50%, thus, reducing the switching activity in the CUT nodes. This will reduce the average power in the CUT, and also it will reduce the peak power that may result while scanning in a test vector. Fig. 4.10 shows the general shape of the BS-LFSR used in scan-based BIST.

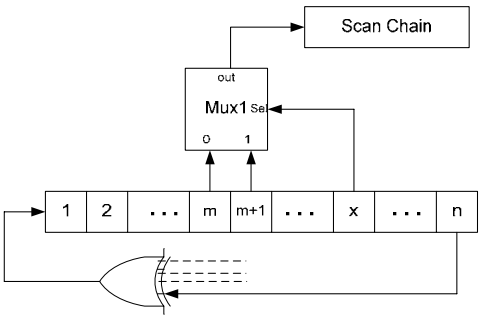

Fig. 4.10 BS-LFSR for test-per-scan BIST.

Table 4.5 shows which sizes of LFSR [7, 152, 153] (in range between 3 to 60 stages) can be configured to satisfy one of the lemmas 4.3 to 4.10 to concentrate the transition savings in one multiplexer output. The designer of the systems normally selects the type of the LFSR to be used in the system (internal or external) and its size and characteristic polynomial, once these parameters are known, then the special configuration to satisfy one of lemmas 4.3 to 4.10 is determined.

TABLE 4.5. LFSRS THAT SATISFY ONE OR MORE OF LEMMAS 3 TO 10

Lemma	LFSR size (in range 3 to 60) that can be configured to satisfy the lemma
Lemma 4.3A	3, 4, 6, 7, 15, 22, 60
Lemma 4.3B	3, 4, 6, 7, 15, 22, 60
Lemma 4.4A	3, 4, 6, 7, 15, 22, 60
Lemma 4.4B	3, 4, 6, 7,15, 22, 60
Lemma 4.5	5, 11, 29, 35
Lemma 4.6	5, 11, 29, 35
Lemma 4.7	3, 4, 6, 7, 9, 10, 11, 12, 13, 14, 15,16, 17, 18, 19, 20, 21, 23,24, 25, 26,27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38,39,41, 42, 43, 44, 45, 46, 47, 48, 50, 51, 52, 53, 54, 55, 56, 58, 59
Lemma 4.8	5, 8, 11,16, 17, 18, 19,21, 22,24, 25, 26, 27, 29, 31, 32, 35, 40, 41, 42, 44, 50, 53, 55, 56, 57, 59, 60
Lemma 4.9	3,4, 6, 7, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 23,24, 25, 26, 27, 28, 29, 30, 31, 33, 34, 35, 36, 37, 38, 39, 41, 42, 43, 44, 45, 46, 47, 48, 50, 51, 52, 53, 54, 55, 56, 58, 59
Lemma 4.10	5, 8, 11, 16, 17, 18, 19, 21, 22,24, 25, 26, 27, 29, 31, 32, 35, 40, 41, 42, 44, 50, 53, 55, 56, 57, 59,60

It is important to note that the proposed BS-LFSR generates the same number of ones and zeros at the output of multiplexers after swapping of two adjacent cells, hence, the probabilities of having a 0 or 1 at a specified cell of the scan chain before applying the test vectors are equal. Hence the proposed design retains an important feature of any random TPG, that of an equi-probable output state. Also, the output of the multiplexer depends on three different cells of the LFSR, each of which contains a pseudorandom value. Hence, the expected value at the output can also be considered to be a pseudorandom value.

4.3 Experimental Results

A group of experiments was performed on the full scan ISCAS'89 benchmark circuits to evaluate the efficiency of the proposed BS-LFSR on average power consumptions and their effect on fault coverage.

In the first set of experiments, the BS-LFSR is evaluated regarding the length of the test sequence needed to achieve a specified fault-coverage. Table 4.6 shows the results for a set of 20 benchmark circuits. The columns labelled n, m, and PI refer to the sizes of the LFSR, the number of flip-flops in the scan-chain, and the number of primary inputs of the CUT, respectively. The column labelled RF indicates the percentage of redundant faults in the CUT and FC indicates the target fault coverage where redundant faults are included, while the column labelled EFC is used to indicate the effective fault coverage where the redundant faults are not included (e.g. in S444 the FC of 96.5% includes the redundant faults which represents 2.95% of the total number of faults, hence 96.5% is equivalent to 99.4% of detectable faults in S444). The last three columns show the test length needed by a deterministic test (i.e. the optimal test vector set is stored in a ROM), a conventional LFSR, and the proposed BS-LFSR for scan-based BIST. The results in table 4.6 show that the BS-LFSR can still achieve better fault coverage than a conventional LFSR for many circuits (better fault coverage in 42% of the tested circuits).

The second set of experiments is used to evaluate the BS-LFSR in reducing average power while scanning a test vector in scan-based circuits. For each benchmark circuit, the same numbers of conventional LFSR and BS-LFSR patterns are applied to the full scan configuration. Table 4.7 shows the obtained results for the same 20 ISCAS'89 benchmark circuits as table 4.6. The column labelled TL refers to the number of test vectors applied to the CUT. The next two columns show the fault coverage (FC), and average weighted switching activity per clock cycle (WSA_{avg}) obtained by using the conventional LFSR. The next two columns show FC and WSA_{avg} obtained by using the BS-LFSR. Finally, the last column shows the savings in average power by using the BS-LFSR. Note that the number of inputs and flip-flops in each of the benchmark circuits and the size of the used LFSR can be found in table 4.6.

In order to provide a comparison with a technique published previously by another author [42], table 4.8 compares the results obtained by the proposed technique with those obtained in [42] where a 2-input AND gate is used since a 3-input AND gate will degrade the FC although it achieves better results regarding WSA reduction (i.e. k= 2, refer to section 2.1.1 for more details). Table 4.8 compares the test length (TL), fault coverage (FC), and average power reduction (WSA_{avg}). It is clear that the proposed method is better for most of the circuits not only in average power

reduction, but also in the test length needed to obtain good fault coverage. The last row in table 4.8 shows that the overall average of TL, and WSA_{avg} savings obtained by the proposed method is better than those obtained in [42] .

TABLE 4.6. TEST LENGTH NEEDED TO GET TARGET FAULT COVERAGE FOR LFSR AND BS-LFSR.

Circuit	n	m	PI	RF%	FC%	EFC%	Test Length		
							Det.	LFSR	BS-LFSR
S208	22	8	11	0	100	100	29	2820	2850
S298	19	14	3	0	100	100	31	680	630
S344	26	15	9	0	100	100	21	160	160
S349	21	15	9	0.57	99.2	99.8	20	930	1070
S420	32	16	19	0	98.0	98.0	51	2450	2510
S444	24	21	3	2.95	96.5	99.4	31	4560	4370
S526	24	21	3	0.18	98.5	98.7	59	5970	6630
S641	32	19	35	0	98.0	98.0	53	5120	4910
S838	32	32	35	0	86.5	86.5	90	8160	8460
S953	30	29	16	0	98.5	98.5	87	6760	7140
S1196	30	18	14	0	97.0	97.0	131	3750	3680
S1238	30	18	14	5.09	91.3	96.2	141	3890	3560
S1423	35	74	17	0.92	99.0	99.9	62	4420	4950
S5378	40	179	35	0.88	98.0	98.9	244	30110	33700
S9234	40	228	19	6.52	90.0	96.3	367	397800	401930
S13207	60	669	31	1.54	95.0	96.5	455	49660	47400
S15850	60	597	14	3.32	91.5	94.6	403	31940	33590
S35932	64	1728	35	10.19	89.8	100	63	18700	16640
S38417	64	1636	28	0.53	96.5	97.0	849	118580	125520
S38584	64	1452	12	4.15	94.0	98.0	632	43530	39660

TABLE 4.7. EXPERIMENTAL RESULTS OF AVERAGE POWER REDUCTION OBTAINED BY USING THE PROPOSED TECHNIQUES.

Circuit	TL	LFSR		BS-LFSR		$\%WSA_{avg}$
		FC%	WSA_{avg}	FC%	WSA_{avg}	Savings
S208	3000	100	18.47	100	10.74	42
S298	700	100	37.05	100	25.62	31
S344	200	100	41.61	100	26.83	36
S349	1000	99.31	54.84	99.18	39.09	29
S420	3000	98.64	25.33	98.62	14.11	44
S444	3000	96.39	43.81	96.21	31.43	28
S526	8000	98.81	62.36	99.09	40.32	35
S641	3000	97.84	48.27	97.51	27.91	42
S838	20000	96.15	45.62	96.08	21.72	52
S953	6000	97.67	36.52	97.45	26.60	27
S1196	2000	95.33	26.92	95.48	14.37	47
S1238	3000	91.11	41.58	90.93	20.17	51
S1423	2000	97.77	180.06	97.77	105.75	41
S5378	40000	98.42	518.32	98.44	421.82	19
S9234	100000	87.27	1347.66	87.26	690.38	49
S13207	100000	96.45	2704.80	96.46	1093.64	60
S15850	100000	94.75	2365.79	94.74	1445.81	39
S35932	200	87.88	3621.43	87.89	1645.58	55
S38417	100000	95.73	5131.52	95.67	2807.46	45
S38584	100000	94.46	6007.38	94.49	3496.20	42

TABLE 4.8 COMPARISON WITH RESULTS OBTAINED IN [42]

Circuit	Results in [42]			Results of proposed method		
	TL	FC	$\%WSA_{av}$	TL	FC	$\%WSA_{av}$
S208	4096	100	19	3000	100	42
S298	1024	100	22	700	100	31
S344	256	100	33	200	100	36
S349	1024	99.27	32	1000	99.18	29
S420	4096	98.24	25	3000	98.62	44
S444	4096	96.39	25	3000	96.21	28
S526	16384	98.82	32	8000	99.09	35
S641	4096	97.72	26	3000	97.51	42
S838	4096	92.73	30	20000	96.08	52
S953	8192	99.32	17	6000	97.45	27
S1196	4096	95.57	11	2000	95.48	47
S1238	4096	89.65	11	3000	90.93	51
S1423	2048	97.81	35	2000	97.77	41
S5378	65536	98.22	26	40000	98.44	19
S9234	524288	91.47	39	100000	87.26	49
S13207	132072	97.33	28	100000	96.46	60
S15850	132072	94.87	36	100000	94.74	39
S35932	128	87.84	33	200	87.89	55
S38417	132072	94.42	34	100000	95.67	45
S38584	132072	95.71	38	100000	94.49	42
AVG	**58792**	**96.26**	**28**	**29755**	**96.16**	**41**

4.4 Conclusion

This chapter started from the two observations described in chapter 3 in order to extend the BS-LFSR proposed there to be applicable for test-per-scan BIST. Furthermore, some new lemmas have been added to show how the savings in the number of transitions while scanning-in a test vector in the scan-chain can be doubled to reach 50%. This significantly reduces the switching activity in the CUT as shown in the experimental results.

The effect of the proposed design on fault coverage, hardware area overhead, and test application time have been shown to be negligible. Comparison between the proposed technique and another one proposed previously by another author [42] shows that the BS-LFSR for test-per-scan circuits can achieve better results in power consumption (90% of the tested benchmark circuit) and test length (Also 90% of them).

CHAPTER 5

RLFSR as a Multi-Degree Smoother for Test-per-Scan BIST

This chapter presents a smoothing technique for the output sequence of LFSRs to reduce power consumption in test-per-scan BIST applications. This chapter generalises the theory and application of the approach of chapter 3 and chapter 4 to use a rotational technique applied to a group of output cells in the LFSR. The swapping techniques presented in chapter 3 and 4 are special cases of rotation where the group contains only 2 bits.

The proposed smoother is implemented by adding one multiplexer between the LFSR and the input to a single scan-chain. The size of the multiplexer is determined by the desired smoothing degree. When the smoothed sequence of the LFSR is used to feed the test patterns in test-per-scan BIST, it reduces the number of transitions that occur at the scan-chain input during scan shift operations by 25% to 50% depending on the smoothing degree, and hence reduces switching activity in the CUT during test application. The proposed technique can be extended to multiple scan-chain BIST, also to test-per-clock applications. Various properties of the proposed technique and the methodology of the design are presented in this chapter. Experimental results for the ISCAS'89 benchmark circuits show that the proposed design can reduce the switching activity while scanning in a test vector by up to 55% with a negligible effect on the fault coverage and test application time. The presented technique is based on some new theoretical concepts and the proposed design can achieve slightly better results when compared with another existing techniques. Also, the regularity of

the proposed design makes it easy to combine it with other techniques for low power test in order to achieve extra power reduction. This chapter is organised as follows: section 5.1 presents the theory and the key ideas about the proposed design; it discusses the main lemmas and the proofs of these lemmas, as well as some illustrative examples. In section 5.2, the methodology to extend the design for multiple scan-chains is presented. The main properties and features of the proposed design are presented in section 5.3. Section 5.4 shows the experimental results obtained using the proposed design; also it compares the obtained results with results of conventional LFSR and with other existing technique. Finally, section 5.5 discusses the main conclusions of this chapter.

5.1 Key Ideas to Reduce the Transitions at the Output Sequences of LFSR

The proposed design for smoothing the output sequence of an LFSR before applying it to the scan-chain input is based on some lemmas about transition counts of LFSRs. Before going into these lemmas, we start with some important definitions.

Definition 5.1: A group of cells (2 cells or more), in an internal or external LFSR, are considered to be adjacent cells if the output of the first cell feeds the input of the second cell directly without an intervening XOR gate, and the output of the second cell feeds the input of the third one directly without an intervening XOR gate and so on. Otherwise, they are not considered to constitute a group of adjacent cells.

Definition 5.2: The smoothing degree of the output sequence of an LFSR is completely dependent on the size of the extra multiplexer. The smoothing degree equals the number of selection lines in the multiplexer. In a conventional system, where no multiplexer is present, one LFSR cell always feeds the scan-chain input and this is considered as smoothing degree 0 (i.e. no smoothing is present). If a 2 x 1 multiplexer is used, where one selection line is needed, then the smoothing degree is 1. When a 2^k x 1 multiplexer is used, where k selection lines are required, then the smoothing degree is k.

This section will develop a general theory of the savings caused by various degrees of smoothing. Zeroth order and first order smoothing correspond to situations already considered in previous chapters, and are briefly recapitulated in lemmas 5.1 and 5.2. This is then followed by the development of a general result for k^{th} order smoothing.

Lemma 5.1 *(Zero order smoothing):* For a conventional scan-based scheme, where one cell of a maximal length n-stage LFSR feeds the scan-chain input as shown in Fig. 5.1 (i.e. smoothing degree 0), then the total number of transitions (moving from 0 to 1 or 1 to 0) at the scan-chain input after running 2^n clock cycles is 2^{n-1} transitions. *This result has already been proved in section 3.1.*

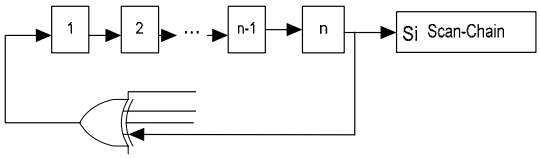

Fig. 5.1. Conventional scan-based BIST where one cell directly feeds the scan-chain

Lemma 5.2 *(smoothing of degree 1):* If a 2x1 multiplexer is used to feed the scan-chain input as shown in Fig. 5.2, where the input lines of the multiplexer are connected with two adjacent cells of the n-stage LFSR, and the selection line is connected with a third cell, then the number of transitions at the output of the multiplexer (scan-chain input) will be 25% less than the number of transitions that will be produced by any LFSR cell.
This result has already been proved in section 3.1 and section 4.1.

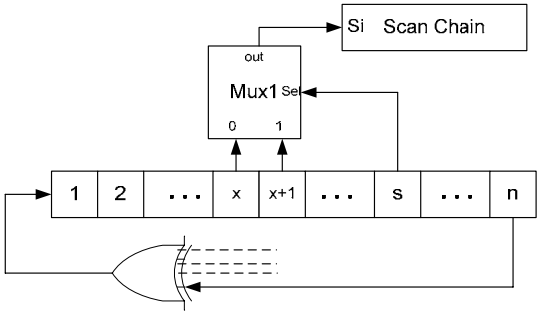

Fig. 5.2. Smoothing degree k =1, where 2 cells feed the scan-chain through a multiplexer.

Lemma 5.3 *(smoothing of degree 2):* If a 4x1 multiplexer is used to feed the scan-chain input as shown in Fig. 5.3, where the input lines of the multiplexer are connected with four adjacent cells of the n-stage LFSR, and the selection lines are connected with another two adjacent cells, then the number of transitions at the output of the multiplexer (the scan-chain input) will be 37.5% less than the number of transitions that will be produced by any LFSR cell.

Note 5.1: The four adjacent cells are connected with the input lines of the multiplexer in a special arrangement that guarantees the reduction of the transitions as shown in Fig. 5.3. The reason for this arrangement is as follows. If the two adjacent selection lines are currently in one of the four possible states (say 00) that selects one of the four cells to feed into the scan-chain (say cell x in Fig. 5.3), then in the next clock cycle these selection lines will take one of two possible values (either 00 or 10. It is impossible to go from 00 to 01 or 11 since the two selection lines are adjacent cells, hence in the next clock cycle the second cell takes the value of the first cell in the present clock cycle). One of these two values (10 in this case, since 00 already selects cell x) should select the subsequent cell to cell x (cell x+1 in Fig. 5.3) in order to have the possibility to save a transition.

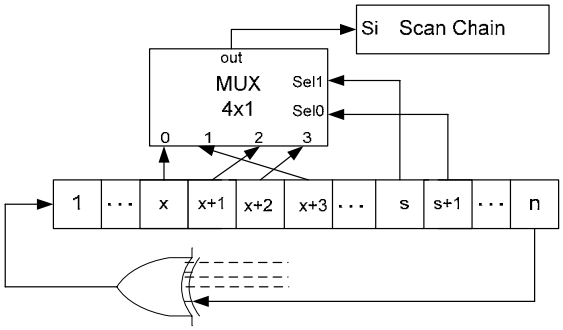

Fig. 5.3. Smoothing degree k =2, where 4 cells feed the scan-chain through a multiplexer.

Proof: A saving in transitions at the scan-chain input in Fig. 5.3 will happen only in three cases. These cases are:

1- When the selection lines in the present clock cycle select cell x to feed the scan chain and selects cell x+1 in the next clock cycle, while at the same time cell x in

the present state has a different value from the next state (i.e. a transition occurs in cell x).

2- When the selection lines select cell x+1in the current cycle and cell x+2 in the next cycle, while at the same time cell x+1 in the present state has a different value from the next state (i.e. a transition occurs in cell x+1).
3- When the selection lines select cell x+2 in the current cycle and cell x+3 in the next cycle, while at the same time cell x+2 in the present state has a different value from the next state (i.e. a transition occurs in cell x+2).

