University of Southampton

Power Conscious Scan-Based Test of Digital VLSI Circuits

by
Paul Rosinger

A thesis submitted for the degree of
Doctor of Philosophy
in the
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science

February 2003

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING AND APPLIED SCIENCE
ELECTRONICS AND COMPUTER SCIENCE
Doctor of Philosophy

POWER CONSCIOUS SCAN TESTING OF DIGITAL VLSI CIRCUITS

by Paul Rosinger

Significant research has been devoted over the past decade to solving power-related challenges in digital integrated circuit design. While power optimisation become almost a push-button design step, the gap between the design and test camps left the power issues associated with the integrated circuit test process unsolved. Ignoring power during test may have serious consequences on product reliability and manufacturing yield. This thesis addresses the problem of power dissipation in scan-based test with the goal of developing and validating a set of power conscious design-for-test methodologies and structures. Four different test development and integration scenarios are investigated and an appropriate solution is proposed in each case.

The first part of this thesis addresses the problem of power dissipation during test in the system integration step. A test power profile manipulation technique, easy to integrate into any existing power constrained test scheduling algorithm, was developed in order to reduce the overall testing time by increasing test concurrency. Extensive experimental results using benchmark circuits, show that the proposed power profile manipulation approach enables testing time reductions up to 41% when compared to existing power constrained test scheduling approaches.

Mixed-mode BIST offers complete fault coverage with short test application times and small test data storage requirements. The second part of this thesis addresses the problem of reducing power dissipation in mixed-mode BIST. A new mixed-mode test pattern generator combining the masking properties of AND/OR composition with LFSR reseeding is proposed. Experimental data shows reductions up to 20% in average power dissipation during test when compared with traditional test pattern generators.

Test data compression/decompression represents an efficient solution to the increasing test data storage requirements on external test equipment. A new test data encoding scheme combined with a new weighted scan latch reordering algorithm are proposed for reducing the test data storage requirements for low power test sets. Experimental results show reductions up to nearly 50% in test data storage requirements and up to 75% in power dissipation when compared with other existing approaches.

The last part of this thesis presents a scan architecture with mutually exclusive scan segment activation which reduces both average and peak power dissipation during test, hence eliminating not only the risks of reliability problems but also the risks of noise-induced test failures. Experimental results show reductions up to 50% in both peak and average power when compared to standard scan architectures.

Contents

List of Figures

List of Tables

Acknowledgements

This PhD project would have not been possible without the support, encouragement and guidance of my supervisor Dr. Bashir Al-Hashimi. I would also like to thank him for his help during the preparation of this thesis and all my publications. I am grateful for the funding and research facilities provided by the Electronics and Computer Science Department of Southampton University, which made possible the work presented in this thesis. Dr. Mark Zwolinski also deserves thanks for his constructive comments during my nine-month and MPhil transfer examinations.

My friend and colleague Dr. Nicola Nicolici deserves thanks for his guidance, feedback and supervision throughout my PhD. I would also like to thank the Department of Electrical and Computer Engineering of McMaster University for providing the necessary research facilities for the three month period I have spent as a visiting researcher at McMaster University.

The encouragements and tolerance of my friends have helped me greatly through this PhD programme. This people include: Marcus, Mauricio, Matheos, Neil, Mircea and others. Special mention deserves Theo Gonciari for his substantial feedback on technical and non-technical matters.

Finally, I would like to thank my mother Ileana, my father Peter and my grandfather Pista for their love and continuous support over the past three years. Pista should be also thanked for persuading me to engage into this PhD.

Chapter 1

Introduction

The rapid advancements in process technologies have enabled the fabrication of chips with tens of millions of transistors operating at frequencies in the gigahertz range. While these advances brought unprecedented performance to electronic products, they also turn effects until recently ignored, such as power dissipation and resistive voltage drop, into critical issues which have to be addressed by integrated circuit design methodologies [Bis02, Gre99].

Significant research has been devoted to the power-related challenges in integrated circuit (IC) design. As a result, several industry leading design tools provide power optimisation, analysis and characterisation capabilities [Syn01c, Gra00a, Cad02]. While power optimisation has become almost a push-button design step, the gap between the design and test camps left unsolved the power issues associated with the IC test process. In most cases, the behaviour of the circuit under test is significantly different than during its normal operation mode due to the different scopes of the two modes. On one hand, low power design techniques [CB95, RP96, Yea98, Ped96] try to shut down all unnecessary blocks during the normal operation of the circuit in order to extend battery life, cut down cooling costs, etc. On the other hand, test scheduling algorithms increase the number of blocks which are tested in parallel in order to reduce test time. The conflicting perspectives of design for testability (DFT) and low power design methodologies make test power a critical design issue. Ignoring power during test may have serious consequences on reliability and yield [BA00, Zor93, SBW01, Whe00].

This thesis addresses the problem of testing digital ICs under power constraints. Various scenarios at system and gate level are investigated with the ultimate goal of developing power conscious DFT methodologies.

This chapter begins with an overview of the test process and of the main DFT techniques. It continues with a section on power dissipation in VLSI circuits outlining the sources of power dissipation and the mostly used power optimisation design techniques. The motivation of the work presented in this thesis is built by explaining the power issues associated with the test process. The chapter concludes with an outline of the thesis and its contributions.

1.1 The Test Process and Design for Testability

The main purpose of the IC test process is to provide a measure of the quality and reliability of a semiconductor stand-alone die or packaged part. *Functional testing* verifies that the circuit performs correctly the intended function it was designed for, over a range of input values. However, the time for exhaustively testing all possible input combinations grows exponentially with the number of inputs. To maintain a reasonable test time, the functional test must focus on the general function and corner cases. *Manufacturing testing* verifies the circuit for manufacturing defects by focusing on circuit structure rather than functional behaviour. Manufacturing defects include problems such as power and ground shorts, open interconnect on the die caused by dust particles, short-circuited source or drain of the transistor caused by metal spike-through, etc. Manufacturing defects might remain undetected by functional testing yet cause undesirable behaviour during circuit operation. Manufacturing test enables the screening out of devices with such structural defects.

1.1.1 Fault Models

When a manufacturing defect occurs, the physical defect has a logical effect on the behaviour of the circuit. For example, a signal shorted to power appears to be permanently high, while a signal shorted to ground appears to be permanently low. *Fault models* increase test generation efficiency by mapping controllable and observable physical defects to mathematical constructs that can be operated upon algorithmically and understood by a software simulator in order to provide a metric for quality measurement.

A fault model is a description of the behaviour and assumptions of how elements in a defective circuit behave. The goal of fault modelling is to model a high percentage of the physical defects that occur in the device at the highest possible level of abstraction. The high level of abstraction reduces the number of individual defects that must be considered and lowers the complexity of the algorithms used in generating the test. The result is that test generation can occur earlier in the design cycle in less time with less expensive computing resources [Gar01].

The gate-level model is widely accepted as the best compromise between abstraction level and the ability to represent most of the defects in the circuit under test (CUT). Register-transfer level (RTL) modelling is too abstract to accurately represent many of the fault types, and switch-level modelling is too computation-intensive [Gar01].

A commonly used metric for quality assessment of digital circuits is the *fault coverage*, given by the ratio:

$$F_{cov} = \frac{N_{DF}}{N_F - N_{UDF}},$$

where N_{DF} is the number of detected faults, N_F is the total number of faults, and N_{UDF} is the number of undetectable faults.

The *single stuck-at fault model* (SSAF) is the most popular fault model, first published in 1961 [Rot61]. It makes the assumption that only one line in the circuit is faulty at one time and that the fault is permanent. The effect of the fault is the same as if the faulty node is tied to V_{dd} (logical 1) or *Gnd* (logical 0), while the other gates in the circuit are not affected by the fault. The SSAF model covers many of the possible manufacturing defects in CMOS circuits, such as missing features, source-drain shorts, diffusion contaminants and metallisation shorts. The International Technology Roadmap for Semiconductors 1999 (ITRS) [Gro99] indicates that SSAF covers about 70% of the possible manufacturing defects in CMOS circuits. However, recent research [SM00] has shown that N-detect test sets for SSAF are very effective for both timing and hard failures. Algorithms for automatic test pattern generation (ATPG) and fault simulation on combinational networks with SSAF are well developed and efficient [BA00, ABF90].

The *multiple stuck-at fault model* (MSAF) [Gar01] makes the same basic assumptions as SSAF, except that it allows two or more lines in the circuit to be faulty at the same time. Although MSAF covers a greater number of defects, it also increases the number of faults which must be analysed under this approach: $3^n - 1$ for n circuit nodes. Furthermore, algorithms for ATPG and fault simulation are much more complex and not as well developed, and commercial simulators do not support MSAF well. Therefore, MSAF fault simulation is rarely used.

The *stuck-open fault model* (SOF) [Gar01] assumes that a single physical line in the circuit is broken and that the resulting open node is not tied to either V_{dd} or *Gnd*. The advantage of this approach is that it covers defects that can not be detected by SSAF and MSAF models but that can be tested with pairs of SAF test vectors.

Even if the circuit does not have a logical defect, it may have some physical defect, such as a process variation, that creates a large enough gate delay to cause problems. The *transition-delay fault model* [Gar01] assumes that the logic function of the circuit under test is error free but that a gate output may be slow to rise or slow to fall and that this time is longer than a required value. If this delay is large enough, the transition-delay fault behaves as a SAF and can be modelled using that method. The primary weaknesses of the transition-fault delay are that two pattern sequences are needed for initialisation and transition detection, and the minimum delay fault size is difficult to determine because of timing hazards.

The *path-delay fault model* [Gar01] is similar to the transition-delay model in assuming that the logic of the circuit under test is error-free. However, instead of modelling the fault as if a single gate delay in the circuit is faulty, this model assumes that the total delay in a path from input to output exceeds some maximum value. The path-delay fault model overcomes a possible problem with the transition-delay model, in which other faster gates in the circuit may compensate for the delay of a faulty gate. A problem with this fault model is that the number of possible paths grows exponentially with the number of nets.

From the above discussion on fault models it may be concluded that there is no universal fault model to cover all possible physical defects. This conclusion is also supported by the results of the experimental work presented in [SM00]. Each of the fault models presented is valuable in certain situations yet also has limitations. For each application, designers have to determine the fault model or, more likely, the combination of fault models which will provide a satisfactory level of defect coverage. The DFT techniques proposed in this thesis, unless otherwise specified, are targeting testing for stuck-at faults, due to the efficiency and wide industrial acceptance of this fault model.

1.1.2 Fault Simulation

Fault simulation determines the fault coverage of a set of test vectors. It can be thought of as performing several logic simulations concurrently - one that represents the fault-free circuit, and the rest that represent the circuits containing faults. A fault is detected each time the output response of the faulty machine differs from the output response of the good machine for a given vector. Fault simulation produces a list of detected faults for each test vector. Fault simulation is useful for determining the fault coverage when the manufacturing test vectors are generated manually or using existing hardware test pattern generators such as linear feedback shift registers (LFSR), etc. For large or complex designs, fault simulation can be very time consuming and often the test sets do not give good fault coverage.

1.1.3 Automatic Test Pattern Generation

Automatic test pattern generation (ATPG) generates test patterns and provides fault coverage statistics for the generated pattern set. ATPG for combinational circuits is well understood, and consequently, in most cases, it is possible to generate test sets with high fault coverage.

Combinational ATPG tools use both random and deterministic techniques to generate test patterns for faults on cell pins [BA00, ABF90]. During random pattern generation, the tool assigns input stimulus in a pseudo-random manner, then it performs fault simulation on the generated vector to determine which faults are detected. As the number of faults detected by successive random patterns decreases, ATPG shifts to a deterministic phase. During the deterministic pattern generation phase, the tool uses a pattern generation algorithm based on path sensitivity concepts to generate a test vector that detects a specific fault in the design. After generating a vector, the tool fault-simulates the vector to determine the complete set of faults detected by the vector. Test pattern generation continues until all faults have either been detected or have been identified as undetectable by this algorithm.

Due to the effects of memory and timing, ATPG for sequential circuits is much more difficult than for combinational circuits. Often, it is not possible to generate high fault coverage test sets for complex sequential designs, even when using sequential ATPG. Sequential ATPG tools use deterministic pattern generation algorithms based on extended applications of the path-sensitivity concepts [BA00, ABF90].

Structural DFT techniques, such as internal scan, simplify the test-pattern generation task for complex sequential designs, resulting in higher fault coverage and reduced testing costs.

1.1.4 Internal Scan

Internal scan design is the most popular DFT technique and has the greatest potential for achieving high fault coverage [BA00]. This technique simplifies the pattern generation problem by dividing complex sequential designs into fully isolated (full-scan design) or semi-isolated combinational blocks (partial-scan design). Internal scan modifies existing sequential elements in the design to support a serial shift capability in addition to their normal functions. This serial shift capability enhances internal node controllability and observability with a minimum of additional I/O pins.

Figure 1.1 shows a D flip-flop modified to support internal scan. Inputs to the multiplexer are the data input of the flip-flop (D) and the scan input signal (scan_in). The active input of the multiplexer is controlled by the scan-enable signal (scan_enable). Input pins are added to the cell for the scan_in and scan_enable signals. One of the data outputs of the flip-flop is used as the scan output signal (scan_out). The scan_out signal is connected to the scan_in signal of another scan cell to form a serial scan (shift) capability.

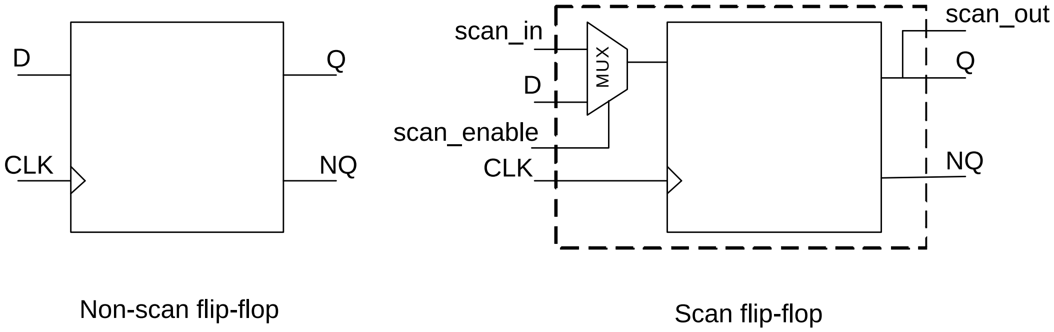


Figure 1.1: D flip-flop with scan capability

The modified sequential cells are chained together to form one or more large shift registers. These shift registers are called scan chains or scan paths. The sequential cells connected in a scan chain are scan controllable and scan observable. A sequential cell is scan controllable when it can be set to a known state by serially shifting in specific logic values. ATPG tools consider scan controllable cells as pseudo-primary inputs of the combinational part of the design. A sequential cell is scan observable when its state can be observed by serially shifting out data. ATPG tools consider scan observable cells as pseudo-primary outputs of the combinational part of the design.

Adding scan circuitry to a design usually has the following effects:

- Design size and power increase slightly because scan cells are usually larger than the non-scan cells they replace, and the nets used for the scan signals use additional area.
- Design performance (speed) decreases marginally because of changes in the electrical characteristics of the scan cells that replace the non-scan cells.
- Global test signals that drive many sequential elements might require buffering to prevent electrical design-rule violations.

Test patterns are applied to a scan-based design through the scan chains. The process is the same for full-scan and partial-scan designs. Scan cells operate in one of two modes: parallel mode (or capture mode) or shift mode. For the multiplexed flip-flop scan style shown in Figure 1.1, the mode is controlled by the scan_enable pin. In parallel mode, the input to each scan element comes from the combinational logic block. In shift mode, the input comes from the output of the previous scan cell or a scan input port. Other scan styles work in a similar fashion.

To apply a scan pattern, the target tester performs the following steps:

1. Selects shift mode by setting the scan_enable port. This test signal is connected to all scan cells.
2. Shifts in the input stimulus for the scan cells (pseudo-primary inputs) at the scan input ports.

3. Selects parallel mode by inverting the scan-enable port.
4. Applies the input stimulus to the primary inputs.
5. Checks the output response at the primary outputs after the circuit has settled and compares it to the expected fault-free response. This process is called parallel measure.
6. Pulses one or more clocks to capture the steady-state output response of the non-scan logic blocks into the scan cells. This process is called parallel capture.
7. Selects shift mode by resetting the scan-enable port.
8. Shifts out the output response of the scan cells (pseudo-primary outputs) at the scan output ports and compares the scan cell contents to the expected fault-free response.

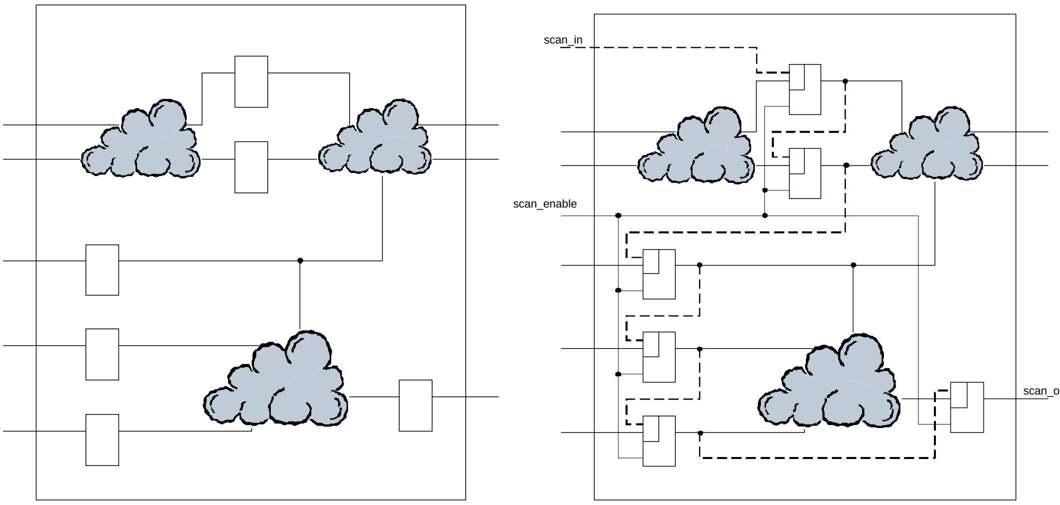
1.1.5 Full-Scan Design vs. Partial-Scan Design

In the full-scan design technique, all sequential cells in the design are modified to perform a serial shift function. Sequential elements that are not scanned are treated as black box cells (cells with unknown function).

Full-scan divides a sequential design into combinational blocks as shown in Figure 1.2. Clouds represent combinational logic; rectangles represent sequential logic. The full-scan diagram shows the scan path through the design.

Through pseudo-primary inputs, the scan path enables direct control of inputs to all combinational blocks. Through pseudo-primary outputs, the scan path enables direct observation of outputs from all combinational blocks. Efficient combinational ATPG algorithms can be used to generate test sets with high fault coverage for the full-scan design.

In the partial-scan design technique, the scan chains contain only a fraction of the sequential cells in the design. The partial-scan technique offers a trade-off



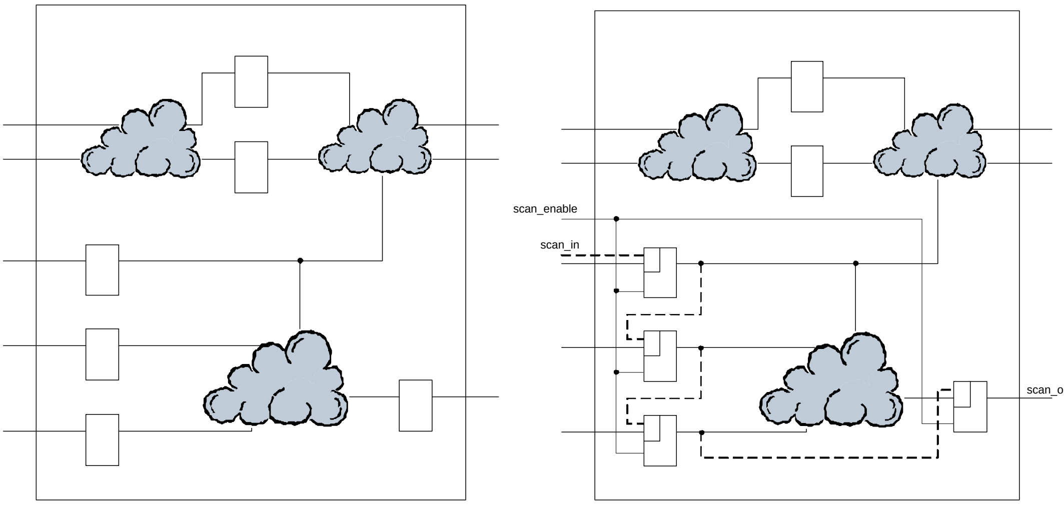
(a) Without scan

(b) With full scan

Figure 1.2: Scan path through a full-scan design

between the maximum achievable fault coverage and the effect on design size and performance [ABF90, Syn01b]. For example, flip-flops on the critical paths of designs with limited timing budget cannot be replaced with scan cells as the cell multiplexer would increase the overall delay of the design. Partial-scan divides a complex sequential design into simpler sequential blocks, as shown in Figure 1.3. The partial-scan diagram shows the scan path through the design.

Sequential ATPG algorithms are needed for partial-scan designs to allow fault propagation through non-scan sequential elements. In general, a partial-scan design can not achieve the fault coverage of the full-scan version of the design. The level of fault coverage for a partial-scan design is related to the location and fraction of scan registers in that design. Moreover, sequential ATPG algorithms are much more complex than combinational ATPG algorithms [BA00, ABF90].



(a) Without scan

(b) With partial scan

Figure 1.3: Scan path through a partial-scan design

1.1.6 Automatic Test Equipment

Automatic test equipment (ATE) is an instrument used to apply test patterns to a device under test (DUT) (also referred to as circuit under test (CUT)), analyse the responses from the DUT, and mark the DUT as good or bad [BA00]. The ATE is controlled by a central computer, and one or more CPUs are built into it to ease control and to enable test data compression. The tester has one or more test heads, which contain buffering electronics local to the DUT, but one mainframe with common instrumentation, power supplies, etc. The ATE is connected to external equipment that mechanically handles the wafers or IC packages being tested. The ATE, unlike a simulator, can operate only at the edge of the design through the package pins, and has real signal and clock delays and signal degradations, chip output loads and thermal considerations. The tester also has edge placement precision, accuracy, and edge-rate (rise and fall time) limitations.

The main component of the cost-of-test is the time a chip spends “in-socket” on the ATE. There is a minimum cost based on the time it takes an automated handler to insert and remove a chip from the tester socket (production throughput). For a typical ATE, the test cost per second is about $0.11 [BRCA01]. Cost reductions to bring the tester time below the handler threshold require testing multiple dies in parallel with the same tester. If the test programs applied on the tester are excessively complex or contain an excessive amount of vector data, then a vector reload may be required. A reload adds a large time overhead to the overall testing process, which increases the "in-socket" time, and consequently the cost of test.

1.1.7 Built-in Self-Test

There is an increasing number of testability problems posed by the increasing complexity of modern chips. The rising logic-to-pin ratio makes it harder to accurately observe signals on the device. VLSI devices are increasingly dense and faster with the emerging deep submicron feature sizes. The time needed to generate and apply test patterns to these chips, as well as the volume of test data which has to be stored on the tester are also increasing. The growing gap between chip internal frequencies and the I/O frequencies makes it very difficult, if not impossible, to perform at-speed testing using external ATE [BA00]. Built-in self-test (BIST) represents a viable solution to most of these problems. BIST moves test pattern generation and test response analysis from the tester to the chip under test. Thus, at the expense of one to three percent area overhead, BIST eliminates the problems associated with the limited pin count and I/O bandwidth.

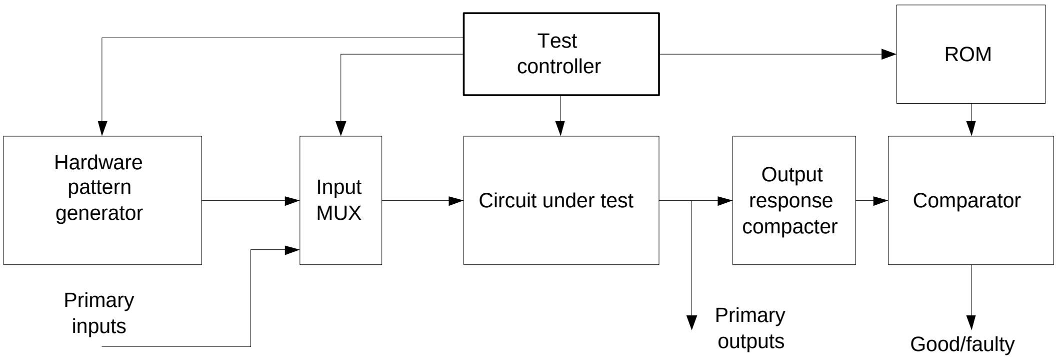


Figure 1.4: Typical BIST scheme

Figure 1.4 shows typical BIST hardware. In test mode, the test controller connects the inputs of the circuit under test to the hardware pattern generator through the input multiplexer. The hardware pattern generators are also referred to as test pattern generators (TPG). The output response compactor transforms the output of the circuit under test into signatures which are compared with reference signatures stored in a ROM.

The most commonly used hardware pattern generators [BA00] include:

1. **ROM**. A precomputed test set is stored in a ROM on the chip. This approach is prohibitively expensive in terms of chip area.

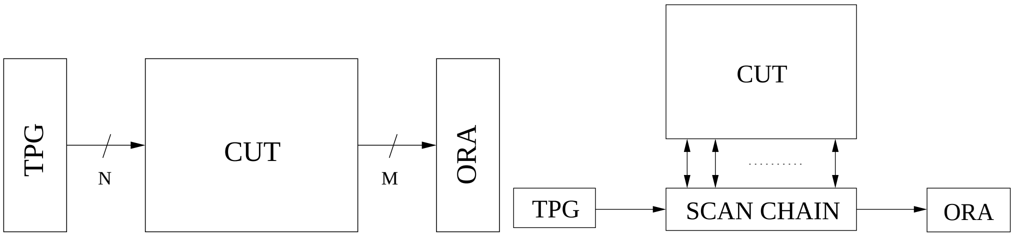
2. **LFSR**. Linear feedback shift registers are used to generate pseudo-random tests. The advantage of this approach is the very low area requirements, however very long test sequences (1 million or more test vectors) are required in order to obtain good fault coverage.

3. **Binary counters**. Binary counters are used to generate pseudo-exhaustive test sequences (the pattern generator is built of several smaller binary counters, each producing an exhaustive sequence at its outputs). The binary counter approach requires more hardware than the LFSR pattern generator.

4. **Modified counters**. Modified counters have also been successful as test pattern generators, but they also require long test sequences for good fault coverage.

5. **LFSR and ROM**. This is one of the most effective approaches, which uses an LFSR to generate a limited number of pseudo-random test vectors to cover the easy-to-detect faults, and generates deterministic test patterns using an ATPG tool for the few remaining random-pattern-resistant faults. The deterministic patterns are embedded in the output of the LFSR, by re-seeding the LFSR with values stored in the ROM, in order to augment the fault coverage. This approach is also known as *mixed-mode BIST* because it uses both pseudo-random and deterministic test vectors [HWH96].

6. **Cellular automaton**. In this approach, each pattern generator cell has a few logic gates, a flip-flop, and connections only to neighbouring cells. The cell is replicated to produce the cellular automaton.



(a) Test-per-clock BIST

(b) Test-per-scan BIST

Figure 1.5: BIST schemes

BIST schemes can be classified into *test-per-clock* (Figure 1.5(a)) and *test-per-scan* schemes (Figure 1.5(b)) according to the method used to apply the test patterns to the CUT. In test per-clock BIST, the outputs of the TPG are connected directly to the CUT inputs and a new test pattern is applied to the CUT with every test clock. In a test-per-scan BIST scheme, the test patterns generated by the TPG are applied to the CUT via one or more internal scan chains, and hence a test pattern is applied to the CUT every $m+1$ test clocks cycles, where m is the number of flip-flops in the scan chain. The test response of the CUT is captured in the scan chain and scanned-out during the next m clock cycles simultaneously with scanning-in of the next test vector. Usually test-per-clock schemes are preferred for memory test, due to their regular structure which makes them easily controllable from the primary input pins, while test-per-scan schemes are commonly used for testing random logic blocks.

1.2 Power Dissipation in Digital VLSI Circuits

Battery life, packaging and cooling costs, reliability are all factors which justify the need for low power design [Ped96]. This section will provide background information on the main sources of power dissipation in digital VLSI circuits, as well as an overview of existing power optimisation techniques. The information provided in this section will be used later in this chapter to explain the conflict between the current DFT techniques and low power design.

1.2.1 Sources of Power Dissipation in Digital VLSI Circuits

The power dissipated in a circuit falls into two broad categories: *static power* and *dynamic power* [Ped96].

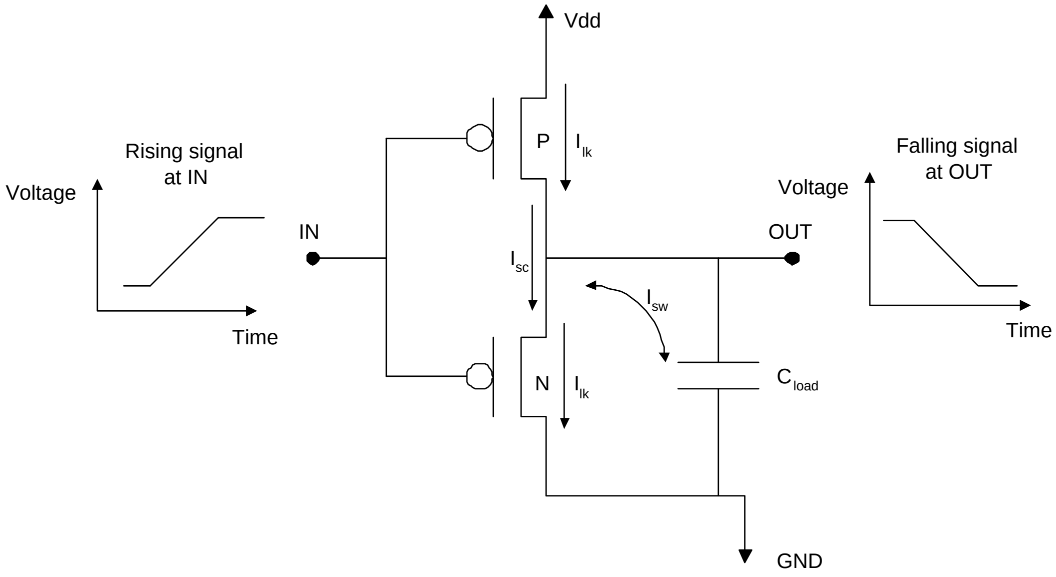


Figure 1.6: Components of power dissipation

Static power is the power dissipated by a gate when it is not switching, that is, when it is inactive or static. The largest fraction of static power results from source-to-drain subthreshold leakage, which is caused by reduced threshold voltages that prevent the gate from completely turning off. Static power is also dissipated when current leaks between the substrate and the diffusion layers.

Dynamic power is the power dissipated when the circuit is active. A circuit is active anytime the voltage on a net changes due to some stimulus applied to the circuit. Because voltage on an input of a cell can change without necessarily resulting in a logic transition on the output, dynamic power is dissipated by the internal transistors of the cell which are switching even when an output net does not change its logic state. The dynamic power of a circuit is composed of two kinds of power: *switching power* and *internal power* [Syn01c].

The switching power of a driving cell is the power dissipated by the charging and discharging of the load capacitance at the output of the cell. The total load capacitance at the output of a driving cell is the sum of the net and gate capacitances on the driving output. Because such charging and discharging are the result of the logic transitions at the output of the cell, switching power increases as logic transitions increase. Therefore, the switching power of a cell is a function of both the total load capacitance at the cell output and the rate of logic transitions. Switching power comprises a large percentage of the power dissipation of an active CMOS circuit. The switching power (P_c) is given by:

$$P_c \;=\; \frac{{V_{dd}}^2}{2} \sum_{\forall_{nets(i)}} C_{loadi} \times TR_i \tag{1.1}$$

where C_{loadi} is the capacitive load of net i, TR_i is the toggle rate of net i as number of transitions per second, and V_{dd} is the supply voltage.