All the three cases above have an equal share in reducing the number of transitions. In the first case the probability that the selection lines select cell x at present state is ¼, and the probability that it will select cell x+1 in the next state is ½ (note 5.1). Also, the probability of a transition at cell x on moving from the present clock cycle to the next clock cycle is ½ (lemma 5.1), then each case has a probability of a transition saving given by $P_{Saving/case(deg.2)}$ in equation (5.1).

$$P_{Saving/case(deg.2)} = ¼ \times ½ \times ½ = 1/16 \quad (5.1)$$

And so the total probability of transition savings is given by $P_{Saving(deg.2)}$ in equation (5.2).

$$P_{Saving(deg.2)} = 3 \times P_{Saving/case(deg.2)} = 3/16 \quad (5.2)$$

Hence, if the LFSR runs for 2^n cycles, going through all of its possible states, the number of saved transitions ($T_{Saved(deg.2)}$) is given by equation (5.3).

$$T_{Saved(deg.2)} = 3/16 \times 2^n = 3 \times 2^{n-4} \quad (5.3)$$

If we divide this figure by the total number of transitions produced by each cell of LFSR (which is 2^{n-1} as shown by lemma 5.1), then $\%T_{Saved(deg.2)}$ will be given by equation (5.4).

$$\%T_{Saved(deg.2)} = (3 \times 2^{n-4}) / 2^{n-1} = 3/8 = 37.5\% \quad (5.4)$$

Lemma 5.4 *(general rule for smoothing of degree k):* If a 2^k x 1 multiplexer, with k selection lines, is used to feed the scan-chain input, where the input lines of the multiplexer are connected with 2^k adjacent cells of the n-stage LFSR, while the selection lines are connected with another k adjacent cells, then the percentage of transitions saved at the output of the multiplexer (scan-chain input) is given by equation (5.5).

$$\%T_{Saved(deg.k)} = \sum_{n=2}^{k+1} 2^{-n} = 2^{-2}+2^{-3}+\ldots+2^{-(k+1)} = (2^k - 1)/2^{k+1} \qquad (5.5)$$

Proof: Firstly, the 2^k adjacent cells are connected with the input lines of the multiplexer in a special arrangement that guarantees a reduction of the transitions (this arrangement is described later, and shown in table 5.1). Since there are k adjacent selection lines, if they are currently in one of the 2^k possible states that selects one of the cells to feed the scan-chain (say cell x), then in the next clock cycle the selection lines will be only in one of two possible values (because all of the selection lines are adjacent and each cell takes its current value from its predecessor cell in the previous clock cycle), one of these two values of the selection lines should select the subsequent cell for cell x (i.e. cell x+1).

Saving in transitions at the scan-chain input will happen in 2^k-1 cases. These cases are:

Case 1: When the selection lines in the present clock cycle select cell x to feed the scan chain and selects cell x+1 in the next clock cycle, while at the same time cell x in the present state has a different value from the next state (a transition occurs in cell x).

Case 2: When the selection lines select cell x+1in the current cycle and cell x+2 in the following cycle, while at the same time cell x+1 in the present state has a different value from the next state (a transition occurs in cell x+1).

… and similarly until…

Case $2^k - 1$: When the selection lines select cell $x+2^k - 2$ in the current cycle and cell $x+2^k - 1$ in the next cycle, while at the same time cell $x+2^k - 2$ in the present state has a different value from the next state (a transition occurs in $x+2^k - 2$).

All the $(2^k - 1)$ cases above have an equal share in reducing the number of transitions. In the first case the probability that the selection lines select cell x at the present state is $1/2^k$, and the probability that it will select cell x+1 in the next state is ½. Also, the probability of a transition at cell x on moving from the present clock cycle to the next clock cycle is ½ (lemma 5.1), then each case has a saving given by $P_{Saving/case(deg.k)}$ in equation (5.6).

$$P_{Saving/case(deg.k)} = 1/2^k \times ½ \times ½ = 1/2^{k+2} \qquad (5.6)$$

And so the total probability of transition savings is given by $P_{Saving(deg.k)}$ in equation (5.7).

$$P_{Saving(deg.k)} = (2^k - 1) \times P_{Saving/case(deg.k)} = (2^k - 1)/2^{k+2} \qquad (5.7)$$

Hence, if the LFSR runs for 2^n cycles to go through all of its possible states, the number of saved transitions ($T_{Saved(deg.k)}$) is given by equation (5.8).

$$T_{Saved(deg.k)} = P_{Saving(deg.k)} \times 2^n = ((2^k - 1)/2^{k+2}) \times 2^n \qquad (5.8)$$

If we divide this figure by the total number of transitions produced by each cell of LFSR (which is 2^{n-1} as stated in lemma 5.1), then $\%T_{Saved(deg.k)}$ will be as given in equation (5.5).

$$\%T_{Saved(deg.k)} = T_{Saved(deg.k)} / 2^{n-1} = (2^k - 1)/2^{k+1} \qquad (5.5)$$

$$= \sum_{n=2}^{k+1} 2^{-n} = 2^{-2}+2^{-3}+\ldots+2^{-(k+1)} \qquad (5.5)$$

Equation (5.5) shows that the percentage transition reduction is exponentially decreasing with k, and the series converges to ½ as k tends to infinity. Hence, in most cases it is sufficient to use a smoothing degree of up to k = 4. Table 5.1 shows for different smoothing degrees one of the possible connections between the input lines of the multiplexer and the adjacent cells that connect with them to guarantee a reduction in the number of transitions in the ratio given by equation (5.5). The 2^k adjacent cells of the LFSR are indicated by cells (C_x to $C_{x+(2^k-1)}$) while the multiplexer inputs are indicated by (In_0 to $In_{(2^k-1)}$). This table can be extended to any smoothing degree; the only consideration that should be observed while doing this is that the connection

relationship is one to one, and the cells are selected to be connected with the multiplexer in a way which is similar to note 5.1.

TABLE 5.1. CONNECTIONS BETWEEN LFSR CELLS AND INPUT LINES OF MULTIPLEXER FOR DIFFERENT VALUES OF K

k	Connections
1	$C_x \rightarrow In_0$, $C_{x+1} \rightarrow In_1$
2	$C_x \rightarrow In_0$, $C_{x+1} \rightarrow In_2$, $C_{x+2} \rightarrow In_3$, $C_{x+3} \rightarrow In_1$
3	$C_x \rightarrow In_0$, $C_{x+1} \rightarrow In_4$, $C_{x+2} \rightarrow In_6$, $C_{x+3} \rightarrow In_7$, $C_{x+4} \rightarrow In_3$, $C_{x+5} \rightarrow In_5$, $C_{x+6} \rightarrow In_2$, $C_{x+7} \rightarrow In_1$
4	$C_x \rightarrow In_0$, $C_{x+1} \rightarrow In_8$, $C_{x+2} \rightarrow In_4$, $C_{x+3} \rightarrow In_2$, $C_{x+4} \rightarrow In_9$, $C_{x+5} \rightarrow In_{12}$, $C_{x+6} \rightarrow In_6$, $C_{x+7} \rightarrow In_{11}$, $C_{x+8} \rightarrow In_5$, $C_{x+9} \rightarrow In_{10}$, $C_{x+10} \rightarrow In_{13}$, $C_{x+11} \rightarrow In_{14}$, $C_{x+12} \rightarrow In_{15}$, $C_{x+13} \rightarrow In_7$, $C_{x+14} \rightarrow In_3$, $C_{x+15} \rightarrow In_1$

Example 5.1: Fig. 5.4 shows an internal 7-bit LFSR that implements prime polynomial $x^7 + x + 1$ with an additional 4x1 multiplexer (smoothing degree k = 2). The selection lines of the multiplexer are connected with adjacent cells ‘b’ and ‘c’ while the input lines are connected with adjacent cells ‘d’, ‘e’, ‘f’, and ‘g’. If the LFSR runs for 2^7 = 128 clock cycles starting from seed “1111111” while assuming that cell ‘g’ is feeding the scan-chain input as in any conventional BIST then ‘g’ will take the sequence
“11111100000010000011000010100011110010001011001110101001111101000011
10001001001101101011011110110001101001011101110011001010101 1”.

This gives a total number of transitions = 64 (exactly as stated in lemma 5.1).

Now if the output of the multiplexer is used to feed the scan chain, then ‘out’ will take the following sequence
“11100000010000001000000100000011110010001110011011110011111000000100
11101111111000011100111100001011000111111111111100010010111”.

This gives a total number of transitions = 40 transitions, which means that 24 transitions have been saved, which is 37.5% of the number of transitions originally produced by cell ‘g’. This is in exact conformity with lemma 5.4.

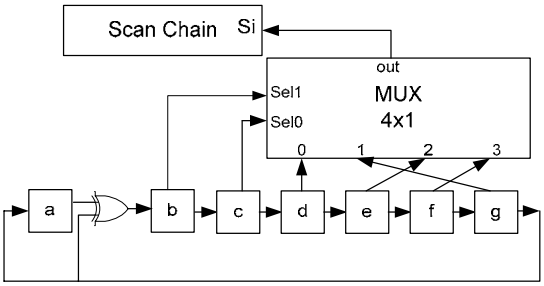

Fig. 5.4 7-bit internal LFSR with 4x1 multiplexer to get a smoothing degree k =2.

5.2 Extending the Design for Multiple Scan-Chain BIST

The proposed design of the previous section can be extended to be used with multiple scan-chain BIST. In this case the number of extra multiplexers needed in the system is equivalent to the number of scan-chains. The presence of the multiplexers between the LFSR cells and the scan-chains breaks the structural dependency of the patterns generated by a conventional LFSR, and hence act as a phase shifter (see section 1.8 for more details about phase shifters). For example, if the LFSR in Fig. 5.4 is extended in order to be used to feed four scan-chains as shown in Fig. 5.5(a), then it saves the number of transitions in each scan-chain as proved in equation (5.5). It is important to note that in Fig. 5.5(a) the same LFSR cells are connected with the selection lines of all multiplexers, while another four cells of the LFSR are connected with the data lines of all multiplexers but with different order in each multiplexer. Fig. 5.5(b) shows that each multiplexer is connected to four cells (no cell has double connection in the same MUX, see columns labelled by Mux1 to Mux4 in Fig. 5.5(b)), and in the same time for any value of the selection lines, the outputs of the four multiplexers will be set to the output of the four different cells, but in a different permutation (i.e. each row in Fig. 5.5(b) has four different cells, and each column has four different cells). Furthermore, each row is the same as its adjacent row with one cell rotation, and each column is the same as its adjacent column with one cell rotation. The top and bottom rows are considered to be adjacent and the left and right columns are considered to be adjacent. The order of the selection lines in this case are 00, 10, 11, and 01 respectively as shown in table 5.1. The design shown in Fig. 5.5(a) is called rotational LFSR (RLFSR).

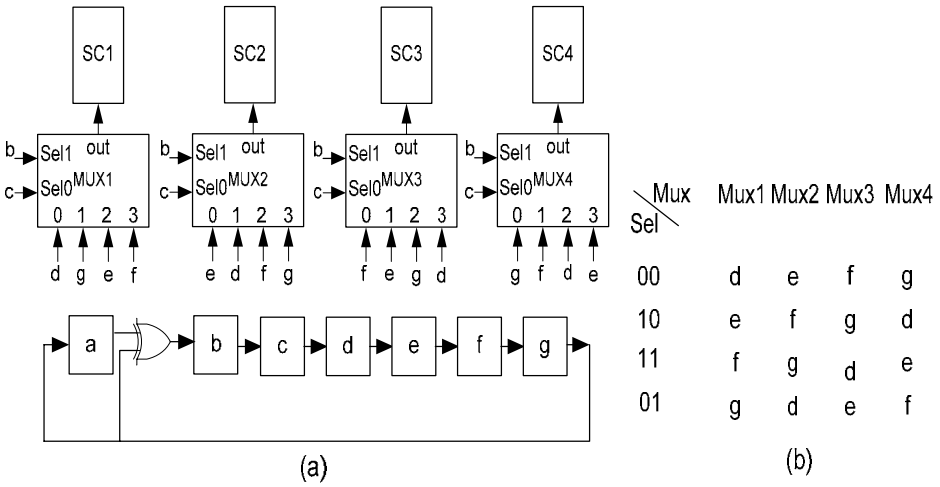

Fig. 5.5 (a) Extending the smoothing technique for multiple scan-chains, (b) the rotational property of the proposed design

Using the proposed design, where the multiplexers are used to feed the scan-chains, decreases the correlation between adjacent scan chains (achieving an effect similar to phase shifting) since the next value to be loaded in the scan-chains depends not only on the value of the cells connected with the multiplexers' data lines, but also on the value of the selection lines.

In order to test how effectively the structural dependencies are removed by the proposed design, a simple correlation test was performed [7] on the sequences *a* and *b* entering the different scan chains. The correlation is defined by mapping the original sequences a_i and b_i whose elements are 1 or 0, into new sequences a'_i and b'_i whose elements are +1 or -1, respectively. The correlation C is then defined as in equation (5.9)

$$C(\tau) = \frac{1}{p}\sum_{i=1}^{p} a'_i \, b'_{i-\tau} \qquad (5.9)$$

where $p=2^{n-1}$ and n is the length of the LFSR and τ is a delay measured in clock cycles.

In the simple correlation test, a 36-bit LFSR was used; where 32-bits are used to feed 32 scan-chains using different smoothing degrees (smoothing degrees of 0, 1, 2, 3, and 4) and 50,000 test patterns were generated. Table 5.2 shows the overall results of this test. The correlation test is performed for C(1) between the contents of scan chain

1 and scan chain 2, C(2) between chain 1 and 3, and so on. The overall average is tabulated. From these results it is clear that the multiplexers significantly reduce the correlation compared with a conventional LFSR (smoothing degree 0), and the RLFSR can be used in multiple scan-chains with a negligible degradation in the fault coverage. Nevertheless, the multiplexers do not behave as an ideal phase-shifter since an ideal phase shifter should have a correlation value of $1/2^{n-1}$ if the correlation is applied to the entire m-sequence.

TABLE 5.2: CORRELATION TEST FOR THE RLFSR WHEN USED IN MULTIPLE SCAN-CHAINS BIST

Smoothing Degree	0	1	2	3	4
Multiplexers Size	No Mux	2x1	4x1	8x1	16x1
Autocorrelation	1	0.034	0.021	0.019	0.036

Previous papers by other authors using smoothing approaches (e.g. [42, 50]) have not addressed the correlation properties of the resulting sequences, so it is not possible to give a comparison with their results.

5.3 Key Properties of the Proposed Design

The properties of the proposed design make the proposed smoothing technique and the proposed RLFSR a good choice when a low power pseudorandom TPG is needed. The most important properties are:

- The proposed design generates the same number of ones and zeros at the output of the multiplexers, hence, the probabilities of having a 0 or 1 at a specific cell of the scan chain before applying the test vectors are equal. Hence the proposed design retains an important feature of any random TPG. Also, the output of the multiplexer depends on many different cells of the LFSR, each of which contains a pseudorandom value. Hence, the expected value at the output can also be considered to be a pseudorandom value.
- If the RLFSR is used to generate test patterns for either test-per-clock BIST or for the primary inputs of a scan-based sequential circuit (assuming that they are directly accessible), then the same exhaustive set of test patterns produced by a conventional LFSR will be produced by the proposed RLFSR with no repetition but with a different order that guarantees reducing the number of transition in each input of the CUT by the value given in equation (5.5). Fig. 5.6 shows an example

where an n-bit RLFSR with smoothing degree k is used to generate test vectors for an n-input CUT in a test-per-clock BIST. In this case *k* LFSR cells (cells *n-k* to *n* in Fig. 5.6) are needed to act as selection lines and connected with the selection lines of all 2^kx1 multiplexers. The cells connected with the selection lines are excluded from rotation (i.e. they are not connected with the data lines of the multiplexers), and all other LFSR cells are divided into groups 2^k adjacent cells, where each group is connected with a group of 2^k multiplexers. Since all possible vectors will be produced at the output of an LFSR exactly once during a complete run of $2^n - 1$ clock cycles, the RLFSR will map each produced vector either to itself or to another vector depends on the value of selection lines. Furthermore, if an original vector (say v1) at LFSR outputs is mapped to another vector (say v2) at the inputs of the CUT, then there is another vector (say v3) at the LFSR outputs that will be mapped to the original vector (v1) at the inputs of the CUT. The mapping relation is one to one (because the selection lines are excluded from swapping and directly connected with the CUT inputs), which guarantees that the exhaustive set can still be applied to the CUT inputs using the RLFSR.

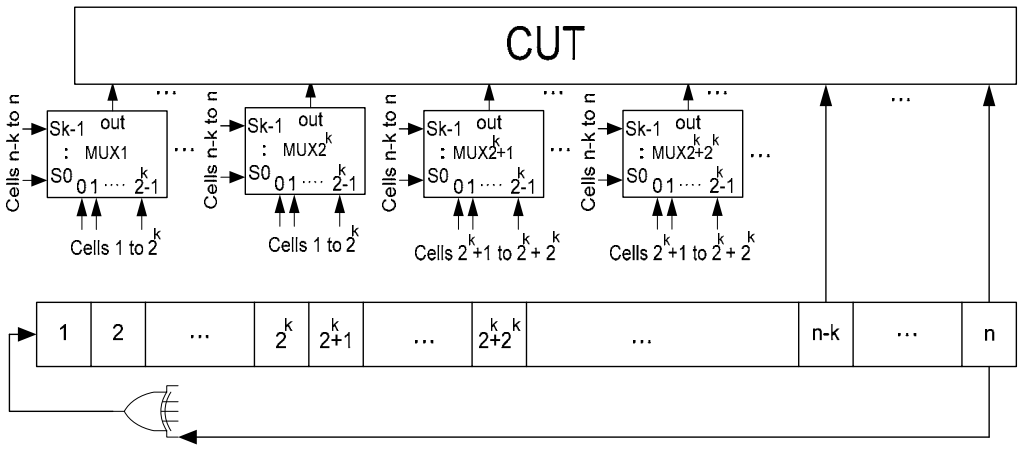

Fig. 5.6 The Architecture of the RLFSR for test-per-clock BIST

For example, if we consider a 6-bit LFSR that generates test vectors to a 6-input CUT, using the RLFSR with smoothing degree 2 (i.e. 2 selection lines – cells 5 and 6 of the LFSR – and four 4x1 multiplexers are needed where cells 1 to 4 of the LFSR are connected with the data lines of the multiplexers in a way similar to the one in Fig. 5.5(a)). Assuming the case where the selection lines (cells 5 and 6 of the LFSR) have the value “01”, then 16 vectors will be produced in the exhaustive set of test patterns

at the output of the conventional LFSR for the value “01” in the selection lines. However, the value in the selection lines will map each vector to another vector (but in both vectors the selection lines still have the same value since they are directly connected with the CUT as shown in Fig. 5.6). Fig. 5.7 shows the mapping between the vectors at the outputs of the conventional LFSR and the vectors at the inputs of the CUT (outputs of the proposed RLFSR). It is clear that all vectors are still produced using the RLFSR when selection lines are 01 but with different order. The same thing can be tested for other values of the selection lines.

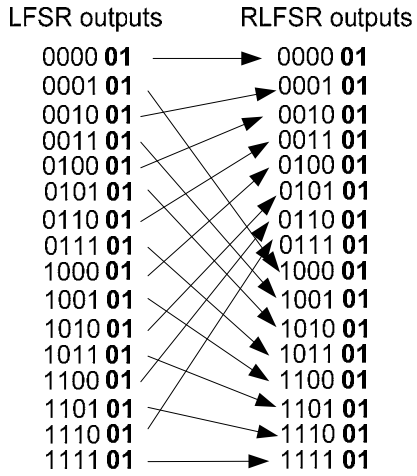

Fig. 5.7 The mapping from LFSR to RLFSR is one-to-one

5.4 Experimental Results

The proposed smoother was evaluated using a set of experiments performed on the ISCAS’89 benchmark circuits. The experiments are not only to evaluate the average power savings, but also to evaluate the effect of the smoothed patterns on fault coverage and test application time.

In the first set of experiments the same number of conventional LFSR and smoothed patterns is applied to each benchmark circuit to evaluate the average power reduction and the achieved fault coverage. Table 5.3 shows the results for different full-scan ISCAS’89 benchmark circuits. For each circuit, the columns labelled by PI and SI are used to refer to the number of primary inputs and scan-chain flip-flops respectively. The number of stages of the LFSR is labelled by column n. The applied test length is indicated by TL. For the normal and the smoothed patterns the columns labelled FC

and WSA are used to refer to the achieved fault coverage and the average weighted switching activity per clock cycle (while scanning in the sequence of test vectors) in the CUT respectively. Finally, for smoothing degrees 3 and 4 (k=3 and k=4), the "WSA Red%" shows the percentage of the WSA savings using a smoother of degree 3 and 4, respectively. It is clear from the table that the smoothed patterns of the proposed design have a negligible effect in fault coverage.

TABLE 5.3. EXPERIMENTAL RESULTS OF THE PROPOSED DESIGN FOR K=3 AND K=4

CKT	PI	SI	n	TL	Conventional patterns		Smoothed Patterns					
							K=3			K=4		
					FC%	WSA	FC%	WSA	WSA Red%	FC%	WSA	WSA Red%
S349	9	15	20	1000	99.17	51.21	99.27	38.14	26	98.97	35.09	31
S420	19	16	24	3000	98.61	27.36	98.59	17.27	37	98.61	16.27	41
S444	3	21	19	3000	96.42	54.82	95.81	40.84	26	95.81	39.39	28
S953	16	29	22	5000	97.61	29.85	97.28	22.53	25	97.19	21.28	29
S1423	17	74	30	2000	97.15	162.07	97.42	105.47	35	97.27	98.08	39
S5378	35	179	40	36000	98.27	580.74	98.27	457.95	21	97.92	436.49	25
S9234	19	228	50	200000	88.83	1341.8	89.06	783.19	42	88.81	735.6	45
S13207	31	669	60	100000	97.22	2271.63	96.31	1169.2	49	95.84	1027.75	55

Table 5.4 compares the results obtained using the proposed design (k = 3) with results obtained in [42] using a 2-input AND gate (refer to section 2.1.1). Table 5.4 compares the test length (TL), fault coverage (FC), and average power reduction (WSA Red %). It is clear that the proposed design in this chapter can achieve slightly better results in WSA reduction. Also it is clear that the proposed design can still get comparable fault coverage with those obtained in [42] while using a shorter test length. This gives an indication that the randomness properties of the proposed design are better than the design presented in [42].

TABLE 5.4: COMPARISON BETWEEN THE RESULTS OBTAINED BY THE PROPOSED DESIGN AND RESULTS FOUND IN [42].

Circuits	Results in [42]			Results of the proposed design		
	TL	FC	WSA Red%	TL	FC	WSA Red%
S349	1,024	99.27	32	1,000	99.27	26
S420	4,096	98.24	25	3,000	98.59	37
S444	4,096	96.39	25	3,000	95.81	26
S953	8,192	99.32	17	5,000	97.28	25
S1423	2,048	97.81	35	2,000	97.42	35
S5378	65,536	98.22	26	36,000	98.27	21
S9234	524,288	91.47	39	200,000	89.06	42
S13207	132,072	97.33	28	100,000	96.31	49

5.5 Conclusion

This chapter has presented a novel technique to smooth the test patterns before applying them to the CUT in scan-based circuits. The proposed technique can achieve multi-level degrees of smoothing with only one multiplexer as hardware area overhead. Also it can be extended to be used in multiple-scan-chains and in test-per-clock BIST. Experimental results on the full scan ISCAS'89 benchmark circuits show an average power reduction of up to 55%. The effect of the proposed technique in the fault coverage, test application time, and hardware area overhead is negligible (only one multiplexer). Comparisons between the proposed design and another previously published method show that the proposed method in this chapter is slightly better regarding average power reduction.

The proposed design is concerned mainly with reducing the transitions at the scan-chain input while scanning in test patterns. It can be easily be combined with algorithms that reduce the transitions while scanning out the captured response such as scan-chain ordering techniques (e.g. [58-62]) to get better results in power savings.

CHAPTER 6

Two-Phase Scan-Chain Ordering Algorithm

This chapter presents a two-phase scan-chain ordering algorithm that aims to reduce average and peak power in the CUT while scanning in a new test vector, scanning out the captured response, or while applying the test vector in the capture cycle. The first phase aims to reduce the scanning (shifting) average and peak power (i.e. while scanning in a test vector or scanning out a captured response), while the second phase reorders some cells in the scan-chain to reduce the capture power in the test cycle. This is in contrast to many techniques proposed in the literature, in which the ordering algorithm either targets scan power or capture power, but not both at the same time.

In this chapter, a new scan-chain ordering algorithm which has two phases is proposed and combined with the BS-LFSR from previous chapters. The BS-LFSR will reduce the average and peak power consumption while scanning in a test vector, also some of its properties will be used in phase 2 of the ordering algorithm. The first phase of the ordering algorithm will reorder the scan-chain cells such as to connect the cells that are most likely to have the same value in the captured response together. This will reduce the number of transitions in the scan-chain while scanning out the captured response. On the other hand, the second phase will use some features of the BS-LFSR to modify the scan-chain order achieved in phase 1 in order to minimise the Hamming distance between the applied test vector and the captured response in the test cycle, and hence reducing the peak power that may result in this cycle.

This chapter is organised as follows: section 6.1 discusses the sources of peak power violation in scan-based testing. Section 6.2 shows the algorithm of phase 1 of the two-phase ordering algorithm. Phase 2 of the scan-chain ordering algorithm is presented in section 6.3. Section 6.4 shows the experimental results and comparison with other previous works, while section 6.5 states the main conclusions of this chapter.

6.1 Sources of Peak Power Violation in Scan Testing

In full scan-based BIST, each test vector needs m clock cycles to be scanned into the scan chain (where m is the number of flip-flops in the full scan-chain); another clock cycle is needed to apply the scanned test vector to the CUT and to capture the response in the scan-chain. Finally, the next m clock cycles are used to scan out the captured response for signature analysis in parallel with scanning in a new test vector. Hence, the peak power violation could be caused by one of the following sources

1. Loading a new test vector: In this case the test vector to be scanned into the scan-chain has many transitions between the consecutive bits as a result of uncorrelated test patterns produced by a random TPG such as the LFSR. If this is the source of peak power violation then the last scanning clock (i.e. the m^{th} clock cycle), while loading a test vector and unloading a captured response, has a number of transitions equal or greater than any other clock cycle in that scan cycle. For example, if the test vector to be loaded into a 6-cell scan-chain is 101010, then in the 6^{th} clock cycle all flip-flops (or all except one, depending on the last bit of the scanned-out captured response) will have transitions which cause a high switching activity in the CUT. In this paper, this problem will be tackled using the BS-LFSR as a low transition TPG.

2. Unloading the captured response: In this case the captured response in the scan-chain has many transitions between the adjacent cells as a result of the applied test vector in the test cycle. If this is the source of peak power violation then the first scanning clock cycle, while unloading the captured response and loading a new test vector, has a number of transitions equal or greater than any other clock cycle in that scan cycle. For example, if the captured response is 010101 in a 6-cell scan chain, then in the first clock cycle all flip-flops (or all except one, depending on

the first bit of the new test vector to be scanned in) will have a transition. This will cause a high switching activity in the CUT. In this paper, this problem will be solved using a scan ordering algorithm that connects the cells that are most likely to have the same response values as each other.

3. Combination between the loaded test vector and the unloaded captured responses: In this case the maximum number of transitions occurs in a clock cycle which is not the first or last clock of the m clock cycles. Normally, if the maximum number of transitions in the flip-flops in the scan-cycle happens before clock cycle m/2 (the midway clock cycle), then the greater component of the transitions comes from the captured response, whereas if it is after the m/2 clock cycle, then the greater component of transitions comes from the loaded test vector. For example, if the captured response in an 8-cell scan-chain is 01010111, while the new test vector is 00011101 (least significant bits in the test vector will be scanned in first, and least significant bits in the captured response will be scanned out first), then in the 1^{st} clock cycle the number of transitions is 6 (scan-chain contents will be 10101011), in the 2^{nd} clock it is 7, in the 3^{rd} it is 8, in the 4^{th} it is 7, in the 5^{th} it is 6, in the 6^{th} it is 6, in the 7^{th} it is 5, and in the 8^{th} it is 4. Since the maximum number of transition does not happen in the 1^{st} clock cycle then it is not caused by the unloaded captured response alone; also since it is not in the 8^{th} clock cycle, then it is not caused by the loaded test vector alone. Hence, the source of peak power violation is a combination of the two, but since it happens in the third clock cycle (i.e. before the 4^{th} clock cycle which represents m/2) then the greater component comes from the captured response. In this paper, this sort of violation will be tackled by using BS-LFSR as a low transition TPG and the scan ordering algorithm.

4. Captured response in the test cycle: In this case the vector scanned and applied to the CUT during the test cycle, and the response captured during the same test cycle have a large Hamming distance. For example, if the scanned test vector in an 8-cell scan-chain is 11100011 and the response for applying this vector to the CUT is 00011100, then in the test cycle (one clock cycle) all flip-flops will change their values which will result in high switching activity in the CUT. In this chapter, this problem will be tackled by modifying the proposed scan-chain

ordering algorithm mentioned in point 2 (phase 2 of the ordering algorithm). In this way, unlike the previous approaches of scan-ordering algorithms which try to solve just one aspect of the peak power violations, the scan-chain ordering algorithm proposed here suggests a two-stages ordering algorithm to solve the peak power that may be produced in the shifting cycles or in the test cycle.

To identify the effect of each component (points 1 to 4) in the peak power consumption, some experiments have been performed on ISCAS'89 benchmark circuits and the results have been tabulated in table 6.1. These results show the maximum percentage of flip-flops that change their values in one clock cycle as a result of scanning in a new test vector (Scan-in) (The conventional LFSR was used in these experiments), scanning out a captured response (Scan-out), or during the test cycle (Capture). The column labelled by "Scan in-&-out" is used to represent the scanning peak power which may be caused by Scan-in alone (if the value equals Scan-in value), Scan-out alone (if the value equals Scan-out value), or both together (if the value is greater than Scan-in and Scan-out values). For most of the circuits it is clear that the results of peak power are slightly higher in scan cycles than test cycle for most of the benchmark circuits. However, to significantly reduce the overall peak power in a CUT, it is required to find techniques that can reduce in parallel the peak power that may be caused while scanning in a test pattern into the scan-chain, and while scanning out a response from the scan-chain, and during the test cycle were a test pattern is applied and a response is captured in the scan-chain.

TABLE 6.1. SOURCES OF PEAK POWER IN SCAN-BASED BIST

Circuit	m	PI	Maximum Transitions in Scan-Chain %			
			Scan-in	Scan-out	Scan in-&-out	Capture
S641	19	35	69%	67%	69%	26%
S1196	18	14	78%	59%	78%	38%
S1423	74	17	65%	59%	68%	56%
S5378	179	35	61%	57%	63%	51%
S9234	228	19	56%	59%	59%	41%
S13207	669	31	57%	60%	61%	54%
S15850	597	14	55%	62%	62%	64%
S35932	1728	35	61%	60%	61%	52%
S38417	1636	28	58%	58%	59%	56%
S38584	1452	12	54%	60%	60%	48%

6.2 Phase 1 of Scan-Chain Ordering Algorithm

The proposed algorithm of phase 1 is similar to the one found in [58, 59] with some modifications. The first modification is the consideration that the test vectors are random (e.g. vectors are generated by the proposed BS-LFSR which behaves as pseudorandom TPG), not a deterministic set of test vectors as considered in most ordering techniques. The consideration of random test vectors gives the algorithm the advantage that it will be more suitable for an arbitrary set of test vectors. Secondly, since the test vectors are very smooth because they are generated by the BS-LFSR, the proposed algorithm will take only the responses that may be captured in the scan-chain as the basis of the scan-chain ordering algorithm. This will lead to the best order of cells that reduces the transitions in the scan-chain while scanning-out the captured response. This is not the case for other ordering algorithms where a deterministic set of test vectors is considered with their responses as the basis of the ordering algorithm. This is because these algorithms try to make a compromise between reducing the number of transitions in the scan-chain while scanning in a test a vector and while scanning out the response because the test vectors and responses are not smooth and may have a very large number of transitions.

Phase 1 of the scan-chain algorithm has the following steps:

1. Apply a sufficient number of random test vectors to the CUT. The test vectors applied to the CUT are randomly selected from a pool of an exhaustive set of test vectors. The reason for not applying all test vectors is the very large time required to apply them when the scan chain has tens of flip-flops (e.g. for a 40-cell scan-chain there are 2^{40} possible test vectors). In order to keep the computational time acceptable, the number of test vectors to be applied to the scan-chain will be between 2^{12} and 2^{14} depending on the CUT size (number of primary inputs and flip-flops). After applying each test vector, the captured response is tabulated in a truth table for later use.

2. Divide the cells in the scan-chain into clusters, where each cluster has between 20 and 30 neighbouring cells.

3. For each cluster, starting from the first cell, search for the best compatible cell from other cells in the same cluster using the results tabulated in the truth table obtained in step 1. The best compatible cell is the one which most often has the same value as cell 1. Let the best compatible cell for cell 1 be denoted as cell x. Cell x will then be connected with cell 1 in the new ordering of the cells in the scan-chain. Hence we now have a chain of two cells; this chain can be further extended from both sides. The next step is to search for the best compatible cell for cell x and for cell 1 amongst the remaining cells in the cluster. Say, for example, that we find that cell y is the best compatible cell for cell 1 with 98 incidences of value match, and cell z is the best compatible cell for cell x with 83 incidences of value match. In this case we will add cell y to the chain. Hence we now have a chain of 3 cells with order cell y, cell 1, and cell x, respectively. Note that cell z has not been added to the chain since it could have a better compatibility on the other side of the chain. The process is then repeated until all cells in that cluster have been allocated. The first and last cells in the chain are connected together so that all cells in the cluster form a ring. This procedure is carried out for all clusters.

4. For each ring in each cluster, a search is done to find the best cells to behave as the scan input and scan output for that cluster. For example, assume we have a cluster with four cells a, b, c, and d that form a ring with order b, d, a, c. We then search whether b is the best scan input (hence, the order of the cells is S_{in}→b→d→a→c→S_{out} or S_{in}→b→c→a→d→S_{out}), or d is the best scan input (hence, the order of the cells is S_{in}→d→a→c→b→S_{out} or S_{in}→d→b→c→a→S_{out}), or a is the best scan-input (hence, the order of the cells is S_{in}→a→c→b→d→S_{out} or S_{in}→a→d→b→c→S_{out}), or finally if c is best scan input (hence, the order of the cells is S_{in} →c→b→d→a→S_{out} or S_{in}→c→a→d→b→S_{out}). The WT in equation (1.8) is used to find the minimum number of weighted transition for the best configuration in this step. This step is an improvement to the one found in [59], since here it assumes that for n cells in the cluster, there are 2×n possible configurations for the connection of S_{in} and S_{out} whereas in [59] the search is done for only n possible configurations because the authors assumed that the direction of the cells is counter clockwise only.

5. Each cluster is connected with its adjacent cluster by connecting the scan-output of the first one with the scan-input of the second, and so on till all the clusters form one scan-chain. The scan-input of the first cluster is connected with scan-input of the CUT (i.e. with the BS-LFSR) and the scan-output of the last cluster is connected with the scan-output of the CUT (i.e. with the signature analyser).

Example 6.1: Assume we have a scan-chain with 5-cells “a”, “b”, “c”, “d”, and “e” (all belong to one cluster) and these cells have initial order S_{in}→a→b→c→d→e→S_{out}. Assume we apply five randomly selected test vectors from a pool of all possible test vectors (32 possible vectors in this case). The responses captured in the scan-chain are R1 to R5 as shown in Fig. 6.1(a).

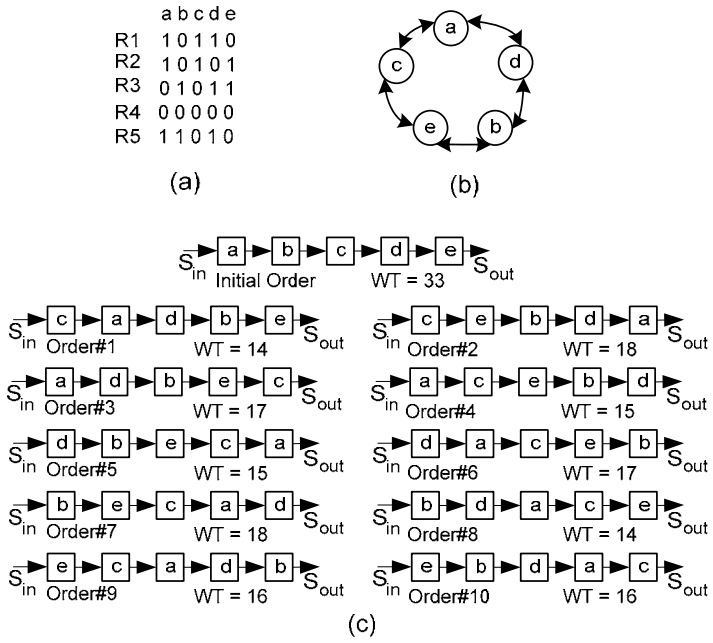

Fig. 6.1 Example of phase 1 of the scan-chain ordering algorithm

If we start with cell “a”, we can find that the best compatible cell is cell “c” since they have the same value in 4 responses. Hence cells “a” and “c” are connected with each other. Next, the best compatible cell for cell a from the remaining cells is cell “d” (3 similar values) and the best compatible value for cell “c” is cell “e” (also 3 similar values). Since both ends of the chain have compatible cells with equal compatibility, we can chose one of them to be connected to the chain at this time (say cell “d”). Hence we have a chain with 3 cells d→a→c, with two ends (“d”, and “c”). Now we

search for the best compatible for cells “d” and “c”, the best compatible for cell “d” is cell “b” (with four similar values) and for cell “c” it is cell “e” (with 3 similar values). In this case cell “b” is added to the chain which will have the order b→d→a→c with two ends (“b” and “c”). Finally since there is only one cell left unconnected; it will inserted between the two ends of the scan chain in order to form a ring as shown in Fig. 6.1(b). Finally, as there are 5 cells in the ring, there are 10 possible orders for the scan-chain as shown in Fig. 6.1(c). Each one of these orders is tested for WT number; the one with the minimum value of WT is the best for reducing the transitions while scanning out the captured response. As shown in Fig. 6.1(c), order#1 and order#8 are the best among the 10 orders because WT = 14 for these two orders. Using one of these orders will reduce the WT from 33 in the initial order to 14 in these suggested orders.