Internal power is any power dissipated within the boundary of a cell. During switching, a circuit dissipates internal power by the charging or discharging of any existing capacitances internal to the cell. Internal power includes power dissipated by a momentary short circuit between the P and N transistors of a gate, called short-circuit power. To illustrate the cause of the short-circuit, consider the simple gate shown in Figure 1.6. As the IN signal transitions from low to high, the N type transistor turns on and the P type transistor turns off. However, for a short time, both the P and N transistors can be on simultaneously. During this time, current I_{sc} flows from V_{dd} to GND, causing the dissipation of short-circuit power. For circuits with fast transition times, short-circuit power can be small. However, for circuits with slow transition times, short-circuit power can account for 30 percent of the total power dissipated by the gate [Syn01c]. Short-circuit power is affected by the dimensions of the transistors and the load capacitance at the output of the cell [Syn01c]. The internal power of a cell is the sum of the internal power of all of the cell's inputs and outputs as modelled in the technology library.

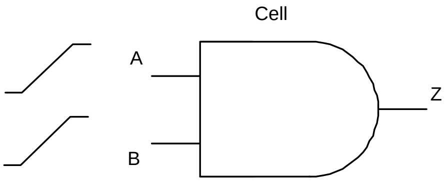


Figure 1.7: Simple cell

For the simple cell shown in Figure 1.7, internal power can be estimated as follows:

$$\begin{aligned} P_{int} &= E_Z \times TR_Z \qquad (1.2) \\ E_Z &= f(C_{load}, WeightAvg(Trans)) \\ WeightAvg(Trans) &= \frac{\sum_{i=A,B} TR_i \times Trans_i}{\sum_{i=A,B} TR_i} \end{aligned}$$

where E_Z is the internal energy for output Z as a function of input transitions and output load (usually defined in the technology library), TR_Z is the toggle rate of output pin Z, TR_i is the toggle rate of input pin i, $Trans_i$ is the transition time of input i, and $WeightAvg(Trans)$ is the weighted average transition time for output Z.

1.2.2 Effects of Excessive Power Dissipation

Power dissipation in a VLSI circuit is usually described in terms of its average and peak values. High average power means a high power consumption sustained for a long period of time. This has the following negative effects on the VLSI device: shortens battery life for mobile applications, increases chip temperature with the risk of exceeding the maximum value tolerated by the package, speeds-up electro-migration, hence leading to reliability problems. High peak power can lead to a drop in power supply voltage, called voltage drop or IR drop. As shown in Figure 1.8, IR drop occurs because of the high instantaneous current I_{dd} flowing through the resistive power network. Its value depends on the resistance R_{line} of the power net, the power grid architecture, the power pad locations and the current drawn by standard cells connected to power net [Bis02]. Thus, for a circuit with a given power grid, the value of the instantaneous current is given by the number of cells which are switching simultaneously. Power net voltage drop affects the performance of a design by increasing cell and interconnect delay. IR drop also reduces noise margins of cells and increases the probability of failures due to crosstalk noise.

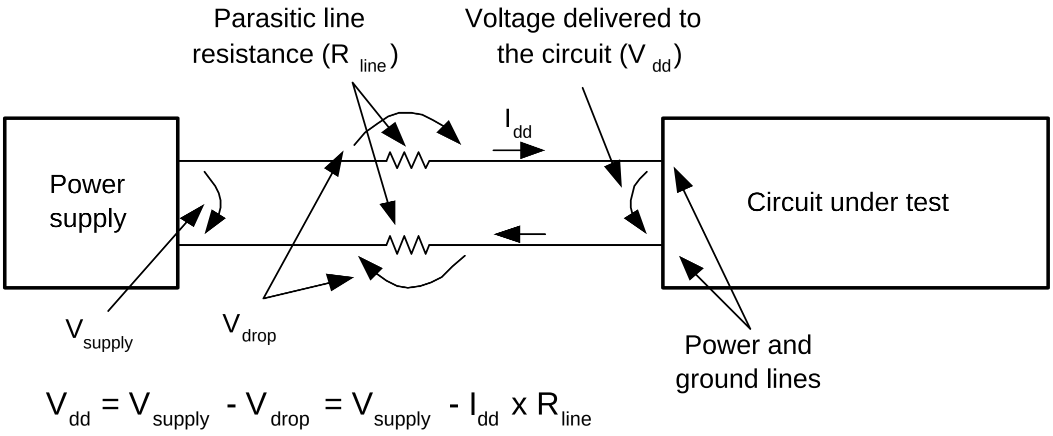

Parasitic line resistance (R_{line})
Voltage delivered to the circuit (V_{dd})
I_{dd}
Power supply
Circuit under test
V_{supply}
V_{drop}
Power and ground lines
$V_{dd} = V_{supply} - V_{drop} = V_{supply} - I_{dd} \times R_{line}$

Figure 1.8: Voltage (IR) drop

1.2.3 Low Power Design

In order to address the challenges posed by the increasing power requirements, various power optimisation methods have been developed [Ped96]:

- **Scaling the supply voltage:** This approach can be very effective in reducing the power dissipation, but often requires new IC fabrication processing.

Supply voltage scaling also requires support circuitry for low-voltage operation including level-converters and DC/DC converters as well as detailed consideration of issues such as signal-to-noise margins.

- **Employing better design techniques:** This approach promises to be very successful because the investment to reduce power by design is relatively small in comparison to other approaches. The most popular solution, adopted by most major synthesis tools is RTL clock gating, which basically ensures that each sequential part of the design is clocked only when it is necessary [Syn01c].

- **Using power management strategies:** The power savings that can be achieved by various static and dynamic power management techniques are very application dependent, but can be significant [BBM00].

RTL Clock Gating

One of most efficient design technique for reducing dynamic power, already provided by most popular commercial synthesis tools, is *RTL clock gating.* RTL clock gating works by identifying groups of flip-flops which share a common enable term (a term which determines that new values will be clocked into the flip-flops). Traditional methodologies use this enable term to control the select on a multiplexer connected to the D port of the flip-flop or to control the clock enable pin on a flip-flop with clock enable capabilities. RTL clock gating uses this enable term to control a clock gating circuit which is connected to the clock ports of all of the flip-flops with the common enable term. Therefore, if a bank of flip-flops which share a common enable term have RTL clock gating implemented, the flip-flops will consume zero dynamic power as long as this enable term is false. The following example shows how RTL clock gating works.

Example 1 *Consider the two pipelined 3-bit counters shown in Figure 1.9. The two counters share the same clock signal (CLK). The overflow pre-condition for COUNTER 1 (all three bits high) is used to trigger the increment of COUNTER 2.*

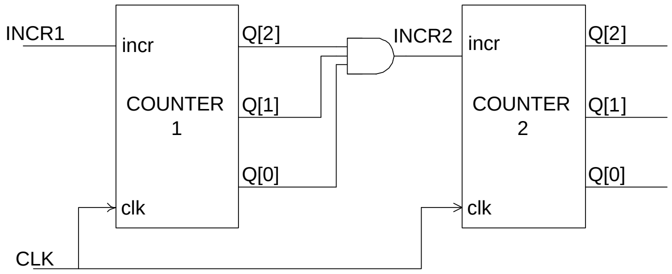


Figure 1.9: Two-counter sample design

Each of the two counters increments on every clock cycle when its corresponding increment signal, INCR1 and INCR2 respectively, is asserted high. As the clock signal is routed directly to both counters, which means that they will be clocked continuously, with the old data recirculated into their flip-flops when the INCR1 and INCR2 signals are low. It should be noted that flip-flops consume power even if their inputs, and therefore, their internal state, do not change. Assuming that INCR1 is held permanently high, COUNTER 1 will increment with every clock while COUNTER 2 will increment only once every 8 clocks, although it will consume power during each clock cycle.

In Figure 1.10, the same circuit is implemented using clock gating. This implementation is similar to the previous one except that two clock gating elements have been inserted into the clock network, which cause the counters to be clocked only when their corresponding increment signals are high. As the increment conditions for the two counters are now used to enable their clocks, when the increment signals are asserted low, the counters are not clocked and hence, they retain their content just like the implementation without clock gating. The difference is that instead of having the two counters clocked every clock cycle and hence, consuming power even if their content does not change, in the optimised implementation, the counters are clocked only *when they have to be incremented. Therefore the situation when the two counters are clocked simultaneously occurs only once every 8 clocks, assuming INCR1 is held permanently high.*

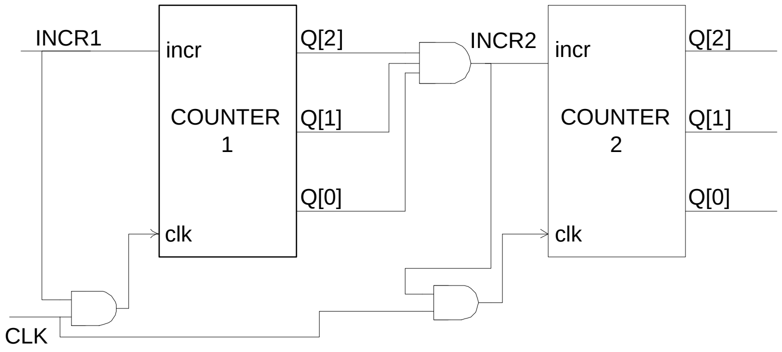


Figure 1.10: Implementation of the two-counter design with clock gating

As illustrated in the previous example, clock gating is a very efficient design technique for reducing the useless switching activity on-chip. Industrial experiences report dynamic power reductions ranging from 50% to 70% achieved through clock gating [EB00].

1.3 Power Issues Affecting the Test Process

As reported in [Zor93], power dissipation during test can be up to three times higher than during normal operation. This section will explain why power consumption tends to be significantly higher during test than during normal operation and what are the problems deriving from that.

The behaviour of a device in test mode is substantially different than its behaviour during normal operation. This is due the different scopes of the two modes. There are two major problems which will be discussed in the following.

- The first problem originates from the contradiction between the goals of system-level power reduction techniques and test scheduling algorithms. Consider, for example, high-performance memory systems. Memories are organised into several blocks of fixed sizes. Under normal system operation, only one memory block is active during each memory access while other blocks are in power down mode to minimise power consumption. During test however, in order to test the memory in the shortest possible time, it is desirable to activate as many memory blocks as possible. Another example is testing of multi-chip modules (MCM) [CSA97]. An attractive approach is to use BIST blocks executing in parallel. This approach reduces significantly the overall testing time but also increases power dissipation during test, while during normal operation not all blocks are activated simultaneously and hence, the inactive blocks do not contribute to power dissipation. There are several reasons why the overall testing time should be kept as short as possible. One is to ensure that the ATE does not de-calibrate before the end of the test session. Another reason is to increase the testing throughput in the case of mass production [BA00]. Based on the facts mentioned above, aggressive test scheduling algorithms which ignore power, may lead to chip overheating. This is extremely undesirable, especially during early production tests performed on the unpackaged dies.

- The second problem arises from the conflict between RTL clock gating and scan testing. This will be explained through the following example.

  Example 2 *Consider the implementation using clock gating of the two-counter design from example 1 in section 1.2.3 (Figure 1.11). The design after scan insertion is shown in Figure 1.12, assuming a full-scan configuration and multiplexed scan style for the scan cells [Syn01b]. Scan insertion has chained all flip-flops in the design into a long shift register. During test, the test enable signal (test_en) is asserted high, hence the clock is enabled simultaneously for all flip-flops, by disabling the effect of the clock gating logic. The scan enable signal (test_se) puts the scan chain into the "shift" (test_se = 1) or "capture" mode (test_se = 1). As the clock is globally enabled during the test mode,* all *six flip-flops in the design will be clocked* every *test clock, and hence the design will consume much more power than during normal operation. As stated in Example 1 in section 1.2.3, during normal operation all flip-flops are clocked simultaneously only once every 8 clocks, assuming INCR1 is held permanently high.*

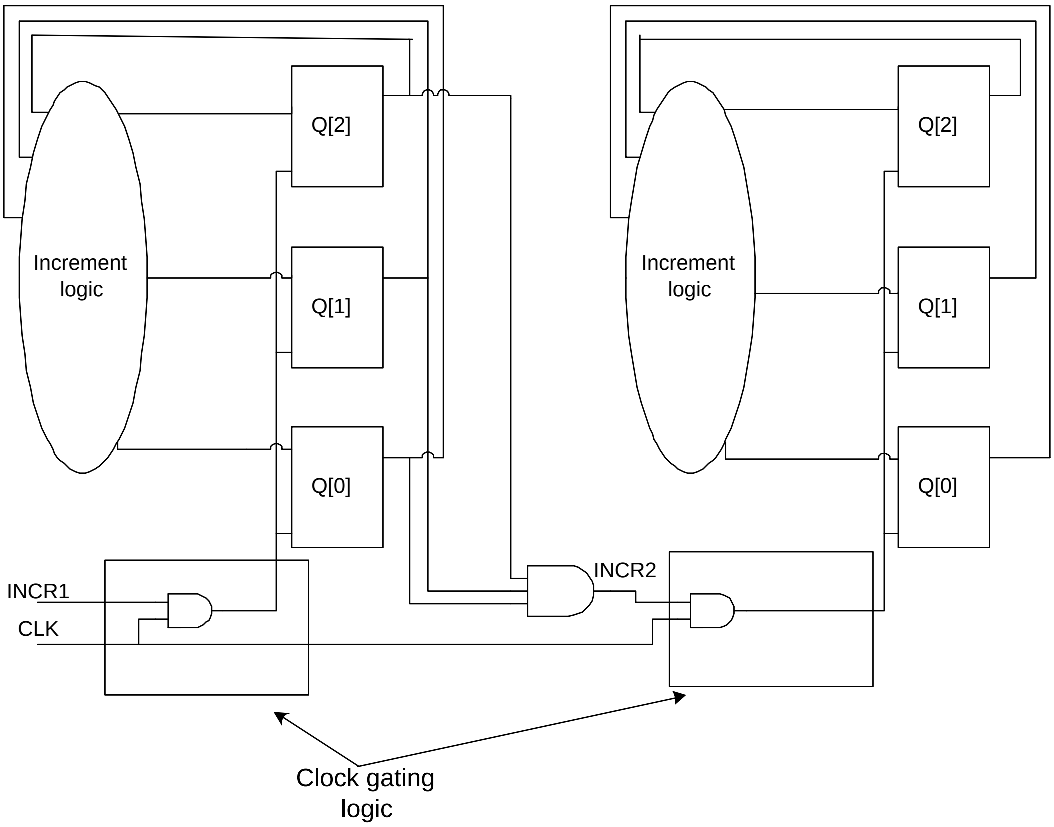


Figure 1.11: Implementation of the two-counter design with clock gating

As shown in the previous example, during testing, a much larger part of the circuit is switching during each clock compared to normal operation. The high current drawn from the power lines by the increased number of simultaneously switching circuit nodes may produce high voltage drops, which would not occur during the normal operation mode. If the amplitude of the voltage drop exceeds the noise margins tolerated by the circuit, one of the following situations can occur:

1. the test stimulus and/or circuit response is corrupted, if the high voltage drop occurs during the shift cycles.
2. the scan chain will capture an erroneous circuit response, if the high voltage drop occurs during the capture cycle.

Moreover, having an increased level of switching activity over a longer period

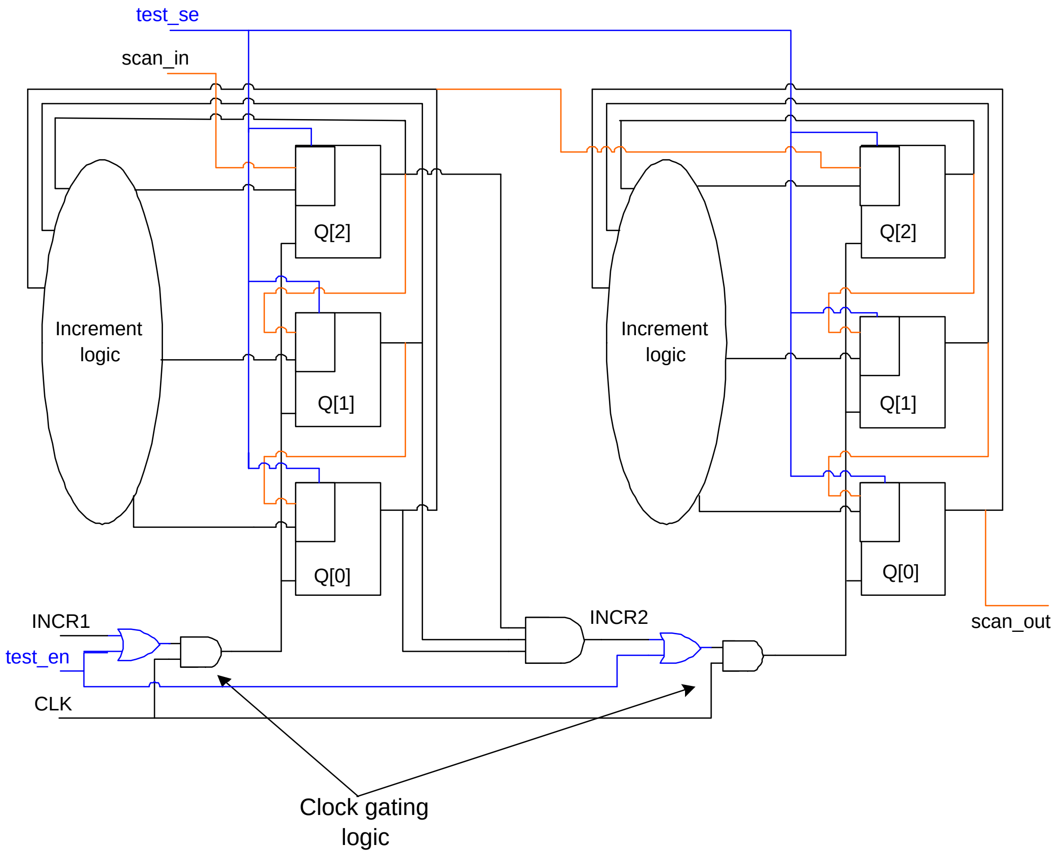


Figure 1.12: Implementation of sample design with clock gating

of time will overheat the circuit which can affect circuit's reliability or even lead to permanent damage of the chip under test.

From the previous discussion it can be concluded that traditional DFT cancels the effect of power conscious design strategies, a fact which can have serious impact on product yield and reliability. Hence, as the fabrication technologies advance and the chips are becoming more and more complex, it is becoming increasingly important to embed power-awareness into existing DFT methodologies.

1.4 Thesis Overview and Contributions

The aim of this thesis is to investigate power-conscious scan-based DFT techniques. As explained in the previous section, power dissipation during test represents a critical issue for the semiconductor industry, which will become even more acute as the gate densities increase. The roots of this problem are the short time-to-market windows and the existing gap between the design and test development flows, which did not allow the advances in the low power design methodology to propagate into the DFT methodology. The work presented in this thesis addresses the problem of reducing power during testing of digital VLSI circuits. Various scenarios are analysed with the goal of developing suitable power conscious DFT techniques. An outline of this thesis is given in the following. Chapters 2, 3, 4 and 5 present the original contributions of this thesis. The relevant background and an outline of the related work is provided in each of these chapters.

Chapter 2 addresses the conflict between test scheduling algorithms and system-level solutions for reducing power during normal operation such as activity-driven block shutdown. As mentioned earlier in this chapter activity driven shutdown powers-down all idle blocks of a system until they become needed again. Test scheduling, on the other hand is trying to increase the number of blocks tested simultaneously in order to reduce test time, which has as a side-effect an undesirably high level of switching activity. Over the past decade, several power constrained test scheduling approaches have been proposed. Chapter 2 presents a power profile manipulation technique which can be used to enhance existing power constrained test scheduling algorithms. The proposed technique performs a set of transformations on the test sequence corresponding to each system block such that their corresponding power profile during testing is lowered and reshaped in order to increase test concurrency without exceeding a given power constraint.

Chapter 3 tackles the problem of power reduction in a BIST environment from the test pattern generation point of view. The proposed test pattern generator (TPG), designed for mixed-mode BIST, produces test sequences leading to less switching activity in the circuit under test, compared to traditional TPGs, while achieving high fault coverage in short test application time.

Present ATE is facing the problem of storing the increasing amounts of test data required for testing complex chips. An attractive solution is to employ test data compression methods in order to reduce the volume of data stored on the tester. Chapter 4 proposes a run-length coding scheme suitable for compressing low power test sets. The efficiency of the proposed scheme is evaluated in conjunction with a recently proposed method for test data compression.

Chapter 5 proposes a low power scan chain architecture. The automated scan chain design technique is based on a scan chain partitioning algorithm which aims to translate clock gating into the testing domain. By enabling exactly one scan chain partition during each test clock, the proposed technique reduces significantly the both peak and average power dissipation during test, hence eliminating problems related to chip overheating, as well as the risk of noise-generated failures.

As it can be concluded from the outline of the thesis presented above, there are several ways to approach the problem of power reduction during test. The methods proposed in chapters 2 and 3 are based on transformations of the test set. Test set transformations do not require any structural changes to the circuit under test, and therefore they are suitable for system integration using intellectual property (IP) cores. Chapter 4 combines test sequence transformations with more power-efficient structural transformations of the circuit. The solution proposed in chapter 5 takes a purely structural, and thus test set independent, approach for test power reduction.

The contributions of the work presented in this thesis have been already published or are under consideration as follows:

1. *Power constrained test scheduling using power profile manipulation* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. - The IEEE International Symposium on Circuits and Systems (ISCAS) 2001, Volume: 5, Page(s): 251-254, vol. 5

2. *Simultaneous reduction in volume of test data and power dissipation for systems-on-a-chip* **Rosinger, P.**; Gonciari, P.T.; Al-Hashimi, B.M.; Nicolici, N. - IEE

Electronics Letters , Volume: 37, Issue: 24, 22 Nov. 2001 Page(s): 1434-1436

3. *Power profile manipulation: a new approach for reducing test application time under power constraints* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. - IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Volume: 21 Issue: 10 , Oct. 2002 Page(s): 1217-1225

4. *Analysing trade-offs in scan power and test data compression for systems-on-a-chip* **Rosinger, P.**; Gonciari, P.T.; Al-Hashimi, B.M.; Nicolici, N. - IEE Proceedings-Computers and Digital Techniques, Volume: 149 Issue: 4 , July 2002 Page(s): 188-196

5. *Low power mixed-mode BIST based on mask pattern generation using dual MP-LFSR reseeding* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. - Proc of the IEEE International Conference on Computer Design (ICCD) 2002, Page(s): 474-479

6. *Scan architecture for shift and capture power reduction* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. Proc of the IEEE Symposium on Defect and Fault Tolerance (DFT) 2002, Page(s): 129-137

7. *Scan architecture with mutually exclusive scan segment activation for shift and capture power reduction* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. - Submitted (2nd revision) to the IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, Nov. 2002.

8. *Dual multiple-polynomial LFSR for low power mixed-mode BIST* **Rosinger, P.**; Al-Hashimi, B.M.; Nicolici, N. - Accepted for publication in the IEE Proceedings-Computers and Digital Techniques, Dec. 2002

Chapter 2

Power Profile Manipulation

Chapter 1 has outlined that a major issue in testing modern systems is the long testing time. With the growing complexity of modern systems, testing time increases rapidly, producing a serious impact on the final cost of the design [BA00]. Test scheduling algorithms are system-level strategies for increasing test concurrency in order to reduce the overall test time. The high power consumption caused by the intense switching activity in the circuit may exceed the specified limits, hence it can damage the system under test or affect its reliability. Consequently, several solutions to the problem of power dissipation during test have been recently proposed [Nic00, Zor93, Gir00, DCPR98, GLPS97, NAHW00, FCN^{+}99, WG98, CSA97, MWMV00, NAH00, LP01, IC01, Cha00, RCV00]. Within these solutions, two main directions can be identified: one considering power dissipation during test an optimisation objective, while the other considers power a design constraint under which other parameters, such as testing time, are improved.

- Test set transformations are examples of techniques which fall into the first category. Test set transformations include test vector reordering [DCPR98, GLPS97, FCN^{+}99, CRRV99], test vector altering and test sequence expansion. Test set transformations methods aim to increase the correlation between successive test patterns and/or between successive bits in the test patterns, and hence decreasing power dissipation during test by reducing the switching activity at the circuit inputs. Test vector reordering and test sequence expansion will be explained later in this chapter. Test vector altering techniques will be discussed in chapters 3 and 4. Usually embedded cores are delivered as IP blocks accompanied with test data, thus the system integrator cannot change their internal structure. Therefore, unless the cores are pre-designed with special scan architectures, the system integrator can control the power dissipation during test *only* by means of test data transformations.

- With the goal of reducing the test time while considering test power dissipation, several power constrained test scheduling (PCTS) algorithms, belonging to the second research direction, were proposed recently. PCTS algorithms [CSA97, MWMV00, NAH00, LP01, IC01, Cha00, RCV00] aim to minimise testing time under a given power constraint imposed by the package type and energy limitations. The basic idea of PCTS algorithms is to maximise the test concurrency, without exceeding the power constraint. Power constrained test scheduling will be discussed later in this chapter.

This chapter proposes a new approach for reducing test time under power constraints by manipulating the test power profiles of the system blocks. The proposed technique aims to reduce power dissipation at block level, and to maximise, with respect to the given power constraint,test concurrency at system level. The proposed technique is addressed to the system integrator without putting any constraints on the scan architectures of the cores. The power profile manipulation technique described in this chapter has the following advantages over existing PCTS algorithms:

1. It allows not only the average and/or peak values of power dissipation to be considered, *but also the shape* of the power profile. The possibility of controlling the size and position within the power profile of the higher and lower power parts would allow existing PCTS algorithms [CSA97, MWMV00, NAH00, LP01, IC01, Cha00, RCV00], to increase test concurrency under specified power constraints.

2. It exploits the *slack time* of short test sequences in order to obtain lower power profiles. A test session from a test schedule usually consists in a number of unequal length tests. The test length differences, referred earlier as "test sequence slack time", can be used to extend shorter test sets in the test session with additional vectors. Careful selection of the additional test vectors can lower the peak power during testing which enables increased test concurrency under a given power constraint.

By manipulating the power profile during test scheduling, the proposed solution is a fusion between the two research directions in low power testing mentioned earlier: techniques for minimising test power dissipation [DCPR98, GLPS97, FCN^{+}99, CRRV99, WG98] and techniques for minimising test time under power constraints [CSA97, MWMV00, NAH00, LP01, IC01, Cha00, RCV00]. In this chapter it will be shown how complementary techniques can be easily combined to achieve high test concurrency under given power constraints. *The proposed power profile manipulation approach is not a test scheduling algorithm, rather it represents a complementary technique meant to enhance the performance of existing power constrained test scheduling algorithms.* The distinctive feature of power profile manipulation is that it is independent of the test scheduling policy. Consequently, it can be *equally* embedded into *any* existing PCTS algorithm to leverage its performance. It should be noted that this methodology targets testing scenarios where test data transformations are possible, such as stuck-at fault testing or skewed-load delay-testing using ATPG generated test sets.

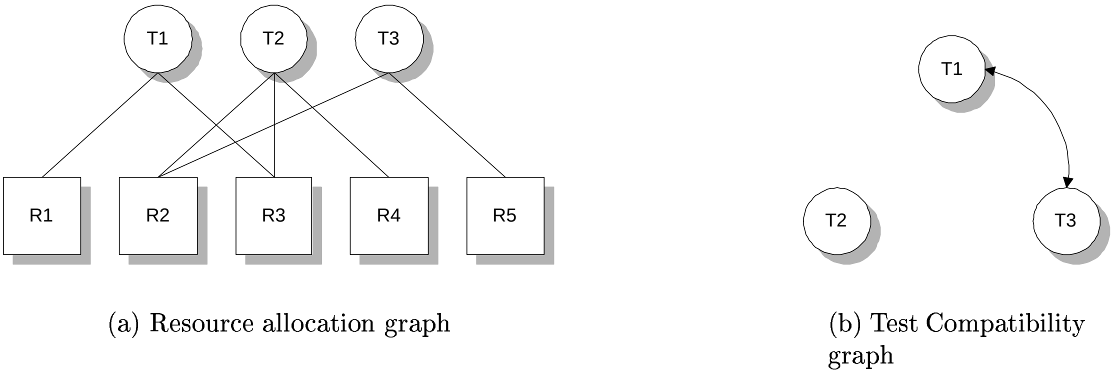
The rest of the chapter is organised as follows. Section 2.1 provides background on test scheduling and a commonly used approximation model for power dissipation during test. The proposed power profile manipulation technique, including a new test power approximation model, is detailed in Section 2.2. Section 2.3 shows through an example how the proposed methodology can be integrated into existing power constrained test scheduling algorithms. Section 2.4 discusses the experiments performed in order to validate the efficiency of the proposed method.

2.1 Background Information

This section provides the terminology and concepts which will be used in the rest of the chapter.

2.1.1 Test Scheduling

Test scheduling algorithms aim to reduce test application time by increasing the concurrency of the testing activities in the system. When ignoring power, maximum test concurrency is limited only by system's resource sharing configuration. The resource sharing configuration of a system can be represented using the *resource allocation graph*. An example of resource allocation graph is shown in Figure 2.1(a). T1, T2 and T3 represent the tests corresponding to the system's blocks, while R1 to R5 represent hardware resources required by the tests. Internal buses, dedicated test I/O pins, BIST test pattern generators and response analysers all constitute examples of hardware resources which might be shared during test. An arc between a test and a resource means that the resource is required by the test. For example, resources R1 and R3 are required in order to perform test T1.



(a) Resource allocation graph

(b) Test Compatibility graph

Figure 2.1: Resource allocation graph and test compatibility graph

The blocks of a system which can be tested simultaneously without generating any resource conflicts are said to be *test compatible*. The tests which are executed at the same time form a *test session.* The tests corresponding to test compatible blocks are said to be *resource compatible tests.* The test compatibility relations among the blocks of a system are represented using the *test compatibility graph* (TCG). Each block and its corresponding test are associated to a node in the TCG. An edge between two tests in the TCG signifies that the two corresponding tests are resource-compatible. Figure 2.1(b) shows the test compatibility graph for the resource allocation graph shown in Figure 2.1(a). Only tests T1 and T3 are compatible as they do not share any hardware resource. T1 is incompatible with T2 as they share resource R3, T2 and T3 are incompatible as they share R2. Thus, from the resource sharing perspective, T1 and T3 could be scheduled into the same test session.

2.1.2 Approximation Model for Power Dissipation During Test

In order to be considered by test scheduling algorithms, the power dissipation during test of a system block needs to be described using mathematical constructs which can be operated upon algorithmically. A *power profile* captures the power dissipation of a circuit over time when a sequence of test vectors is applied to the circuit. Power profiles represent cycle-accurate descriptions of power dissipation which makes them too complex to be considered by the test scheduling algorithms. Therefore simpler yet reliable approximate power models are needed. The following section analyses a commonly used power approximation model and justifies the need for a new power approximation model for power constrained test scheduling.

2.1.3 Global Peak Power Approximation Model

The power approximation model used by existing PCTS algorithms [CSA97, IC01, MWMV00, NAH00, LP01, Cha00, RCV00] will be referred in the rest of this chapter as the global peak power approximation model (GP-PAM). As shown in Figure 2.2, the GP-PAM basically flattens the power profile of a circuit to the

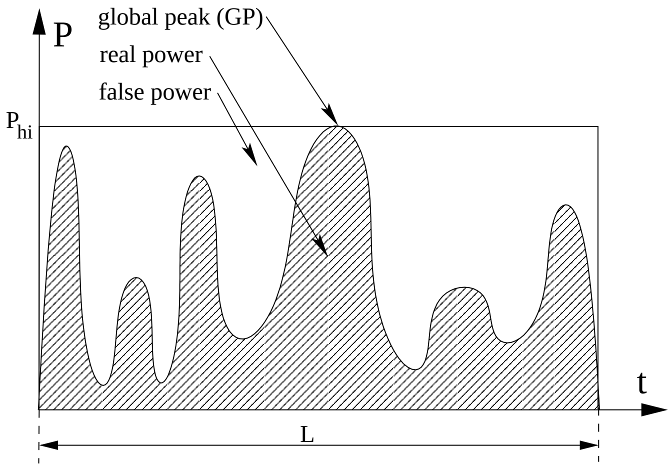


Figure 2.2: Global peak power approximation model

worst-case power dissipation value, i.e. its peak value. According to this model, the power profile of a block under test is described by the pair (P_{hi}, L), where P_{hi} is the global peak value of the power profile, and L is the length of the test sequence. This simple approximation model, although it guarantees that power dissipation is not under-estimated at any time, it introduces a high approximation error, indicated by the *false power* from Figure 2.2. The false power component introduced by the power approximation model leads to sub-optimal test concurrency, and hence to longer testing times.