6.3 Phase 2 of Scan Ordering Algorithm

In this phase of the scan-chain ordering algorithm, some cells of the ordered scan-chain in phase 1 will be reordered again in order to reduce the peak power which may result during the test cycle. This phase mainly depends on an important property of the BS-LFSR proposed in previous chapters (The special cases configuration in chapter 4, where 50% of transitions will be saved). This property states that if two cells are connected with each other then the probability that they have the same value at any clock cycle is 0.75. (In a conventional LFSR where the transition probability is 0.5, two adjacent cells will have the same value in 50% of the clocks; for a BS-LFSR that reduces the number of transition of an LFSR by 50%, so the transition probability is 0.25, and hence, two adjacent cells will have the same value in 75% of clock cycles). Thus, for two connected cells (cell j and cell k), if we apply a sufficient number of test vectors to the CUT, then the values of cells j and k are similar in 75% of the applied vectors.

Now, assume we have cell x which is a function of many cells including at least cells y and z. If the value that cell x will assume in the captured response is the same as its value in the applied test vector (i.e. no transition will happen for this cell in the test cycle) in the majority of cases where cells y and z have the same value in the applied

test vector, then we connect cells y and z together on the scan chain, since they will have the same value in 75% of cases. This reduces the possibility that cell x will undergo a transition in the test cycle.

The steps in this phase include:

1. Simulate the CUT for the test patterns generated by the BS-LFSR.
2. Identify the group of vectors and responses that violate the peak power (i.e. the capture peak power exceeds the scan peak power after phase 1 of scan-chain ordering).
3. In these vectors identify the cells that mostly change their values in the test cycle and cause the peak power violation.
4. For each cell found in step 3, identify the cells that play the key role in the value of this cell in the test cycle.
5. If it is found that when two cells have a similar value in the applied test vector, the concerned cell will most probably have no transition in the test cycle, then connect these cells together. If it is found that when two cells have a different value, the cell under consideration will most probably have no transitions in the test cycle, then connect these cells together through an inverter.

It is important to note that step 2 states that we have to identify the test cycles (applied vectors and captured responses) that violate the peak power. Thus, if there is no violation, then there is no need to apply the algorithm for this phase. For example, if we have a CUT with initial scan peak power of 50 mW, and capture peak power of 20 mW. Then, if the scan peak power becomes 25 mW after we use the BS-LFSR and phase 1 of the scan-chain ordering algorithm, while the capture peak power is still below 25 mW (i.e. less than the new scan peak power), then phase 2 is not needed. This is because while phase 2 decreases the power in test cycle, it will slightly increase the power in scanning cycles (since the 2 phases are orthogonal). But if we find that the capture power is greater than the achieved scan peak power after using the BS-LFSR and phase 1, then phase 2 will be implemented to reduce the test cycle peak power.

Example 6.2: Assume that we have a CUT with a scan-chain that contains 7 cells. Fig. 6.2(a) shows a part of the logic circuit of the CUT. It is clear that the response captured in cell S6 (represented as S6+) is a function of S4, S6, and S7. Also the response captured at cell S7 (represented as S7+) is a function of the bits of the test vector scanned into S4, S5 and S7. Fig. 6.2(b) shows the logic equations and truth table for S6+ and S7+. If we examine the truth table of S6+, we see that in 4 rows the value of S6+ is the same as the value of S6 (i.e. in these locations no transition will happen to S6 in the test cycle), and it is clear that in 3 of these rows the values of S6 and S4 are similar in the applied test vector. Hence, if the scanned test vector guarantees that the value inserted in cell S6 is similar to the value inserted in S4, then the probability that S6+ has the same value as S6 will be increased. Thus power dissipation during the test cycle decreases. This can be achieved by connecting cells S4 and S6 together and using the BS-LFSR as a TPG. On the other hand, the truth table for S7+ shows that in 5 rows the value of S7+ is the same as of S7. In 4 of these rows the value of S7 is the complement to the value in S5. Hence, if cells S7 and S5 are connected with each other through an inverter then they will have different values for most of the applied test vectors that are generated by the BS-LFSR.

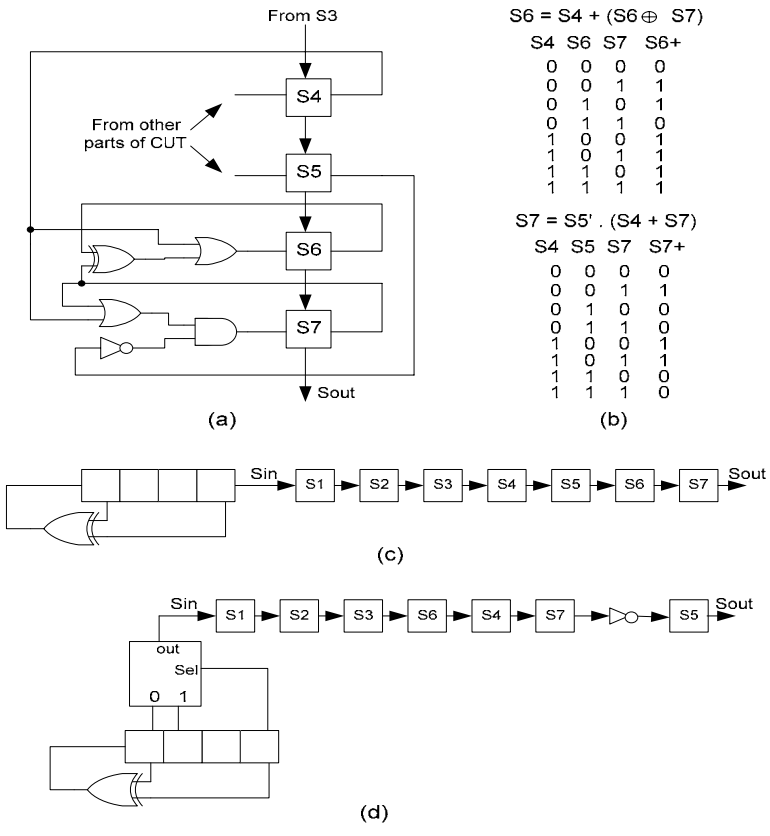

Fig. 6.2 Example of phase 2 of scan-chain ordering algorithm.

To show the effect of the proposed reordering of cells S4, S5, S6, and S7, Fig. 6.2(c) shows a conventional 4-bit LFSR connected with the default scan-chain. Fifteen test vectors are applied to the CUT (the initial seed in the LFSR is “1111”). These vectors cause a total of 8 transitions in S6 and 6 transitions in S7 during the 15 test cycles. Finally the BS-LFSR is used to generate 15 vectors to the modified scan-chain as shown in Fig. 6.2(d). These vectors cause 4 transitions in S6 (50% less than default order) and 2 transition in S7 (67% less than default order).

To justify phase two of the scan-chain ordering algorithm, example 6.3 presents a real example where a part of one of the ISCAS’89 benchmark circuits is used, and some cells are reordered to reduce the capture peak power in the test cycle.

Example 6.3: Fig. 6.3 shows part of s838 benchmark circuit. It is clear from the figure that this part contains five cells (G0, G10, G11, G12, and G13). The cell under consideration is cell G11, in which its value in the next clock cycle in normal operation mode is a function of its current value as well as the values stored in cells G0, G10, G12, and G13. Assuming that G11 is found to be one of the cells that has a transition in the capture cycle after phase 1 of the scan-chain ordering algorithm in many test patterns, and we want to reduce its effect in capture peak power as much as we can by reordering some cells. First, a truth table for the next value of cell G11 is done as shown in table 6.2.

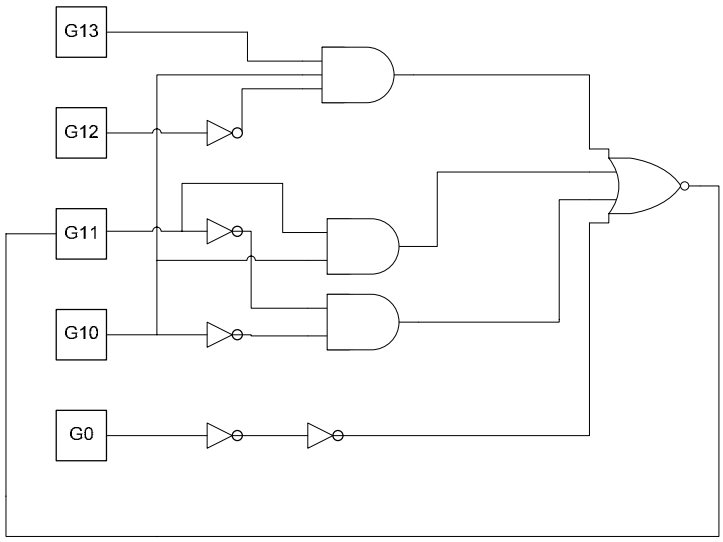

Fig. 6.3 Part of s838 ISCAS’89 benchmark circuit.

TABLE 6.2 TRUTH TABLE FOR THE CIRCUIT SHOWN IN FIG. 6.3 WHERE THE NEXT VALUE OF G11 (I.E. G11+) IS THE OUTPUT

G0	G10	G11	G12	G13	G11+	G11+ =? G11
0	**0**	**0**	**0**	**0**	**0**	**Yes**
0	**0**	**0**	**0**	**1**	**0**	**Yes**
0	**0**	**0**	**1**	**0**	**0**	**Yes**
0	**0**	**0**	**1**	**1**	**0**	**Yes**
0	**0**	**1**	**0**	**0**	**1**	**Yes**
0	**0**	**1**	**0**	**1**	**1**	**Yes**
0	**0**	**1**	**1**	**0**	**1**	**Yes**
0	**0**	**1**	**1**	**1**	**1**	**Yes**
0	1	0	0	0	1	No
0	**1**	**0**	**0**	**1**	**0**	**Yes**
0	1	0	1	0	1	No
0	1	0	1	1	1	No
0	1	1	0	0	0	No
0	1	1	0	1	0	No
0	1	1	1	0	0	No
0	1	1	1	1	0	No
1	**0**	**0**	**0**	**0**	**0**	**Yes**
1	**0**	**0**	**0**	**1**	**0**	**Yes**
1	**0**	**0**	**1**	**0**	**0**	**Yes**
1	**0**	**0**	**1**	**1**	**0**	**Yes**
1	0	1	0	0	0	No
1	0	1	0	1	0	No
1	0	1	1	0	0	No
1	0	1	1	1	0	No
1	**1**	**0**	**0**	**0**	**0**	**Yes**
1	**1**	**0**	**0**	**1**	**0**	**Yes**
1	**1**	**0**	**1**	**0**	**0**	**Yes**
1	**1**	**0**	**1**	**1**	**0**	**Yes**
1	1	1	0	0	0	No
1	1	1	0	1	0	No
1	1	1	1	0	0	No
1	1	1	1	1	0	No

It is clear from the truth table that the next value of cell G11 (i.e. G11+) is same as cell G11 in 17 locations (i.e. no transition occurred in cell G11). Statistically, by using random patterns, then for every 32 patterns, 15 transitions will occur at cell G11. However, if we look in the rows where the value of G11+ equals the value of G11 (bold rows in table 6.2) we found that in 5 cases of them the value of G11 is similar to the value of G0, and different in 12 cases. Similarly, the value of G11 and G10 are similar in 8 cases and different in 9 cases. Also the value of G11 and G12 are similar in 9 cases and different in 8 cases. Finally, the value of G11 and G13 are similar in 9 cases and different in 8 cases. The same is repeated for cells (G0,G10), (G0,G12), (G0,G13), (G10,G12), (G10,G13), and (G12,G13). The highest number of these data is found to be 12, this is when cell G0 and G11 have different values.

Using these data we can conclude that if we insert different values in G0 and G11, then that will statistically decrease the probability that G11 will have a transition in the next cycle. Thus, if we connect cell G0 with cell G11 through an ***inverter***, and use the BS-LFSR to generate test patterns, then, as shown previously, the value of G0 will be different from the value of G11 in 75% of the cases. So if we run the BS-LFSR to generate 32 test vectors, then the statistical expectation is that in 24 patterns the value of G0 is different from the value of G11. Using the data in the truth table (table 6.2), we found that if G0 and G11 have similar value (16 cases) then G11 will have a transition in 11 cases (i.e. $P_{t\text{-similar}} = 11/16$), while when G0 and G11 have different values (16 cases) then G11 will have a transition in 4 cases (i.e. $P_{t\text{-different}} = 4/16$). Use these statistics when the BS-LFSR is used and cell G0 and G11 are connected with each other through an inverter, then running the BS-LFSR for 32 test vectors will cause a number of transitions as given in equation 6.1

$$\text{Number of transitions} = P_{(G11=G0)} \times P_{t\text{-similar}} + P_{(G11!=G0)} \times P_{t\text{-different}} \qquad (6.1)$$

$$= 8 \times (11/16) + 24 \times (4/16)$$

$$= 5.5 + 6 = 11.5 \text{ transitions}$$

This shows how the number of transitions is reduced by an average of 3.5 transitions every 32 test patterns.

There are some points to be discussed regarding phase 2 of the scan-chain ordering algorithm. Firstly, it is important to note that the inverters will have no effect on the number of transitions in the cells of the scan-chain. For example if a vector of 111 is to be scanned into two cells, where the second cell is connected with the first one through an inverter, then for the first cell, three consecutive ones will be scanned which will not cause any transition. For the second cell, all the bits will be inverted before passing through the cell; hence 3 consecutive zeros will be scanned into the second cell which will not cause any transition in this cell as well. Secondly, the presence of the inverters should be taken into consideration in computing the stored signature for signature analysis purposes; this step is needed also in any scan ordering algorithm. Finally, the second phase of the ordering algorithm will affect the order in phase 1 which will cause the transitions during scanning out the captured response to increase. However, since the second phase will only reorder a small number of cells

(compared to the total number of cells in the scan-chain), then it will have only a very small effect on the phase 1 results.

6.4 Experimental Results

A group of experiments was performed on full scan ISCAS'89 benchmark circuits to evaluate the efficiency of the proposed scan-chain ordering algorithm on average and peak power consumptions and the effect on fault coverage. The proposed BS-LFSR in chapter 4 was used as a TPG for the CUTs in order to reduce the number of transitions in the generated test vectors; any other low transition TPG such as RLFSR can be used as well.

A set of experiments is used to evaluate the BS-LFSR together with the proposed scan-chain ordering algorithm in reducing average and peak power. For each benchmark circuit, the same numbers of conventional LFSR and BS-LFSR patterns are applied to the full scan configuration. The LFSR patterns are applied to the CUT before applying the proposed scan-chain ordering algorithm, while the BS-LFSR patterns are applied to the ordered scan-chain. Table 6.3 shows the obtained results for 20 ISCAS'89 benchmark circuits. The column labelled by n refers to the size of the used LFSR or BS-LFSR that generates the test vectors to the CUT. The column labelled by m refers to the number of flip-flops in the scan-chain while the column labelled by PI refers to the number of primary inputs in each benchmark circuits. The column labelled by TL refers to the number of test vectors applied to the CUT. The next three columns show the fault coverage (FC), average weighted switching activity per clock cycle (WSA_{avg}), and the maximum WSA in a clock cycle (WSA_{pk}) for patterns applied using the conventional LFSR. The next three columns show FC, WSA_{avg}, and WSA_{pk} for the BS-LFSR with ordered scan-chain. Finally, the last two columns show the savings in average and peak power by using the BS-LFSR with the scan-chain ordering algorithm.

TABLE 6.3 EXPERIMENTAL RESULTS OF AVERAGE AND PEAK POWER REDUCTION OBTAINED BY USING THE PROPOSED TECHNIQUES.

Circuit	n	m	PI	TL	LFSR			BS-LFSR with cell ordering			%Savings of BS-LFSR	
					FC%	WSA_{avg}	WSA_{pk}	FC%	WSA_{avg}	WSA_{pk}	WSA_{av}	WSA_{pk}
S208	22	8	11	3000	100	31.35	47	100	14.36	26	54	45
S298	19	14	3	700	100	78.74	136	100	42.83	85	46	38
S344	26	15	9	200	100	98.11	156	100	47.73	79	51	49
S349	21	15	9	1000	99.31	105.65	171	99.18	49.17	92	53	46
S420	32	16	19	3000	98.64	46.12	74	98.68	19.14	48	58	35
S444	24	21	3	3000	96.39	99.42	178	96.21	56.14	112	44	37
S526	24	21	3	8000	98.81	134.06	196	99.17	71.06	119	47	39
S641	32	19	35	3000	97.84	97.78	153	97.54	42.20	84	57	45
S838	32	32	35	20000	96.15	81.91	151	96.21	33.14	83	60	45
S953	30	29	16	6000	97.67	57.34	101	98.31	27.81	68	51	33
S1196	30	18	14	2000	95.33	53.18	74	95.51	21.52	42	60	43
S1238	30	18	14	3000	91.11	61.20	97	90.97	34.80	59	43	39
S1423	35	74	17	2000	97.77	318.33	486	97.77	160.07	349	50	28
S5378	40	179	35	40000	98.42	1143.24	1639	98.40	625.28	993	45	39
S9234	40	228	19	100000	87.27	2817.45	3988	87.28	1108.93	2197	61	45
S13207	60	669	31	100000	96.45	4611.67	7108	96.39	1897.33	4172	59	41
S15850	60	597	14	100000	94.75	4907.29	7244	94.72	2533.49	4904	48	32
S35932	64	1728	35	200	87.88	7945.81	12592	87.89	2793.16	5723	65	55
S38417	64	1636	28	100000	95.73	10965.50	16380	95.68	5022.30	10017	54	39
S38584	64	1452	12	100000	94.46	11194.65	15974	94.48	5682.72	7851	49	51

In order to provide a comparison with another technique published previously by other authors, table 6.4 compares the results obtained by the proposed technique with those obtained in [50]. In [50] the authors proposed an approach that combines a low-transition TPG with scan-chain reordering algorithm (i.e. this is similar to the approach presented in this chapter where the BS-LFSR is combined with a scan-chain ordering algorithm). The proposed low power TPG in [50] has 2 different smoothing degrees: 2-bit smoother and 3-bit smoother. When the 3-bit smoother is used, although it can achieve better reduction in average power consumption, there is a significant degradation to the fault coverage as well as an increase to the hardware area overhead. Thus comparison will be with the 2-bit smoother TPG. Table 6.4 compares the test length (TL), fault coverage (FC), average power reduction (RED WSA_{avg}), and peak power reduction (RED WSA_{pk}). It is clear that the proposed method has comparable results with the technique proposed in [50] regarding the TL, FC, while WSA_{avg} reduction is slightly better in [50] than in the proposed method. However, the proposed technique is much better regarding peak power reduction for all benchmark circuits. Furthermore, there is no indication in [50] that the capture power has been considered in calculation of power consumption.