The false power can be minimised by modifying the test power profiles and using a more accurate power approximation model to describe them. Section 2.2 will show how test concurrency can be increased by reducing the false power component of the power profile.

2.2 Power Profile Manipulation Technique

The previous section has shown that, regardless of its simplicity and reliability, global peak power approximation model leads to large approximation errors and consequently to low test concurrency. This can be avoided if the shape of the power profile can be changed such that it can be described using a more accurate yet simpler model. This section presents a power profile manipulation technique which enables increased test concurrency under power constraints. This technique consists of the following steps:

- **Test vector reordering (Section 2.2.1)** - initially, power profiles for the block tests are lowered by increasing the correlation between successive vectors; test vector reordering is used for peak power reduction, as well as for power profile reshaping;
- **Test sequence expansion (Section 2.2.2)** - additional test vectors are added to test sequences in order to lower further their power profiles. Only the test sequences which do not influence the test session length are extended, in order to preserve the total test time;
- **New power approximation model (Section 2.2.3)** - a power approximation model which is capable of exploiting the shape of power profiles corresponding to reordered test sequences is used to reduce the false power;
- **test sequence rotation (Section 2.2.4)** - finally, the low power profiles are rotated and piled up together such that the high power parts do not overlap with each other in order to obtain improved usage of the power constraint;

As it will be shown in the following sections, power profile manipulation performs low complexity operations on simple data structures, thus even for large amounts of test data corresponding to real-life circuits the required computational effort can be handled by typical desktop PCs.

2.2.1 Test Vector Reordering

As mentioned in Chapter 1, dynamic power represents one of the main components of power dissipation in CMOS circuits. The dynamic power dissipation is dependent on the switching activity, i.e. the average number of gate transitions per clock period [RP00]. The number of gate transitions depends on the switching activities at the inputs of the gate. Thus, the order in which test patterns are applied to the primary and pseudo-inputs influences the power dissipation in the circuit. Automatic testing equipment (ATE) as well as ROM-based BIST test pattern generators allow the order in which test vectors are applied to be changed.

The test vector reordering algorithm described below targets the following two objectives: $p$eak power minimisation, and achieving a power profile suitable for simple, reliable and accurate characterisation. Test sequences with lower power profiles allow higher test concurrency under a given power constraint. The use of accurate descriptions of power profiles can also increase the test concurrency under power constraints as it reduces the false power component, which is equivalent with having test sequences with lower power profiles.

The input to the test vector reordering algorithm is the input transition graph described below. Consider a test sequence TS with N test vectors and let $ITG = (\Psi, E)$ be the corresponding input transition graph. ITG is a complete directed graph with $|\Psi| = N$ nodes and $|E| = N(N-1)$ edges, where each node $V_i \in \Psi$ represents a vector in TS and each edge $(V_i, V_j) \in E$ represents the succession at the primary inputs of V_i and V_j. The ITG edges are labelled according to the amount of power dissipated in the circuit by the corresponding input transitions. The edge weights are computed differently depending on the test application scheme: test-per-clock or test-per-scan [AKS93b]:

- In a test-per-clock testing scheme, every clock cycle a test vector is applied to the primary inputs of the circuit. Each edge (V_i, V_j) in ITG is weighted with the power P consumed in the circuit during the transition of the primary inputs from V_i to V_j: $Weight(V_i, V_j) = P(V_i, V_j)$. All pairs $(V_i, V_j), i \neq j, V_i, V_j \in \Psi$ have to be simulated using a power estimation tool in order to compute the ITG edge weights.
- In a test-per-scan testing scheme, a test vector is first shifted into the scan chain during m clock cycles, where m is the length of the scan chain, the loaded stimulus data is applied to the combinational part of the block during

clock cycle $m+1$, and the circuit response is shifted-out during the next m clock cycles, while the next test vector is simultaneously shifted into the scan chain. Edge (V_i, V_j) in ITG is weighted with the power consumed by the simultaneous scan-out of V_i and scan-in of V_j. Cycle-accurate power simulation of all possible pairs of test vectors for test-per-scan schemes is very time consuming for large circuits. Therefore, a simpler power estimation method is needed. It was shown in [SOT00] that the *weighted transition count* (WTC) defined below is well correlated with the real power dissipation. The *weighted transition counts* (WTC) at scan-in and respectively scan-out corresponding to a m-bit wide test vector V_i are given by:

$$WTC_{scan-in}(V_i^{in}) = \sum_{j=1}^{m-1} (V_i^{in}(j) \oplus V_i^{in}(j+1))(m-j) \qquad (2.1)$$

$$WTC_{scan-out}(V_i^{out}) = \sum_{j=1}^{m-1} (V_i^{out}(j) \oplus V_i^{out}(j+1))j \qquad (2.2)$$

where $V_i(j)$ represents the j^{th} bit from vector V_i. V_i^{in} and V_i^{out} represent the scan-in vector and its test response (the scan-out vector). The WTC represents the number of transitions generated in the scan chain by shifting-in a test vector and shifting-out its test response. Figure 2.3 explains the weighted transition metric. The transitions in the test vector are marked with a $\wedge$ sign. The number of transitions generated in the scan chain during shift is reported for each transition in the test vector. For example, the transition between bits 5 and 6 causes 6 transitions in the scan chain as it ripples over the first 6 scan cells, while the transition between bits 0 and 1 causes only one transition in the scan chain, more specifically on the output of the first scan cell. The authors of [SOT00] have performed a set of experiments to prove the correlation between the WTC and power dissipation. A series of test vectors was applied to a circuit using a digital circuit simulator that stimulates the scan-in and scan-out operations. As test vectors were applied, the transitions in the circuit were counted. The number of transitions in the

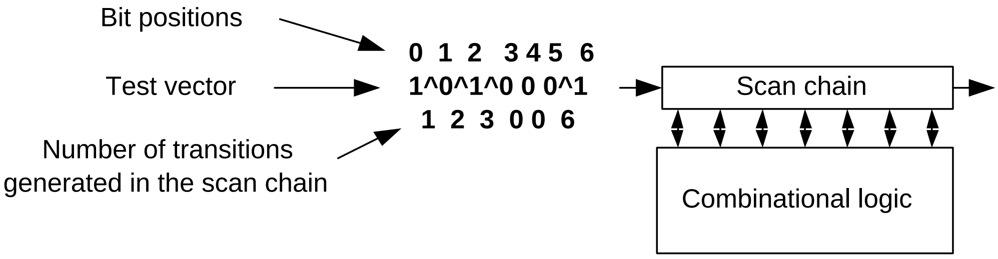


Figure 2.3: Weighted transition metric explained

circuit is a measure of the power consumed by the circuit. Figure 2.4 shows a plot of the sum of average weighted scan-in transitions and average weighted scan-out transitions for one of the ISCAS89 benchmark circuits. Due to the strong correlation between WTC and the power dissipation during shift, the strong correlation between WTC and the power dissipation during shift, reducing the WTC leads implicitly to a reduction in power dissipation during shift.

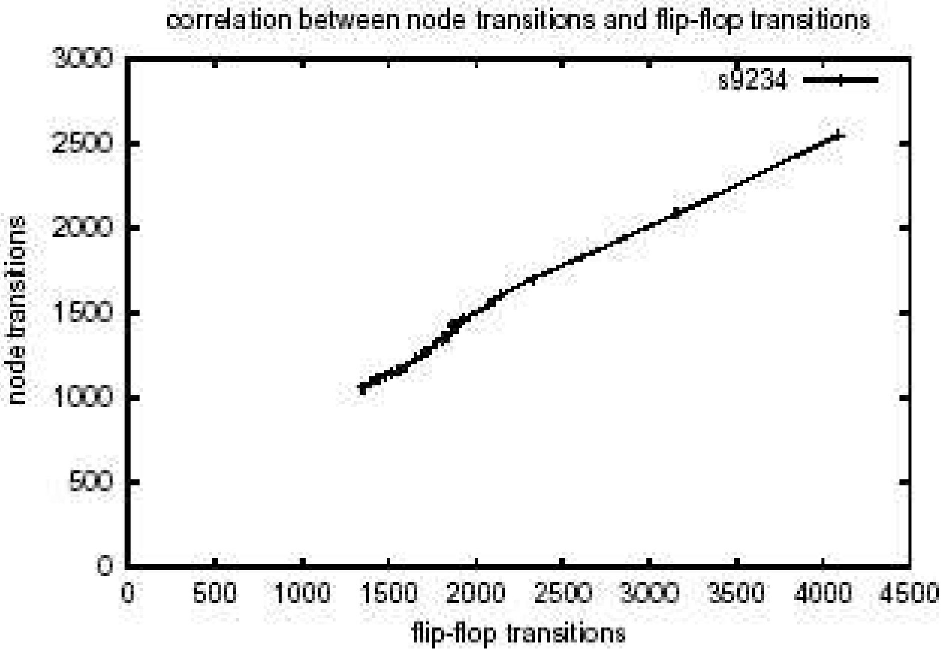


Figure 2.4: Correlation between internal node transition count and WTC(flip-flop transitions) [SOT00]

In a real manufacturing flow, accurate values for the ITG edge weights can be determined through direct measurement of the power dissipated by each of the blocks of a prototype system put under test conditions. Direct power consumption measurement eliminates the need of relying on approximate models such as the WTC. In this thesis however, the WTC was used as no prototype of the considered systems was available for conducting direct measurements. It should be noted that implementing the proposed power profile manipulation technique is completely external to the chip (both the vector order and the test schedule are handled entirely by the external tester), thus the chip can be manufactured prior to determining its power conscious test schedule.

Having computed the ITG edge weights, reordering the test sequence to reduce the switching activity at circuit's inputs can be formulated as the problem of finding a low cost Hamiltonian tour in ITG. As ITG is a complete directed graph, finding a low cost Hamiltonian tour in it represents an instance of the asymmetric travelling salesman problem, known as being NP-hard. Hence, a greedy search heuristic was implemented to determine a good solution to this problem. The algorithm starts from a randomly selected vector in the test sequence and, at each iteration, it selects the neighbouring node which generates the lowest power dissipation, i.e. the outgoing edge with the lowest weight. *Due to the greedy nature of the method adopted for traversing the ITG, the power profile corresponding to resulting path will exhibit an initial long low power part followed by a short high power part towards the end of the sequence.* This is because the edges with lower weights are added to the path in early iterations, leaving the edges with higher weights to the end of the profile. The shape of this power profile has the following features:

- The power profile of the reordered test sequence has lower peak power compared to a random path in the graph (such as the initial test sequence), as shown later in this chapter;
- The power profile of the reordered test sequence can be accurately described using a new power approximation model, presented in section 2.2.3, as shown later in Table 2.1 in Section 2.4;

The complexity of the reordering heuristic procedure is $\mathcal{O}(Nlog(N))$, where N is the number of test patterns.

2.2.2 Test Sequence Expansion

It was shown that power can be reduced by increasing the correlation between consecutive test vectors. In the previous section, the correlation between consecutive test vectors was improved by reordering the vectors in the test sequence. Another way of increasing the correlation in the test set is to find pairs of consecutive vectors which cause high power dissipation and insert additional vectors between them such that the number of transitions at the inputs of the circuit is reduced. The following example demonstrates test sequence expansion for test-per-clock and test-per-scan application schemes.

Example 3 *Consider a test-per-clock scheme with the following test vector pair: V_1 = 00001111 and V_2 = 11110000. The sequence (V1, V2) will produce 8 transitions at the inputs of the circuit. However, by inserting V^* = 00111100 between V_1 and V_2 will produce only 4 transitions in each of the two clock cycles. Consider now a test-per-scan scheme, and the following test vectors: V_1 = 1010 and V_2 = 0101, where V_1 is the test response which needs to be shifted-out from the scan chain and V_2 is the next test vector to be loaded into the scan-chain. Shifting the sequence (V1, V2) will produce 12 (6+6) transitions at the inputs of the circuit (see Equations 2.1 and 2.2 from Section 2.2.1). However, by inserting between V_1 and V_2 V^*_{in} = 0001, V^*_{out} = 1000, where V^*_{out} is the circuit test response for V^*_{in} will produce only in 7 transitions on circuits inputs during each of the two shifting cycles.*

The previous example has shown how switching activity at circuit's inputs can be reduced by inserting carefully selected test vectors in the test sequence. The additional test vectors are test vectors from the original test set which can be identified in linear time ($\mathcal{O}(N)$) based on the ITG edge weights. Test sequence expansion is suitable for non-partitioning test schemes for unequal test lengths

[CKS88]. Length differences among test sequences can be used to extend test sequences which are shorter than their test session's length in order to lower their power profiles without increasing the total test time. Low power profiles enable high test concurrency under a given power constraint, which leads to shorter test times. It should be noted that the amount of additional vectors has to be kept low as it affects the test data storage requirements.

2.2.3 Improved Power Approximation Model

Section 2.2.1 has shown how, test vector reordering can generate a test sequence with a regular power profile which has an initial long low power part followed by a short high power part towards the end of the sequence. This regular shaped power profiles can be accurately described using the simple approximation model shown in Figure 2.5.

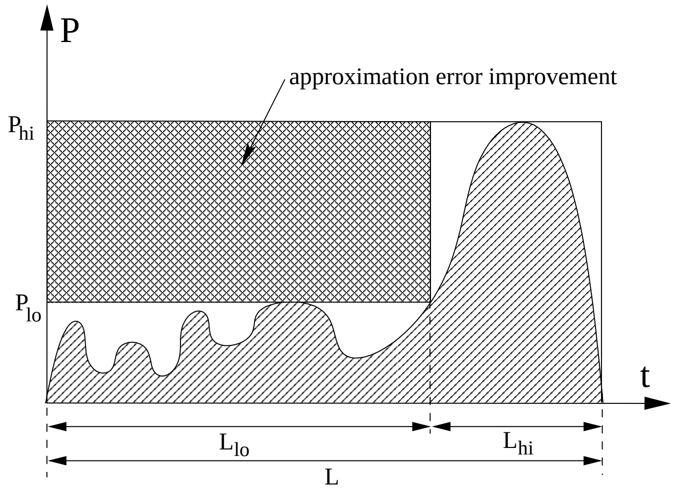


Figure 2.5: Two local peak power approximation model

By describing the low and high power parts of the profile based on their local peaks and lengths $((P_{lo}, L_{lo})$ and $(P_{hi}, L_{hi}))$, the value, position and size of each part of the profile are made available to power constrained test scheduling algorithms. The improvement in approximation accuracy over the GP-PAM, represented by the crosshatched rectangle in Figure 2.5, is given by $\Delta_{approx.improv} = (P_{hi} - P_{lo})L_{lo}$.

This power approximation model will be further referred to as the *two local peak power approximation model* (2LP-PAM) and will be represented by the 4-tuple $(P_{lo}, L_{lo}, P_{hi}, L_{hi})$. While P_{hi} is the same with the global peak value, several values for P_{lo} can be derived by changing the L_{lo} and L_{hi} ratio. Thus several 4-tuple descriptions are possible for the same power profile, however the optimum is the one with the largest $\Delta_{approx.improv}$. The optimum 4-tuple approximation can be computed in linear time ($\mathcal{O}(N)$) by varying L_{lo} in the $[0, L]$ interval.

2.2.4 Test Sequence Rotation

This section explains how several resource-compatible tests, can be combined into a test session by using test sequence rotation in order to exploit the particular shape of 2LP-PAM power profiles. Since 2LP-PAM offers information on the position and size of both low and high power parts, the power profiles can be rotated such that when added to a test session their high power parts do not overlap with the high power parts of power profiles of test sequences which are already in the test session. *This leads to higher test concurrency under power constraints* as illustrated in the following example.

Example 4 *Consider the power profiles shown in Figure 2.6 that belong to two compatible test sequences TS1 and TS2 that can be merged into the same test session. Figures 2.6(a) and 2.6(b) show the 2LP-PAM power profiles corresponding to TS1 and TS2. First, TS2 is added to the empty test session. Then, TS1 is rotated left by* L_{hi2} *vectors, as shown in Figure 2.6(c). The joint power profile obtained by adding the rotated TS1 to the test session is shown in Figure 2.6(d). Unlike the GP-PAM based approach where the maximum power dissipation for the test session composed of the two tests would be given by*

$$P_{Session}(GP-PAM) = \sum_{t_i \in Session} P_{hi}(t_i) = P_{hi_1} + P_{hi_2}$$

by using the 2LP-PAM, the maximum test session power dissipation becomes

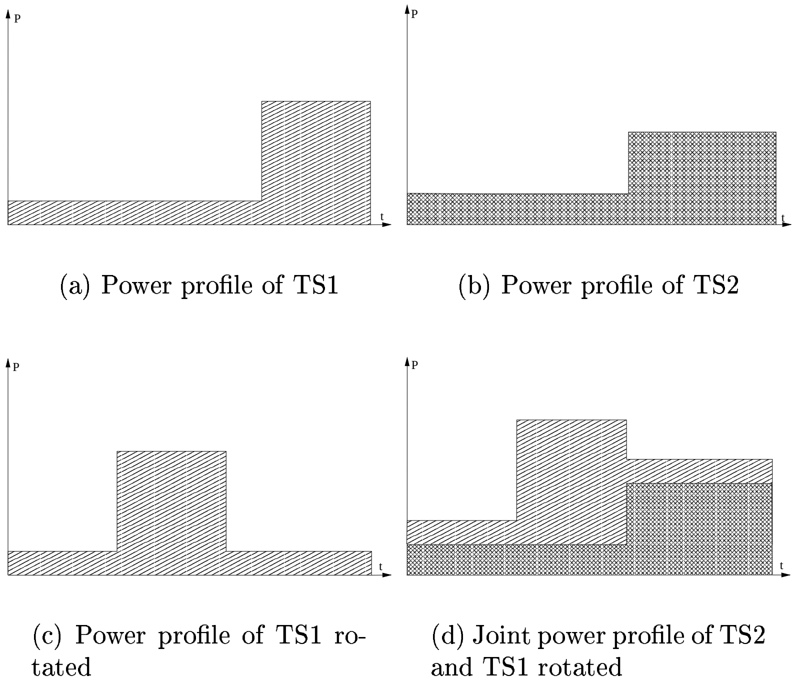
$$\begin{aligned} P_{Session}(2LP - PAM) &= \max_{t_i \in Session} (P_{hi}(t_i) + \sum_{t_j \in Session, t_j \neq t_i} P_{lo}(t_j)) = \\ &= max(P_{hi_1} + P_{lo_2}, P_{hi_2} + P_{lo_1}) < P_{hi_1} + P_{hi_2}. \end{aligned}$$

Thus, $P_{Session}(2LP-PAM) < P_{Session}(GP-PAM)$*. This example has shown how by controlled rotation of test sequences before adding them to a test session, the high power parts of their power profiles are uniformly spread over the entire test session length, rather than being piled up on top of each other, as in the case of the GP-PAM approach. Therefore, the joint power profile of the test session when using the 2LP-PAM, becomes lower and flatter, and hence more tests can be accommodated under the same power constraint. Test sequence rotation does not affect the peak or average power of a test sequence; rather, it helps test scheduling algorithms perform better allocation of tests in test sessions under power constraints, which leads to short test times.*

It should be noted, that *cyclic power profiles* are needed for test sequence rotation. A cyclic power profile starts and ends with the same test vector. Test sequence rotation consists in assigning a value to an *offset* parameter which specifies the initial vector in the rotated test sequence. Therefore, the computational effort associated with test sequence rotation is virtually non-existent.

2.3 Power Constrained Test Scheduling Using Power Profile Manipulation

Power profile manipulation was introduced in the previous section using the following elements: test vector reordering, test sequence expansion, two local peak power approximation model, and test sequence rotation. This section shows how power profile manipulation can be integrated into existing power constrained test scheduling algorithms.



(a) Power profile of TS1

(b) Power profile of TS2

(c) Power profile of TS1 rotated

(d) Joint power profile of TS2 and TS1 rotated

Figure 2.6: Test sequence rotation

The non-partitioning test scheduling algorithm for unequal test lengths proposed in [CSA97] will be extended for use in conjunction with power profile manipulation. As in [CSA97], it is assumed that a new test session cannot start before all tests in the current test session are completed, even if the resources required for the next test session are already available. A practical reason for this restriction is that enabling interruption of a test session to start new tests will increase the complexity of the test controller. The extended PCTS algorithm starts with a preprocessing step (lines 1 to 5 in Algorithm 1), where all test sequences are *reordered* and *extended* with a small, fixed number (5 was used for the experiments reported in this chapter in order to show that a very small number of additional test vectors is enough in order to achieve significant reductions in peak power) of additional test vectors. The resulting power profiles for both the reordered and the reordered-and-extended test sequences are modelled using the 2LP-PAM. The high power parts of the 2LP-PAM power profiles are then moved to the beginning of the test sequence by rotating them to the left by L_{lo} vectors (line 4).

Algorithm 1 PCTS Using the Proposed Power Profile Manipulation

```
INPUT: test compatibility graph TCG, power constraint $P_{constr}$
   and additional vector count AVC
OUTPUT: power constrained test schedule
1  for every test sequence $TS_i${
2      reorder $TS_i$ and extend by AVC vectors the reordered test sequence
3      compute the 2LP-PAM approximations for both reordered
                   and reordered-and-extended test sequences
4      rotate left the power profile by $L_{lo}$ vectors
5  }
6  compute clique set $\Omega$ for TCG
7  for every clique $C_i \in \Omega$
8      compute power compatible lists $PCL_i$ for $C_i$ and $P_{constr}$
                   using Algorithm 2
9  compute power constrained test schedule
                   as a minimum cost cover of the PCLs set
```

Next, the algorithm shown in Figure 1 determines all cliques (i.e. completely connected subgraphs) of TCG (line 6). The TCG cliques represent maximal groups of test compatible blocks. For each TCG clique, the algorithm computes all maximal ordered subsets which comply with the given power constraint, refers to them as the *power compatible lists* (PCLs) (lines 7 and 8). The following example explains PCLs.

Example 5 *Consider the system with the TCG and 2LP-PAM power profiles shown in Figure 2.7. The cliques in this case are* (T_2, T_5)*,* (T_1, T_4, T_5)*,* (T_3, T_4, T_5)*. The maximality requirement of the PCLs means that no other test can be added to them without exceeding the power constraint. The tests in a PCL are arranged in the descending order of their length. Consider the test compatible clique composed of tests* (T_3, T_4, T_5)*. Figures 2.8(a) and 2.8(b) show the PCLs corresponding to the test sequences in the clique for a power constraint of 10 using 2LP-PAM and respectively GP-PAM approximations. The ordered list of test sequences is* (T_5, eT_3, eT_4)*, where* eT_3 *and* eT_4 *represent the* reordered-and-extended *test sequences* T_3 *and* T_4*.* T_5 *is the longest test sequence in the clique and hence it determines the length of the test session. Test sequence* T_5 *is not extended in order to preserve the original*

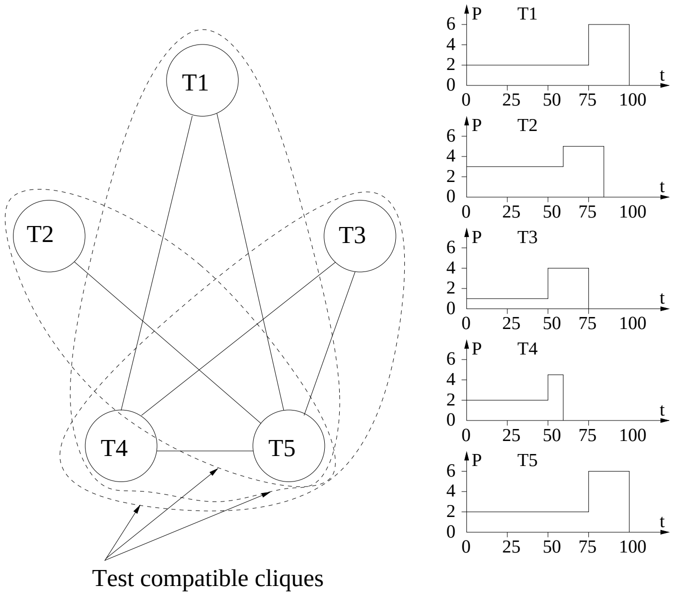
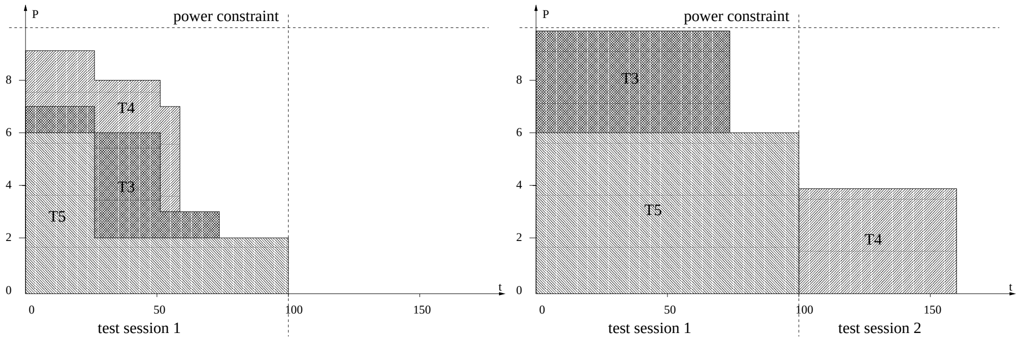


Figure 2.7: Example: Test Compatibility Graph



(a) PCL for (T_3, T_4, T_5) using the 2LP-PAM

(b) PCL for (T_3, T_4, T_5) using the GP-PAM

Figure 2.8: Example: Power compatible lists under the 2LP-PAM and GP-PAM

length of the test session. However, T_3 and T_4 can be extended as their lengths are smaller than the length of the session to which they are about to be assigned to. T_5 is added to the empty test session. The next test in the ordered list, eT_3, is rotated right such that its high power part does not overlap with the high power part of T_5, and then it is added the test session. Finally, test eT_4 is rotated right such that its high power part does not overlap with the high power parts of the two test sequences

Algorithm 2 Power Compatible Lists

```
ALGORITHM: Power Compatible Lists
INPUT: test compatible clique C and the power constraint $P_{constr}$
OUTPUT: power compatible lists PCL for C and $P_{constr}$
PCL = $\phi$
for every subset $S_i \subset$ C {
    Offset = 0; Session = $\phi$
    compute $pcl_i$ by sorting tests in $S_i$ in descending order of their lengths
    for every test $T_j \in pcl_i$ {
        if ($L_{loj} \geq$ Offset) rotate right $T_j$ by Offset vectors else Offset = 0
        Session = Session $\cup T_j$
    }
    compute maximum power dissipation $P_{max}$ for Session
    if ($P_{max} < P_{constr}$) then {
      MaximalSet = TRUE
      for every $T_j \in C$ and $T_j \notin pcl_i$ shorter than
            the longest test sequence in $pcl_i$ {
          if ($L_{loj} \geq$ Offset) rotate right $T_j$ by Offset vectors
          Session' = Session $\cup T_j$
          compute maximum power dissipation $P'_{max}$ for Session'
          if ($P'_{max} \leq P_{constr}$) MaximalSet = FALSE;
      }
      if (MaximalSet = TRUE) PCL = PCL $\cup pcl_j$
}
```

already in the test session. The maximum value of the resulting power profile is $P_{max} = P_{hi5} + P_{lo3} + P_{lo4} = 9$*, which is less than the power constraint. This means that by using 2LP-PAM power profiles all tests in the clique can be scheduled in the same 100 clock cycle test session under the specified power constraint, while by using GP-PAM power profiles, two test sessions, summing up 160 clock cycles, are required to cover all tests in the clique.*

The PCLs for each clique C from TCG are computed in compliance with the given power constraint using the algorithm shown in Algorithm 2. For each subset of C the algorithm computes the optimum arrangement of its tests making use of the test sequence rotation method described in Section 2.2.4. The Offset variable guides the rotation of the test sequences to be inserted into the current test session.

For the longest test(s) in the test subset, the original *reordered-only* test sequence is used in order to preserve the length of the test session, while shorter test sequences are *reordered and extended* in order to lower their power profiles without increasing the length of the test session. The maximal power compatible subsets are then added to the set of PCLs.

Finally, finding the optimum test schedule under the given power constraint is reduced to the problem of finding a minimum cost cover for the PCLs set (line 9 in Algorithm 1), where the cost associated to each PCL is the length of the longest test in the PCL, i.e. the test session length. The minimum cost covering problem can be formulated as an integer linear programming (ILP) problem and solved using *lp_solve* [Sch97].

This section has shown how power profile manipulation can be integrated into power constrained test scheduling algorithms. Although the integration was detailed for the algorithm presented in [CSA97], the proposed approach can be included into *any* other existing power constrained test scheduling algorithm [MWMV00, NAH00, LP01, IC01, Cha00, RCV00], to leverage its performance.

2.4 Experimental Results

This section describes the experiments performed to asses the efficiency of the power profile manipulation technique. The algorithms were implemented in C++ and ran on an AMD 1.2Ghz Linux workstation with 384Mb RAM. Due to the simplicity of the WTC model, determining the ITG edge weights does not require a high computational effort. For example, the ITG weights for a test set of 7000 vectors, 1000-bit wide for a test-per-scan scheme were computed in less than 120 seconds. Reordering the vectors of the same sequence was performed in 71 seconds, while expanding the sequences by 5 test vectors as well as computing the two local peak approximation were each performed in less than one second.

The first set of experiments compares the 2LP-PAM with the GP-PAM in terms

Circuit	$\Delta_{approx.improv}$(%)		**Peak pow. reduction (%)**
	test-per-clock	**test-per-scan**	**by extending with 5 vec.**
c1355	3.97	10.93	13.89
c1908	9.59	7.52	8.3
c432	14.04	23.23	13.7
c499	7.36	8.83	5.14
c6288	2.76	13.53	7.24
c7552	21.94	4.35	11.51
s5378	N/A	8.4	9.02
s9234	N/A	6.75	3.53
s13207	N/A	14.41	4.75
s15850	N/A	11.77	7.5
s35932	N/A	31.51	38.16
s38417	N/A	0	7.87
s38584	N/A	0	2.75

Table 2.1: Approximation accuracy improvement for test-per-clock and test-per-scan testing schemes

of approximation accuracy improvement, which basically shows how much "false power" is saved by using 2LP-PAM power profiles. Test-per-clock and, where suitable, test-per-scan test sequences for the largest ISCAS [bl] benchmark circuits, generated using ATALANTA [Teca], were reordered using the algorithm described in section 2.2.1. The power profiles of the reordered test sequences were approximated using the 2LP-PAM and the GP-PAM. Columns 2 and 3 in Table 2.1 show the improvement in approximation accuracy of the 2LP-PAM over the GP-PAM for all benchmark circuits used in the experiments. Next, the efficiency of the test sequence expanding method is evaluated. The reordered test sequences were extended with 5 test vectors which led to the peak power reductions reported in column 4 in Table 2.1.