TABLE 6.4 COMPARISON WITH RESULTS OBTAINED IN [50]

Circuit	Results in [50]				Results of proposed method			
	TL	FC	%RED WSA_{av}	%RED WSA_{pk}	TL	FC	%RED WSA_{av}	%RED WSA_{pk}
S641	4096	95.91	57.41	NA	3000	97.54	57	45
S953	8192	97.97	55.47	NA	6000	98.31	51	33
S1196	4096	92.71	56.58	NA	2000	95.51	60	43
S1423	2048	98.59	57.25	NA	2000	97.77	50	28
S5378	65536	98.56	58.51	13.31	40000	98.40	45	39
S9234	131072	91.40	59.37	13.75	100000	87.28	61	45
S38417	81984	95.42	60.37	13.78	100000	95.68	54	39
S38584	82055	95.00	60.87	13.12	100000	94.48	49	51
AVG	**47,385**	**95.69**	**58.22**	**13.49**	**44,125**	**95.62**	**53.38**	**41.5**

Finally, in order to provide more comparisons regarding peak power reduction, table 6.5 compares the results obtained by the proposed technique for peak power reduction with those obtained in [73] where the authors proposed a technique to reduce the switching activity during scan shift cycles by assigning identical values to adjacent scan inputs, and they reduce the switching activity during capture cycle by limiting the number of scan-chain cells that capture responses, hence a reduction to average and peak power consumption can be achieved. It is clear from table 6.5 that the proposed method has slightly better overall result than those in [73] for the tested benchmark circuits.

TABLE 6.5 COMPARISON OF PEAK POWER REDUCTIONS WITH RESULTS IN [73]

Circuit	Results in [73] WSA_{pk} Savings %	Proposed Method WSA_{pk} Savings %
S5378	36.6	39
S9234	38.9	45
S13207	46.1	41
S15850	42.2	32
S38417	40.1	39
S38584	35.9	51
AVG	**39.97**	**41.17**

6.5 Conclusions

In this chapter a novel algorithm for a two-phase scan-chain ordering has been presented. The proposed ordering algorithms are combined with the BS-LFSR presented in chapter 4 in order to reduce the switching activity in the CUT while scanning in a test vector; also some properties of the BS-LFSR have been used in phase 2 of the ordering algorithm.

The chapter started with a study of the possible sources of peak power in a scan-based test. It reached the conclusion that in order to significantly reduce the peak power we should reduce power while scanning in a test vector (this is achieved by using the BS-LFSR), while scanning out a captured response, and also during the test cycle (the capture power).

The aim of the first scan-chain ordering phase is to reduce the switching activity while scanning out a response to the signature analyser. On the other hand, the second phase modifies the order achieved in phase 1 in order to reduce the capture peak power that results during the test cycle. When the BS-LFSR is used together with the proposed two-phase scan-chain ordering algorithm, the average and peak power are substantially reduced when compared with the conventional LFSR results. The effect of the proposed design on the fault coverage, test application time, and hardware area overhead is negligible. Comparisons between the proposed design and other previously published methods show that the proposed design can achieve comparable or slightly better results.

CHAPTER 7

A High Fault Coverage Technique for Test-per-Scan BIST

This chapter presents a new technique that increases fault coverage and decreases test application time in test-per-scan BIST. The proposed design, called multi-output LFSR (MO-LFSR), uses the output of several D-FF cells in the LFSR, selecting one output to be used throughout each scan cycle to feed the scan chain input in a full-scan BIST. Some bits of the LFSR are used at the beginning of the new scan cycle to select which output to be used for this cycle in such a way that the patterns generated by this output can detect some of the random pattern resistant faults (hard to detect faults), hence, increasing the fault coverage and decreasing the test length at the same time. Experimental results on ISCAS'89 benchmark circuits show that the proposed technique can achieve a high fault coverage using simple logic functions and relatively small number of test patterns compared with the conventional LFSR or other existing techniques. The area overhead is relatively small.

This chapter is organised as follows: Section 7.1 presents the main features of the proposed design and describes the main idea behind the proposed technique. Section 7.2 discusses the main steps of the algorithm to achieve high fault converge using the proposed design. The experimental results are presented in section 7.3. Finally, section 7.4 states the main conclusions of this chapter.

7.1 Key Features of the Proposed Design

7.1.1 Key Idea

In conventional test-per-scan scheme, an n-stage LFSR is used to generate test vectors for an m-flip-flop scan chain where a single bit (e.g. bit n) of the LFSR feeds the scan chain input (as shown in Fig. 7.1 where cell 5 of the 5-stage LFSR is used to feed the scan-chain input). For targeted fault coverage, the LFSR runs till the test patterns generated by the bit which feeds the scan chain achieve this coverage. In fact, the output sequences of all other flip-flops within the LFSR are ignored despite the fact that in most cases they generate different sequences other than those produced by the bit which is connected to the scan chain input. In the technique presented in this chapter, an algorithm is proposed to use many outputs of the LFSR cells to achieve targeted fault coverage with a relatively small number of test vectors. Thus, it is possible to detect a greater number of hard to detect faults since a hard to detect fault with respect to patterns generated by a specific output could be easier with respect to another.

Example 7.1: Fig. 7.1 shows a five-bit LFSR used to generate test vectors for a CUT with a scan chain of five flip-flops (scan-chain is not shown in Fig. 7.1), where bit 5 (O_5 in Fig. 7.1) is used to feed the scan chain. Let the test patterns shown in table 7.1 to be the targeted patterns used to detect the hard to detect faults of the CUT. Table 7.2 shows the 31 states of the LFSR (assuming initial seed is 11111), and table 7.3 shows the generated patterns by O_5 which are fed and applied to the CUT with each scan cycle. Finally, for the first six scan cycles, table 7.4 shows the patterns generated by each output (O_1 to O_5 in Fig. 7.1) of the LFSR if it is used to feed the scan-chain.

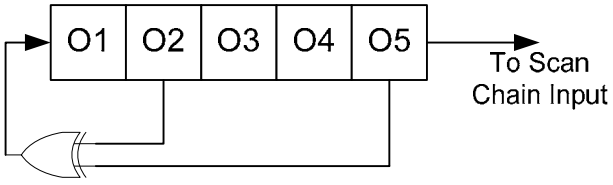

Fig. 7.1 5-bit LFSR where bit 5 feeds the scan chain.

TABLE 7.1 TEST PATTERNS THAT DETECT HARD-TO-DETECT FAULTS

Targeted Patterns
00101
11111
01100
10100

TABLE 7.2 5-BIT LFSR STATES.

LFSR state	O_1	O_2	O_3	O_4	O_5	LFSR state	O_1	O_2	O_3	O_4	O_5
Q_1	1	1	1	1	1	Q_{17}	1	0	1	0	0
Q_2	0	1	1	1	1	Q_{18}	0	1	0	1	0
Q_3	0	0	1	1	1	Q_{19}	1	0	1	0	1
Q_4	1	0	0	1	1	Q_{20}	1	1	0	1	0
Q_5	1	1	0	0	1	Q_{21}	1	1	1	0	1
Q_6	0	1	1	0	0	Q_{22}	0	1	1	1	0
Q_7	1	0	1	1	0	Q_{23}	1	0	1	1	1
Q_8	0	1	0	1	1	Q_{24}	1	1	0	1	1
Q_9	0	0	1	0	1	Q_{25}	0	1	1	0	1
Q_{10}	1	0	0	1	0	Q_{26}	0	0	1	1	0
Q_{11}	0	1	0	0	1	Q_{27}	0	0	0	1	1
Q_{12}	0	0	1	0	0	Q_{28}	1	0	0	0	1
Q_{13}	0	0	0	1	0	Q_{29}	1	1	0	0	0
Q_{14}	0	0	0	0	1	Q_{30}	1	1	1	0	0
Q_{15}	1	0	0	0	0	Q_{31}	1	1	1	1	0
Q_{16}	0	1	0	0	0	Back again to Q_1					

TABLE 7.3 TEST PATTERNS FED INTO SCAN CHAIN USING LFSR IN FIG. 3.

Scan cycle	Test pattern	Scan cycle	Test pattern
1	**11111**	17	10101
2	00110	18	11011
3	10010	19	00011
4	00010	20	11100
5	10111	21	11010
6	**01100**	22	01000
7	01111	23	01010
8	10011	24	11101
9	01001	25	10001
10	00001	26	11110
11	01011	27	01101
12	10110	28	00100
13	00111	29	**00101**
14	11001	30	01110
15	**10100**	31	11000
16	10000	Back to pattern 1	

TABLE 7.4 FIRST 6 TEST PATTERNS OF ALL LFSR OUTPUTS.

Scan cycle	LFSR initial value	O_1 patterns	O_2 patterns	O_3 patterns	O_4 patterns	O_5 patterns
1	11111	10011	11001	11100	11110	**11111**
2	01100	01001	**10100**	11010	01101	00110
3	01001	00001	10000	01000	00100	10010
4	01000	01011	10101	01010	**00101**	00010
5	11101	10110	11011	11101	01110	10111
6	00110	00111	00011	10001	11000	**01100**

It is clear from table 7.3 that if we start with the LFSR seed of "11111" then we need to go through 29 scan cycles to get the patterns listed in table 7.1, hence applying 29 test patterns to the CUT, which requires 174 clock cycles. On the other hand, table 7.4 shows that the test patterns listed in table 7.1 can be found within the first 6 scan cycles if different outputs of the LFSR are allowed to feed the scan chain input for different scan cycles. For this simple example, it is only O_2, O_4, and O_5 that generate useful patterns. Table 7.4 shows that the initial value of the LFSR at the beginning of the scan cycle for each useful pattern is 01100 when the useful pattern is generated by O_2, 01000 for the one generated by O_4, and (11111 and 00110) for the two patterns generated by O_5. If bit 3 and bit 4 of the LFSR are used to select which output to feed the scan chain at the beginning of each scan cycle as shown in Fig. 7.2, then it is possible to get all the patterns listed in table 7.1 with only 6 scan cycles, which require 36 clock cycles. Bit 3 and 4 are used as selection line because they have the value "10" in LFSR initial value in table 7.4 when the targeted pattern is generated by O_2, "00" when the useful pattern is generated by O_4, and "11" when the useful patterns are generated by O_5. Thus O_2, O_4, and O_5 can be connected with multiplexer inputs in_2, in_0, and in_3, respectively (in_1 of the MUX can be connected with any output in this case). It is important to note that in Fig. 7.2 the new scan cycle signal comes from a Mod (m+1) counter which is needed in the control logic for any test-per-scan BIST [143].

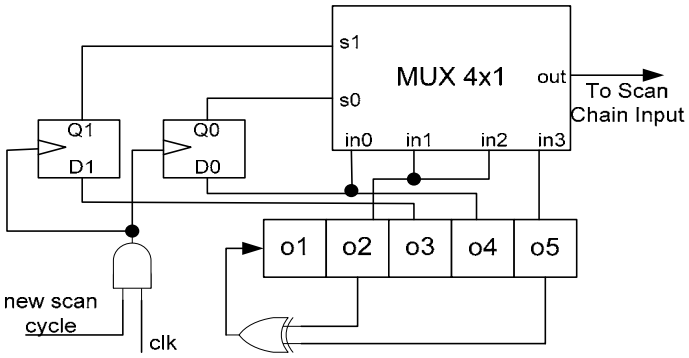

Fig. 7.2 Suggested design of MO-LFSR.

7.1.2 Key Property of the Sequences Generated by Different Outputs of the LFSR

An important feature of the sequences generated by an n-bit LFSR is that the sequence generated by output 1 is the same as the one generated by output 2 but shifted by one clock cycle. Similarly, it is the same as the sequence generated by output 3 but with a two clock shift, and so on. This means that the same test pattern is generated by output 1 if the scan cycle starts from an initial LFSR state of Q_1, as would be generated by output 2 if the scan cycle starts from an initial LFSR state of Q_2. Similarly, the same pattern will be generated by output 3 if the scan cycle starts with the LFSR in initial state Q_3, and so on. So, how is it useful to use more than one output to feed the scan chain if they will produce the same patterns?

The answer to this question depends on how many scan cycles will be run (i.e. how many test patterns will be generated for test-per-scan BIST) before we reach a state where a test pattern generated by a specific output can be generated by another output. In other words, if an m-bit test pattern is generated by output 1 when the LFSR starts from state Q_1, then how many test patterns will be generated before the LFSR reaches state Q_2 as the beginning of a new scan cycle, with the result that output 2 can generate the same pattern as generated by output 1. This normally depends on the size of LFSR (n), and the size of the scan chain (m).

For most BIST circuits, the size of the LFSR is smaller than or equal to the size of the scan chain (i.e. $n \leq m$). If a maximal length LFSR is used then it will move through the states Q_1 to $Q_2{}^n{}_{-1}$ and then cycle (i.e. it follows the sequence Q_1, Q_2, Q_3,…, $Q_2{}^n{}_{-2}$, $Q_2{}^n{}_{-1}$, Q_1, Q_2, …). If the first scan cycle starts at LFSR state Q_1, then the second scan

cycle will start at LFSR state Q_{1+m} , and the i^{th} scan cycle will start at LFSR state $Q_{1+(i*m) \bmod 2^n-1}$. At any scan cycle, in order for output 2 to produce the same pattern previously produced by output 1, a scan cycle has to start at LFSR state Q_2. Similarly, a scan cycle would need to start at state Q_3 in order for output 3 to generate the same pattern, or with state Q_n for output n. Since m > n, then no output will produce any pattern that is produced by other outputs until $Q_{1+(i*m) \bmod 2^n-1}$ equals Q_2 or Q_3, … or Q_n. This will not occur until L scan cycles have been run (and L test patterns are generated), where the minimum value of L is given by equation (7.1).

$$L = \left\lfloor \frac{2^n - 1}{m} \right\rfloor \tag{7.1}$$

Table 7.5 shows for different values of n and m, the value of L which indicates how many distinct test patterns can be generated by each output before any possible repetition, and Σ refers to the total number of different test patterns that can be generated by all outputs, which equals L × n. In example 7.1 above, n = m =5, thus L = 6. That means, up to 6 scan cycles, each of the five outputs can produce six different test patterns from those generated by other outputs. This is shown in table 7.4, where there are 30 different test vectors generated by the 5 outputs.

TABLE 7.5 L FOR DIFFERENT VALUES OF M AND N.

n	m	L	∑
16	17	3855	61,680
16	29	2259	36,144
32	37	$\approx 116 \times 10^6$	$\approx 3.71 \times 10^9$
32	1,000	$\approx 4 \times 10^6$	$\approx 137 \times 10^6$
64	2,000	$\approx 9.22 \times 10^{15}$	$\approx 5.90 \times 10^{17}$
64	10,000	$\approx 1.84 \times 10^{15}$	$\approx 1.18 \times 10^{17}$

The other case is where n > m. In this case, in addition to the possible repetition in test patterns even if only one output is used to feed test patterns to the scan chain input, there also exist some outputs that will repeat the patterns of other outputs. The number depends on how much smaller m is than n. For example if n = 8 and m = 6, then the patterns produced by output 1 will be produced by output 7, and those produced by output 2 will be generated by output 8 as well. This means output 7 and output 8 are not useful in the generation of new patterns that can detect hard faults. However, outputs 1 to 6 are still useful until we reach L in equation (7.1).

7.2 The Proposed Design and Algorithm

As shown in example 7.1 and Fig. 7.2, the basic structure of the proposed design has three components: an LFSR, a small number of extra D-FFs, and a small size multiplexer. The goal of the proposed algorithm is to obtain high fault coverage using a small (but not necessarily minimal) number of test patterns.

Given a CUT, the main steps involved in the proposed algorithm of the suggested design procedure are:

1. For a specified test length, identify the hard to detect faults of the CUT.

2. Run the LFSR for the specified test length, and simulate the CUT for each output of the LFSR, then select the minimum number of LFSR outputs that can detect all of the listed faults up to that point, or a targeted percentage of these faults.

3. Search for the best size multiplexer to be used, and search for the best LFSR outputs to act as selection lines.

4. Use the new design (LFSR plus multiplexer) to simulate the CUT to determine the final fault coverage for a targeted test length.

7.2.1 Identify the Hard to Detect Faults

For a specified test length, the hard to detect faults can be identified by different theoretical and experimental methods. The method used in this chapter is an experimental method which is based on the generation of six different sets of pseudorandom patterns corresponding to different initial seeds. Then a simulation is performed on the CUT and the faults that are detected by three or more of the pseudorandom patterns set are considered to be easy to detect faults and the remaining faults are considered as hard to detect faults.

7.2.2 Selection of Minimum Number of Outputs that Achieves High Fault Coverage

The next step after identifying the hard to detect faults is to run the LFSR for the specified test length, and then the CUT is simulated for each output bit of the LFSR separately assuming that this output is the one which feeds the scan chain. Each detected fault from the list for an LFSR output is stored in array related to that output, together with the initial value of the LFSR at the beginning of the scan cycle in which that fault was detected.

The output that has an array that contains the largest number of detected faults is considered to be the best output to feed the scan chain. All the faults detected by this output are deleted from all other arrays related to other outputs. Then for the remaining outputs, the one which has an array with the largest number of detected faults is considered to be the best output to share the previous selected output to feed the scan chain input, and the faults detected by this output are deleted from all other arrays. This process is repeated until all outputs are categorised according to how many new faults they can detect.

Each time an output selected as the best among the remaining outputs, all the faults detected by this output will be omitted from all other outputs that detect them. Thus, the number of detected faults stored in the arrays related to the remaining outputs rapidly decreases. And so, the first few best outputs, in most cases, give a large number of detected faults. This makes them sufficient to achieve a high fault-coverage.

7.2.3 Selection of Best Multiplexer and Best Selection Lines

In order to keep the design simple, no more than 16 outputs of the LFSR are selected for use in the design to feed the scan chain input. After selecting the best outputs to serve as data inputs of the multiplexer, the next step is to search for best outputs of the LFSR to be used as selection lines for the multiplexer.