The next set of experiments evaluates the performance improvement which can be achieved by integrating power profile manipulation into the power constrained test scheduling algorithm presented in [CSA97]. The modified and the original algorithms were applied on hypothetical systems. Each system was represented as a set of embedded blocks and a randomly generated test compatibility graph. The embedded blocks were selected randomly from the ISCAS benchmarks [bl]. More details on the ISCAS benchmarks are given in Appendix D. Systems with 8 to 16 blocks were considered in the experiments. The original PCTS algorithm from [CSA97] was applied on GP-PAM power profiles of the unordered test sequences. The resulting test times for a wide range of power constraints are reported in

Block cnt.	**Power constr.(mW)**	$T_{orig.PCTS}$ **(clks)**	$T_{modifPCTS}$ **(clks)**	**Test time red. (%)**
8	330	594	477	19.7
8	346.5	594	477	19.7
8	363	594	477	19.7
10	412.5	720	604	16.11
10	429	720	604	16.11
10	445.5	720	604	16.11
12	445.5	1069	714	33.21
12	561	952	598	37.18
12	577.5	831	598	28.04
14	363	1070	836	21.87
14	445.5	948	715	24.58
14	544.5	831	598	28.04
16	412.5	1545	1074	30.49
16	511.5	1428	952	33.33
16	544.5	1428	836	41.46

Table 2.2: Experimental results for the test-per-clock testing scheme (ISCAS85 benchmarks)

Block cnt.	**Power constr.(WTC)**	$T_{orig.PCTS}$ **(clks)**	$T_{modifPCTS}$ **(clks)**	**Test time red.(%)**
8	450	870514	545696	37.31
8	550	718986	521173	27.51
8	600	718986	466877	35.06
8	650	533826	406419	23.86
10	500	958550	624515	34.84
10	650	642928	545696	15.12
10	750	588524	460715	21.71
10	900	533826	381896	28.46
12	500	991888	649038	34.56
12	600	916124	649038	29.15
12	800	642928	545696	15.12
12	850	567164	491400	13.35
14	650	903852	664288	26.50
14	700	794750	612645	22.91
14	750	794750	579307	27.10
14	850	664288	555186	16.42
16	700	903852	700681	22.47
16	800	903852	667343	26.16
16	850	828088	649038	21.62
16	900	828088	570219	31.14

Table 2.3: Experimental results for the test-per-scan testing scheme (ISCAS89 benchmarks)

column 3 of Tables 2.2 and 2.3 respectively. Integration of power profile manipulation into the PCTS algorithm from [CSA97] reduced the previous test times to the values reported in column 4 of Tables 2.2 and 2.3 respectively. The test time reductions achieved by using power profile manipulation in conjunction with the original algorithm are reported in column 5. As shown in the experimental results, power profile manipulation reduced test time by up to 41%.

2.5 Concluding Remarks

Two main research directions can be identified in the area of low power test. The first research direction considers power dissipation during test an optimisation objective. The second direction considers power dissipation as a design constraint, while test time becomes the minimisation objective. The power profile manipulation technique presented in this chapter bridges these two directions. Test vector reordering is used to lower and reshape the test power profiles. Test sequence expansion further lowers the power profiles of shorter tests which do not affect the total test time. The two local peak power approximation model provides accurate descriptions for power profiles corresponding to reordered and expanded test sequences. The 2LP-PAM power profiles are exploited by test sequence rotation in order to increase test concurrency under a specified power constraint. Since the power profile manipulation technique is orthogonal to the test scheduling policy and the test set values, the distinctive feature of the proposed solution is that it can be embedded into existing power constrained test scheduling algorithm to improve its performance.

Chapter 3

Low Power Mixed-Mode BIST

In the previous chapter it was shown how test set transformations such as test vector reordering, test sequence rotation and test sequence expansion can be used to enhance existing power constrained test scheduling algorithms, thus reducing the overall test time. This chapter will present another test set transformation for reducing power dissipation during test.

Testing modern chips using external testing equipment is increasingly expensive due to the raising disproportion between chip pad counts and circuit complexity [ZDR00a]. Hence, there is an increasing need for low-cost test systems which enhance the traditional test methods in terms of reduced dependency on physical probes, at-speed test capabilities and test portability. Built-in self-test (BIST) represents a possible solution to this problem. In a BIST framework, on-chip hardware generates test vectors and evaluates test response data [ABF90, AKS93a, AKS93b, BMS86, BA00], hence eliminating test access problems, limitations of the I/O channel and the need for expensive external test equipment. A practical BIST scheme is required to guarantee complete fault coverage while minimising the following three parameters: test application time, area overhead and test data storage. Several test generation schemes have been proposed to accomplish various trade-offs between these parameters. These solutions range from exhaustive [BCR83, WH92, WM86] and pseudorandom techniques [BMS86] to deterministic automatic test pattern generation (ATPG) techniques [AJ89, DPA84, DM81]. Exhaustive and pseudorandom techniques do not use any storage but have long test application times. Deterministic techniques achieve complete fault coverage in a relatively short time but require significant test data storage.

Mixed-mode test generation [DG91, HRT$^+$95, RTZ98, Koe91, TM96, WK96] overcomes the limitations of previously mentioned approaches. In a mixed-mode BIST scheme, a limited number of pseudorandom vectors are used to cover the easy-to-detect faults, while the few remaining random pattern resistant faults are detected with a small number of deterministic vectors. Unlike other approaches, such as test point insertion [TCLB98, TM99], mixed-mode techniques achieve complete fault coverage without any circuit modifications which may affect the performance of the circuit. Moreover, the trade-off between test data storage and testing time can be tuned by varying the ratio between the number of deterministic and pseudorandom patterns. Various implementations of mixed-mode BIST TPGs have been proposed recently. Bit-fixing [TM01] and bit-flipping [WK96] techniques generate deterministic test patterns by altering some bits in the output sequence of an LFSR. Although these techniques provide high quality tests, the corresponding BIST hardware is very dependent on the test set, thus any change of the test set requires a complete re-synthesis of the BIST hardware. The work presented in [Koe91] uses the same LFSR to generate both pseudorandom and deterministic patterns. The deterministic set of patterns is encoded as LFSR seeds computed for test cubes (incompletely specified test patterns) of random pattern resistant faults. Hellebrand et al. [HRT$^+$95] extended the LFSR re-seeding technique to multiple-polynomial LFSRs (MP-LFSR) which reduces the storage requirements and the LFSR length when compared to the re-seeding of single-polynomial LFSRs. In the method proposed in [HRT$^+$95], a concatenated group of test cubes with a maximum of s specified bits is encoded using approximately s bits representing a seed and a feedback polynomial identifier. Rajski et. al. [RTZ98] adapted the MP-LFSR TPG for multiple scan-chain designs with boundary scan chains.

Although mixed-mode test BIST test pattern generators (TPG) achieve complete fault coverage with short test sequences, they cause intense switching activity in the circuit under test while shifting the test patterns into the scan chains. This is because LFSR-generated sequences contain a large number of transitions between consecutive bits which ripple on the inputs of the circuit under test during shift. As mentioned in Chapter 1, sustained intense switching activity causes overheating and electro-migration which can permanently damage the chip under test or seriously affect its reliability [Zor93, Whe00, BA00, NAH02]. Several techniques have been recently proposed for reducing switching activity during test in BIST environments. Some of these techniques require modifications of the circuit under test, while others do not. This chapter discusses only the techniques which do not require modification of the circuit under test. The most relevant techniques in this category are summarised in the following. The TPG proposed in [WG99] reduces the number of transitions between consecutive bits in the scan test patterns by combining k stages of an LFSR through a k-input AND gate and a T flip-flop. Although this method reduces significantly the power dissipation during shift, it requires long test application times in order to achieve reasonably high fault coverage. The TPG proposed [MGL+99], filters the non-detecting vectors from a LFSR sequence, hence reducing the energy consumption of the entire test sequence. However, this TPG does not reduce power dissipation on a per test pattern basis.

This chapter presents a low power mixed-mode TPG architecture which aims to overcome the shortcomings of previous low power BIST TPGs. MP-LFSR reseeding, explained in Section 3.1, method proved to be very effective in encoding deterministic test data necessary for achieving complete fault coverage with short test sequences. Hence, it was used as the basis for the proposed architecture. The output sequences of two MP-LFSR structures operated in parallel are combined through an AND or OR gate to produce a sequence with consistently 25% less transitions compared to the original sequences. An extensive set of experiments has been conducted on several benchmark designs using commercial synthesis and simulation tools in order to assess the efficiency of the proposed TPG.

3.1 LFSR Re-Seeding

Given an LFSR and the initial values of its flip-flops, referred to as a LFSR seed, clocking the LFSR produces a deterministic sequence at its output according to the polynomial function implemented by the feedback network of the LFSR and the initial seed. This suggested the idea that the output sequence of an LFSR could be manipulated by changing the initial seed and/or feedback network.

Let $h(x) = x^k + \sum_{i=0}^{k-1} h_i x^i$ be the feedback polynomial of the LFSR and

$$A(t) = \begin{bmatrix} a_0(t) \\ \vdots \\ a_{k-1}(t) \end{bmatrix}$$

the state of the shift register at clock t. The system can be described as

$$A(t+1) = T_S A(t)$$

where

$$T_S = \begin{bmatrix} 0 & 1 & 0 & \dots & 0 & 0 \\ 0 & 0 & 1 & \dots & 0 & 0 \\ \vdots & \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & \dots & 1 & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ 1 & h_1 & h_2 & \dots & h_{k-2} & h_{k-1} \end{bmatrix}$$

represents the state transition matrix for the given LFSR. Let $C = (c_0, ..., c_{m-1}) \in \{0, 1, X\}^m$ be a test cube and $S(C) = \{i \mid c_i \neq X\}$ the set of specified bits of C. C can be generated using the LFSR described by T_S if the following system of linear equations is consistent:

$$c_i = a_i = \left[{T_S}^i A(0)\right]_1, \quad for\ all\ i \in S(C) \tag{3.1}$$

where $\left[{T_S}^i A(0)\right]_r$ denotes the r-th component of $\left[{T_S}^i A(0)\right]$. The solution of this system of equations, if any, represents the initial seed for which the LFSR described by $h(x)$ will generate a test vector covered by cube C. The system of equations 3.1 can be solved using the Gauss-Jordan elimination, thus the complexity of finding an initial seed is $\mathcal{O}(n^3)$, where n is the degree of the feedback polynomial. Under certain conditions it may be impossible to solve the above system of linear equations. This situation, known as *test pattern lockout*, is very likely to occur if the number of specified bits in the test cube exceeds the number of seed bits, i.e. the number of derived equations exceeds the number of independent variables. Test pattern lockout can also occur even if the number of equations is smaller than the number of independent variables, if two or more equations are conflicting, i.e. their left hand sides are dependent and their requested (right hand side) values differ. However, the system of equations has a solution when the equations are dependent if their requested values match the dependency. Hence, the probability of a test pattern lockout is upper bounded by the probability of dependency in the system of linear equations. In order to ensure full fault coverage, it is necessary to avoid test pattern lockouts. This can be achieved by increasing the size of the LFSR until it exceeds the maximum expected number of specified bits per test cube. It was shown in [Koe91] that the lockout probability for $n = s + 20$ is smaller than 10^{-6}. Thus, when using LFSR re-seeding, the storage requirements for encoding a test cube are determined only by the number of specified bits in the test cube. The following example illustrates the procedure for computing the initial seed for a given test cube and feedback polynomial.

Example 6 *Consider test cube $C = (X,1,1,X,X,0,X,X,1,X)$ and a 4-stage LFSR with the characteristic polynomial given by $h(x) = x^4 + x^3 + 1$. The state transition matrix corresponding to this LFSR will be*

$$T_S = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 \end{bmatrix}$$

The equations which need to be solved in order to find the initial seed are

$$\begin{bmatrix} c_0 = X \\ c_1 = 1 \\ c_2 = 1 \\ c_3 = X \end{bmatrix} = {T_S}^0 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

and

$$\begin{bmatrix} c_5 = 0 \\ c_6 = X \\ c_7 = X \\ c_8 = 1 \end{bmatrix} = {T_S}^5 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 0 \\ 0 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Solving the equations above will produce the following solution, i.e. initial seed

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ a_3 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 1 \end{bmatrix}$$

The test pattern generated by the given LFSR starting from the computed initial seed will be $P = (0, \underline{1}, \underline{1}, 1, 1, \underline{0}, 1, 0, \underline{1}, 1)$*, where the underlined positions represent the specified bits from the original test cube. Hence, a 10-bit test cube can be encoded as a 4-bit LFSR seed.*

The LFSR re-seeding scheme was extended in [HRT$^+$95] to MP-LFSR re-seeding. A typical MP-LFSR is shown in Figure 3.1. Different feedback configurations, i.e. different LFSRs, can be selected by setting the appropriate values on the $S[0 : k-1]$ lines. Up to 2^q configurations can be encoded using a q-bit polynomial identifier. Theoretical analysis presented in [HRT$^+$95] shows that, for a feedback network which can implement 16 polynomials, the probability of not finding a seed for a test cube C with $s = |S(C)|$ specified bits using a $s + 4$ bit LFSR is less

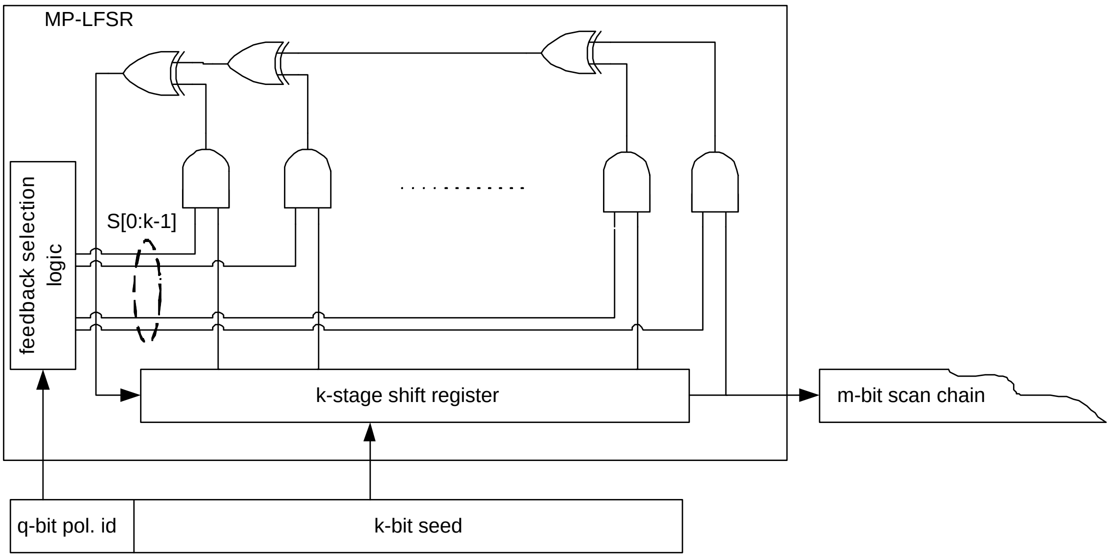


Figure 3.1: Multiple-polynomial LFSR architecture [HRT+95]

than 10^{-6}. In the MP-LFSR re-seeding approach, a deterministic test cube C is encoded as a q-bit polynomial identifier and a k-bit seed.

3.2 AND/OR Masking

LFSR-generated sequences contain a large number of transitions between consecutive bits due to their pseudorandom nature. While shifting the test patterns into the scan chain, these transitions ripple on the inputs of the circuit, hence causing intense switching activity in the combinational logic. This sections presents a method which exploits the "masking" properties of AND and OR gates in order to reduce the number of transitions occurring in the scan chain during shift.

Definition 1 Given a logic signal S, the *signal probability* $P_1(S)$ represents the average fraction of clock cycles in which signal S is 1. Analogously, $P_0(S)$ represents the average fraction of clock cycles when signal S is 0.

S is a binary signal, thus $P_0(S) + P_1(S) = 1$. If signal S is generated by a random source, its 0 and 1 signal probabilities are equal $P_0(S) = P_1(S) = 0.5$

Definition 2 The *transition probability* of a signal S, $P_{tr}(S)$ represents the average fraction of clock cycles when the current value of S is different than its previous value.

Assuming temporal independence between consecutive values of S, its transition probability can be computed as

$$P_{tr}(S) = P_0(S) \times P_1(S) + P_1(S) \times P_0(S) \tag{3.2}$$

The transition probability of a random logic signal S is

$$P_{tr}(S) = 0.5 \times 0.5 + 0.5 \times 0.5 = 0.5 \tag{3.3}$$

Assuming two random and mutually independent signals S_A and S_B, and their AND composition $S_{AND} = S_A \ AND \ S_B$. Given the mutual independence of S_A and S_B, the signal probabilities of S_{AND} can be computed as follows:

$$\begin{aligned} P_1(S_{AND}) &= P_1(S_A) \times P_1(S_B) = 0.25 \\ P_0(S_{AND}) &= 1 - P_1(S_{AND}) = 0.75 \end{aligned}$$

According to equation 3.2, the transition probability of S_{AND} is given by

$$P_{tr}(S_{AND}) = 2 \times P_1(S_{AND}) \times P_0(S_{AND}) = 0.375 \tag{3.4}$$

The signal and transition probabilities of the OR composition of signals S_A and S_B, $S_{OR} = S_A \ OR \ S_B$ can be computed in a similar way:

$$\begin{aligned} P_0(S_{OR}) &= P_0(S_A) \times P_0(S_B) = 0.25 \\ P_1(S_{OR}) &= 1 - P_0(S_{OR}) = 0.75 \\ P_{tr}(S_{OR}) &= 0.375 \end{aligned} \tag{3.5}$$

From equations (3.3), (3.4) and (3.5) it can be concluded that both AND and OR compositions of two independent random signals have 25% lower transition probability than the original signals. Therefore, these composition functions will be also referred to as AND/OR masking because they "mask" a quarter of the transitions of the "source" signals. The randomness of LFSR-generated sequences associated with the previous observation suggested the integration of AND/OR masking into BIST TPG in order to reduce the number of transitions in the scan chain during shift, and consequently the power dissipation in the circuit under test.

3.3 AND/OR Masking in Mixed-Mode BIST

In the previous section it was shown how AND/OR composition applied on LFSR-generated sequences can be used to produce sequences with lower number of transitions. This section presents a methodology which combines AND/OR composition with mixed-mode test generation in order to obtain a low power TPG achieving complete fault coverage with short test sequences.

There are two problems which need to be addressed:

- The first problem is to ensure that test patterns resulted from AND/OR composition still have enough randomness for covering the easy-to-detect faults within a reasonable amount of time.
- The second problem is to guarantee that AND/OR composition can generate test patterns covered by precomputed test cubes for random pattern resistant faults, in order to achieve complete fault coverage.

While the first problem is solved easily by constraining the two sequences which are combined to be mutually independent, the second problem requires a more elaborate solution which is detailed in the following.

A test pattern covered by a given test cube C preserves the specified bits of C. In order to obtain such a test pattern through AND/OR composition, the following constraints must be imposed for two "source" patterns which will be combined:

1. Each *controlling value*(with respect to the chosen composition function) specified in the precomputed test cube has to appear at least in one of the "source" patterns.
2. Each *non-controlling value*(with respect to the chosen composition function) specified in the test cube must appear in both "source" patterns.

Definition 3 The *non-controlling mask cube* ($NM(C)$) of a given test cube C with respect to a given composition function f_{comp} (AND or OR) is the cube which has as specified bits only the non-controlling values (with respect to f_{comp}) specified in C.

Assuming $f(X, X) = X$, where X represents a "don't care"(unspecified) bit and f_{comp} is either AND or OR, the following equation holds for a given test cube C:

$$f_{comp}(C, NM(C)) \;=\; C \tag{3.6}$$

Theorem 1 Assuming C and $NM(C)$ is a precomputed test cube and, respectively, its non-controlling mask cube with respect to a given composition function f_{comp}, any two patterns P and MP (the "mask pattern") covered by C and $NM(C)$ respectively will produce through f_{comp} composition a pattern covered by C.

The proof for Theorem 1 is immediate. P preserves both controlling and non-controlling values of C with respect to f_{comp}, while MP preserves the non-controlling values of C with respect to f_{comp}, hence the result of the composition will preserve all specified bits of C. P and MP can be generated by re-seeding two LFSR structures with seeds derived from C and $NM(C)$ as explained in Section 3.1.

The following example illustrates the procedure for generating the "mask pattern" for a given test cube and composition function.

Example 7 *This example shows how to generate the "mask pattern" MP corresponding to the AND composition for the deterministic cube C =* (X, 1, 1, X, X, 0, X, X, 1, X) *from Example 6. The "mask cube" for AND composition is $NM(C)$*

= (X, 1, 1, X, X, X, X, X, 1, X). *Consider* $h_m(x) = x^3 + x + 1$ *as the characteristic polynomial LFSR which will be used to generate* MP. *The initial seed for* $NM(C)$ *is computed as explained in the Section 3.1. The state transition matrix associated to the LFSR is:*

$$T_{Sm} = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}$$

The equations needed to compute the initial seed are:

$$\begin{bmatrix} c_0 = X \\ c_1 = 1 \\ c_2 = 1 \end{bmatrix} = {T_{Sm}}^0 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

and

$$\begin{bmatrix} c_6 = X \\ c_7 = X \\ c_8 = 1 \end{bmatrix} = {T_{Sm}}^6 \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix}$$

Solving these equations will lead to the following seed:

$$\begin{bmatrix} a_0 \\ a_1 \\ a_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$$

The "mask pattern" which will be generated using this seed is MP = (1, $\underline{1}$, $\underline{1}$, 0, 0, 1, 0, 1, $\underline{1}$, 1), *where the underlined positions represent the specified bits from* $NM(C)$. *The pattern resulted by AND composition of* P, *computed in Example 6, and* MP *is* P' = (0, $\underline{1}$, $\underline{1}$, 0, 0, $\underline{0}$, 0, 0, $\underline{1}$, 1), *where the underlined positions show the specified bits from the original test cube* C, *preserved through AND composition.*

The pattern P *from Example 6 generated using a single LFSR TPG contains 5 transitions between consecutive bits, while the pattern* P' *produced by the proposed method for the same deterministic cube* C *by applying AND masking on the patterns generated by two LFSRs (one for* C *and one for* $NM(C)$*) contains only 3* *transitions between consecutive bits.*

Thus, for a given test cube C, LFSR re-seeding can be used to generate a suitable pair of "source" patterns for AND/OR masking in order to reduce the number of transitions in the scan chain while preserving the specified bits of C, and hence its fault coverage. The following Section presents a TPG architecture which implements this method.

3.4 Dual MP-LFSR Test Pattern Generator

In Section 3.2 it was shown how AND/OR composition of independent pseudorandom "source" sequences produces sequences with lower transition count. Next, in Section 3.3 it was shown how AND/OR composition can be combined with LFSR re-seeding in order to generate test patterns covered by precomputed test cubes. This section will present a mixed-mode BIST TPG architecture which combines AND/OR composition and LFSR re-seeding.

The basic idea is to have two distinct LFSRs which are operated in parallel in order to generate the pair of "source" sequences for AND/OR composition. The randomness of the test patterns resulted from the AND/OR composition, which ensures rapid coverage of easy-to-detect faults, can be achieved by using different primitive characteristic polynomials for the two LFSRs. Test patterns covered by deterministic test cubes for random pattern resistant faults are generated by re-seeding the two LFSRs. One LFSR, referred to as the "main LFSR", generates patterns covered by the precomputed test cubes, while the "secondary LFSR" generates patterns covered by the non-controlling mask cubes of the precomputed cubes with respect to the chosen composition function. Hence, in this approach, each deterministic test cube is encoded using two initial seeds, one for each LFSR. In Section 3.1 it was shown that an MP-LFSR with 16 feedback polynomials and $s + 4$ flip-flops can be used to encode a set of test cubes with a maximum of s specified bits per test cube, which would otherwise require a $s + 20$-bit single polynomial LFSR. Hence, in order to reduce the required length of the two LFSRs, and implicitly the test data storage requirements, the LFSRs are replaced with shorter MP-LFSRs capable of covering the same deterministic test cubes. The dual MP-LFSR architecture encodes each deterministic test cube as two *(polynomial identifier, initial seed)* pairs, one for each MP-LFSR, and a mask selection bit which is used to select between AND and OR composition. The complete TPG architecture with two MP-LFSR structure and selectable AND/OR composition is shown in Figure 3.2.

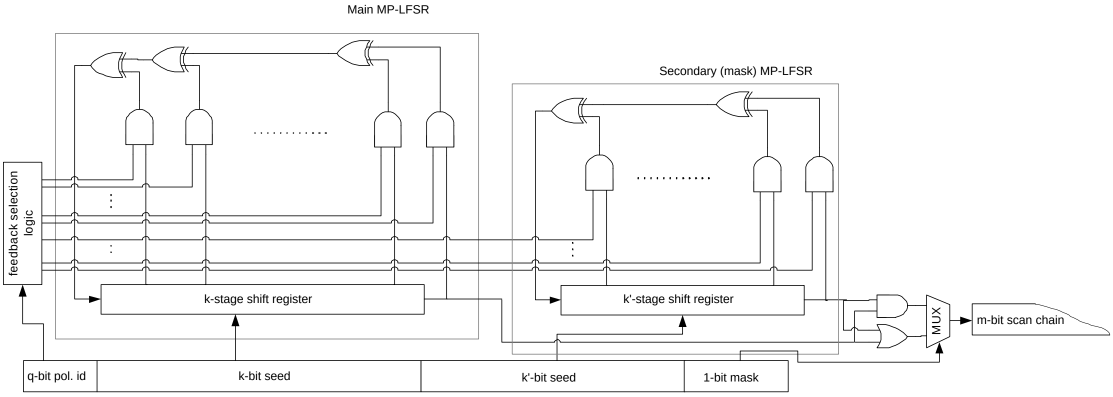


Figure 3.2: Proposed dual MP-LFSR for low power mixed-mode BIST

3.5 Test Set Pre-processing

The minimum length of the MP-LFSR, and consequently the size of the memory for storing the initial seeds, is given by the maximum number of specified bits in the cubes which have to be encoded, as explained in Section 3.1. Hence, the size of the main MP-LFSR, k in Figure 3.2, is determined by the maximum number s of specified bits per cube in the precomputed test set. In order to minimise the number of specified bits per test cube, compaction is disabled during the ATPG process, which results in “relaxed” (or un-compacted) test sets. Analysis of several relaxed test sets of test cubes showed that only for a small fraction of the test cubes in the set the number of specified bits is close to s, while the remaining test cubes have much less than s specified bits. This observation can be exploited in order to reduce the test data storage requirements by compressing the precomputed set of test cubes in two steps: cube compaction and cube concatenation.

Cube compaction. In this step, several “compatible” test cubes in the precomputed set are merged together in order to reduce the number of test cubes which have to be encoded. Two test cubes are “compatible” for compaction if all their overlapping specified bits match. If s is the maximum number of specified bits per test cube for the precomputed set, two test cubes are merged into a single cube only if the number of specified bits of the “compacted” cube does not exceed s. The following example illustrates the use of cube compaction.

Example 8 *Consider the following set of precomputed test cubes:*

c1	1	0	1	X	1	X	(4)
c2	X	0	X	1	X	X	(2)
c3	0	0	1	X	X	X	(3)
c4	X	1	1	0	X	X	(3)

The maximum number s of specified bits per test cube for this test set is 4. Hence, test cubes c1 *and* c2*, although they are compatible, are not merged because the resulting cube would then have 5 specified bits. However,* c2 *and* c3 *can be merged as the resulting compacted cube* $c2'$ *will have only 4 specified bits. The test set after compaction will be:*

c1	1	0	1	X	1	X	(4)
$\mathbf{c2'} = \mathbf{c2}\vert\mathbf{c3}$	0	0	1	1	X	X	(4)
c4	X	1	1	0	X	X	(3)

Hence, the number of test cubes in the test set was decreased from 4 to 3 without increasing the maximum number of specified bits per test cube, and hence the required LFSR length.

Cube concatenation. After cube compaction it may happen that there are cubes in the test set with very few specified bits, but which could not be compacted due to differences in their overlapping specified bits. During this step, several test cubes can be concatenated in order to be generated using a single seed (instead of using one seed per test cube). The following two constraints are imposed during test cube concatenation:

- The number of cubes in a concatenated cube should not exceed a user-specified value, which determines the intervals between successive re-seed operations.
- The number of specified bits in a concatenated cube should not exceed the maximum number of specified bits per test cube in the original test set. This condition ensures that cube concatenation will not increase the length of the LFSR required to encode the original test set.

The following example illustrates the use of cube compaction.

Example 9 *Consider the following set of deterministic test cubes:*

c1	1	0	0	X	1	X	(4)
c2	X	X	0	1	X	X	(2)
c3	0	X	1	X	X	X	(2)
c4	1	X	X	0	X	X	(2)

Assuming that a concatenated cube can contain at most two initial cubes and given the maximum of 4 number of specified bits per cube, the only allowable cube concatenations are $(c2, c3)$*,* $(c3, c4)$ *and* $(c2, c4)$*. The test set after concatenation will be:*

$(\mathbf{c1}, \mathbf{X})$	1	0	0	X	1	X	X	X	X	X	X	X	(4)
$(\mathbf{c2}, \mathbf{c3})$	X	X	0	1	X	X	0	X	1	X	X	X	(4)
$(\mathbf{c4}, \mathbf{X})$	1	X	X	0	X	X	X	X	X	X	X	X	(2)

The cubes which could not be concatenated with other cubes are concatenated with *dummy cubes (all Xs) in order to match the default length of concatenated cubes, in this case 12 bits. The initial test set with 4 test cubes was transformed into an equivalent set with 3 concatenated cubes, without exceeding the maximum number of specified bits per cube corresponding to the initial set.*

The size of the secondary MP-LFSR is given by the maximum number of specified bits in the mask cubes. However, each test cube has two possible mask cubes, one for each composition function (AND and OR). This fact can be exploited in order to further reduce the size of the secondary MP-LFSR, and implicitly the storage requirements for its seeds, by selecting for each test cube the composition function which leads to the mask cube with less specified bits. The mask bit from Figure 3.2 is used to select the appropriate composition function for each test cube. The following example illustrates the selection of the appropriate composition function.

Example 10 *Consider the following test cubes with their mask cubes for AND, and respectively OR composition, where the number of specified bits of each cube appears in brackets:*

	Test cube							***AND-mask cube***							***OR-mask cube***							***Fn.***
c1	1	0	0	1	1	*X*	(5)	1	*X*	*X*	1	1	*X*	(3)	**X**	**0**	**0**	**X**	**X**	**X**	**(2)**	*OR*
c2	0	0	0	1	*X*	*X*	(4)	**X**	**X**	**X**	**1**	**X**	**X**	**(1)**	0	0	0	*X*	*X*	*X*	(3)	*AND*
c3	0	*X*	1	0	*X*	0	(4)	**X**	**X**	**1**	**X**	**X**	**X**	**(1)**	0	*X*	*X*	0	*X*	0	(3)	*AND*
c4	1	*X*	1	0	*X*	1	(4)	1	*X*	1	*X*	*X*	1	(3)	**X**	**X**	**X**	**0**	**X**	**X**	**(1)**	*OR*

The OR-mask cubes of cubes c1 *and* c4 *have less specified bits than their AND-mask cubes, hence OR composition is selected in these cases (column **Fn.**). The AND-mask cubes of* c2 *and* c3 *have less specified bits than the corresponding OR-mask cubes, hence OR composition is selected for* c2 *and* c3 *(column **Fn.**). The set of mask cubes with selectable composition function has maximum 2 specified bits per cube, while both sets of mask cubes for a fixed composition (AND or OR) function have maximum 3 specified bits per test cube. Hence, providing the flexibility of selecting the appropriate composition function on a per-cube basis has significant impact on the length of the mask LFSR, and consequently on the mask seed storage requirement.*

An upper bound for the maximum number of specified bits in mask cubes will be derived in the following for the case when the composition function (AND or OR) can be selected for each test cube.

Given a test cube C, let $M_{AND}(C)$ and $M_{OR}(C)$ be the sets of specified bits of the mask cubes corresponding to AND and OR composition respectively. The mask cube which will be selected in order to minimise the MP-LFSR size will have $s' = min(|M_{AND}(C)|, |M_{OR}(C)|)$ specified bits. As the sets of specified bits of $M_{AND}(C)$ and $M_{OR}(C)$ are disjoint, and their union is equal to $S(C)$, the set of specified bits of C, the following relations hold:

$$
\begin{aligned}
|M_{AND}(C)| &+ |M_{OR}(C)| = |S(C)| && (3.7) \\
s' &\leq |M_{AND}(C)| && (3.8) \\
s' &\leq |M_{OR}(C)| && (3.9)
\end{aligned}
$$

Adding inequalities (3.8) and (3.9) and using equation (3.7) will lead to:

$$s' \leq \frac{|S(C)|}{2} \tag{3.10}$$

Thus, the upper bound for the length of the secondary MP-LFSR, k' from Figure 3.2, is given by $\frac{|S(C)|}{2} + l$, where l is a small constant ($l = 2$ proved to be large enough for all experiments described later in this chapter), which ensures that the probability of finding an initial seed for each mask cube is high, and consequently the total number of feedback polynomials is small. Thus, although intuitively it may seem that the storage requirements have to be doubled by the addition of the secondary MP-LFSR to the TPG, inequality (3.10) shows that the length of the secondary (mask) MP-LFSR from Figure 3.2, and consequently the storage requirement for its seeds, are only approximately 50% of the values corresponding to the main MP-LFSR, whose length is dependent on $|S(C)|$, as explained in Section 3.1.

3.6 Experimental Results

Several experiments using full scan versions of the ISCAS89 benchmark circuits [bl] were performed in order to assess the reduction in power dissipation achievable using the proposed dual MP-LFSR TPG versus the amount of additional hardware and test data storage when compared with a traditional single MP-LFSR TPG [HRT$^+$95]. The experiments for each circuit go through the following steps:

1. An initial set of deterministic test cubes was computed using ATALANTA [Teca] for the faults undetected by a sequence of 512 pseudorandom test patterns. This set was used only for determining the lengths of main and secondary MP-LFSRs, based on the maximum number of specified bits per test cube and respectively mask cube.

2. Having determined the length of the two MP-LFSRs, two pseudorandom sequences of L patterns each, with $L \in$\{1k, 2k, 4k, 8k and 16k\}, were generated using some default primitive polynomials for the two MP-LFSRs. The first test sequence was generated by the main MP-LFSR, and hence corresponds to the traditional TPG, while the second was generated by AND composition of the sequences generated by the two MP-LFSRs.

3. The two pseudorandom test sequences generated at the previous step were fault-simulated using FSIM [Tecb] on the target design in order to estimate their fault coverage and to determine the lists of undetected faults.

4. ATALANTA was used to determine two sets of deterministic test cubes, one for each of the two fault lists computed in the previous step.

5. The set of (polynomial identifier, seed) pairs for the cube set corresponding to the main MP-LFSR was computed using a tool implementing the method described in Section 3.1. The same tool was used to encode the cubes corresponding to the dual MP-LFSR structure using the 5-tuple (main polynomial identifier(polID), main seed, secondary polID, secondary seed, mask selection bit). This step was necessary in order to determine the number of

polynomials and the storage requirements for the single [HRT+95] and dual MP-LFSR TPGs.

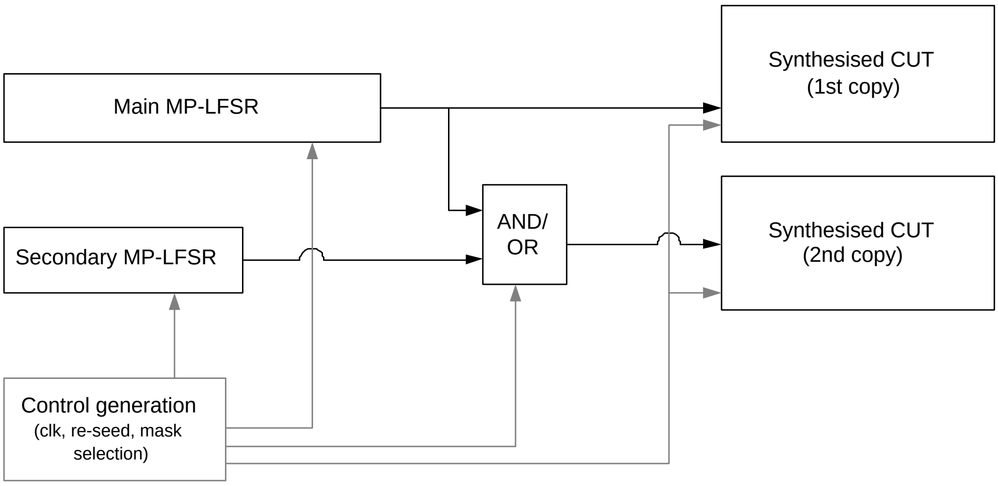

Synthesised CUT (1st copy)
Main MP-LFSR
AND/ OR
Secondary MP-LFSR
Synthesised CUT (2nd copy)
Control generation
(clk, re-seed, mask selection)

Figure 3.3: Circuit configuration for simulation

6. The targeted design was synthesised with Synopsys' Design Compiler [Syn01a] using an Alcatel 0.5μ technology library with power pre-characterised cells. The synthesised design was then simulated using Mentor Graphics' Modelsim [Gra00b] with 5 pseudorandom test patterns generated by the main MP-LFSR and 5 patterns generated by the dual MP-LFSR structure. Only 5 test patterns were considered to keep the simulation time within reasonable limits. Figure 3.3 shows the circuit configuration used for simulations. The switching activities corresponding to these two simulations were back-annotated to the synthesised design and Synopsys' Power Compiler [Syn01c] was used to estimate the average power dissipation in each case.

More details on the tools and benchmark circuits used in these experiments can be found in Appendix D.

Table 3.1 shows the reductions in scan-in transition count and average power dissipation achieved using the dual MP-LFSR TPG compared to the traditional single MP-LFSR TPG [HRT+95]. Columns **Single TC** and **Dual TC** report the number of scan-in transitions generated by the traditional, and respectively

	Single [HRT^{+}95]		**Dual (proposed)**		**Reduction**	
CUT	**TC**	**Pavg**	**TC**	**Pavg**	**TC red (%)**	**Pavg red(%)**
s526	152027	1.25	116369	1.00	23.46	19.66
s641	753730	1.26	577057	1.05	23.44	16.31
s713	753044	1.22	573862	1.09	23.79	10.29
s820	143475	0.67	108245	0.67	24.55	0.30
s832	143284	0.61	108090	0.61	24.56	0.33
s838	1202210	1.04	904106	0.95	24.80	8.45
s953	532909	0.49	404086	0.42	24.17	14.57
s1196	294182	1.83	227156	1.57	22.78	14.47
s1238	295494	1.65	228405	1.50	22.70	9.09
s1423	2137027	2.93	1611456	2.53	24.59	13.53
s5378	13592997	9.66	10188661	9.06	25.04	6.25
s9234	18707043	17.55	14152373	15.42	24.35	12.10
s13207	161446001	31.17	120663103	26.17	25.26	16.03
s15850	115238803	32.40	86268594	27.50	25.14	15.13
s38417	1027848936	387.86	771889958	370.19	24.90	4.56
s38584	722223634	355.36	543841106	342.59	24.70	3.59
Average	-	-	-	-	24.27	10.29

Table 3.1: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT^{+}95] in terms of scan-in transition count and average power dissipation

the proposed TPG. Column **TC red (%)** gives the relative reduction in scan-in transition count achieved by the proposed TPG. The experimental values are very close to the relative reduction of 25% in transition count predicted in Section 3.2. Columns **Single Pavg** and **Dual Pavg** show the average power dissipation during test corresponding to the traditional, and respectively the proposed TPG. Column **Pavg red(%)** gives the relative reduction in average power dissipation obtained when using the proposed TPG. Reductions up to nearly 20% in average power dissipation compared to the traditional single MP-LFSR [HRT^{+}95] were be obtained when using the proposed dual MP-LFSR. The last row in Table 3.1 shows the average reductions in scan-in transition count and average power dissipation for the entire set of benchmark circuits. The nearly 25% reduction in scan-in transition count corresponds to a typical reduction of 10% in average power dissipation during test.

Table 3.2 shows the area overhead associated with the proposed TPG when compared with the traditional single MP-LFSR TPG [HRT^{+}95]. Columns **NPm** and

	Dual (proposed)				
	Single [HRT$^+$95]				
CUT	**NPm**	**PDm**	**NPs**	**PDs**	**xArea%**
s526	2	15	1	8	53.33
s641	1	24	2	10	41.67
s713	2	24	2	12	50
s820	5	15	1	8	53.33
s832	4	15	1	8	53.33
s838	1	38	1	8	21.05
s953	1	17	3	9	52.94
s1196	1	19	4	9	47.37
s1238	1	19	4	9	47.37
s1423	1	28	2	13	46.43
s5378	2	29	5	15	51.72
s9234	2	53	2	27	50.94
s13207	3	24	3	13	54.17
s15850	3	40	2	21	52.5
s38417	3	88	2	45	51.14
s38584	4	56	2	29	51.79

Table 3.2: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT$^+$95] in terms of area overhead

NPs show the number of polynomials needed by the main and secondary MP-LFSRs to cover the deterministic test cubes and respectively their corresponding mask cubes. Columns **PDm** and **PDs** give the size of the main, and respectively secondary MP-LFSR. Column **xArea%** shows the area overhead introduced by the secondary MP-LFSR, relative to the area of the main MP-LFSR. It can be seen that the values for area overhead obtained from the experiments are consistent with the theoretical upper bound of nearly 50% determined in section 3.3. For example, in the case of circuit s38584, 4 feedback polynomials of degree 56 are needed in order to encode the 315 test cubes, while the set of mask cubes can be encoded using only two polynomials of degree 29.

As outlined in the beginning of this chapter, a mixed mode test set consists of a pseudorandom sequence and a deterministic sequence. The following two aspects will be analysed next:

1. the relation between the length of the pseudorandom sequence and the test data storage requirements for complete fault coverage, and

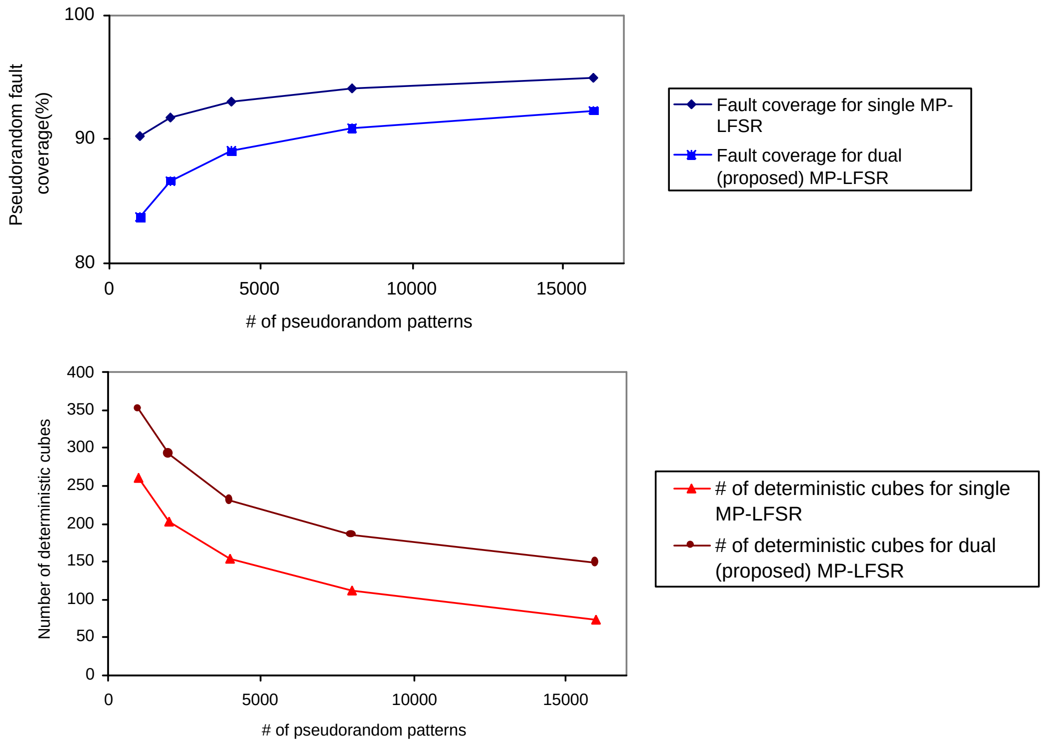


Figure 3.4: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT+95] in terms of pseudorandom fault coverage and test data storage requirements (circuit s38584)

2. the effect of AND/OR masking on the fault coverage of the pseudorandom sequence generated by the proposed TPG when compared to the traditional single MP-LFSR [HRT+95].

Figure 3.4 illustrates the previously mentioned aspects on experimental data obtained from the largest benchmark circuit considered, s38584. It can be seen that the fault coverage of the pseudorandom sequences increases logarithmically with the length of the sequences. For example, increasing the lengths of the pseudorandom sequences from 1k patterns to 4k patterns resulted in 3% and 6% increase in fault coverage for the traditional and the proposed TPG, respectively. Further increasing the lengths of the pseudorandom test sequences to 16k patterns increased the fault coverage by only 1% and 3% respectively, for the traditional

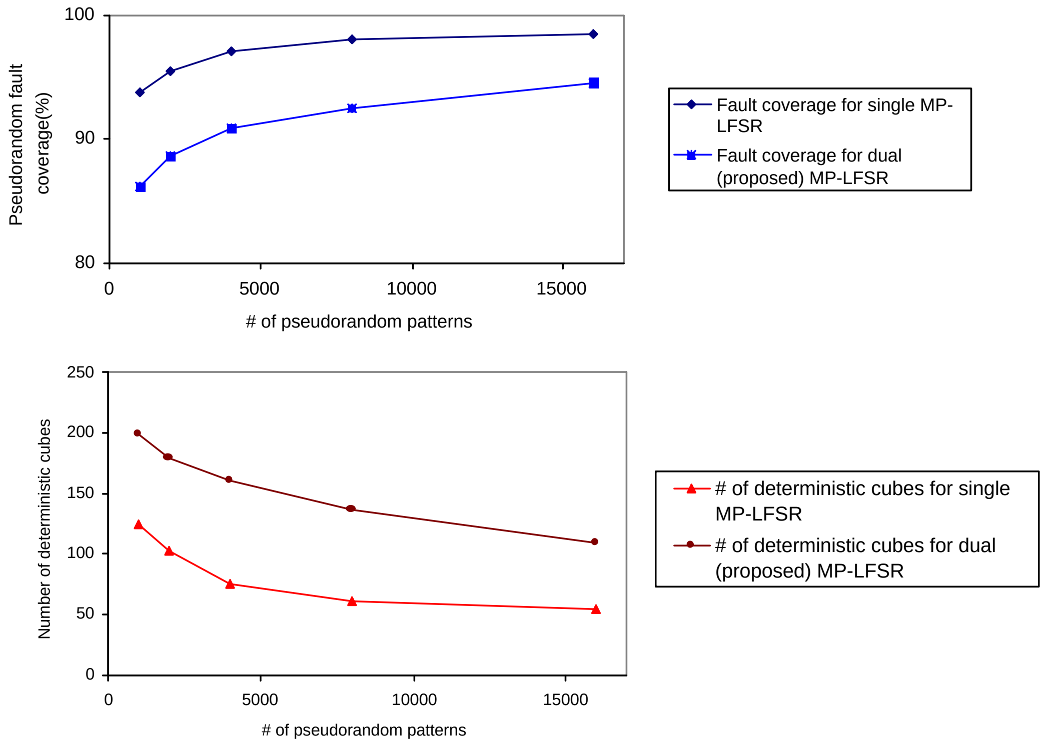


Figure 3.5: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT^{+}95] in terms of pseudorandom fault coverage and test data storage requirements (circuit s5378)

and proposed TPG. The fault coverage of the pseudorandom sequence determines the number of faults which remain to be detected using deterministic patterns. As it can be seen, a linear increase in the lengths of the pseudorandom sequences determines a logarithmic decrease in the number of deterministic test patterns required for complete fault coverage, and hence in the test data storage requirements. Thus, the storage requirements can be controlled by varying the length of the non-deterministic sequence. For example, increasing the length of the pseudorandom sequence from 1k to 8k patterns, decreased the number of deterministic test cubes by 57% for the single MP-LFSR TPG [HRT^{+}95] and by 47% for the dual MP-LFSR TPG. However, expanding the pseudorandom test sequences to 16k pattern led to an additional reduction of only 14%, and respectively 10%, in the number of deterministic test cubes corresponding to the two TPGs.

Although AND/OR compositions reduce the scan-in transition count, and hence the power dissipation during shift, they depreciate the randomness characteristic to LFSR-generated test sequences consequently affecting their fault coverage. As shown in Figure 3.4, the lower fault coverage of the pseudorandom sequence generated by the dual MP-LFSR TPG, when compared to the traditional TPG, has to be compensated with a higher number of deterministic cubes in order to preserve the complete fault coverage. Similar behaviour can be observed on all other benchmark circuits, another example being shown in Figure 3.5. The complete set of experimental results are reported in Tables 3.3, 3.4 and 3.5, where columns 2 to 7 show data corresponding to the standard single MP-LFSR architecture, columns 8 to 14 hold the experimental data for the dual MP-LFSR architecture and the last three columns compare the results of the two approaches. Columns $\#prVec$ and $prFC$ give the length of the pseudorandom sequence and its fault coverage. Columns $\#c$ and $\#cC$ show the number of deterministic cubes, before and after compaction respectively, required for full fault coverage. Columns $\#px$ and PDx, where $x \in \{m, s\}$, show the number of feedback polynomials and their degrees, for the main (m) and secondary (s) MP-LFSRs. Column $TCr(\%)$ shows the scan-in transition count reduction obtained using the dual MP-LFSR architecture, when compared to the single MP-LFSR architecture. Column xC shows the number of extra deterministic test cubes required by the dual MP-LFSR TPG to compensate the loss in fault coverage due to masking, when compared to the single MP-LFSR architecture. In some cases (circuit s15850 for 1000 pseudorandom test vectors, for example), this value is negative, which means that the dual MP-LFSR TPG requires less deterministic test cubes than the single MP-LFSR TPG for achieving full fault coverage. Column $xTAT(\%)$ shows the variation of the overall test time introduced by the additional deterministic cubes, where applicable. It can be seen that the overall test time increases by at most 10% due to the extra test cubes.

3.7 Concluding Remarks

The work presented in this chapter has addressed the problem of reducing power dissipation during test in a mixed-mode BIST environment. The dual MP-LFSR TPG was proposed in order to overcome the shortcomings of previously proposed low power TPGs, such as incomplete fault coverage and long test times. The proposed TPG combines masking properties of AND/OR composition and LFSR re-seeding in order to achieve complete fault coverage with short test sequences while reducing the average power dissipation during test, when compared to the traditional single MP-LFSR approach [HRT+95]. Moreover, as the power reduction is achieved only through test set transformations, the proposed TPG does not require any modification of the circuit under test, hence preserving its original performance. Extensive experimental data obtained using commercial synthesis and simulation tools showed that this TPG consistently reduces the number of scan-in transitions by 25%, which can offer reductions up to nearly 20% in average power dissipation. These power savings are achieved at the cost of an increased area overhead and test data storage requirement when compared to the traditional single MP-LFSR TPG [HRT+95]. This overhead is justified by the reduction in power dissipation, necessary for decreasing the risks of reliability problems and manufacturing yield loss.

	Single MP-LFSR						**Dual MP-LFSR**							**Comparison**		
CUT	#prVec	prFC	#c	#cC	#p	PD	prFC	#c	#cC	#pm	PDm	#ps	PDs	TCr(%)	xC	xTAT(%)
s526	1000	95.68	28	17	3	15	86.85	41	32	2	15	1	8	23.45	15	1.47
	2000	96.40	25	15	4	15	90.81	31	26	2	15	1	8	23.86	11	0.55
	4000	97.12	21	14	2	15	92.79	25	21	2	15	1	8	24.41	7	0.17
	8000	97.66	18	14	2	15	95.50	17	16	2	15	1	8	24.87	2	0.02
	16000	97.84	17	14	3	15	96.58	15	14	2	15	1	8	24.84	0	0.00
s641	1000	96.54	27	14	1	24	88.99	39	23	1	24	2	10	23.44	9	0.89
	2000	97.62	22	12	1	24	91.36	32	21	1	24	2	11	24.54	9	0.45
	4000	98.06	20	11	1	24	92.01	30	21	1	24	3	11	24.86	10	0.25
	8000	98.06	20	11	1	24	92.87	28	21	1	24	3	11	24.97	10	0.12
	16000	98.27	19	11	1	24	93.74	24	20	1	24	2	10	24.74	9	0.06
s713	1000	90.71	27	14	1	24	84.34	39	22	2	24	2	12	23.79	8	0.79
	2000	91.57	22	12	1	24	86.58	32	21	1	24	2	12	24.22	9	0.45
	4000	91.91	20	11	1	24	87.09	30	21	1	24	1	12	24.83	10	0.25
	8000	91.91	20	11	1	24	87.78	28	19	1	24	2	11	24.97	8	0.10
	16000	92.08	19	11	1	24	88.47	24	18	1	24	2	11	24.97	7	0.04
s820	1000	87.06	77	47	5	15	81.53	77	53	5	15	1	8	24.55	6	0.57
	2000	89.77	64	38	5	15	87.06	56	40	5	15	1	8	24.46	2	0.10
	4000	90.47	61	37	5	15	89.65	48	35	5	15	1	8	24.45	-2	-0.05
	8000	90.47	61	37	5	15	90.59	44	31	5	15	1	8	24.52	-6	-0.07
	16000	90.47	61	37	5	15	94.12	34	27	5	15	1	8	24.53	-10	-0.06
s832	1000	85.40	74	47	5	15	79.66	77	52	4	15	1	8	24.56	5	0.48
	2000	88.05	61	38	5	15	85.17	55	40	5	15	1	8	24.47	2	0.10
	4000	88.74	58	37	5	15	87.82	47	35	4	15	1	8	24.46	-2	-0.05
	8000	88.74	58	37	5	15	88.85	42	30	4	15	1	8	24.54	-7	-0.09
	16000	88.74	58	37	5	15	92.30	32	26	4	15	1	8	24.53	-11	-0.07

Table 3.3: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT$^+$95]

	Single MP-LFSR						Dual MP-LFSR							Comparison		
CUT	#prVec	prFC	#c	#cC	#p	PD	prFC	#c	#cC	#pm	PDm	#ps	PDs	TCr(%)	xC	xTAT(%)
s838	1000	56.50	168	131	1	38	59.40	168	131	1	38	1	8	24.80	0	0.00
	2000	57.14	168	131	1	38	61.87	168	131	1	38	1	8	24.74	0	0.00
	4000	60.69	168	131	1	38	65.63	168	131	1	38	1	8	24.89	0	0.00
	8000	62.30	168	131	1	38	67.88	168	131	1	38	1	8	24.74	0	0.00
	16000	64.77	168	131	1	38	70.68	168	131	1	38	1	8	24.79	0	0.00
s953	1000	84.71	66	41	1	19	91.47	44	39	1	17	3	9	24.17	-2	-0.19
	2000	93.14	44	26	1	19	93.98	35	32	2	16	2	9	24.24	6	0.30
	4000	96.29	33	21	1	19	96.85	24	24	2	16	1	8	24.19	3	0.07
	8000	96.29	33	21	1	19	96.94	23	23	2	16	1	8	24.24	2	0.02
	16000	96.29	33	21	1	19	96.94	23	23	2	16	1	8	24.23	2	0.01
s1196	1000	90.34	94	66	1	19	73.91	148	124	1	19	4	9	22.78	58	5.44
	2000	93.40	69	47	1	19	79.31	125	105	1	19	4	9	24.66	58	2.83
	4000	96.86	53	38	1	19	83.17	113	95	1	19	4	9	25.41	57	1.41
	8000	98.47	42	33	1	19	88.25	90	78	1	19	4	9	25.01	45	0.56
	16000	99.36	34	30	1	19	89.94	77	66	1	19	4	9	24.94	36	0.22
s1238	1000	85.02	106	71	1	19	68.49	162	130	1	19	4	9	22.70	59	5.51
	2000	87.75	81	52	1	19	73.43	137	109	1	19	4	9	24.62	57	2.78
	4000	91.29	63	44	1	19	76.75	120	97	1	19	4	9	25.40	53	1.31
	8000	93.36	49	37	1	19	81.85	100	84	1	19	4	9	25.00	47	0.58
	16000	94.24	41	33	1	19	83.62	88	74	1	19	4	9	24.92	41	0.26
s1423	1000	97.49	33	18	1	28	93.33	43	28	1	28	2	13	24.59	10	0.98
	2000	98.22	25	15	1	28	95.12	34	20	1	28	1	14	25.16	5	0.25
	4000	98.61	21	13	1	28	95.71	29	18	1	28	1	14	24.94	5	0.12
	8000	98.94	17	13	1	28	97.43	20	16	1	28	1	14	24.82	3	0.04
	16000	99.01	16	13	1	28	97.49	20	16	1	28	1	14	24.93	3	0.02

Table 3.4: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT$^+$95]

	Single MP-LFSR						**Dual MP-LFSR**							**Comparison**		
CUT	#prVec	prFC	#c	#cC	#p	PD	prFC	#c	#cC	#pm	PDm	#ps	PDs	TCr(%)	xC	xTAT(%)
s5378	1000	93.80	222	124	2	29	86.18	357	199	2	29	5	15	25.04	75	6.67
	2000	95.58	161	103	1	25	88.57	310	179	2	29	4	15	25.23	76	3.61
	4000	97.14	116	75	1	25	90.82	248	161	2	29	4	15	25.04	86	2.11
	8000	98.07	88	61	1	25	92.57	196	136	2	29	3	15	25.02	75	0.93
	16000	98.51	70	55	1	25	94.55	152	109	1	29	3	15	24.99	54	0.34
s9234	1000	72.01	1061	261	2	53	63.71	626	243	2	53	2	27	24.35	-18	-1.43
	2000	76.90	909	231	2	53	67.03	584	230	4	53	2	27	24.69	-1	-0.04
	4000	80.27	795	215	2	53	70.52	543	215	2	53	2	27	24.75	0	0.00
	8000	82.46	718	200	2	53	72.76	519	210	2	53	2	27	24.89	10	0.12
	16000	84.61	645	187	2	53	74.94	491	202	2	53	2	27	25.01	15	0.09
s13207	1000	77.55	1260	202	2	123	81.14	791	319	3	24	3	13	25.26	117	9.73
	2000	83.32	1036	166	1	121	84.46	698	277	3	24	2	13	25.27	111	5.12
	4000	87.09	857	220	2	32	86.65	630	252	3	24	2	13	25.14	32	0.76
	8000	91.84	615	176	2	28	89.48	541	216	2	24	1	13	25.02	40	0.49
	16000	95.19	417	130	2	24	91.77	450	196	2	22	2	12	25.01	66	0.41
s15850	1000	83.95	961	262	3	40	84.66	573	238	3	40	2	21	25.14	-24	-1.90
	2000	87.55	792	233	2	40	86.62	505	224	4	40	2	21	25.10	-9	-0.40
	4000	89.60	664	215	2	40	88.22	461	213	3	40	2	21	25.00	-2	-0.05
	8000	90.58	611	203	2	40	89.49	414	210	3	40	2	21	25.03	7	0.09
	16000	91.63	560	200	3	40	91.55	365	200	2	40	2	21	24.99	0	0.00
s38417	1000	86.52	2734	618	4	88	84.89	1410	498	3	88	2	45	24.90	-120	-7.42
	2000	87.95	2535	576	4	88	86.36	1348	484	3	88	2	45	24.93	-92	-3.57
	4000	90.24	2295	523	4	88	87.13	1314	467	2	88	1	45	24.94	-56	-1.24
	8000	91.86	2077	456	3	88	88.30	1258	457	2	88	2	45	24.95	1	0.01
	16000	93.41	1875	401	3	88	88.86	1231	448	3	88	2	45	24.97	47	0.29
s38584	1000	90.18	1546	260	3	56	83.75	2085	351	4	56	2	29	24.70	91	7.22
	2000	91.72	1208	202	2	56	86.62	1706	291	3	56	2	29	24.78	89	4.04
	4000	93.06	908	153	3	56	89.03	1338	231	2	56	3	29	24.96	78	1.88
	8000	94.15	680	111	3	56	90.88	1039	186	2	56	2	29	24.97	75	0.92
	16000	94.99	477	73	3	56	92.30	806	149	3	56	2	29	24.96	76	0.47

Table 3.5: Comparison between the proposed dual MP-LFSR TPG and the traditional single MP-LFSR TPG [HRT$^+$95]

Chapter 4

Low Power Test Data Compression

Chapters 2 and 3 have presented two methods for reducing power dissipation during test using test set transformations. This chapter shows how by combining test set transformations with structural modifications of the circuit under test, even higher reductions in power dissipation during test can be obtained.

Design for testability (DFT) based on scan and automatic test pattern generation (ATPG) has been broadly accepted as a methodology that provides very high test coverage. The process of scan insertion and test generation is automated and guarantees high quality results. Conventional ATPG tools can generate test sets achieving almost complete fault coverage for various fault models. Figure 4.1 shows statistical data from a recent study [Bly01] capturing the growth of test data volume caused by the increasing gate counts. The volume of test data is one of the main parameters that determine the test time and the tester memory requirements [ZDR00b, RTK$^+$02], and consequently has affects the test cost. For a standard scan-based test configuration as the one shown in Figure 4.2, the test data volume (TDV) can be approximated by:

$$TDV = NSC \times NSP \tag{4.1}$$

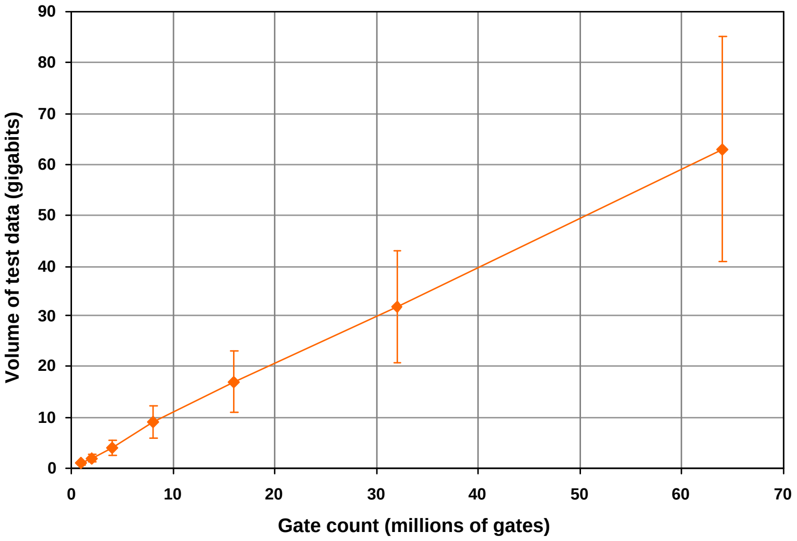


Figure 4.1: Volume of test data vs. gate count

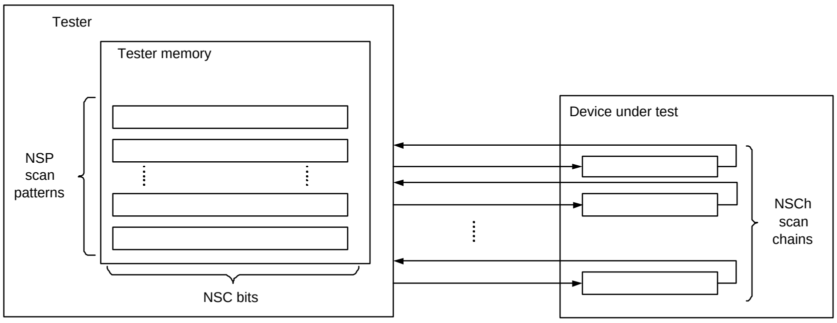


Figure 4.2: Tester and device under test

where NSC is the total number of scan cells of the circuit and NSP is the number of scan patterns. Assuming equal length scan chains and a continuous data stream between the tester and the device under test, the test time (TT) can be expressed as a function of the test data volume as follows:

$$TT = \frac{TDV}{NSCh \times F} \tag{4.2}$$

where $NSCh$ is the number of scan chains of the circuit and F is the scan operating frequency. The last equation shows three options available for reducing test time:

1. increasing the test data transfer frequency (F), i.e. speeding-up the tester channels,
2. increasing the number of scan chains ($NSCh$), and consequently the number of tester channels, and
3. reducing the test data volume (TDV).