If we search for k bits to act as selection lines for a multiplexer from an n bit LFSR, then we have a number of different combinations given by equation (7.2).

$$Number\ of\ combinations = \binom{n}{k} = \frac{n!}{(n-k)!\times k!} \tag{7.2}$$

So, for example, if we have an LFSR with 16 bits and it is found that 8 of its outputs are sufficient to get a targeted fault-coverage with an acceptable test length, then the minimum size multiplexer to be used is 8x1. In this case we will search for the best 3 selection lines out of 16 bits; hence we have 560 different combinations. For this example, the method will construct 8 arrays, one for each LFSR output that will be connected with a multiplexer data line. The arrays contain the initial value of the LFSR at the scan cycle corresponding to each of the target faults that is detected by that output. The best three bits are selected by using the array that contains the LFSR initial values corresponding to the best output. Then for the first possible combination (e.g. bit1,bit2,bit3) it searches through all the stored initial values and counts how many different values this combination has (the minimum is 1 and maximum is 8). This process is then repeated for all possible different combinations (i.e. (bit1, bit2, bit4), (bit1, bit2, bit5)…(bit n-2, bit n-1, bit n)). The combination that has the minimum count of different values among the stored initial values of LFSR is considered as the best one to act as the selection lines of the multiplexer.

In general, if we have 8 outputs of the LFSR to be connected with the multiplexer data lines to feed the scan chain, then 3 selection lines are sufficient to detect the faults stored in all of these arrays if it is possible to find a set of three bits combination that has one fixed value for all of initial values of LFSR for the best output, a second fixed value for the second best output, and so on. Since in most scenarios this will not be possible, it is often better to use a larger multiplexer, in which each of the best 8 outputs of the LFSR can be connected with more than one data line, hence increasing the flexibility of the system. In the proposed design a decision was taken to use multiplexer sizes up to 64x1 in order to keep the area overhead relatively small, hence the search will be performed on up to six-bit combinations.

There are two problems to be solved. Firstly, the number of combinations that need to be examined may be infeasibly large. For example, for six-bit combinations and a 32 bit LFSR there are 906,192 different combinations. However, for an LFSR size of 64

the number of combinations increases to 74,974,368. This requires an excessive computation time. A decision was therefore taken to search for combinations of the first 32 bits only.

The second problem is the possibility of conflict between two or more different outputs for a specific value of the selection lines. This problem is solved by assigning the value of the selection lines to the output that detects more faults with respect to this value. In the case of excessive conflicts between outputs for a combination of selection lines, then a new search for a better combination is done.

7.2.4 Determination of the Final Fault Coverage for the New Design

The final step to be done is to apply the test vectors generated by the MO-LFSR design to the CUT to get the final fault coverage. This step is needed to check how many faults are not detected as a result of some conflicts between different outputs related to the value of selection lines.

Example 7.2: The aim of this example is to explain the algorithm step by step. This is not a real example, but it is used for the sake of illustration only. Assume we have a CUT with 100 faults (F_1, F_2, F_3,… ,F_{100}). A 6-bit LFSR (assume the cell names are A, B, C, D, E, and F) is used to generate test vectors to test the faults of this CUT. The following steps are the steps included in the algorithm described above to get high fault coverage:

Step 1: Identify the hard to detect faults among the 100 faults. This is obtained by simulating the CUT for 6 different seeds, and the faults detected in 3 or more case are not considered as hard to detect faults. The remaining faults are considered as hard to detect faults. For the sake of illustration, assume that faults F_1 to F_{10} are found to be hard to detect faults.

Step 2: We simulate the CUT for each LFSR output (A to F) to find which of the hard to detect faults (F=1 to F_{10}) will be detected by each one. Table 7.6 shows the faults detected by each output. It is clear that the output taken from cell D has the best fault among other outputs. Then we want to find the second best output among the

remaining outputs (i.e. outputs of cells A, B, C, E, and F). To find this, we have to delete the faults detected by cell D from the fault list detected by each output, table 7.7 shows the new list, the faults that already detected by cell D are shown in gray. In the same way, table 7.8 shows the third and the fourth output to participate in feeding the input of the scan-chain.

TABLE 7.6: FAULTS DETECTED BY EACH OUTPUT

A	**B**	**C**	**D**	**E**	**F**
F_1	F_2	F_2	F_2	F_4	F_3
F_4	F_5	F_3	F_4	F_5	F_8
F_6	F_8	F_4	F_5		F_{10}
		F_9	F_6		
			F_7		
			1st		

TABLE 7.7 SEARCHING FOR THE 2ND BEST OUTPUT

A	**B**	**C**	**D**	**E**	**F**
F_1	F_2	F_2	F_2	F_4	F_3
F_4	F_5	F_3	F_4	F_5	F_8
F_6	F_8	F_4	F_5		F_{10}
		F_9	F_6		
			F_7		
			1st		**2nd**

TABLE 7.8 FINDING THE BEST OUTPUTS TO DETECT ALL THE REQUIRED FAULTS

A	**B**	**C**	**D**	**E**	**F**
F_1	F_2	F_2	F_2	F_4	F_3
F_4	F_5	F_3	F_4	F_5	F_8
F_6	F_8	F_4	F_5		F_{10}
		F_9	F_6		
			F_7		
3ed		**4th**	**1st**		**2nd**

Step 3: The task now is to find the best size multiplexer and the best cells to act as selection lines for this multiplexer. Table 7.9 shows – for sake of illustration – the initial value inside the LFSR for the scan cycles where output D is used to feed the scan-chain.

TABLE 7.9 OUTPUT D FAULTS AND LFSR INITIAL VALUES

Output D						
Fault	**LFSR Initial Value**					
	A	**B**	**C**	**D**	**E**	**F**
F_2	1	0	1	0	1	0
F_4	0	0	1	0	1	1
F_5	1	1	1	1	1	1
F_6	1	1	0	1	0	0
F_7	0	0	0	0	1	0

Since we have 4 outputs that detect all faults, then the minimum size multiplexer that can be used is 4x1. Hence, 2 selection lines are required. But in order to connect each of the 4 outputs to the four data lines (i.e. each output will be connected to one data line only) then the selection line should be 00 for one output, 01 for the second, 10 for the third and 11 for the last. For example, for output D in table 7.9, if selection lines AB are used, the output D will be connected to 3 data lines of the multiplexer. This is because for F_2 detected by output D, the value of AB is 10 (i.e. data line # 2 will be reserved for output D), and for F_4, the value of AB is 00 (i.e. data line # 0 will be reserved for output D), and for F_5, the value of AB is 11 (i.e. data line # 3 will be reserved for output D), and for F_6, the value of AB is 11 (same as F_5). Finally, for F_7 the value of AB is 00 (same as F_4). Hence, 3 data lines are reserved for output D, and the remaining 3 outputs must fit into one data line, which is impossible.

Thus, it is better to use an 8x1 multiplexer. In this case, each output can be connected to more than one data line since we have 4 outputs and 8 data lines. Let us start with the best output (i.e. output D) to search for the best 3 cells to act as selection lines. Since we have 6 cells, and we have to search for a 3 cell combination, then we have 20 different combinations. For example, if we select the combination ABC, then we can see that in table 7.9 that this combination has the value 101 for F_2, 001 for F_4, 111 for F_5, 110 for F_6, and 000 for F_7, hence, output D will reserve 5 data lines of the 8x1 multiplexer. But we need the combination with the minimum number of values in order to keep more data lines for the other outputs. Table 7.10 shows for each of the 20 combinations, the number of data lines that will be reserved for output D. It is clear from table 7.10 that if we use the combination ABD or BDE then 3 data lines of the 8x1 multiplexer will be reserved for output D, hence it is recommended to use one of these combinations in order to keep 5 data lines for the other outputs.

TABLE 7.10 NUMBER OF DATA LINES THAT WILL BE RESERVED FOR OUTPUT D FOR DIFFERENT COMBINATIONS

Combinations	ABC	ABD	ABE	ABF	ACD
Diff. Values	5	3	4	5	5
Combinations	ACE	ACF	ADE	ADF	AEF
Diff. Values	4	5	4	5	5
Combinations	BCD	BCE	BCF	BDE	BDF
Diff. Values	4	4	5	3	4
Combinations	BEF	CDE	CDF	CEF	DEF
Diff. Values	5	4	5	4	4

Let us choose the combination ABD for selection lines, then output D will be connected to data line 0 (for F_4, F_7), line 4 (for F_2), and line 7 (for F_5, F_6). Finally, Table 7.11 shows the initial value of the LFSR for the faults detected by output F, output A, and output C. Since we have chosen the combination ABD to act as a selection line, the output F will be connected with data line 6 (for F_3), data line 5 (for F_{10}), and data line 0 (for F_8). Also output A will be connected with data line 1 (for F_1), and output C will be connected with data line 2 (for F_9). Note that we have a conflict at data line 0, since we want to connect it with output D and output F. This conflict can be solved in many ways. One approach is to leave a fault undetected and connect one output to this line. Another way is to search for another combination other than the lines ABD, for example we can use the lines BDE. Also we can use a bigger multiplexer (e.g. 16x1) in order to allow more data lines to be connected with each output and minimising the number of conflicts between different outputs.

TABLE 7.11 INITIAL VALUES OF LFSR FOR OUTPUTS F, A, AND C

Output F						
Fault	**LFSR Initial Value**					
	a	**b**	**c**	**d**	**e**	**f**
F_3	1	1	1	0	1	0
F_8	0	0	0	0	0	1
F_{10}	1	0	1	1	0	1
Output A						
F_1	0	1	0	1	1	0
Output C						
F_9	0	0	1	1	1	0

7.3 Experimental Results

The algorithm described in the previous section was used to design the MO-LFSR which contains an LFSR, a multiplexer, and extra D-FFs. The smaller the multiplexer size the lower the hardware area overhead, since the number of gates and the number of selection lines is reduced, and each selection line requires a D-FF to store its value

throughout the scan cycle. On the other hand, a larger multiplexer size gives more flexibility to achieve better fault coverage, since each selected output that feeds the scan chain can be connected to more data lines.

First, for a specified test length, some ISCAS'89 benchmark circuits have been simulated in order to identify the hard to detect faults. Then for each LFSR output, we identify a list from the hard faults that can be detected if this output is used to feed the scan chain. Then we search for the best outputs of the LFSR to feed the scan chain through a multiplexer and to search for the best outputs to act as selection lines of the multiplexer. Finally, The MO-LFSR is used to generate test patterns for full scan ISCAS'89 sequential circuits.

Table 7.12 shows the experimental results for some of the ISCAS'89 benchmark circuits. The first column states the name of the benchmark circuit used in the experiments. The second column is used to specify the scan chain size, where primary inputs and flip-flops in each benchmark circuit were configured to act as a full scan chain. The third column specifies the size of the LFSR used as the TPG. The fourth column shows how many outputs of the LFSR feed the scan chain input through a multiplexer so as to get a high fault-coverage. The fifth column states the size of the MUX used. Then the next 3 columns show for a target test length (TL_T), the effective fault coverage (EFC) obtained by using the conventional LFSR and the MO-LFSR, respectively. Alternatively, the last three columns show for a target EFC (EFC_T) the test length (TL) needed by a conventional LFSR and by the proposed MO-LFSR, respectively.

TABLE 7.12 EXPERIMENTAL RESULTS ON ISCAS'89 BENCHMARK CIRCUITS USING THE MO-LFSR

CKT	m	n	# of outputs	MUX Size	TL_T	EFC%		EFC_T%	TL	
						LFSR	MO-LFSR		LFSR	MO-LFSR
S420	35	64	5	16x1	512	86.74	98.37	98.37	8200	512
S641	54	32	7	32x1	256	81.86	97.41	97.41	2400	256
S713	54	32	6	32x1	256	84.53	97.42	97.42	2700	256
S820	23	32	9	32x1	1024	92.94	99.65	99.65	13600	1024
S953	45	32	12	32x1	1024	88.97	99.26	99.26	23600	1024
S1196	32	32	10	32x1	1024	88.00	99.52	99.52	22800	1024
S1238	32	32	11	32x1	1024	87.63	99.22	99.22	17300	1024
S1423	91	64	6	16x1	512	96.60	99.53	99.53	4700	512
S5378	214	64	13	64x1	2048	83.53	99.78	99.78	56400	2048
S9234	247	32	16	64x1	8192	75.08	95.75	95.75	172000	8192
S13207	700	32	8	64x1	4096	68.71	98.31	98.31	184000	4096
S15850	611	64	14	64x1	4096	74.51	96.04	96.04	161000	4096

From table 7.12 it is clear that the results of EFC and TL obtained by using the MO-LFSR are significantly better than those obtained by using a conventional LFSR. Another comparison between the MO-LFSR technique and an existing technique found in [128] is shown in table 7.13. Where the technique used in [128] is a reseeding technique for LFSR-based BIST application, in which multiple seeds are produced by the TPG itself to deal with hard-to-detect faults without using ROM to store the seeds. The first column in table 7.13 shows the circuit name. The second and third columns show the TL and EFC obtained by [128], while the next two columns show the TL and EFC of the MO-LFSR design. Finally, the last two columns show the improvements obtained by the MO-LFSR regarding TL and EFC. TL% column shows how shorter is the test length in the proposed design over the method in [128] and EFC% column shows how much higher is the effective fault coverage. As shown in the table, the proposed design achieves higher effective fault coverage for all the circuits while it uses a shorter test length for all of the circuits except S820.

TABLE 7.13 COMPARING MO-LFSR RESULTS WITH THOSE OBTAINED BY [128]

CKT	Results in [128]		Proposed Design		Improvement	
	TL	EFC%	TL	EFC%	TL%	EFC%
S420	2132	92.2	512	98.37	76	6.17
S641	1399	97.4	256	97.41	82	0.01
S713	1899	92.1	256	97.42	87	5.32
S820	978	91.9	1024	99.65	-5	7.75
S953	3218	99.0	1024	99.26	68	0.26
S1196	11762	99.2	1024	99.52	91	0.32
S1238	7146	94.5	1024	99.22	86	4.72
S1423	1244	98.3	512	99.53	59	1.23
S5378	9103	98.3	2048	99.78	78	1.48
S9234	12192	91.9	8192	95.75	33	3.85

7.4 Conclusion

This chapter has presented a new technique to achieve high fault coverage with a short test length for test-per-scan BIST. The technique is based on the use of more than one cell output of the LFSR to feed the scan chain using a multiplexer such that hard to detect faults can be detected with a short test sequence. Experimental results on ISCAS'89 benchmark circuits show that the proposed method is efficient in reducing test length and increasing fault coverage with a modest hardware area overhead. Comparison with conventional LFSR and another proposed technique shows that the MO-LFSR is much better for most benchmark circuits.

CHAPTER 8

Conclusions and Future Work

The increasing demand for portable electronic devices with long lifetime battery and reliable functionality has led to increased interest in low power design. However, many faults may arise in digital circuits either during fabrication or during operational lifetime. If these faults remain undetected, then there is no way to distinguish good chips from faulty chips. Hence, test is a necessary part of the manufacturing process. However, it is found that the power consumption during test is higher than during normal operation. Thus, it is very important to develop techniques for low power testing.

On the other hand, in order for testing to be a reliable and cost-efficient process, it should detect all or most of the faults (stuck-at-faults on this thesis) that may occur in digital circuits within an acceptable test length (and, therefore, test application time) and acceptable storage space. Hence, deterministic tests (which store test vectors in memory) are not the optimal solution since they need a huge storage space for large circuits. However, random TPGs are not good since there are some faults (known as random pattern resistant faults) that need a very long sequence of test vectors (i.e. an extremely long test application time). Thus, it is very important to produce techniques that compromise between hardware overhead, test application time, and the obtained fault coverage.

This thesis has focused on reducing the power consumption during test application, and increasing the fault coverage with a short test length.

8.1 Summary of Thesis Contribution and Main Conclusions

This thesis has concentrated on the improvement of many test parameters in order to improve quality and cost of testing of digital VLSI circuits. The main contributions of this thesis are summarised in the following subsections.

8.1.1 Average Power Minimisation in Test-per-Clock BIST using Low Transition LFSR

The first contribution of this thesis is reduction of the power consumption of test-per-clock BIST by using the BS-LFSR as a low transition TPG. Furthermore, the BS-LFSR design is based on new theoretical concepts derived from the properties of the LFSR and proved to be valid for any LFSR, regardless of the number of stages and the characteristic polynomial. The BS-LFSR can be used as a new TPG for test-per-clock BIST to achieve moderate results in power savings. The results obtained from the BS-LFSR can be further improved by combining it with other low power testing technique that uses the conventional LFSR, replacing the LFSR by the BS-LFSR. Also, the BS-LFSR can be extended to the rotational LFSR (RLFSR) for test-per-clock applications. In this way further power saving can be achieved. The main drawback of the test-per-clock BS-LFSR and RLFSR is that the hardware area overhead depends on their size: an extra multiplexer is needed for each D-FF cell used.

8.1.2 Average Power Minimisation in Test-per-Scan BIST using Low Transition LFSR

Another contribution of this thesis is the reduction of average power consumption in scan-based BIST. This is achieved by using the test-per-scan version of the BS-LFSR or RLFSR. It has been shown that the BS-LFSR can be configured in special configurations that reduce the number of transition inside the scan-chain by 50% with an area overhead of only one extra 2x1 multiplexer.

On the other hand, the RLFSR, of which the BS-LFSR is a special case, can be used to reduce the number of transitions in single scan-chain circuits as well as in the

multiple scan-chain circuits. Furthermore, using the RLFSR with multiple scan paths circuit can significantly reduce the correlation between the test vectors loaded into different paths, thus removing the need for the phase shifter that is required when a conventional LFSR is used to feed multiple scan chains.

The presented low transition LFSRs can replace the conventional LFSR in many low power methods in order to achieve further reduction to the power consumption during test.

8.1.3 Scan and Capture Peak Power Minimisation in Scan-Based BIST using BS-LFSR, and 2-Phase Scan-Chain Ordering Algorithm

A study has been presented about sources of peak power violations in test-per-scan BIST. It has been shown experimentally that in order to significantly reduce the overall peak power in scan-based BIST, three different components should be taken in consideration simultaneously.

Firstly, the peak power may arise as a result of scanning in the test vector because in some test vectors it could be that most of the consecutive bits have different values (i.e. there is a transition between them). This source of peak power violation is eliminated or reduced by using the BS-LFSR for test-per-scan BIST which reduces the number of transitions by 50% in a test vector.

The second source of peak power violation is while scanning out a captured response. This will happen if in the captured response, most of the consecutive cells in the scan-chain have different values (i.e. transition will occur with each scan shift). This problem has been solved using a cell ordering algorithm that aims to connect the cells which will probably have similar values in the captured response with each other. In this way this source of peak power violation will be significantly reduced.

The peak power that may arise in the test cycle (called capture power), is the third source of peak power violations in scan-based BIST. This happens when the applied

test vector (which exists inside the scan-chain cells at the beginning of the test cycle) and the captured response (which exists inside the scan-chain cells at the end of the test cycle) have a large Hamming distance. This problem is solved by a second phase of scan cell ordering that uses one of the properties of the BS-LFSR that every pair of consecutive cells will probably have the same value in the scanned test vector in most of the cases since the BS-LFSR is a low transition TPG. Then if a cell in the scan chain will probably have the same value during the test cycle (i.e. its value in the applied vector is the same as in the captured response) if two other cells have the same value in the applied test vector, then the last two cells are connected with each other in the scan-chain. This will reduce the capture peak power.

Using these three algorithms together in one design will give substantial reductions in the overall peak power in the CUT.

8.1.4 Increasing Fault Coverage in Scan-Based BIST using a Multi-Output LFSR

Normally, when random TPGs generate and apply test vectors, the fault coverage rises at first, and then levels off. Application of further test vectors then provides little further increase in fault coverage. This is because the CUT has faults called random pattern resistant faults (or hard to detect faults). In order to achieve a perfect (or near-perfect) fault coverage with these TPGs such as the LFSR, the test length will be extremely high (in some cases it will be in term of years).

A final contribution of this thesis is to introduce a new technique to achieve a high fault-coverage within an acceptable test length. This technique is based on using more than one cell of the LFSR to feed the scan-chain input in each scan cycle. The cell that detects some of the hard to detect faults will be chosen at the beginning of the scan cycle. This technique significantly improves the fault coverage with a low-to-moderate hardware area overhead.

8.2 Suggestions for Future Work

While this dissertation has studied some properties of the LFSR and developed some techniques for low power testing and high fault coverage, there are several points that can be investigated further in order to develop other techniques that are an improvement on existing techniques. In this section, a number of future research directions in the field of testing will be briefly described.

8.2.1 Investigation of LFSRS' Properties

Since many of the techniques presented in this thesis are based on new properties and observations about the behaviour of LFSRs, searching for other properties and observations may help in developing new techniques that improve the fault coverage, power consumption and any other parameters related to test. Furthermore, since the LFSRs have many applications other than testing of digital VLSI circuits, many of the properties may be useful in these applications such as cryptography.

8.2.2 Low Power Delay Test

All of the techniques described in this thesis are based on the stuck-at fault model, which is the most-widely used fault model. However, delay testing, which is based on a delay fault model, is becoming increasingly important in the test flow. Delay testing is used to verify that a circuit meets its timing specification. While several low power approaches have been proposed for stuck-at fault model tests, fewer techniques of low power delay testing have been proposed despite its importance in today's systems. Thus, further techniques and methodologies to address this problem should be considered in future research.

8.2.3 System-on-a-Chip (SOC) Test

Recent developments in semiconductor technology and design techniques make it possible to integrate millions of transistors in a single chip to form system-on-a-chip (SOC). However, problems related to SOC test still existed. These problems include test time, test storage, and test power. The test time problem and test power problem

are highly related to each other. In SOC testing, each core is tested independently and the interconnect wires between the cores are tested as well. If several cores are tested serially, then this will reduce the average power consumption but will increase the test time. If all or some of the cores are tested concurrently, then this will decrease the test time but will increase the average power consumption. Therefore, test power consumption restricts the number of cores that can be tested concurrently. (There are additional factors that restrict the number of cores that can be tested concurrently, but they are not the concern here). Thus, if power consumption in SOC can be reduced, a larger number of cores can be tested concurrently thereby reducing test time.

8.2.4 Random Access Memory (RAM) Test

Despite the fact that there are many proposed techniques to minimise power consumption during test, there appear to be 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 [154]. 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.

Appendix A

Maximal-Length LFSR
Primitive Polynomial/Tap Sequence

A maximal length LFSR can be either represented by the primitive polynomial it implements, or by the location of the tap sequences. For example, the primitive polynomial in equation (A.1)

$$P(x) = x^8+x^6+x^5+x+1 \qquad (A.1)$$

can be implemented by an 8-bit external-XOR LFSR as shown in Fig. A.1, or internal-XOR LFSR as shown in Fig. A.2. Alternatively, the maximal length 8-bit LFSR can be denoted by numbers that indicate the tap sequence (or the locations of feedback). Thus, the 8-bit LFSR can be represented in terms of tap sequence as (8 6 5 1).

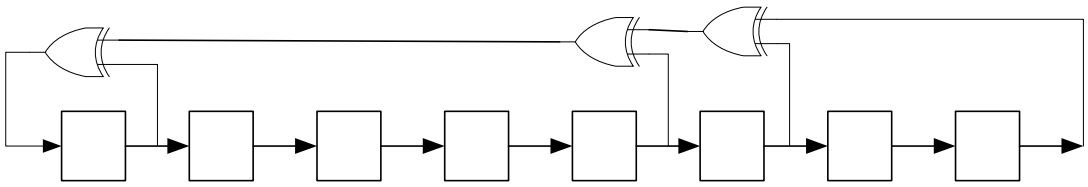

Fig. A.1 8-bit External LFSR that implements primitive polynomial $P(x) = x^8+x^6+x^5+x+1$

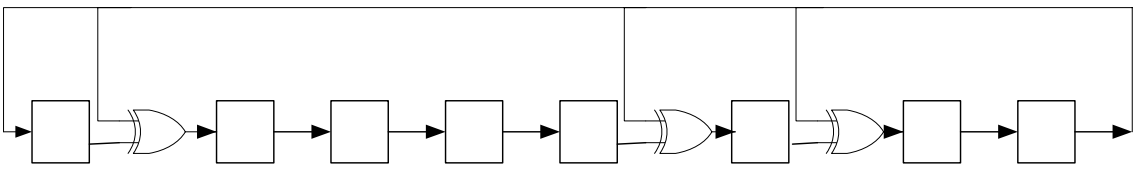

Fig. A.2 8-bit Internal LFSR that implements primitive polynomial $P(x) = x^8+x^6+x^5+x+1$

It is important to note that the primitive polynomials for a degree are not unique, hence, for any number of required stages there are many different LFSRs that can be used to generate the maximal length sequence (also called m-sequence). Table A.1 shows for the LFSR stages from 2 to 64 the tap sequences (up to 3 choices different

tap sequences) for a maximal length LFSR [7, 152, 153]. Note that table A.1 states more than one LFSR tap sequence for maximal length LFSR, one using 2 tap sequences (if any), and one or two with 4 tap sequences.

Another important note is that if, for a specific degree, there is a known primitive polynomial, then it is easy to directly find another one which is called the reciprocal polynomial. For example, if $P_1(x) = x^8+x^6+x^5+x+1$ is a primitive polynomial, then $P_2(x) = x^8 + x^2+x^3+x^7+1$ is also primitive polynomial.

In general, if $P_1(x) = x^n+ x^{d1}+x^{d2}+x^{d3}+1$ is a primitive polynomial of degree n, then its reciprocal primitive polynomial is $P_2(x) = x^n+x^{n-d1}+x^{n-d2}+x^{n-d3}+1$. Thus, in table A.1 it is possible to find more choices for tap sequences by using the reciprocal primitive polynomials.

Table A.1 Different tap sequences for maximal-length LFSR

n	*Tap Sequence*									
	Choice 1 (2 taps)		*Choice 2 (4 taps)*				*Choice 3 (4 taps)*			
2	2	1								
3	3	2								
4	4	3								
5	5	3	5	4	3	2				
6	6	5	6	5	3	2				
7	7	6	7	6	5	4				
8			8	6	5	4				
9	9	5	9	8	6	5				
10	10	7	10	9	7	6				
11	11	9	11	10	9	7				
12			12	6	4	1				
13			13	4	3	1				
14			14	5	3	1				
15	15	14	15	14	13	11				
16			16	15	13	4	16	14	13	11
17	17	14	17	16	15	14				
18	18	11	18	17	16	13				
19			19	6	2	1	19	18	17	14
20	20	17	20	19	16	14				
21	21	19	21	20	19	16				
22	22	21	22	19	18	17				
23	23	18	23	22	20	18				
24			24	23	22	17	24	23	21	20
25	25	22	25	24	23	22				
26			26	6	2	1				
27			27	5	2	1	27	26	25	22
28	28	25	28	27	24	22				
29	29	27	29	28	27	25				
30			30	6	4	1				
31	31	28	31	30	29	28				

32			32	22	2	1	32	30	26	25
33	33	20	33	32	29	27				
34			34	27	2	1	34	31	30	26
35	35	33	35	34	28	27				
36	36	25	36	35	29	28				
37			37	36	33	31				
38			38	6	5	1				
39	39	35	39	38	35	32				
40			40	38	21	19	40	37	36	35
41	41	38	41	40	39	38				
42			42	41	20	19	42	40	37	35
43			43	42	38	37				
44			44	43	18	17	44	42	39	38
45			45	44	42	41				
46			46	45	26	25	46	40	39	38
47	47	42	47	46	43	42				
48			48	47	21	20	48	44	41	39
49	49	40	49	45	44	43				
50			50	49	24	23	50	48	47	46
51			51	50	36	35	51	50	48	45
52	52	49	52	51	49	46				
53			53	52	38	37	53	52	51	47
54			54	53	18	17	54	51	48	46
55	55	31	55	54	53	49				
56			56	55	35	34	56	54	52	49
57	57	50	57	55	54	52				
58	58	39	58	57	53	52				
59			59	58	38	37	59	57	55	52
60	60	59	60	58	56	55				
61			61	60	46	45	61	60	59	56
62			62	61	6	5	62	59	57	56
63	63	62	63	62	59	58				
64			64	63	61	60				

References

1. M. Bushnell and V. Agrawal, *Essentials of Electronic Testing for Digital Memory, and Mixed Signal VLSI Circuits*, Kluwer Academic Publishers, 2002.
2. W.M. Needham, "Nanometer technology challenges for test and test equipment", *Computer*, **32**(11), pp. 52-57, November 1999.
3. M. Abramovici, M.A. Breuer, and A.D. Friedman, *Digital Systems Testing and Testable Design*, IEEE Press, 1990.
4. V. Agrawal, C.R. Kime, and K. Saluja, "A tutorial on built-in self-test - part 1: principles", *IEEE Design and Test of Computers,* **10**(1), pp. 73-82, March 1993.
5. V. Agrawal, C.R. Kime, and K.K. Saluja, "A tutorial on built-in self test - part 2: applications", *IEEE Design and Test of Computers*, **10**(1), pp. 69-77, June 1993.
6. V. Agrawal, C.-J. Lin, P.W. Rutkowski, S. Wu, and Y. Zorian, "Built-in self-test for digital integrated circuits", *AT&T Technical Journal*, pp. 30-39, March 1994.
7. P.H. Bardell, W.H. McAnney, and J. Savir, *Built-in test for VLSI: pseudorandom techniques*, John Wiley & Sons, 1987.
8. J. Rajski, N. Tamaparalli, and J. Tyszer, "Automated synthesis of phase shifters for built-in self-test applications", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **19**(10), 2000.
9. R. Aitken, "Nanometer technology effects on fault models for IC testing", *Computer*, **32**(11), pp. 46-51, November 1999.
10. R. Garcia, "Rethink fault models for submicron-IC test", *Test and Measurement World*, **21**(12), pp. 35-40, October 2001.
11. H.K. Lee and D.S. Ha, "An efficient forward fault simulation algorithm based on the parallel pattern single fault propagation", *Proceedings of the 1991 International Test Conference*, pp. 946-955, October 1991.
12. H.K. Lee and D.S. Ha, "On the generation of test patterns for combinational circuits", Virginia Polytechnic Institute and State University, 1993
13. P.H. Bardell and W.H. McAnney, "Pseudorandom arrays for built-in test", *IEEE Transactions on Computers*, **C-35**(7), pp. 653-658, July 1986.
14. P.H. Bardell, "Design consideration for parallel pseudorandom pattern generator", *Journal of Electronic Testing: Theory and Application (JETTA)*, **1**(1), pp. 73-87, 1990.
15. A.P. Chandrakasan, S. Sheng, and R.W. Brodersen, "Low power CMOS digital design", *IEEE Journal of Solid-State Circuits* **27**(4), pp. 473-484, April 1992.
16. 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.
17. N. Nicolici and X. Wen, "Embedded tutorial on low power test", *IEEE European Test Symposium*, pp. 202-210, May 2007.
18. W.C. Athas, L.J. Svensson, J.G. Koller, N. Tzartzanis, and E. Ying-Chin Chou, "Low power digital systems based on adiabatic-switching principles", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **2**(4), pp. 398-416, December 1994.

19. P. Girard, "Survey of low-power testing of VLSI circuits", *IEEE Design and Test of Computers*, **19**(3), pp. 80-90, June 2002.
20. 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.
21. C.Y. Wang and K. Roy, "Maximum power estimation for CMOS circuits using deterministic and statistical approaches", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **6**(1), pp. 134-140, March 1998.
22. 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.
23. 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.
24. Y. Zorian, "A distributed BIST control scheme for complex VLSI devices", *Proceedings of 11th IEEE VLSI Test Symposium (VTS 93)*, pp. 4-9, April 1993.
25. N.H. Weste and K. Eshraghian, *Principles of CMOS VLSI design: A system Perspective*. 2nd ed, Reading, MA, Addison-Wesley, 1993.
26. S. Wang and S. Gupta, "DS-LFSR: A BIST TPG for low switching activity", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **21**(7), pp. 842-851, July 2002.
27. J. Altet and A. Rubio, *Thermal testing of integrated circuits*, Boston, Springer, 2002.
28. A. Hertwig and H.J. Wunderlich, "Low power serial built-in self-test", *IEEE European Test Workshop*, pp. 49-53, May 1998.
29. B. Pouya and A. Crouch, "Optimization trade-offs for vector volume and test power", *Proceedings of International Test Conference*, pp. 873-881, October 2000.
30. K. Roy and S. Prasad, *Low-Power CMOS VLSI Circuit Design*, John Wiley & Sons, 2000.
31. P. Rosinger, "Power Conscious Scan-Based Test of Digital VLSI Circuits", in *Electronics and Computer Science*, University of Southampton, Southampton. pp. 182, February 2003
32. J. Rajski and J. Tyszer, *Arithmetic Built-In Self-Test for Embedded Systems*, Englewood Cliffs, NJ, Prentice-Hall, 1998.
33. J. Saxena, K.M. Butler, V.B. Jayaram, S. Kundu, N.V. Arvind, P. Sreeprakash, and M. Hachinger, "A case study of IR-drop in structured at speed testing", *IEEE International Test Conference*, pp. 1098-1104, September 2003.
34. N. Nicolici and B.M. Al-Hashimi, *Power-Constrained Testing of VLSI Circuits*, Boston, Springer, 2003.
35. A.S. Abu-Issa and S.F. Quigley, "LT-PRPG: Power minimization technique for test-per-scan BIST", *IEEE International Conference in Design and Technology of Integrated Systems (DTIS'08)*, pp. 1-5, March 2008.
36. A.S. Abu-Issa and S.F. Quigley, "Bit-swapping LFSR for low-power BIST", *IET Electronics Letters*, **44**(6), pp. 401-402, March 2008.
37. A.S. Abu-Issa and S.F. Quigley, "A multi-output technique for high fault coverage in test-per-scan BIST", *IEEE International Conference in Design and Technology of Integrated Systems*, pp. 1-6, March 2008.
38. A.S. Abu-Issa and S.F. Quigley, "Bit-swapping LFSR and scan-chain ordering: A novel technique for peak - and average - power reduction in scan-

based BIST", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **28**(5), pp. 5, May 2009.

39. J. Monzel, S. Chakravarty, V. Agrawal, R. Aitken, J. Braden, J. Figueras, S. Kumar, H. Wunderlich, and Y. Zorian, "Power dissipation during testing: Should we worry about it?" *15th IEEE VLSI Test Symposium*, pp. Panel Session, April 1997.
40. J. Saxena, K.M. Butler, and L. Whetsel, "An Analysis of power reduction techniques in scan testing", *Proceedings of International Test Conference*, pp. 670-677, October 2001.
41. S. Wang, "A BIST TPG for low power dissipation and high fault coverage", *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, **15**(7), pp. 777-789, July 2007.
42. S. Wang and S. Gupta, "LT-RTPG: A new test-per-scan BIST TPG for low switching activity", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **25**(8), pp. 1565-1574, August 2006.
43. H. Ronghui, L. Xiaowei, and G. Yunzhan, "A low power BIST TPG design", *Proceedings of 5th International Conference on ASIC*, pp. 1136-1139, October 2003.
44. L. Jie, Y. Jun, L. Rui, and W. Chao, "A New BIST structure for low power testing", *Proceedings 5th International Conference on ASIC*, pp. 1183-1185, October 2003.
45. M. Tehranipoor, M. Nourani, and N. Ahmed, "Low transition LFSR for BIST-based applications", *Proceedings of the 14th Asian Test Symposium (ATS'05)*, pp. 138-143, December 2005.
46. P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and H. Wunderlich, "A modified clock scheme for a low power BIST test pattern generator", *Proceedings of VLSI Test Symposium*, pp. 306-311, May 2001.
47. X. Zhang, K. Roy, and S. Bhawmik, "POWERTEST: A tool for energy conscious weighted random pattern testing ", *Proceedings of International Conference on VLSI Design*, pp. 416-422, January 1999.
48. Y. Kim, M.-H. Yang, Y. Lee, and S. Kang, "A new low power test pattern generator using a transition monitoring windows based on BIST architecture", *Asian Test Symposium*, pp. 230-235, December 2005.
49. F. Corno, M. Rebaudengo, M. Reorda, G. Squillero, and M. Violante, "Low power BIST via non-linear hybrid cellular automata", *Proceedings of IEEE VLSI Test Symposium*, pp. 29-34, May 2000.
50. N. Lai, S. Wang, and Y. Fu, "Low power BIST with a smoother and scan-chain reorder under optimal cluster size", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **25**(11), pp. 2586-2594, November 2006.
51. S. Chakravarty and V. Dabholkar, "Two techniques for minimizing power dissipation in scan circuits during test application", *Proceedings of Asian Test Symposium*, pp. 324-329, November 1994.
52. P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac, "Reducing power consumption during test application by test vector ordering", *Proceedings of IEEE International Symposium on Circuits and Systems*, pp. 296-299, June 1998.
53. P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A test vector ordering technique for switching activity reduction during test operation", *Proceedings of Great Lakes Symposium on VLSI*, pp. 24-27, March 1999.