The first two options require expensive test equipment and offer in exchange only a reduction in test time. However, the third option, i.e. reducing the test data volume, does not require any special features of the tester and reduces not only the test time but also the memory requirements for the tester. Consequently, reduction of test data volume has been targeted by several recently proposed test data compression methods [AN98, CC01c, CC01b, YTIH97].

A typical configuration for scan-based test using test data compression is shown in Figure 4.3. The test data, stored in a compressed form on the tester, is sent to the device under test, where a small decoding unit decompresses the data before shifting it into the scan chains. The test responses shifted-out from the scan chains are compressed into signatures which are sent back to the tester.

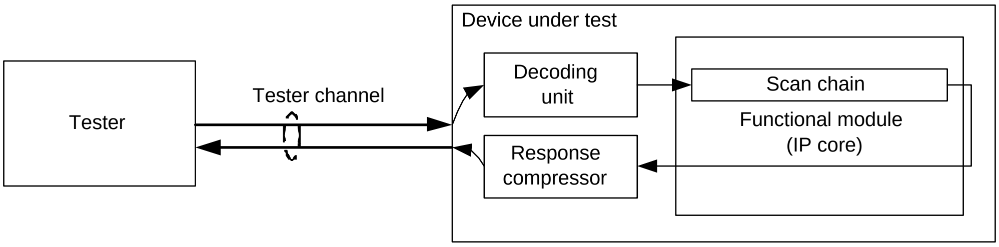


Figure 4.3: Scan-based test using a test data compression scheme

Scan-based test solutions using test data compression bring the following advantages:

- no tester modifications are required,
- test time is reduced as less data is sent at the slow I/O frequency, while on-chip decompression runs at the internal clock frequency,

- tester memory requirements are lower compared to standard scan-based test architectures,
- no circuit modifications are required (the decoding unit is a separate logic block).

These advantages are achieved at the expense of a small area overhead required by the decoding and response compressor units.

Generally, for a target fault, the ATPG tool specifies only a small number of scan cells, while the remaining scan positions are filled with random values. Such fully specified test patterns are more likely to detect additional faults, and can be stored on a tester. However, a side effect of the random fill is that the test patterns are over-specified [RTK$^+$02]. Test data compression methods exploit the presence of unspecified ("don't care" (DC)) bits in ATPG-generated test sets in order to compress the test sets. Data compression methods based on run-length encoding have received special attention because of their simplicity. Several run-length encoding schemes have been recently proposed for compressing test data [AN98, CC01c, CC01b]. A common characteristic of all these approaches is that they use asymmetric codes, i.e. runs of 0s are encoded differently in terms of encoding length from runs of 1s.

Methods such as "don't care" fill for minimum transition count [FCN$^+$99] and scan latch reordering [DCPR98] have been employed for reducing power dissipation during test by reducing the number of transitions in the test sets. Thus, these methods generate with equal probability runs of 1s and runs of 0s. In consequence, the resulting test sets cannot be compressed efficiently using asymmetric coding schemes, as it will be shown later in this chapter. This observation suggested the idea of developing a symmetric run-length coding scheme, i.e. providing equal length codes for runs of 0s and runs of 1s of the same length, suitable for compressing low power test sets.

The first objective of this chapter is to introduce a general method for converting asymmetric run-length coding schemes into symmetric coding schemes. The effectiveness of the proposed transformation will be demonstrated for Golomb codes [Gol66]. Next, the chapter continues with an analysis of the effect of scan latch reordering on the compression efficiency. Based on the conclusions drawn from this analysis, a parametrisable scan latch reordering algorithm is proposed which offers the possibility of controlling the trade-off between power dissipation during test and the volume of compressed test data, according to the requirements of each specific application.

4.1 Background Information

The *weighted transition metric*, described in Chapter 2, will serve as a measure of the average power dissipation during test. Due to the strong correlation between WTC and the power dissipation during shift, reducing the WTC leads implicitly to a reduction in power dissipation during shift.

This section provides theoretical background on Golomb codes.

4.1.1 Golomb Codes

In Golomb's coding scheme [Gol66], each codeword corresponds to a *run-length*. The binary coder receives a *binary input sequence* of symbols. A sequence of 0s and 1s can be treated as a sequence of binary events C and S, where C *continues* the run and S *stops* the run. The distribution of the *run-length* random variable value, i.e. 0, 1, 2, …, is characterised by the parameter p_c: the "run continues" probability. The "run stop" probability, p_s, is calculated as $1 - p_c$. A sequence of bits may be parsed into run-lengths of zero or more C events followed by a single S event.

Golomb's parameterised family of codes converts ranges of probability value $p_c \geq 1/2$ to a single parameter g, also known as the *Golomb* group size, with a positive integer value such that the following approximation holds:

Binary string	Run-length	Golomb code
1	0	000
01	1	001
001	2	010
0001	3	011
00001	4	**1**000
000001	5	**1**001
0000001	6	**1**010
00000001	7	**1**011
000000001	8	**11**000
0000000001	9	**11**001
00000000001	10	**11**010

Table 4.1: Golomb codes for group size $g = 4$

$$(p_c)^g \approx \frac{1}{2} \tag{4.3}$$

The Golomb code corresponding to each Golomb group size g and run-length r has two components:

- the first component is the *unary* or *base 1* encoding of the integer-valued *quotient* $\mathcal{Q} = r/g$. This component denotes the *prefix* of the Golomb code, encoded as a unary number of the form of zero or more 1s followed by a *delimiting* 0;

- the second component is the codeword of the integer *remainder* $\mathcal{R} = r \; mod \; g$. This component denotes the *tail* of the code, encoded on $log_2 g$ bits in a binary base.

For example the binary string 000001 would be encoded using $g = 4$ as 10’01, where $\mathcal{Q}$ and $\mathcal{R}$ are encoded as 10_1 and 01_2 respectively. The Golomb codes for run-lengths from 0 to 10 are listed in column 3 of Table 4.1 where the prefix of each code, if any, is shown in bold.

4.2 Compression Efficiency of Standard Golomb Codes

This section explains why compression methods based on asymmetric coding schemes, specifically Golomb codes, fail to compress efficiently low power test sets.

As shown in Table 4.2, asymmetric coding schemes, such as Golomb codes, associate short codes to runs of C events finished by an S event. The problem arises when a run of S events occurs in the binary string. Each S in this run is seen as a zero length run of C events, and hence a Golomb code is assigned to each S event in the run, rather than to the entire run. Thus, while asymmetric coding schemes can compress efficiently data with high density of C events, their compression efficiency decreases as the density of S events increases.

Attempting to exploit the asymmetry of standard Golomb codes, the approach proposed in [CC01a] suggested filling the DCs of incompletely specified test sets with values corresponding to the C event in order to improve the compression efficiency. However, filling all DCs with the same value has two undesired effects on the weighed transition count, and consequently on the power dissipation during shift:

- It may introduce unnecessary transitions in the scan-in vectors. This situation occurs when the fill value differs from the specified values adjacent to unspecified bits.
- Has an uncontrollable effect on the test responses in terms of scan-out transition count.

Low transition test sets, obtained through transformations such as scan latch reordering or DC fill for minimum transition count, contain with equal probability runs of 1s and runs of 0s. Therefore, for any of the two possible values of the C and S events, (0,1) and (1,0), asymmetric coding schemes will fail to compress efficiently such test sets.

The following section presents a symmetric version of the Golomb coding scheme for efficient compression of low transition count test sets.

4.3 Symmetric Golomb Coding Scheme

From the observations made in the previous section it can be concluded that the shortcoming of the standard (asymmetric) run-length coding schemes such as the Golomb coding, arises from the fixed assignment of the C and S values. In order to compress efficiently test sets with balanced densities of runs of 1s and runs of 0s using run-length coding schemes it should be possible to dynamically change the values of the C and S events. This section presents a general method of converting asymmetric run-length coding schemes into symmetric coding schemes with improved compression efficiency for low power test sets.

In a nutshell, the proposed symmetric coding scheme adds a "sign bit" in front of the standard asymmetric code which will specify the (C,S) value pair used for the current code. This way, runs of 1s and runs of 0s of the same length will have the same encoding, differing only in the value of the "sign bit". While an asymmetric coding scheme compresses only runs of 0s and it "expands" runs of 1s, assuming (C,S) = (0,1), the proposed symmetric variant will compress both runs of 1s and runs of 0s.

Particularising the asymmetric-to-symmetric coding scheme transformation for Golomb codes, a run of one or more C events ended by a S event will be encoded using C as the "sign-bit" followed by the asymmetric Golomb code corresponding to the given run-length. Columns 5 to 7 in Table 4.2 shows the "sign-bit", the symmetric Golomb code and its size for run-lengths in the 0 to 8 range, assuming (C,S) = (0,1). From columns 4 and 7 in Table 4.2, it can be observed that the symmetric codes for runs of 0s are 1 bit longer than the standard asymmetric codes. However, the symmetric codes for runs of 1s are significantly shorter than the asymmetric Golomb encodings. The proposed symmetric coding scheme overcomes the disadvantage of standard Golomb codes by treating runs of 0s and runs

		Standard Golomb coding scheme		**Proposed coding scheme**		
Input	**Run-length**	**Code**	**Code size**	**Sign-bit**	**Code**	**Code size**
1	0	000(α)	3	0	*0*000	4
01	1	001	3	0	*0*001	4
001	2	010	3	0	*0*010	4
0001	3	011	3	0	*0*011	4
00001	4	1000	4	0	*0*1000	5
000001	5	1001	4	0	*0*1001	5
0000001	6	1010	4	0	*0*1010	5
00000001	7	1011	4	0	*0*1011	5
000000001	8	11000	4	0	*0*11000	5
110	2	N/A (2 x α = 000000)	6	1	*1*010	4
1110	3	N/A (3 x α = ...)	9	1	*1*011	4
11110	4	N/A (4 x α = ...)	12	1	*1*1000	5
111110	5	N/A (5 x α = ...)	15	1	*1*1001	5
1111110	6	N/A (6 x α = ...)	18	1	*1*1010	5
11111110	7	N/A (7 x α = ...)	21	1	*1*1011	5
111111110	8	N/A (8 x α = ...)	24	1	*1*1100	5

Table 4.2: Asymmetric and symmetric Golomb codes for group size 4

of 1s uniformly.

The following example demonstrates the improved compression efficiency of the symmetric coding scheme compared to the standard Golomb coding scheme for low power test sets.

Example 11 *Consider the following 17-bit test vector: 11111110000000001. The standard Golomb encoding of this vector, assuming a group size of 4, will be 000' 000' 000' 000' 000' 000' 000' 11001. Hence, the standard Golomb coding scheme failed to compress the test vector by "expanding" the initial run-length from 17 bits to 26 bits. The encoding of the same test vector using the symmetric Golomb coding scheme will be 11011' 011000, where the underlined positions represent the "sign-bits". The encoding in this case is 11 bits long, that is 15 bits shorter than the standard asymmetric Golomb encoding length and 6 bits shorter than the original run-length.*

Figure 4.4 compares the code sizes for the asymmetric and symmetric Golomb coding schemes. On one hand, when using the asymmetric Golomb scheme, the code size for runs of 1s increases linearly with the run length, while short codes are assigned to runs of 0s. On the other hand, the proposed symmetric coding scheme produces equally short codes for both runs of 1s and runs of 0s, a feature which can be exploited to efficiently compress low transition count test sets. The "uncompressed run size" line in Figure 4.4 represents the hypothetical case when the coding size is equal to the run-length. Above this line no compression is achieved since the encoding size exceeds the run-length. It should be noted that, unlike standard Golomb coding which compresses only runs of 0s with lengths greater than 3 while "expanding" runs of 1s, the proposed symmetric coding scheme compresses both runs of 1s and runs of 0s for lengths greater than 4. From Figure 4.4 it can also be observed that the compression efficiency increases with the run-length. Increasing the run-lengths is equivalent with reducing the scan-in transition count, therefore it can be concluded that *reducing the scan-in transition count improves the compression efficiency of symmetric run-length coding schemes.*

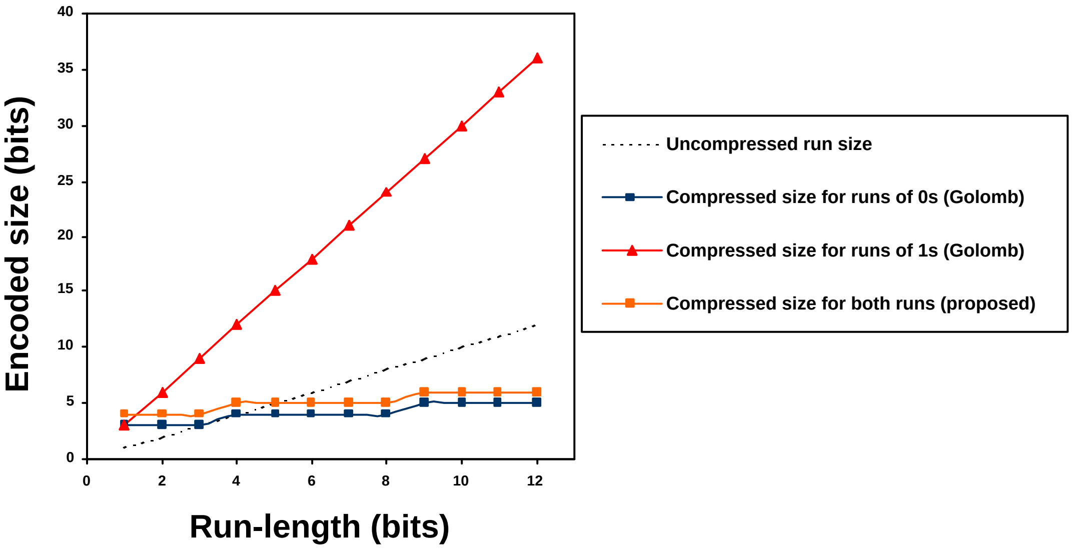


Figure 4.4: Code size comparison for the asymmetric and symmetric Golomb coding schemes for a group size of 4

It should be noted that, although illustrated for Golomb codes, the transformation of an asymmetric code into a symmetric one by augmenting the codewords with a *sign bit*, is generally applicable to **any** asymmetric run-length coding scheme.

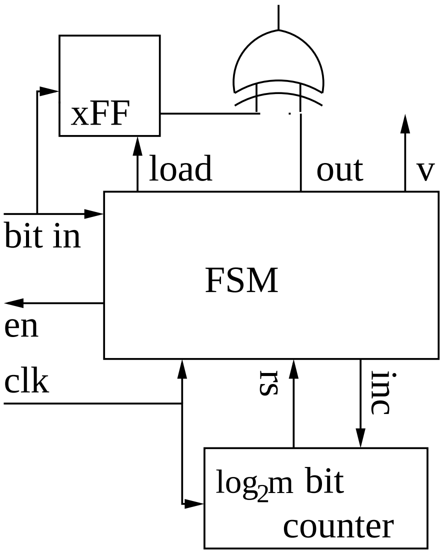


Figure 4.5: Decoding unit for the symmetric Golomb coding scheme

4.4 Decompression Unit for Symmetric Golomb Codes

This section describes the behaviour and architecture of the on-chip decompression unit for the proposed symmetric coding scheme. The starting point for the proposed on-chip decompression unit is the decoder for Golomb codes presented in [CC01a], which can be implemented using a finite state machine (FSM) and a counter [CC01a]. As mentioned earlier, the symmetric coding scheme adds a sign-bit in front of the standard Golomb code in order to provide a short and unified encoding for runs of 0s and runs of 1s. The changes made to the asymmetric Golomb decoder in order to account for the sign-bit consist of:

- one extra state in the decoder FSM,
- one extra flip-flop (xFF), and
- one XOR gate added to the decoder architecture.

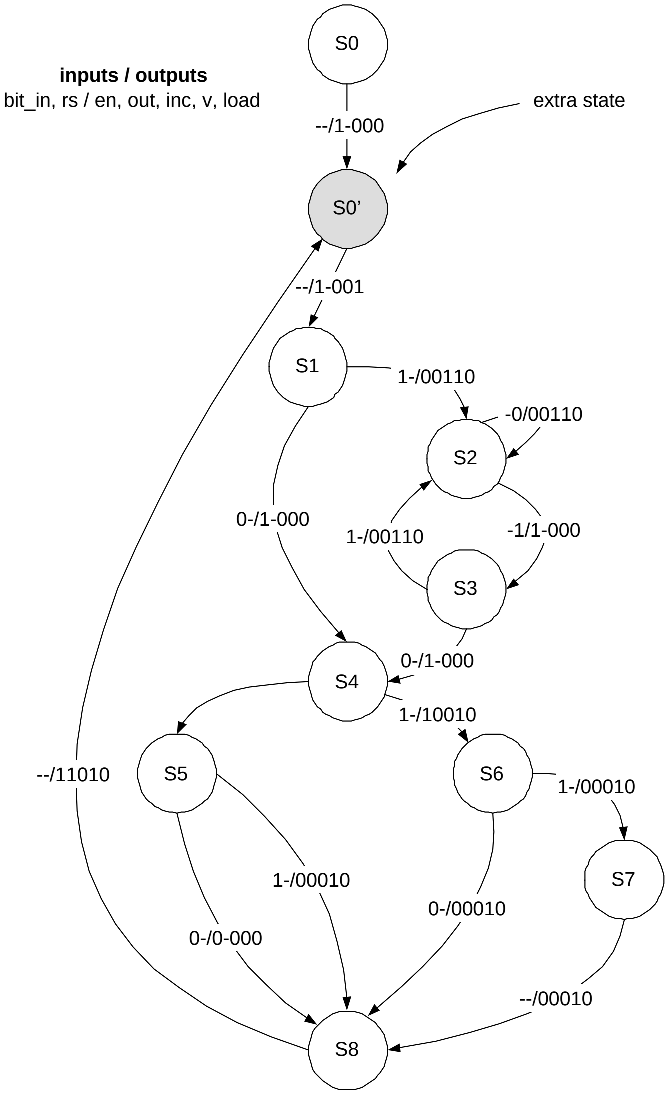
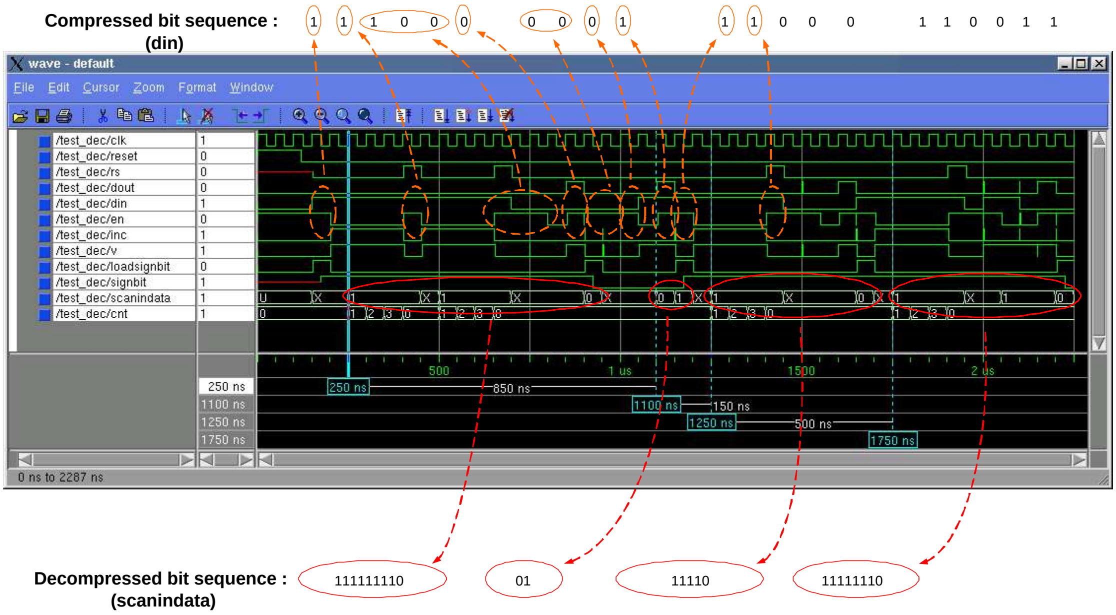


Figure 4.6: FSM for the symmetric Golomb code decoder (group size 4)

The architecture of the modified decoder is shown in Figure 4.5 and the extended FSM is shown in Figure 4.6, where the notations are the same with those used in [CC01a]. The *bit_in* line is the input of the FSM for the compressed binary string. The *en* line signals when the decoding unit is ready to load a new bit. The *inc* line is used to increment the counter and the *rs* signal indicates that the counter has finished counting. The *out* line is the decoded output and the *v* line indicates when the output is valid. The additional *load* signal is used to load the *sign-bit* into xFF. *S0'* marked in Figure 4.6 represents the extra FSM state. For each new

Golomb code, the extra state will set the value of xFF to 0 or 1 according to the sign-bit. After loading the sign-bit into xFF, the FSM loads from the *bit in* line the remaining bits of the code, corresponding to the asymmetric Golomb code, and starts the decoding process. States S0, S0', S1, S2, S3 and S4 correspond to the decoding of the Golomb code prefix, .i.e. the quotient $\mathcal{Q}$ defined in Section 4.1.1. States S5 to S8 handle the decoding of the Golomb code tail, i.e. the remainder $\mathcal{R}$ defined in Section 4.1.1. As the Golomb code is decoded, a run of 0s of the appropriate length is generated by the FSM on the *out* line. The value of the *out* line is XOR-ed with the value of the sign-bit stored in xFF in order to produce the correct binary string which will be fed into the scan chain.

Figure 4.7 shows simulation waveforms for a symmetric Golomb decoding unit for group size 4. The *scanindata* signal represents the output of the XOR gate in Figure 4.5 and the *cnt* signal shows the content of the log_2m-bit counter.



/test_dec/clk	1
/test_dec/reset	0
/test_dec/rs	0
/test_dec/dout	0
/test_dec/din	1
/test_dec/en	0
/test_dec/inc	1
/test_dec/v	1
/test_dec/loadsignbit	0
/test_dec/signbit	1
/test_dec/scanindata	1
/test_dec/cnt	1
	250 ns
	1100 ns
	1250 ns
	1750 ns

Figure 4.7: Simulation waveforms for the symmetric Golomb decompression unit

Post-synthesis reports show that the area overhead of the decompression logic for the symmetric coding scheme for group size 4, when compared to the architecture presented in [CC01a], is 10 gates and 1 flip-flop, which represents an approximately

30% increase in the area of the decoder. This fraction decreases as the group size increases.

4.5 Weighted Scan Latch Reordering (W-SLR)

In the previous sections it was shown that symmetric coding schemes are efficient for compressing scan-in vectors with low transition count. Unlike compression, power dissipation during shift depends not only on the scan-in transition count but also on the scan-out transition count. Although DC fill methods can be used to reduce the scan-in transition count, their effect on the scan-out transition count, and implicitly on the amount of power dissipated during shift, is uncontrollable. Therefore, in order to reduce power dissipation during shift, both scan-in and scan-out vectors need to be considered. Scan latch reordering (SLR) algorithms [DCPR98] have been proposed to reduce the power dissipation during shift. Existing SLR algorithms target reduction of the overall (scan-in and scan-out) transition count in the scan chain, which does not necessarily imply minimum scan-in transition count. It was shown in Section 4.3 that reducing the scan-in transition count, i.e. increasing in length the runs of 0s and 1s improves the compression when using a symmetric run-length coding scheme. Therefore, a SLR algorithm which would target both overall and scan-in transition count reduction would reduce power dissipation during shift and simultaneously improve the compression efficiency, thus reducing the tester memory requirements. A parameterisable SLR algorithm which allows *explicit control over the ratio of scan-in and scan-out transition counts*, and consequently over the compression efficiency, is presented in the following.

Due to the strong correlation between the weighted transition metric defined in Chapter 2 and the average power dissipation during shift, reducing the average power dissipation during shift is equivalent to reducing the average scan-in and scan-out weighted transition counts. Since scan latch reordering is known to be a $NP - hard$ problem [DCPR98], exact algorithms for solving it are intractable.

Therefore, a greedy heuristic will be used to find a close-to-optimal solution to this problem. The proposed heuristic consists of reordering the columns in the test set such that the Hamming distance between adjacent columns is reduced. The suitability of this heuristic is proven by representing the average WTC in terms of the Hamming distance between the adjacent columns in the test set:

$$\begin{aligned} Avg(WTC_{scan-x}) &= \sum_{i=1}^{N} WTC_{scan-x}(V_i^x)/N \\ &= \sum_{i=1}^{N}\sum_{j=1}^{m-1} (V_i^x(j) \oplus V_i^x(j+1))(m-j)/N \\ &= \sum_{j=1}^{m-1}\sum_{i=1}^{N} (V_i^x(j) \oplus V_i^x(j+1))(m-j)/N \\ &= \sum_{j=1}^{m-1} Hamm(Col_j, Col_{(j+1)})(m-j)/N \qquad (4.4) \end{aligned}$$

where Col_j represents the j^{th} column in the test set (the transposed vector containing the bits on the j^{th} position in the test vectors), $Hamm(Col_j, Col_{(j+1)})$ is the Hamming distance between columns Col_j and $Col_{(j+1)}$, and *scan-x* stands for *scan-in* or *scan-out*. The boundaries of the solution space defined by the test data compression efficiency and the average power dissipation during shift can be identified by the following two cases:

- *maximum power reduction* - achieved when scan-in and scan-out transitions are considered with equal weights during scan latch reordering; in this case, lower compression may be achieved since the SLR algorithm reduces the overall number of transitions (scan-in and scan-out), which does not necessarily lead to the minimum transition count in the scan-in vector set;

- *maximum compression* - attained when the scan-out transitions are completely neglected during scan latch reordering, the algorithm in this case aiming to minimise only the scan-in transition count; the scan power dissipation in this case may be higher than in the previous case due to the undetermined effect of scan-in-only driven scan latch reordering on the scan-out transition count;

In order to explore this solution space, the proposed SLR algorithm uses a parameterised cost function. It was shown in Section 4.2 that scan-in transitions influence the compression efficiency and the scan-in component of the power dissipation during shift, while the scan-out transitions contribute only to the scan-out component of the power dissipation during shift. Therefore, a variable weight $W \in [0,1]$ is introduced as parameter of the cost function to control the effect of the scan-out transitions on the minimisation objective:

$$Objective(W) = Avg(WTC_{scan-in}) + W \times Avg(WTC_{scan-out}) \qquad (4.5)$$

The proposed weighted scan latch reordering (W-SLR) algorithm is described in Algorithm 3. It starts by selecting the first column, Col_0, from the initial test set TS_{init} as the starting point for the reordered test set TS_{reord}. In each of the following iterations, the algorithm searches the columns which are not already in TS_{reord} and selects the column Col_j which minimises the distance $Dist(Col_{Pos}, Col_j, W)$ to the last column Col_{Pos} added to TS_{reord}. The distance between two columns Col_i and Col_j is computed using

$$\begin{aligned} Dist(Col_i, Col_j, W) &= \sum_{k \in scan-inset} (Col_i(k) \oplus Col_j(k)) + \\ & W \times \sum_{k \in scan-outset} (Col_i(k) \oplus Col_j(k)) \end{aligned} \qquad (4.6)$$

The algorithm stops when all the columns in TS_{init} had been added to TS_{reord}. The complexity of the W-SLR algorithm is $\mathcal{O}(nmlog(m))$, where n is the number of test patterns and m is the width of the test patterns.

When $W = 1$, the scan-out transitions have equal contribution with the scan-in ones to the minimisation objective. Therefore, in this case the W-SLR algorithm will minimise the overall transition count, $Avg(WTC_{scan-in}) + Avg(WTC_{scan-out})$. Maximum reduction in the overall transition count, and consequently in power dis-

Algorithm 3 Proposed scan latch reordering algorithm

```
INPUT: initial test set $TS_{init}$
OUTPUT: reordered test set $TS_{reord}$

$TS_{reord} = \emptyset$
**add** $Col_0$ from $TS_{init}$ to $TS_{reord}$
$Pos \leftarrow 0$
**while** $\exists c | c \in TS_{init}$ AND $c \notin TS_{reord}$
    **find** $Col_j \notin TS_{reord}$ such that
        $Dist(Col_{Pos}, Col_j, W) = \min_{Col_k \notin TS_{reord}} Dist(Col_{Pos}, Col_j, W)$
    **add** $Col_j$ to $TS_{reord}$
    $Pos \leftarrow Pos + 1$
**end while**
```

sipation, is achieved, however, at the expense of potentially lower compression. At the other extreme, $W = 0$, the W-SLR algorithm considers only the scan-in transitions, which leads to the minimum scan-in transition count, $Avg(WTC_{scan-in})$, and thus maximum compression. Intermediate values of W represent solutions between these two extremes.

Since test data compression targets only the scan-in test set, while transitions in both the scan-in and scan-out test sets contribute to the average power dissipation during shift, the following conclusions can be derived. Firstly, symmetric run-length coding schemes exceed standard asymmetric coding schemes in terms of compression efficiency. Secondly, controlling the transition distribution between the scan-in and scan-out test sets during scan latch reordering is essential for improving the compression efficiency simultaneously with reducing power dissipation during shift.

4.6 Experimental Results

This section presents an experimental analysis of the symmetric Golomb coding scheme and of the weighted scan latch reordering algorithm (W-SLR). Several experiments have been performed on the full-scan versions of the largest six ISCAS89 benchmarks [BBK89] and their test sets generated using MinTest [IGA] and 0 DC fill. Two software tools have been developed in order to produce the experimental results reported in this chapter. The first tool performs test data compression using the asymmetric and symmetric Golomb coding schemes. The second tool performs the W-SLR algorithm on precomputed test sets.

Several low transition count test sets have been derived from the 0 DC filled MinTest sets using the W-SLR algorithm for a wide range of values of the weighting parameter $W \in [1/30, 1]$. $W = 1/30$ was chosen as lower bound for W, based on the empirical observation that test data compression hardly improves for values of W below this limit. In order to compare the compression efficiency of the standard asymmetric Golomb coding scheme used in [CC01a] and the symmetric Golomb coding scheme presented in Section 4.3, both methods have been applied on all test sets generated using the W-SLR algorithm. The compression ratios obtained using the asymmetric ($gCmp$) and symmetric ($sCmp$) Golomb coding schemes, the scan-in WTC ($WTCin$) and total WTC ($WTCall$) for each of these experiments are reported in Table 4.3. The scan-in WTC ($WTCin$) is reported in order to illustrate the effect of the weighting parameter W on the distribution of transitions between the scan-in and scan-out sets. The last two columns show the improvements in compression ratio and WTC obtained using the proposed W-SLR algorithm in conjunction with the symmetric Golomb coding scheme when compared to the results of the method proposed in [CC01a], shown in the second and third column. For example, in the case of circuit s35932, the combination between the W-SLR algorithm and the symmetric Golomb coding scheme improves by 97% the compression ratio and by 87% the reduction in the overall WTC over the results of the method in [CC01a]. Columns 5 and 6 of Table 4.3 show the compression ratios achieved using the symmetric and respectively asymmetric Golomb coding schemes. The compression ratio achieved using the symmetric coding scheme is always higher than the compression ratio corresponding to the standard asymmetric coding scheme, which proves that the symmetric scheme is more efficient in compressing low power test sets than the standard Golomb coding scheme. Columns 5 and 7 in Table 4.3 show the relation between the scan-in WTC and the compression efficiency of the symmetric coding scheme. The compression achieved using the symmetric scheme improves as the scan-in WTC decreases.