54. X. Kavousianos, D. Bakalis, M. Bellos, and D. Nikolos, "An efficient test vector ordering method for low power testing", *Proceedings of IEEE Computer Society Annual Symposium on VLSI*, pp. 285-288, February 2004.
55. W. Tseng and L.-J. Lee, "Reduction of power dissipation during scan testing by test vector ordering", *International Workshop on Microprocessor Test and Verification*, pp. 15-21, December 2007.
56. P. Flores, J. Costa, H. Neto, J. Monteiro, and J. Marques-Silve, "Assignment and reordering of incompletely specified patterns sequences targeting minimum power dissipation", *Proceedings of International Conference on VLSI Design*, pp. 37-41, January 1999.
57. V. Dabholkar, S. Chakravarty, I. Pomeranz, and S.M. Reddy, "Techniques for minimizing power dissipation in scan and combinational circuits during test applications", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **17**(12), pp. 1325-1333, December 1998.
58. Y. Bonhomme, P. Girard, C. Landrault, and S. Pravossoudovitch, "Power driven chaining of flip-flops in scan architectures", *International Test Conference*, pp. 796-803, October 2002.
59. Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, and V. Virazel, "Design of routing-constrained low power scan chains", *Proceedings of the Design, Automation, and Test in Europe*, pp. 62-67, February 2004.
60. W. Tseng, "Scan chain ordering techniques for switching activity reduction during scan test", *IEE Proceedings in Computer and Digital Techniques*, **152**(5), pp. 609-617, September 2005.
61. C. Giri, B. Kumar, and S. Chattopadhyay, "Scan flip-flop ordering with delay and power minimization during testing", *Annual IEEE INDICON*, pp. 467-471, December 2005.
62. M. Bellos, D. Bakalis, and D. Nikolos, "Scan cell ordering for low power BIST", *IEEE Computer Society Annual Symposium on VLSI* pp. 281-284, February 2004.
63. K.V.A. Reddy and S. Chattopadhyay, "An efficient algorithm to reduce test power consumption by scan cell and scan vector reordering", *Proceedings of the IEEE INDICON 2004*, pp. 373-376, December 2004.
64. N. Badereddine, P. Girard, S. Pravossoudovitch, A. Virazel, and C. Landrault, "Scan cell reordering for peak power reduction during test cycles", *IFIP International Federation for Information Processing, Springer Boston*, pp. 267-281, October 2007.
65. S. Ghosh, S. Basu, and N. Touba, "Joint minimization of power and area in scan testing by scan cell reordering", *IEEE Computer Society Annual Symposium on VLSI*, pp. 246-249, February 2003.
66. X.L. Huang and J.I. Huang, "A routability constrained scan chain ordering technique for test power reduction", *Asian South Pacific Conference on Design Automation*, pp. 1-5, January 2006.
67. P. Wohl, J. Waicukauski, S. Patel, and M. Amin. "Efficient compression and application of deterministic patterns in logic BIST architecture", *Proceedings of Design and Automation Conference*, pp. 566 - 569, June 2003.
68. T. Hiraide, K. Boateng, H. Konishi, K. Itaya, M. Emori, H. Yamanaka, and T. Mochiyama, "BIST-aided scan test: A new method for test cost reduction", *VLSI Test Symposium*, pp. 359-364, May 2003.

69. K.M. Butler, J. Saxena, T. Fryars, G. Hetherington, A. Jain, and J. Lewis, "Minimizing power consumption in scan testing: Pattern generation and DFT techniques", *Proceedings of International Test Conference*, pp. 355-364, 2004.
70. T.K. Maiti and S. Chattopadhyay, "Don't care filling for power minimization in VLSI circuit testing", *IEEE International Symposium on Circuits and Systems*, pp. 2637-2640, May 2008.
71. A. Chandra and R. Kapur, "Bounded adjacent fill for low capture power scan testing", *IEEE VLSI Test Symposium*, pp. 131-138, April 2008.
72. J. Li, Q. Xu, Y. Hu, and X. Li, "iFill: An impact-oriented X-filling method for shift and capture power reduction in at speed scan-based testing ", *Design, Automation and Test in Europe*, pp. 1184-1189, March 2008.
73. S. Wang and W. Wei, "A technique to reduce peak current and average power dissipation in scan designs by limited capture", *Asia and South Pacific Design Automation Conference*, pp. 810-816, January 2007.
74. S. Remersaro, X. Lin, Z. Zhang, S.M. Reddy, I. Pomeranz, and J. Rajski, "Preferred fill: A scalable method to reduce capture power for scan based designs", *IEEE International Test Conference*, pp. 1-10, October 2006.
75. X. Wen, Y. Yamashita, S. Kajihara, L. Wang, K. Saluja, and K. Kinoshita, "On low-capture-power test generation for scan testing", *Proceedings of 23rd IEEE VLSI Test Symposium*, pp. 265-270, May 2005.
76. X. Wen, Y. Yamashita, S. Morishima, S. Kajihara, L.-T. Wang, K. Saluja, and K. Kinoshita, "Low-capture-power test generation for scan-based at-speed testing", *Proceedings of IEEE International Test Conference*, pp. 1-10, November 2005.
77. X. Wen, K. Miyase, T. Suzuki, Y. Yamato, S. Kajihara, Y. Ohsumi, and K. Saluja, "A highly-guided X-filling method for effective low-capture-power scan test generation", *International Conference on Computer Design*, pp. 251-258, October 2006.
78. 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", *Design and Test of Integrated Systems in Nanoscale Technology*, pp. 359-364, 2006.
79. F. Corno, M. Rebaudengo, M. Sonza Reorda, and M. Violante, "A new BIST architecture for low power circuits", *Proceedings of European Test Workshop*, pp. 160-164, May 1999.
80. S. Gerstendorfer and H. Wunderlich, "Minimizing power consumption for scan-based BIST", *Proceedings of International Test Conference*, pp. 77-84, September 1999.
81. S. Manich, A. Gabarro, M. Lopez, J. Figueras, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, P. Teixeira, and M. Santos, "Low power BIST by filtering non-detecting vectors", *Journal of Electronic Testing: Theory and Application (JETTA)*, **16**(3), pp. 193-202, June 2000.
82. T.-C. Huang and K.-J. Lee, "Reduction of power consumption in scan-based circuits during test application by an input control technique", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **20**(7), pp. 911-917, July 2001.
83. N. Nicolici, B.M. Al-Hashimi, and A.C. Williams, "Minimisation of power dissipation during test application in full-scan sequential circuits using primary input freezing", *IEE Proceedings in Computer and Digital Techniques*, **147**(5), pp. 313-322, September 2000.

84. N.-C. Lai and S.-Y. Wang, "Low-capture-power test generation by specifying a minimum set of controlling inputs", *Asian Test Symposium*, pp. 413-418, October 2007.
85. P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, J. Figueras, S. Manich, P. Teixeira, and M. Santos, "Low energy BIST design: Impact of the LFSR TPG parameters on the weighted switching activity", *Proceedings of International Symposium on Circuits and Systems*, pp. 110-113, June 1999.
86. M. Brazzarola and F. Fummi, "Power characterization of LFSRs", *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 139-147, November 1999.
87. P.-H. Wu, T.-T. Chen, W.-L. Li, and J.-C. Rau, "An efficient test-data compaction for low power VLSI testing", *IEEE International Conference on Electro/Information Technology*, pp. 237-241, May 2008.
88. J.-Y. Lee, Y. Hu, R. Majumdar, and L. He, "Simultaneous test pattern compaction, ordering and X-filling for testing power reduction", *Quality of Electronic Design*, pp. 702-707, March 2009.
89. L. Whetsel, "Adapting scan architectures for low power operation", *Proceedings of International Test Conference*, pp. 863-872, October 2000
90. P. Rosinger, B.M. Al-Hashimi, and N. Nicolici, "Scan architecture with mutually exclusive scan segment activation for shift and capture-power reduction", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **23**(7), pp. 1142-1153, July 2004.
91. O. Sinanoglu, I. Bayraktaroglu, and A. Orailoglu, "Dynamic test data transformations for average and peak power reductions", *Proceedings of European Test Symposium*, pp. 113-118, May 2002.
92. K.-J. Lee, T.-C. Huang, and J.-J. Chen, "Peak power reduction for multiple scan circuits during test application", *IEEE Asian Test Symposium*, pp. 453-458, December 2000.
93. T.-C. Huang and K.-J. Lee, "A token scan architecture for low power testing", *Proceedings of International Test Conference*, pp. 660-669, October 2001.
94. N. Nicolici and B.M. Al-Hashimi, "Multiple scan chains for power minimization during test application in sequential circuits", *IEEE Transactions on Computers*, **51**(6), pp. 721-734, June 2002.
95. D. Ghosh, S. Bhunia, and K. Roy, "Multiple scan chain design technique for power reduction during test application in BIST", *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pp. 191-198, November 2003.
96. K.Y. Cho, S. Mitra, and E. McCluskey, "California scan architecture for high quality and low power testing", *IEEE International Test Conference*, pp. 1-10, October 2007.
97. Y. Bonhomme, P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch, "A gated clock scheme for low power scan testing of logic ICs or embedded cores", *Proceedings of Asian Test Symposium*, pp. 253-258, November 2001.
98. R. Sankaralingam and N. Touba, "Multi-phase shifting to reducing instantaneous peak power during scan", *Proceedings of Latin American Test Workshop*, pp. 78-83, February 2003.
99. J. Zhang, T. Zhang, and Q. Zuo, "Multi-phase clock scan technique for low test power", *International Symposium on High Density Packaging and Microsystem Integration*, pp. 1-5, June 2007.

100. L.-T. Wang, C.-W. Wu, and X. Wen, *VLSI Test Principles and Architectures: Design for Testability*, San Francisco, Morgan Kaufmann, 2006.
101. A. Chandra and K. Chakrabarty, "Combining low-power scan testing and test data compression for system on-a-chip", *Proceedings of Design Automation Conference*, pp. 166-169, June 2001.
102. A. Chandra and K. Chakrabarty, "Reduction of SOC test data volume, scan power and testing time using alternating run-length codes", *Proceedings of Design Automation Conference*, pp. 673-678, June 2002.
103. C. Giri and S. Chattopadhyay, "Power optimized dictionary coding for test data compression", *IEEE International Conference on Industrial Technology*, pp. 2541-2545, December 2006.
104. Y. Han, Y. Xu, and X. Li, "Co-optimization for test data compression and testing power based on variable-tail code", *Proceedings of International Conference on ASIC*, pp. 105-108, October 2003.
105. Y. shi, S. Kimura, N. Togawa, M. Yanagisawa, and T. Ohysuki, "Alternative run-length coding through scan chain reconfiguration for joint minimization of test data volume and power consumption in scan test", *Asian Test Symposium*, pp. 432-437, November 2004.
106. B. Konemann, "LFSR-coded test patterns for scan-designs", *IEEE European Test Conference*, pp. 237-242, 1991.
107. P. Rosinger, B.M. Al-Hashimi, and N. Nicolici, "Low power mixed-mode BIST based on mask pattern generation using dual LFSR-reseeding", *Proceedings of International Conference on Computer Design*, pp. 474-479, 2002.
108. J. Lee and N. Touba, "Low power test data compression based on LFSR reseeding", *Proceedings of International Conference on Computer Design*, pp. 180-185, October 2004.
109. I. Hamzaoglu and J. Patel, "Reducing test application time for full scan embedded cores", *Proceedings of Design Automation Conference*, pp. 260-267, June 1999.
110. A. Al-Yamani, E. Chmelar, and M. Grinchuck, "Segmented addressable scan architecture", *Proceedings of VLSI Test Symposium*, pp. 405-411, May 2005.
111. S. Wang and S. Gupta, "ATPG for heat dissipation minimization during test application", *IEEE Transactions on Computers*, **47**(2), pp. 256-262, 1998.
112. F. Corno, P. Prinetto, M. Rebaudengo, and M. Sonza Reorda, "A test pattern generator methodology for low power consumption", *IEEE VLSI Test Symposium*, pp. 453-457, April 1998.
113. X. Wen, S. Kajihara, K. Miyase, T. Suzuki, K.K. Saluja, L.-T. Wang, K.S. Abdelhafez, and K. Kinoshita, "A new ATPG method for efficient capture power reduction during scan testing ", *IEEE VLSI Test Symposium*, pp. 1-6, May 2006.
114. S. Wang and S. Gupta, "An automatic test pattern generator for minimizing switching activity during scan testing activity", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **21**(8), pp. 954-968, August 2002.
115. R. Dandapani, J. Patel, and J. Abraham, "Design of test pattern generator for built-in test", *Proceedings of International Test Conference*, pp. 315-319, October 1984.

116. C. Dufaza, H. Viallon, and C. Chevalier, "BIST hardware generator for mixed test scheme", *Proceedings of European Design and Test Conference*, pp. 424-430, 1995.
117. L. Li and Y. Min, "An efficient BIST design using LFSR-ROM architecture", *Proceedings of the Ninth Asian Test Symposium*, pp. 386-390, December 2000.
118. D. Das and N. Touba, "Reducing test data volume using external/LBIST hybrid test patterns", *International Test Conference*, pp. 115-122, October 2000.
119. V. Agrawal and E. Cerny, "Store and generate built-in testing approach", *Proceedings of Fault Tolerant Computing Symposium*, pp. 35-40, June 1981.
120. G. Edirisooriya and J.P. Robinson, "Design of low cost ROM based test generators", *Proceedings of VLSI Test Symposium*, pp. 61-66, April 1992.
121. M.E. Aboulhamid and E. Cerny, "A class of test generators for built-in testing", *IEEE Transactions on Computers*, **C-32**(10), pp. 957-959, October 1983.
122. A. Al-Yamani, S. Mitra, and E. McCluskey, "Optimized reseeding by seed ordering and encoding", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **24**(2), pp. 264-270, February 2005.
123. G. Jervan, E. Orasson, H. Kruus, and R. Ubar, "Hybrid BIST optimization using reseeding and test set compaction", *10th Euromicro Conference on Digital System Design Architecture, Methods, and Tools*, pp. 596-603, August 2007.
124. S. Neophytou, M. Michael, and S. Tragoudas, "Efficient deterministic test generation for BIST schemes with LFSR reseeding", *12th IEEE International On-Line Testing Symposium*, pp. 1-6, July 2006.
125. A. Al-Yamani, S. Mitra, and E. McCluskey, "BIST reseeding with very few seeds", *Proceedings of 21st VLSI Test Symposium*, pp. 69-74, May 2003.
126. E. Kalligeros, X. Kavousianos, D. Bakalis, and D. Nikolos, "An efficient seeds selection method for LFSR-based test-per-clock BIST", *Proceedings International Symposium on Quality Electronic Design*, pp. 261-266, March 2002.
127. Y. Yan, H. Wang, Z. Yang, and S. Yang, "A new LFSR reseeding method for BIST", *8th International Conference on Solid-State and Integrated Circuits Technology*, pp. 2145-2147, 2006.
128. N. Lai and S. Wang, "A reseeding technique for LFSR-based BIST applications", *Proceedings of the 11th Asian Test Symposium*, pp. 200-205, November 2002.
129. C. Krishna and N. Touba, "Reducing test data volume using LFSR reseeding with seed compression", *Proceedings of International Test Conference*, pp. 321-330, October 2002.
130. S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois, "Generation of vector patterns through reseeding of multiple-polynomial linear feedback shift registers", *IEEE Transactions on Computers*, **19**(3), pp. 223-233, February 1995.
131. S. Hellebrand, B. Reeb, S. Tarnick, and H.-J. Wunderlich, "Pattern generation for a deterministic BIST scheme ", *Proceedings of International Conference on Computer-Aided Design*, pp. 88-94, November 1995.

132. J. Rajski, J. Tyszer, and N. Zacharia, "Test data compression for multiple scan designs with boundary scan", *IEEE Transactions on Computers*, **47**(11), pp. 1188-1200, November 1998.
133. H.-J. Liang, S. Hellebrand, and H.-J. Wunderlich, "Two-dimensional test data compression for scan-based deterministic BIST", *Proceedings of International Test Conference*, pp. 894-902, September 2001.
134. C. Yu, S. Reddy, and I. Pomeranz, "Weighted pseudo-random BIST for n-detection of single stuck at faults", *13th Asian Test Symposium*, pp. 178-183, November 2004.
135. D. Neebel and C. Kime, "Cellular automata for weighted random pattern generation", *IEEE Transactions on Computers*, **46**(11), pp. 1219-1229, November 1997.
136. A. Jas, C. Krishna, and N. Touba, "Weighted pseudorandom hybrid BIST", **12**(12), pp. 1277-1283, December 2004.
137. M. AlShaibi and C. Kime, "Fixed-biased pseudorandom built-in self-test for random pattern resistant circuits", *Proceedings International Test Conference*, pp. 929-938, October 1994.
138. S. Wang, K. Balakrishnan, and S. Chakradhar, "XWRC: externally-loaded weighted random pattern testing for input test data compression", *IEEE International Test Conference*, pp. 1-10, November 2005.
139. L. Lai, J. Patel, T. Rinderknecht, and W.-T. Cheng, "Hardware efficient LBIST with complementary weights", *Proceedings of International Conference on Computer Design*, pp. 479-481, October 2005.
140. H.-J. Wunderlich, "Self test using unequiprobable random patterns", *Proceedings of Fault-Tolerant Computing Symposium*, pp. 258-263, July 1987.
141. H.D. Schnurmann, E. Lindbloom, and R.G. Carpenter, "The weighted random test pattern generator", *IEEE Transactions on Computers*, **24**(7), pp. 695-700, July 1975.
142. M. Chatterjee and D. Pradhan, "A BIST pattern generator design for near-perfect fault coverage", *IEEE Transactions on Computers*, **52**(12), pp. 1543-1557, December 2003.
143. N. Touba and E. McCluskey, "Bit-fixing in pseudorandom sequences for scan BIST", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **20**(4), pp. 545-555, April 2001.
144. C. Chen and S. Gupta, "Efficient BIST TPG design and test set compaction via input reduction", *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, **17**(8), pp. 692-705, August 1998.
145. H. Wunderlich and G. Kiefer, "Bit-flipping BIST", *International Conference on Computer-Aided Design*, pp. 337-343, November 1996.
146. V. Iyengar and D. Brand, "Synthesis of pseudo-random pattern testable designs", *Proceedings of International Test Conference*, pp. 501-508, August 1989.
147. N. Touba and E. McCluskey, "Test point insertion based on path tracing ", *Proceedings of VLSI Test Symposium*, pp. 2-8, April 1996.
148. N. Tamarapalli and J. Rajski, "Constructive multi-phase test point insertion for scan-based BIST", *Proceedings of International Test Conference*, pp. 410-416, October 1996.

149. F. Brglez and H. Fujiwara, "A neutral netlist of 10 combinational benchmark circuits and a target translator in FORTRAN", *IEEE International Symposium on Circuits and Systems*, pp., June 1985.
150. F. Brglez, D. Bryan, and C. Kozminski, "Combinational profiles of sequential benchmark circuits", *IEEE International Symposium on Circuits and Systems*, pp. 1929-1934, May 1989.
151. R. David, *Random Testing of Digital Circuits, Theory and Applications*, Marcel Dekker Inc., 1998.
152. P. Alfke, "Efficient shift registers, LFSR counters, and long pseudo random sequence generator", July 1996
153. R. Ward and T. Molteno, "Table of linear feedback shift register", University of Otago, October 2008
154. 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.