	0 DC [CC01a]		0 DC & W-SLR					Improvement	
CUT				sCmp	gCmp			gCmp	WTCall
	gCmp(%)	WTCall	W	(%)	(%)	WTCin	WTCall	(%)	(%)
s5378	37.11	10964	1	41.93	36.12	2352	4811	4.82	56.12
			1/2	44.19	36.04	2152	5005	7.08	54.35
			1/3	46.61	35.82	2066	5159	9.50	52.94
			1/4	47.01	35.82	1973	5425	9.90	50.51
			1/5	47.46	35.87	1956	5553	10.35	49.35
			1/6	48.49	35.79	1912	5651	11.38	48.45
			1/7	48.59	35.76	1878	5885	11.48	46.32
			1/8	48.64	35.79	1882	6053	11.53	44.79
			1/9	49.03	35.73	1866	5920	11.92	46.00
			1/10	49.39	35.77	1841	5895	12.28	46.23
			1/15	49.62	35.76	1820	6562	12.51	40.14
			1/20	49.67	35.75	1864	6997	12.56	36.18
			1/25	49.99	35.74	1845	7405	12.88	32.46
			1/30	50.21	35.64	1835	7996	13.10	27.07
s9234	45.26	17207	1	48.43	43.52	2892	6830	3.17	60.30
			1/2	49.38	43.32	2607	6986	4.12	59.40
			1/3	51.49	43.20	2457	7261	6.23	57.80
			1/4	52.27	43.22	2609	7832	7.01	54.48
			1/5	52.63	43.17	2521	7751	7.37	54.95
			1/6	52.82	43.15	2523	8089	7.56	52.99
			1/7	53.22	43.15	2463	8609	7.96	49.96
			1/8	53.31	43.16	2308	8267	8.05	51.95
			1/9	53.58	43.11	2436	8709	8.32	49.38
			1/10	54.59	42.99	2564	8764	9.33	49.06
			1/15	54.56	42.98	2548	9935	9.30	42.26
			1/20	54.64	43.00	2538	9774	9.38	43.19
			1/25	54.91	42.96	2522	9886	9.65	42.54
			1/30	55.04	42.95	2507	10588	9.78	38.46
s13207	79.74	48235	1	83.95	79.22	3937	11671	4.21	75.80
			1/2	85.34	79.13	3293	12395	5.60	74.30
			1/3	85.57	79.13	3119	13035	5.83	72.97
			1/4	86.00	79.11	2966	13285	6.26	72.45
			1/5	86.09	79.09	2965	13570	6.35	71.86
			1/6	86.15	79.09	2933	13743	6.41	71.50
			1/7	86.14	79.09	2938	13896	6.40	71.19
			1/8	86.16	79.08	2920	14981	6.42	68.94
			1/9	86.25	79.08	2883	14848	6.51	69.21
			1/10	86.31	79.08	2840	15061	6.57	68.77
			1/15	86.47	79.09	2794	14780	6.73	69.35

continued on next page

	0 DC [CC01a]		0 DC & W-SLR					Improvement	
CUT				sCmp	gCmp			gCmp	WTCall
	gCmp(%)	WTCall	W	(%)	(%)	WTCin	WTCall	(%)	(%)
			1/20	86.57	79.07	2763	16059	6.83	66.70
			1/25	86.47	79.09	2754	15908	6.73	67.01
			1/30	86.49	79.08	2748	16353	6.75	66.09
s15850	62.83	49632	1	71.55	61.50	6339	18406	8.72	62.91
			1/2	73.65	61.34	5514	19114	10.82	61.48
			1/3	74.45	61.32	5394	20186	11.62	59.32
			1/4	74.69	61.30	5287	20797	11.86	58.09
			1/5	74.89	61.28	5200	21258	12.06	57.16
			1/6	75.11	61.27	5071	21624	12.28	56.43
			1/7	74.69	61.29	5119	20765	11.86	58.16
			1/8	75.09	61.25	5055	21403	12.26	56.87
			1/9	75.00	61.28	5085	21792	12.17	56.09
			1/10	75.26	61.28	5028	22025	12.43	55.62
			1/15	75.59	61.26	4967	22982	12.76	53.69
			1/20	75.97	61.25	4849	25071	13.14	49.48
			1/25	75.82	61.25	4826	25391	12.99	48.84
			1/30	75.84	61.26	4897	24110	13.01	51.42
s35932	-2.63	113481	1	93.95	-6.39	5204	14422	96.58	87.29
			1/2	95.01	-6.38	4517	14473	97.64	87.24
			1/3	95.11	-6.38	5300	15657	97.74	86.20
			1/4	95.11	-6.38	5300	15657	97.74	86.20
			1/5	95.11	-6.38	5300	15657	97.74	86.20
			1/6	95.11	-6.38	5300	15657	97.74	86.20
			1/7	95.11	-6.38	5300	15657	97.74	86.20
			1/8	95.11	-6.38	5300	15657	97.74	86.20
			1/9	95.11	-6.38	5300	15657	97.74	86.20
			1/10	95.11	-6.38	5300	15657	97.74	86.20
			1/15	95.11	-6.38	5300	15657	97.74	86.20
			1/20	95.11	-6.38	5300	15657	97.74	86.20
			1/25	95.11	-6.38	5300	15657	97.74	86.20
			1/30	95.11	-6.38	5300	15657	97.74	86.20
s38417	28.38	502791	1	72.01	26.21	40198	131984	43.63	73.74
			1/2	74.03	26.11	32605	134734	45.65	73.20
			1/3	74.67	26.08	32234	139328	46.29	72.28
			1/4	74.84	26.08	31674	142387	46.46	71.68
			1/5	75.02	26.08	30923	145764	46.64	71.00
			1/6	75.35	26.09	30149	147974	46.97	70.56
			1/7	75.33	26.08	30586	150202	46.95	70.12
			1/8	75.54	26.06	28989	153943	47.16	69.38
			1/9	75.62	26.05	30636	152950	47.24	69.57
			1/10	75.72	26.04	29180	156363	47.34	68.90
			1/15	75.90	26.03	28931	167707	47.52	66.64

continued on next page

CUT	0 DC [CC01a]		0 DC & W-SLR					Improvement	
	gCmp(%)	WTCall	W	sCmp (%)	gCmp (%)	WTCin	WTCall	gCmp (%)	WTCall (%)
			1/20	76.04	26.04	27930	176566	47.66	64.88
			1/25	75.99	26.03	28423	176898	47.61	64.81
			1/30	76.05	26.04	28124	177848	47.67	64.62

Table 4.3: Experimental results

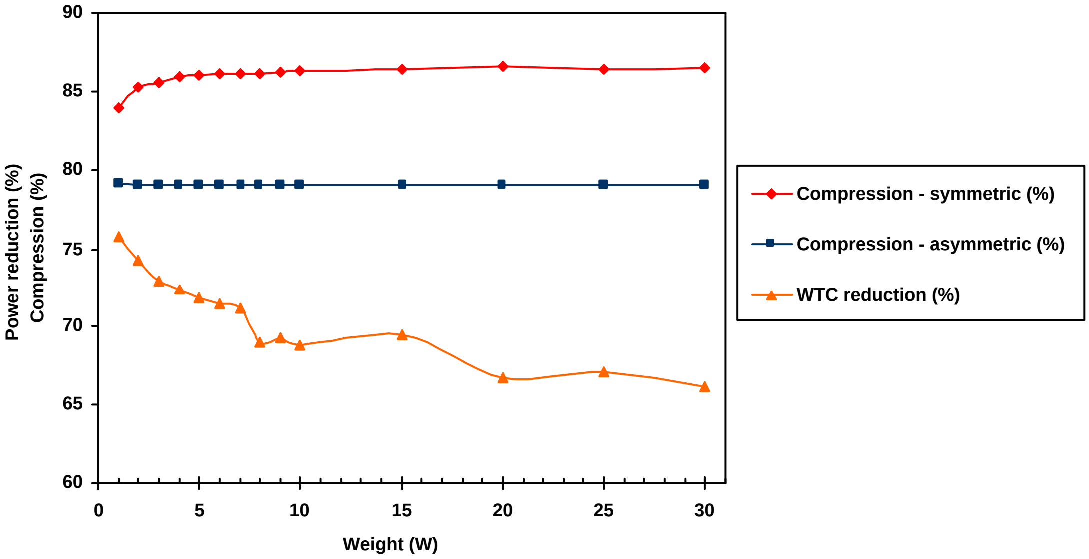


Figure 4.8: Experimental results for circuit s13207

Figure 4.8 illustrates for circuit s13207 the relation between the compression ratios achieved with the asymmetric and the symmetric coding schemes, and the reduction in the overall WTC achieved through W-SLR. It should be noted that W-SLR has a minimum impact on the compression efficiency of the standard asymmetric coding scheme. This is because in this case, the compression is mostly dependent on the density of S events in the test set, which is not affected by the W-SLR algorithm. However, the compression efficiency of the symmetric coding scheme improves asymptotically as W increases. As the improvement in the overall WTC decreases almost linearly with the variation of W, it can be concluded that the optimum solution in terms of both compression efficiency and power dissipation is obtained for $W \in [1/5, 1/3]$ in this case, i.e. the point where the steep increase of the compression ratio ends.

4.7 Concluding Remarks

This chapter has addressed the problem of compressing low power test sets. It was shown that symmetric run-length coding schemes are more efficient than asymmetric schemes in compressing low power test sets. A general method for transforming asymmetric run-length coding schemes into symmetric ones has been presented. Moreover, the parametrised scan latch reordering algorithm presented in Section 4.5 allows the core designer to select the optimum scan latch order in terms of compression and power dissipation during test.

Chapter 5

Low Power Scan Architecture

Chapters 2 and 3 have presented methods for reducing power dissipation during test based on test set transformations. In Chapter 4, test set transformations are combined with structural modifications of the circuit under test, in order to reduce power dissipation during test. This chapter will present a purely structural, and thus test set independent, method for reducing power dissipation during test.

Scan architectures represent an attractive solution for both built-in and external testing of digital integrated circuits (IC). As explained in Chapter 1, a scan-based test cycle has two distinct cycles: shift and capture. Shifting a test pattern into the scan chain occurs simultaneously with the shifting out of the circuit response for the previous test pattern. In the capture cycle, the test pattern, loaded in the the scan chain during the shift cycle, is applied to the circuit under test and the response of the circuit is captured into the scan chain.

Limited battery capacity, high cooling costs and circuit reliability are only some of the factors which made necessary to consider power consumption during IC design [Ped96]. Clock gating is probably the most efficient and commonly used approach for reducing power dissipation at register-transfer and logic level [EB00], [Syn01c]. However, traditional scan insertion cancels during test the effect of clock gating logic [EB00]. During scan testing, clock gating logic is disabled and hence all flip-flops in the design are clocked in every clock cycle. During normal operation only the flip-flops which have to be updated are clocked, while all remaining flip-flops are disabled by clock gating logic. Hence, internal switching activity during testing can exceed the level corresponding to the normal operation of the circuit. Sustained intense switching activity causes overheating and electro-migration which can permanently damage the chip under test or seriously affect its reliability. Moreover, the effects of parasitic resistance on power supply rails - combined with the large current drawn from the power grid by the large number of internal nodes which switch at the same time - reduce the voltage delivered to cells. Ignoring the effect of this reduction in voltage - referred to as IR drop - increases the probability of noise-induced test failures. Fixing IR drop-related problems requires redesigning the power grid, and hence a design re-spin. Given today's tight market windows and high costs of re-spin it is desirable that such late design failures are preempted if possible from early design stages.

Several methods aiming to solve power-related problems associated with scan-based test have been proposed recently. They fall into the following broad categories:

Minimum transition don't care fill [CC01d]. These methods assign values to don't cares in test patterns in order to reduce the number of transitions in the scan-in vectors, and hence the shift power component caused by scan-in transitions. These methods have no direct control over the number of transitions in the scan-out vectors, thus overall reduction in power cannot be guaranteed. Moreover, these methods do not address peak power problems during the capture cycle.

Power conscious ATPG algorithms [WG98, SOT00, ST02a]. These are special ATPG algorithms which aim to decrease the number of transitions in scan-in and scan-out vectors for shift power reduction, and also to decrease the Hamming distance between test stimulus vectors and the corresponding test response vectors for capture cycle power reduction. These ATPG algorithms, while overcoming the shortcomings of minimum transition don't care filling methods, are very complex and the resulting test sets are generally much larger compared to test sets generated with regular ATPG algorithms.

Special scan cells [GW99, ST02b]. The approach proposed in [GW99] inserts blocking logic on the outputs of the scan cells in order to block the shift ripple at the inputs of the circuit. Although this method reduces substantially power dissipation during the shift cycle, it introduces undesired delay on the data path due to the blocking logic which has a negative impact on circuit's performance. The work presented in [ST02b] improves the solution from [GW99] by inserting blocking logic only on the outputs of a limited number of flip-flops which are not on critical paths. The blocking logic is disabled/enabled in two additional clock cycles inserted before/after the capture clock. This way the switching caused by enabling/disabling the blocking logic does not add to the switching caused by the test response capture. Neither of these approaches addresses the problem of peak power during capture cycles.

Scan chain partitioning [BGG$^+$01, Whe00, SBW01, NAH02]. The method proposed in [BGG$^+$01] uses two non-overlapping clocks running at half the frequency of the main clock, to operate the odd and the even scan cells of the scan chain. This technique reduces shift power dissipation by a factor of approximately two, without affecting the testing time or the performance of the circuit. The approach proposed in [NAH02] splits the scan chain into multiple segments based on a compatibility relation between the flip-flops and activates only one segment in each shift clock. An extra test vector, computed using a special ATPG algorithm, is applied during the shift cycle to the primary inputs of the circuit under test in order to further reduce switching due to the shift ripple. A simpler yet very efficient approach, first proposed in [Whe00] and extended later in [SBW01], splits the scan chain into length-balanced segments and enables only one segment in each shift clock. The maximum number of scan cell outputs which are rippling in each shift clock can be tuned by selecting the appropriate number of scan segments. No blocking logic is inserted on the stimulus path, thus the performance of the design is not affected. Moreover, this method reuses test sets generated for standard scan architectures, hence it does not require special ATPG algorithms. Operating only during shift cycles (which dominate the overall testing time), these methods reduce the average power dissipation, hence eliminating the risks of overheating and electro-migration. However, in all these approaches the capture clock is applied simultaneously to all scan cells, leaving the designs prone to noise-induced test failures during capture cycles.

New approaches, easily integrable into existing automated design flows, for reducing switching activity not only during shift cycles but also during capture cycles are needed in order to provide a comprehensive and practical solution to the power-related problems associated with scan-based testing. Methods based on scan chain partitioning [BGG$^+$01, Whe00, SBW01, NAH02] appear to be efficient solutions in terms of shift power reduction vs. area overhead and integrability into existing design flows. Hence these methods deserve further investigation and provide the foundation for the work presented in this chapter. In Section 5.1, a new scan architecture is proposed with the aim of reducing both shift and capture power dissipation. Basically, the scan chain is split into a given number of length-balanced segments, and only one segment is enabled during each test clock (shift or capture) through the use of a clock gating scheme. Unlike standard scan architectures and previously proposed low power scan architectures [GW99, SBW01, Whe00, NAH02], which apply the capture clock at the same time to all scan cells, the proposed scan architecture applies sequentially the capture clocks to the segments of the scan chain. The test response is captured over a number of clocks, given by the number of scan segments. The proposed architecture reduces not only shift power but also capture power, thus eliminating the risks of overheating and electro-migration as well as the risk of high IR drops during capture cycles. An algorithmic procedure for assigning flip-flops to scan chain segments enables reuse of test vectors generated for single-clock capture. Hence, the proposed low power scan architecture does not require special ATPG algorithms to handle the multi-clock capture cycle. Section 5.2 explains how the proposed method can be applied to multiple scan chain designs. Section 5.3 presents experimental results on several benchmark circuits.

5.1 Scan Architecture with Mutually Exclusive Scan Segment Activation

With the goal of reducing the number of scan cells which are switching simultaneously during testing, the method presented in this chapter splits the scan chain into a given number of length-balanced segments, and enables only one scan segment during each test clock. At each shift clock, a test stimulus bit is shifted into the active scan segment while a test response bit from the previous test pattern is shifted out from the scan segment. Unlike all previously proposed methods based on scan chain partitioning, instead of applying the same single capture clock to all flip-flops in the design, this scan architecture captures the test response for each test pattern over a sequence of clocks cycles, one for each scan segment. Hence, only a fraction - given by the length of the scan segments - of the flip-flops in the design will be active in each test clock. This limits the maximum number of flip-flops which can toggle simultaneously, and consequently both shift and capture clock cycles will generate only a limited amount of switching activity in the circuit. This method, replacing standard scan insertion, reduces both average and peak power dissipation during test. This enables shifting of test data at high frequencies without the risk of overheating the chip under test, and also eliminates the risk of noise-induced test failures, hence avoiding unnecessary re-spins of the design.

Figure 5.1 presents the proposed low power scan architecture. The scan chain is divided into N length-balanced segments. If the number of scan cells is not a multiple of N, the sum of the differences between the scan lengths is upper bounded by N-1 (the maximum remainder of division by N). In order to account for the small length differences between the scan segments without increasing the complexity of the scan control unit, the test vectors are padded with dummy bits corresponding to the “missing” flip-flops from the shorter scan segments, as shown in Figure 5.2. The maximum number of padding bits, i.e. N-1, is much smaller than the length of the scan chain, hence these extra bits will not affect severely neither the testing time nor the test data storage requirements. All scan

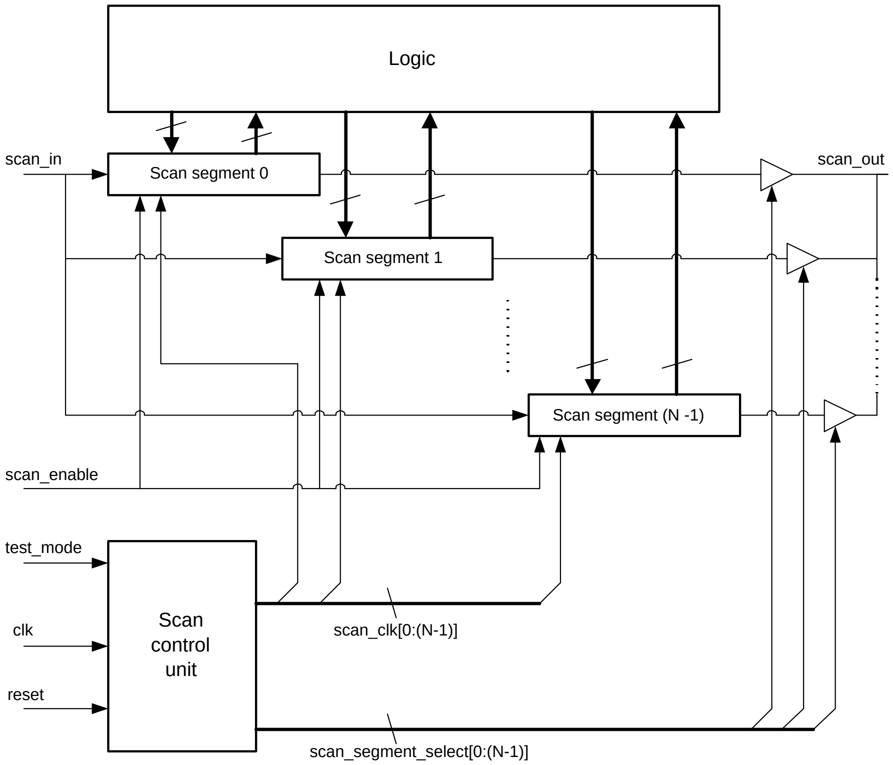


Figure 5.1: Scan architecture with mutually exclusive scan segment activation

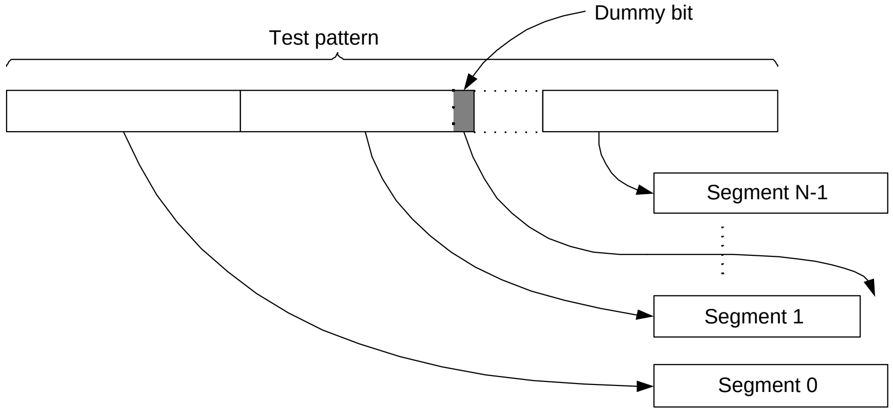
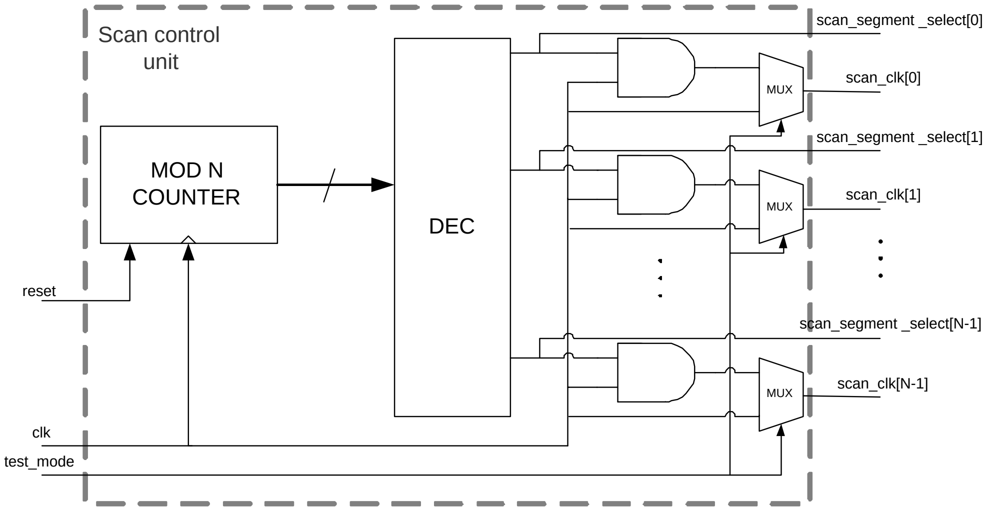


Figure 5.2: Compensating length differences among scan segments

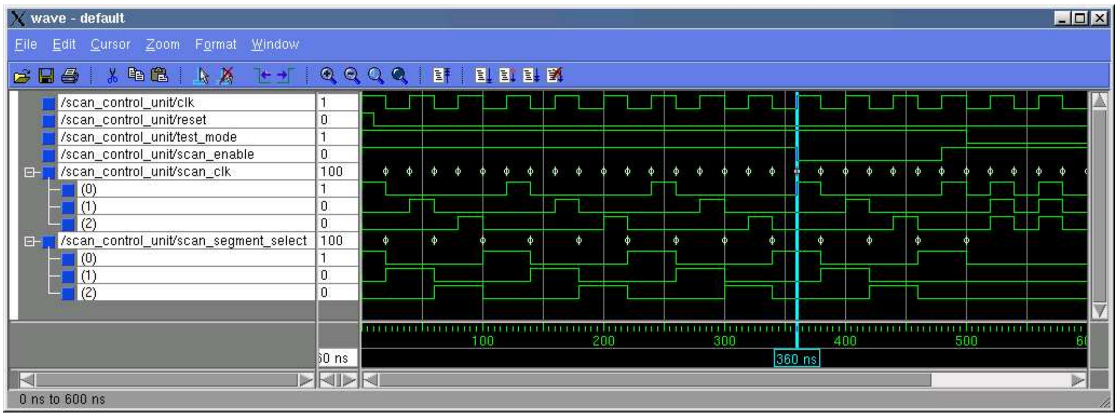
segments share the same scan-in (*scan_in*) and scan-enable (*scan_enable*) signals, but each scan segment has a separate clock signal(*scan_clk[i], i=0,N-1*). The scan segment outputs are connected to the global scan-out line (*scan_out*) through tri-state buffers controlled by mutually exclusive signals (*scan_segment_select[i], i=0,N-1*). Hence, at the boundaries of the circuit, the low power scan chain appears as a normal scan chain, with standard scan-specific I/O signals (*scan_in*, *scan_enable* and *scan_out*). The scan control unit generates N mutually-exclusive clock signals for the N scan segments. A possible implementation of the scan control unit [1], shown in Figure 5.3(a), consists of a modulo N counter, a decoder and clock gating logic. At each test clock the active output of the decoder selects the scan segment which will receive the shift or capture clock. Mutually exclusive clock signals (*scan_clk[0-(N-1)]*) for the scan segments are generated by AND-ing the system clock (*clk*) with the segment selection signals (*scan_segment_select[0-(N-1)]*). The size of the scan control unit depends only on the number of scan segments, thus it is not affected by the size of the design.

Figure 5.3(b) shows the simulation waveforms for the scan control unit for a low power scan chain architecture with three scan segments. During test mode (*test_mode* = 1), the three clock signals are mutually exclusive during both shift and capture. Initially (t = 0ns), the scan chain is in shift mode (*scan_enable* = 1). The scan segments are clocked in a cyclic sequence (segment 0, segment 1, segment 2, segment 0, ...) until all bits of a test pattern are loaded into the scan chain. At t = 360ns, the test pattern has been fully loaded into the scan chain, and the architecture is put into capture mode by asserting low the *scan_enable* signal. Three capture clocks are applied in sequence, one for each scan segment. After the first capture clock (*scan_clk[0]* = 0 - 1 - 0), the first third of the circuit response is latched into segment 0, in the second capture clock (*scan_clk[1]* = 0 - 1 - 0), another third of the test response is stored into segment 1, and in the third and last capture clock (*scan_clk[2]* = 0 - 1 - 0), the last part of the circuit response is stored into segment 2. The multi-clock capture cycle is the fundamental difference between this approach and all previously proposed low power scan architectures, which capture the entire test response in a single clock. While in the case of single-clock capture, all flip-flops in the scan chain can change their

[1] One possible solution for making the scan control unit testable is to scan its sequential part, i.e. the flip-flops of the modulo-N counter, and add observation points on its outputs (*scan_clk[0-(N-1)]* and *scan_segment_select[0-(N-1)]*).



(a) Scan control unit



/scan_control_unit/clk	1
/scan_control_unit/reset	0
/scan_control_unit/test_mode	1
/scan_control_unit/scan_enable	0
/scan_control_unit/scan_clk	100
(0)	1
(1)	0
(2)	0
/scan_control_unit/scan_segment_select	100
(0)	1
(1)	0
(2)	0

(b) Timing diagram for the scan control unit

Figure 5.3: Control unit for low power scan chain architectures

values simultaneously, the multi-clock capture cycle allows at most 1/N of the flip-flops in the design to change their value simultaneously. After N capture clocks, the entire test response is available in the scan chain, and a new shift cycle can start. During normal operation ($test\_mode = 0$), all three clocks are mapped to the system clock. Clock gating circuitry corresponding to the normal operation mode should be built on the *scan_clk* signals and it should be disabled by asserting high the *test_mode* signal.

As the scan segments are length-balanced and only one scan segment is active during each test clock, the number of simultaneously active flip-flops - i.e. the sources of switching activity in the circuit under test - can be tuned at scan insertion by selecting the appropriate number of segments for the scan chain. It should be noted that increasing the number of scan segments also increases the number of capture clocks, and hence the overall test time. However, the increase in test time is insignificant for circuits with long scan chains where the test time is dominated by the shift cycles. For example, for a circuit with a 1,000 flip-flop scan chain, partitioning the scan chain into two segments will reduce the number of simultaneously active scan cells to 50% while increasing the length of a test cycle by only one extra capture clock, which represents 0.1% of the original test time.

5.1.1 Structural Dependencies and Capture Violations

In order to reuse test stimulus and test responses generated using traditional ATPG tools for single-clock capture cycles, it is necessary to ensure throughout the multi-clock capture cycle that stimulus data bits are overwritten with test response bits only after they have become unnecessary. In this section, the structural dependencies between the flip-flops in the design are analysed in order to identify potential problems. Consider the situation shown in Figure 5.4. The test response bit which will be captured in flip-flop FF2 depends on test stimulus bit hold by flip-flop FF1 due to the existing combinational path from FF1 to FF2.

Definition *Flip-flop FF2 is said to "depend" on flip-flop FF1 if there is a combinational and/or sequential path from the output of FF1 to the input of FF2.*

Definition *A test stimulus bit hold by a flip-flop FF is said to be "necessary" if there are other flip-flops in the design which depend on FF and which have not received their capture clocks in the current capture cycle.*

According to the timing diagram shown in Figure 5.4, FF1 and FF2 are assigned to different scan segments, and hence their capture clocks do not occur simultaneously. As FF2 depends on FF1, after applying the capture clock to FF1 (*clk1* = 0 - 1 - 0), the value held by FF1, representing stimulus data for FF2, is overwritten with the test response bit.

Definition *The situation when a capture clock applied to a flip-flop in the design overwrites a necessary stimulus bit is referred to as a "capture violation".*

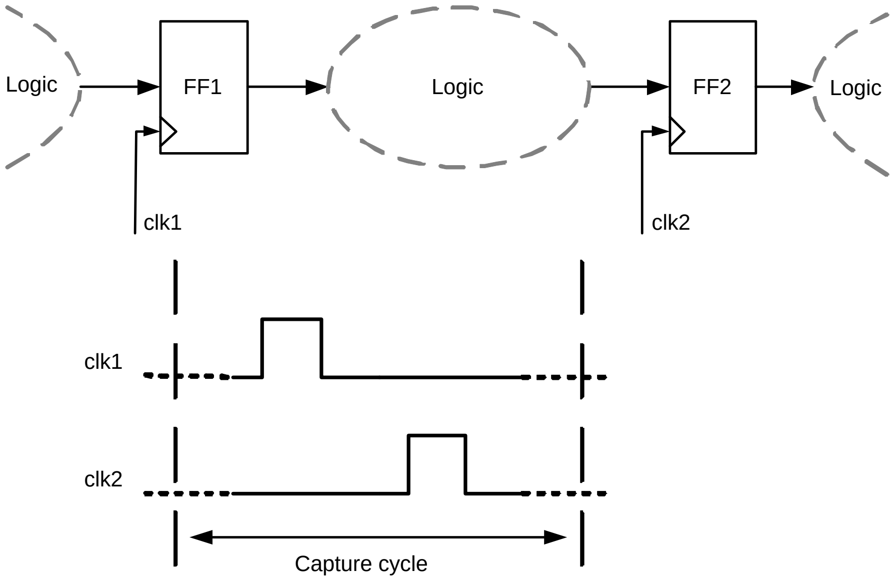


Figure 5.4: Capture violation example

The structural dependencies between flip-flops in the design have to be analysed in order to identify all possible "capture violation" situations. For this purpose, a structural dependency graph (SDG) can be derived from the net list of the design. Each node in the SDG corresponds to a flip-flop in the design, and a directed edge from node V_i to node V_j means there is a combinational path from flip-flop V_i to flip-flop V_j. According to the SDG model, V_i depends on V_j, if there is a path in the SDG from V_j to V_i. In case of a bidirectional dependency between two nodes V_i and V_j, i.e. V_i and V_j belong to a cycle in the SDG, flip-flops V_i and V_j must receive the same capture clock in order to avoid a "capture violation" situation. Generalising this observation, all nodes from a *strongly connected component* (or simply *strong component*) [Gib99] of the SDG must share the same capture clock, as there is a path between each pair of nodes of a strong component. Consider for example the SDG shown in Figure 5.5. Nodes FF4, FF5, FF6, FF7 and FF8 form a strong component as there is a path between each ordered pair of them. Applying the capture clock to one of these flip-flops before applying it to the others will result in a capture violation. For example, capturing first in FF4 will overwrite the test stimulus needed by FF5, FF6 and FF8, and so on. Therefore, flip-flops FF4, FF5, FF6, FF7 and FF8 must be assigned to the same scan segment in order to receive the same capture clock.

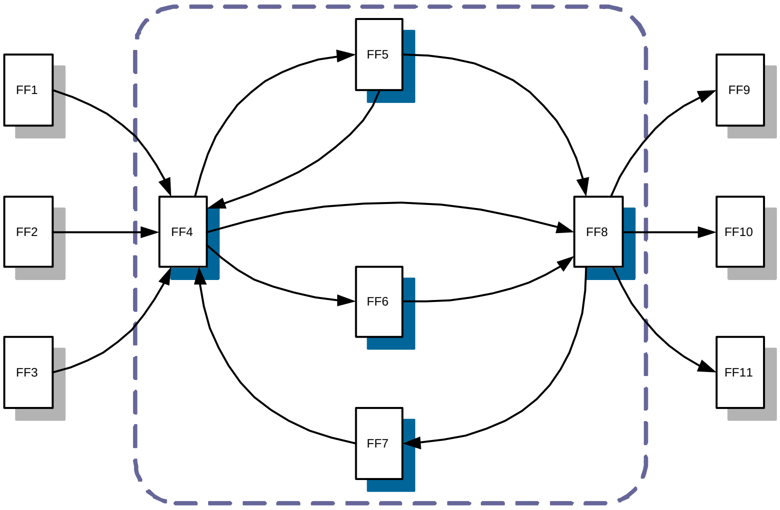


Figure 5.5: Structural dependency graph

From the above discussion it can be concluded that structural dependencies between flip-flops have to be taken into account when assigning flip-flops to scan segments in order to reuse test stimulus and test response vectors computed for single-clock capture. The following section presents a systematic method for partitioning the flip-flops in the design into equal-length scan segments and scheduling segment capture clocks while avoiding "capture violations".

5.1.2 Scan Chain Partitioning

Partitioning the flip-flops in the design into scan segments must meet the following two constraints:

- The scan segments have to be length-balanced;
- There must exist at least one ordering of the segment capture clocks which does not lead to any “capture violations” between the scan segments.

According to the low power scan architecture presented in Figure 5.1, all flip-flops assigned to a scan segment share the same clock signal. As explained earlier, all flip-flops covered by a strong component in the SDG must share the same capture clock in order to avoid “capture violations” and consequently they must be all assigned to the same scan segment. This implies that the length of the scan segments will be lower bounded by the size of the largest strong component in SDG. However, the scan segment length is imposed by the given number of scan segment, as the scan segments are length-balanced. It might happen that the size of the largest strong component in the SDG exceeds the scan segment length imposed by the number of scan segments. In this case it is necessary to "break" the largest strong component into smaller ones, which could be fitted into scan segments of the desired length. "Breaking" a strong component means removing some of the bidirectional dependencies between two or more nodes in the strong component. This can be achieved by replacing a node in the strong component with a pair of nodes: an *input-only* node and an *output-only* node. This pair will be further referred to as an “extended node”. The *input-only* node holds the stimulus bit for the fan-out logic cone, while the *output-only* node captures the test response bit from the fan-in logic cone. As between the *input-only* node and the *output-only* node there is just an unidirectional dependency, more precisely the *output-only* node depends on the *input-only* node, the two nodes can have different capture clocks, thus they can be assigned to different scan segments.

There are two alternatives for implementing extended nodes in hardware. The first possible solution is illustrated in Figure 5.6. Flip-flop SFF1 corresponds to a node selected for breaking the largest strong component in the SDG. In this extended node implementation, SFF1 is used as an *input-only* node, and an extra flip-flop SFF2 is added to act as the corresponding *output-only* node. In this solution the performance of the original circuit remains unaffected as no extra logic is added on the functional data path. For this implementation of extended nodes, the test vectors have to be padded with dummy bits on the positions corresponding to *output-only* nodes, as these nodes are used only for test response capture.

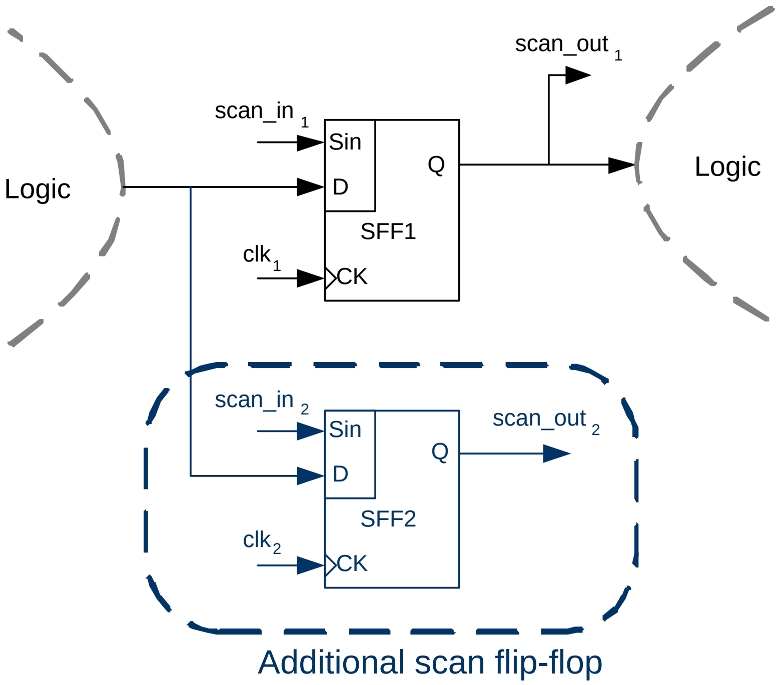


Figure 5.6: Implementing an extended node using an extra scan flip-flop

If the performance of the circuit is not critical, another solution is to implement the pair of nodes using a scan-hold flip-flop [BA00] as shown in Figure 5.7. This solution incurs less area overhead compared to the first implementation, at the cost of the extra delay introduced on the functional data path by the “hold” latch. The HOLD line of the scan-hold flip-flop is driven by to the scan enable signal. During the shift cycle ($scan\_enable = 1$), HOLD is asserted to 1 and hence, the “hold” latch is ”transparent” to the output value of the D flip-flop. During the capture cycle($scan\_enable = 0$), HOLD becomes 0, blocking the stimulus bit into the ”hold” latch, while allowing the D flip-flop to capture the test response bit without causing a “capture violation”.

The scan chain partitioning algorithm (Algorithm 4) operates on the SDG derived from the net-list of the design. The algorithm starts by computing the length of the

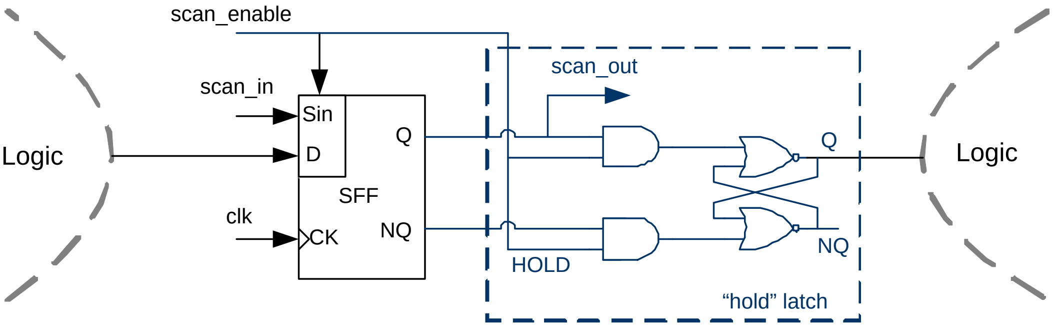


Figure 5.7: Implementing an extended node using a scan-hold flip-flop

Algorithm 4 Scan chain partitioning algorithm

INPUT: SDG, number of scan segments N_{seg}
OUTPUT: Scan segments as lists of SDG nodes

```
1  compute $L_{seg} = \lfloor N_{nodes}/N_{seg} \rfloor$, where $N_{nodes}$ is the number of nodes in $SDG$
2  compute $SSC = \{ sc \mid sc$ is a stong component in $SDG \}$
3  compute $lsc \in SSC$ where $|lsc| = \max_{sc \in SSC} |sc|$
4  if $|lsc| <= L_{seg}$ then go to line 6
5  break $lsc$ and go to line 2
6  $i = 0, C_{nodes} = \emptyset$
7  $Sseg_i = \emptyset$
8  while $|C_{nodes}| < N_{nodes}$ do {
9          find $sc \in SSC$,
               where $\forall$ fan-out node $n$ of $sc | n \in C_{nodes}$
10         if $|sc| + |Sseg_i| > L_{seg}$ then $C_{nodes} = C_{nodes} \bigcup Sseg_i$, $i = i + 1$, $Sseg_i = \emptyset$
11         $Sseg_i = Sseg_i \bigcup sc$
12 } // end while
13 if $i = N_{seg} - 1$ then STOP; else goto line 5
```

scan segments L_{seg} based on the number of flip-flops in the design N_{nodes} and the given number of scan segments N_{seg} (line 1). Next, the set of strong components SSC of SDG are identified (line 2) using a linear time search algorithm [Gib99]. If the size of largest strong component $|lsc|$ exceeds the scan segment length L_{seg} imposed by the given number of scan segments, the largest component is broken into smaller ones by replacing one of its nodes with an extended node (line 5). This step is repeated until the size of the largest strong component in the SDG becomes less than the required scan segment length L_{seg}. Once the sizes of strong components in the SDG have been adjusted according to the segment length, the algorithm proceeds to assigning the nodes in the SDG to scan segments (line 6). The set of covered nodes C_{nodes} and the first scan segment $Sseg_0$ are initialised to empty sets (lines 6 and 7). An iterative procedure starts to assign flip-flops to scan segments. At each iteration, the algorithm identifies the strong component *sc* in the SDG which has all fan-out nodes, if any, already covered, i.e. in the covered node set C_{nodes}, and adds the nodes in *sc* to the current scan segment (lines 9 to 11). Hence, during the first iterations, the primary outputs of the design, which include also the *output only* parts of extended nodes, will be assigned to the first scan segment as they do not have any fan-out nodes, i.e. no flip-flops in the design depend on them. When the number of nodes in the current scan segment reaches the scan segment length L_{seg} (line 10), the nodes in the current segment are marked as covered and a new empty segment is started. This process is repeated until all nodes in the SDG have been assigned to scan segments. If not all nodes could be fitted into the given number of scan segments (line 13), the algorithm breaks the largest strong component in the SDG and repeats the procedure of assigning strong components to scan segments. The order in which the capture clocks will be applied is the order in which the scan segments were created. This will ensure that each capture clock will overwrite only stimulus data which became unnecessary for the current capture cycle. The following example shows how scan chain partitioning works.

Example 12 *Consider the SDG shown in Figure 5.5 where nodes FF1, FF2 and FF3 are primary inputs, nodes FF9, FF10 and FF11 are primary outputs, and nodes FF4, FF5, FF6, FF7 and FF8 represent internal flip-flops. The largest strong component in this case contains five nodes, FF4, FF5, FF6, FF7 and FF8, as there is a path between each ordered pair of these nodes. Assuming the given number of scan segments N_{seg} is four, the scan segment length is three. It can be seen that for the original SDG, the size of the largest strong component exceeds the scan segment length.*

The algorithm selects node FF7 as "breaking" node for the largest strong component

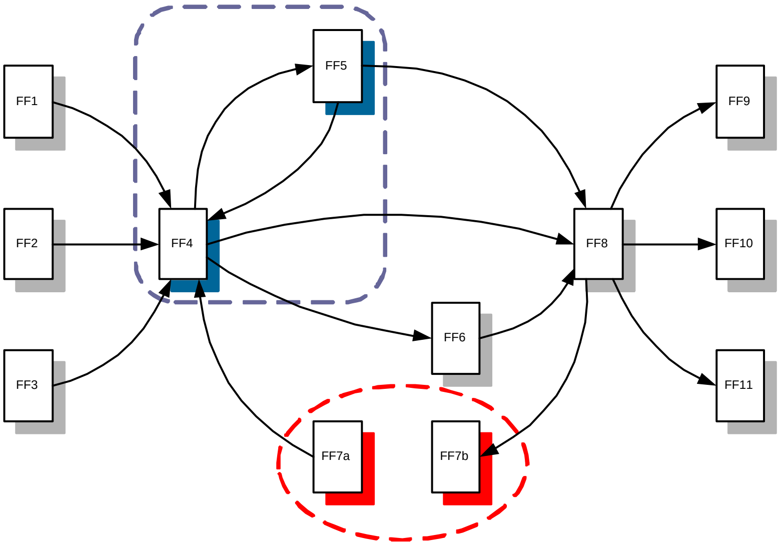


Figure 5.8: Breaking the largest strong component

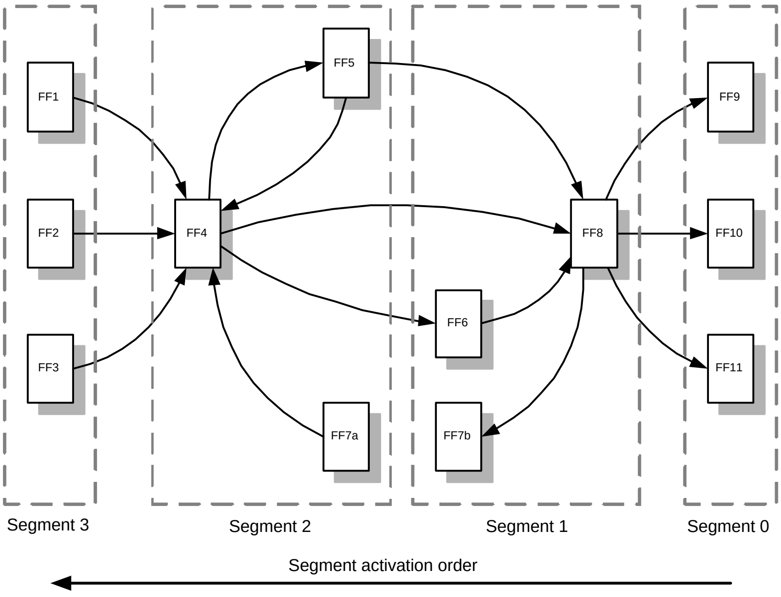


Figure 5.9: Scan segments

in SDG. Thus, node 7 will be replaced with an extended node comprising the pair (FF7a, FF7b) (Figure 5.8), where FF7a is the output only *node, while FF7b is the* input only *node. The largest strong component has now only two nodes, FF4 and FF5, which already complies with the imposed scan segment length. Analysis of the* *resulting SDG, shown in Figure 5.8, shows that flip-flops FF7a, FF4 and FF5, and FF7b, FF6 and FF8 respectively, can be assigned to different scan segments without causing "capture violations", as long as the first three flip-flops receive the capture clock after the latter three. The scan chain partitioning algorithm continues with assigning nodes to scan segments. As initially the set of covered nodes is empty, the algorithm assigns the three primary output-nodes, FF9, FF10 and FF11, to the first scan segment (Figure 5.9). This segment will receive the first capture clock in the multi-clock capture cycle as none of the remaining flip-flops in the design depends on the values of the primary outputs, and therefore no "capture violation" can occur. Next, the algorithm assigns nodes FF6, FF7b and FF8 to the following scan segment as only nodes in Segment 0 depend on them, and Segment 0 has been already scheduled for earlier capture. In a similar fashion, the algorithm assigns nodes FF4, FF5 and FF7a to Segment 2, and nodes FF1, FF2 and FF3, to Segment 3 respectively. From examining Figure 5.9 it can be observed that by applying capture clocks to Segment 0, Segment 1, Segment 2 and Segment 3 in this order, no necessary stimulus bits will be overwritten, and consequently no "capture violation" will occur.*

5.2 Low Power Multiple Scan Chain Architecture

Multiple scan chains architectures are commonly used to reduce the test time by loading in parallel a number of scan chains of the design, rather than having just one long scan chain filled serially with test data. This section explains how the low power scan chain architecture described in Section 5.1 can be applied to multiple scan chain designs.

Figure 5.10 shows a multiple scan chain architecture adapted for low power operation. The same scan control unit used for single scan chain designs generates the control signals which will be shared by all scan chains in the design. Initially, all flip-flops in the design are partitioned into scan segments, as in the case of a

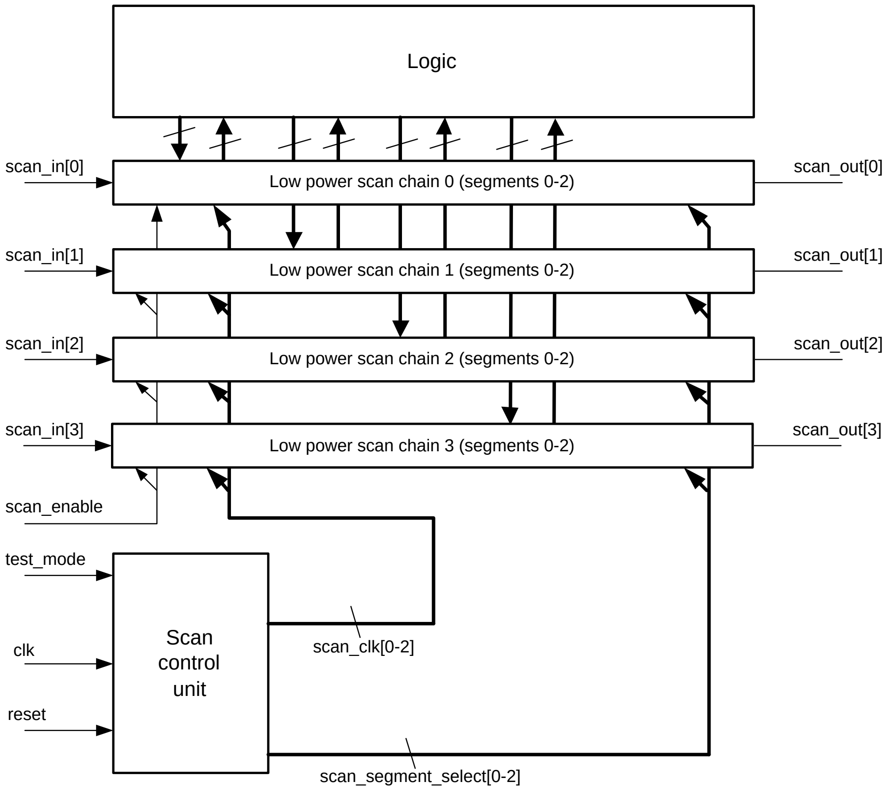


Figure 5.10: Low power multiple scan chain architecture

single scan chain architecture. The flip-flops of each scan segment are then equally distributed between the scan chains of the design. Assuming that initially there are three scan segments Segment 0, Segment 1 and Segment 2 and the design has four scan chains, each scan chain will be composed of a fourth of Segment 0, a fourth of Segment 1 and a fourth of Segment 2, which all share the scan-in line of the scan chain. The segment components of each scan chain are connected to the corresponding scan-out line through tri-state buffers controlled by the select signals generated by the scan control unit. Thus, each of the four scan chains appears as a low power scan chain as the one shown in Figure 5.1.

5.3 Experimental Results

The efficiency of the low power scan architecture described in Section 5.1, was validated by running two sets of experiments using the largest seven ISCAS89 benchmark circuits [bl]. Ten additional designs have been generated by concatenating two to seven of the largest ISCAS89 circuits, in order to assess the scalability of the proposed approach to larger designs. The number of flip-flops in the designs considered in the experiments ranged from 300 to 7000.

The goal of the first set of experiments was to estimate the reduction in average power dissipation which can be achieved using the proposed method. Six experiments were performed for each design: one experiment using standard scan chain insertion, and five experiments using the proposed scan architecture with two to six scan segments. The following flow was used in each experiment:

1. Each design was been synthesised using Alcatel MTC35000 technology library.
2. The appropriate type of scan chain (standard or low power) was inserted into the synthesised design.
3. The design was simulated using Mentor Graphics' ModelSim [Gra00b] simulator using five pseudo-randomly generated scan patterns in order to capture the toggle activity of internal nodes.
4. The toggle activity was back-annotated to the synthesised design, and an average power estimation was obtained using Synopsys' Power Compiler [Syn01c].

Table 5.1 shows the relation between the average power dissipation and the number of scan chain partitions. Column 2 corresponds to the standard single-segment scan chain, while the remaining columns show the results for the proposed low power scan chain architecture using two to six scan segments. For each of the five versions of the low power architecture, Table 5.1 reports the average power

	Std.	**2 segments**		**3 segments**		**4 segments**		**5 segments**		**6 segments**	
CUT	**Pavg**	**Pavg**	**%red**	**Pavg**	**%red**	**Pavg**	**%red**	**Pavg**	**%red**	**Pavg**	**%red**
s5378	13.23	5.62	57.53	3.75	71.62	2.85	78.47	2.28	82.78	1.82	86.23
s9234	20.70	9.24	55.36	6.42	68.99	4.72	77.22	3.83	81.50	3.14	84.85
s13207	37.35	17.39	53.43	11.43	69.39	8.75	76.56	7.01	81.24	5.84	84.37
s15850	41.06	18.79	54.25	12.50	69.56	9.34	77.27	7.34	82.13	6.21	84.88
s35932	418.74	200.53	52.11	41.82	90.01	29.30	93	23.11	94.48	20.64	95.07
s38417	402.92	196.68	51.19	36.63	90.91	27.35	93.21	21.49	94.67	17.46	95.67
s38584	372.65	185.76	50.15	28.43	92.37	21.30	94.29	16.65	95.53	13.85	96.28
cut1	397.15	193	51.40	32.81	91.74	24.80	93.76	19.55	95.08	16.48	95.85
cut2	490.60	238.65	51.36	207.10	57.79	190.42	61.19	37.36	92.38	30.64	93.76
cut3	535.62	257.44	51.94	218.31	59.24	202.41	62.21	43.48	91.88	37.31	93.03
cut4	566.58	262.09	53.74	226.39	60.04	207.47	63.38	48.08	91.51	41.26	92.72
cut5	574.39	271.66	52.70	229.98	59.96	206.29	64.09	198.89	65.37	44.77	92.20
cut6	566.58	267.60	52.77	222.99	60.64	207.49	63.38	194.51	65.67	40.71	92.81
cut7	597.92	288.53	51.74	237.04	60.35	214.67	64.10	198.57	66.79	44.56	92.55
cut8	626.19	297.10	52.55	241.32	61.46	221.94	64.56	203.26	67.54	49.61	92.08
cut9	628.51	291.09	53.69	242.13	61.48	218.42	65.25	205	67.38	192.53	69.37
cut10	681.51	325.69	52.21	262.00	61.56	234.13	65.65	217.80	68.04	192.53	69.37
Avg.	-	-	52.83	-	69.83	-	73.97	-	81.41	-	90.69
Worst	-	-	50.15	-	57.79	-	61.19	-	65.37	-	69.37

Table 5.1: Average power dissipation vs. the number of scan segments

	Std.	**2 segments**		**3 segments**		**4 segments**		**5 segments**		**6 segments**		
CUT	**FF**	**xFF**	**%**	**xFF**	**%**	**xFF**	**%**	**xFF**	**%**	**xFF**	**%**	**CPU**
s5378	314	14	4.46	5	1.59	14	4.46	19	6.05	14	4.46	0.84
s9234	389	0	0	7	1.80	6	1.54	9	2.31	9	2.31	4.82
s13207	1057	12	1.14	14	1.32	12	1.14	13	1.23	14	1.32	14.17
s15850	941	93	9.88	44	4.68	93	9.88	71	7.55	93	9.88	50.40
s35932	2242	10	0.45	11	0.49	11	0.49	12	0.54	10	0.45	92.31
s38417	2139	0	0	0	0	1	0.05	12	0.56	13	0.61	217.5
s38584	2173	253	11.64	253	11.64	253	11.64	257	11.83	253	11.64	715.64
cut1	2533	0	0	0	0	0	0	82	3.24	82	3.24	678.54
cut2	3899	0	0	71	1.82	225	5.77	207	5.31	241	6.18	791.74
cut3	4001	1	0.02	0	0	3	0.07	1	0.02	1	0.02	417.92
cut4	4864	1	0.02	0	0	1	0.02	1	0.02	0	0	593.24
cut5	5054	0	0	0	0	0	0	0	0	0	0	414.88
cut6	5062	0	0	0	0	0	0	0	0	0	0	728.86
cut7	5572	1	0.02	0	0	1	0.02	0	0	2	0.04	950.06
cut8	5922	0	0	0	0	0	0	0	0	0	0	1007.72
cut9	6159	0	0	0	0	0	0	0	0	0	0	1105.10
cut10	7080	5	0.07	0	0	0	0	2	0.03	136	1.92	1688.82
Avg.	-	-	1.63	-	1.37	-	2.06	-	2.28	-	2.48	-
Worst	-	-	11.64	-	11.64	-	11.64	-	11.83	-	11.64	-

Table 5.2: Number of extended nodes vs. the number of scan segments

dissipation(**Pavg**) as well as the relative reduction in average power dissipation (**%red**) obtained over the standard scan chain. For example, for circuit s38584, the proposed scan architecture with two scan segments reduced the average power by 50% compared to the standard scan architecture. The three scan segment architecture further reduces the average power dissipation by an additional 42%, which represents a reduction of 92% over the standard scan architecture. The last two rows in Table 5.1 show the average and worst case reductions in average power dissipation.

Table 5.2 shows the overhead associated with the proposed low power scan architecture. The increase in testing time due to the multi-clock capture cycle can be derived from the number of scan segments and the total number of flip-flops in the design. The number of flip-flops in the original designs is shown in column 2 (**FF**). Columns **xFF** show the number of extended nodes needed to implement the proposed low power scan chain for each experiment. Columns **%** show the number of extended nodes as a percentage of the total number of flip-flops in the original design. Depending on the solution used to implement the extended node, the number of these nodes represents:

- The number of extra scan cells which have to be added to the design, and also the number of additional shift clocks per test pattern,when extended nodes are implemented using extra scan flip-flops (Figure 5.6).
- The number of scan cells which have to be replaced with scan-hold flip-flops when extended nodes are implemented using scan-hold flip-flops (Figure 5.7). To be noted that in this case the total number of scan cells in the design does not increase.

Generally, the percentage of extended nodes decreases and can get as low as 0, as the number of flip-flops in the design increases. This is because ,for large designs, the length of the scan segments tend to be much higher than the size of the largest strong component in the SDG and thus only few or no extended nodes are necessary during scan chain partitioning. The last two rows in Table 5.2 report the average and worst case percentages of extended nodes. Even for the worst case scenarios, reductions up to nearly 70% in average power dissipation can be achieved using the proposed low power scan architecture at the cost of changing at most 12% of the total number of flip-flops in the design into extended nodes. The last column in Table 5.2 shows the worst case CPU times (in seconds) required to perform the scan chain partitioning algorithm and to insert the resulting scan chain into the designs. The proposed scan chain partitioning and scan insertion were performed using a tool written in C++ running on a Linux Pentium 4, 1.6GHz with 512MB of RAM.

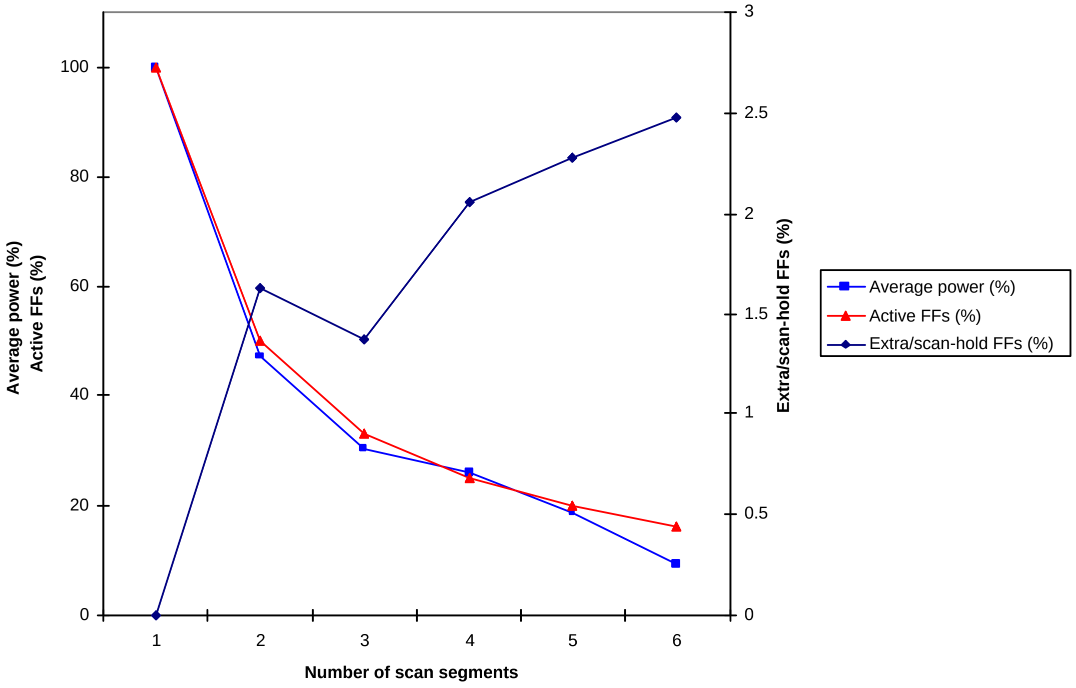


Figure 5.11: Average trends for average power and number of extended nodes

Figure 5.11 shows average trends, derived from the experimental data shown in Tables 5.1 and 5.2, for average power dissipation, number of simultaneously active flip-flops and percentage of extended nodes with respect to the number of scan segments. It is interesting to note that the average power dissipation follows closely the 1/N ratio (where N is the number of scan segments). The 1/N ratio also gives fraction of the total number of flip-flops which are simultaneously active during testing. Reduction to 1/N of the average power and of the maximum number of simultaneous switching flip-flops in the circuit is achievable at a relatively small overhead incurred by an average of 2.5% (12% for the worst case) of extended nodes.

A second set of experiments was performed in order to measure the reduction

	Std.	**2 segments**		**3 segments**		**4 segments**		**5 segments**		**6 segments**	
CUT	**mTC**	**mTC**	**%red**	**mTC**	**%red**	**mTC**	**%red**	**mTC**	**%red**	**mTC**	**%red**
s5378	2771	1676	39.52	1206	56.48	1195	56.87	982	64.56	963	65.25
s9234	4031	2540	36.99	2516	37.58	1889	53.14	1787	55.67	1697	57.90
s13207	6042	3659	39.44	2804	53.59	1760	70.87	1707	71.75	1776	70.61
s15850	10071	5161	48.75	3452	65.72	3313	67.10	2935	70.86	2498	75.20
s35932	24458	21786	10.92	14619	40.23	18566	24.09	4454	81.79	11184	54.27
s38417	17295	12918	25.31	11413	34.01	9168	46.99	6664	61.47	6148	64.45
s38584	15840	11738	25.90	9101	42.54	7812	50.68	6682	57.82	5820	63.26
cut1	21282	10483	50.74	8077	62.05	7078	66.74	5684	73.29	4203	80.25
cut2	32737	16540	49.48	19647	39.99	18079	44.78	14476	55.78	11716	64.21
cut3	40472	27635	31.72	22615	44.12	27583	31.85	11983	70.39	10382	74.35
cut4	47194	37198	21.18	18034	61.79	16094	65.90	22281	52.79	11871	74.85
cut5	47191	22944	51.38	26055	44.79	23154	50.94	18104	61.64	12984	72.49
cut6	50353	29258	41.89	20427	59.43	15876	68.47	21522	57.26	11348	77.46
cut7	54901	27084	50.67	24180	55.96	17377	68.35	22901	58.29	11766	78.57
cut8	57721	44358	23.15	24809	57.02	20444	64.58	13920	75.88	19441	66.32
cut9	59604	40023	32.85	36062	39.49	25682	56.91	14838	75.10	14267	76.06
cut10	73651	41511	43.63	39491	46.38	33874	54.00	23693	67.83	19064	74.11
Avg.	-	-	36.67	-	49.48	-	55.42	-	65.64	-	69.97
Worst	-	-	10.92	-	34.01	-	24.09	-	52.79	-	54.27

Table 5.3: Maximum number of transitions per clock vs. the number of scan segments

in the number of simultaneous transitions in the circuit under test achieved using the proposed scan architecture. The six versions of each design (standard scan architecture and proposed scan architectures with two to five scan segments), were simulated using pseudorandom test patterns in order to capture the number of transitions occurring in the circuit during each clock. Column 2 in Table 5.3 shows the maximum number of transitions per clock for each design when a standard scan architecture was used. The following columns give the maximum number of transitions per clock (**mTC**) and the relative reductions (**%red**) with respect to the values in column 2, obtained with the proposed scan chain architecture with two to six scan segments. For example, for circuit s38584, the proposed scan architecture with two segments has reduced the maximum number of simultaneous transitions by 25%. Increasing the number of scan segments to three further reduces the maximum number of transitions per clock by an additional 17%. The six segment scan chain design reduces the maximum number of simultaneous transitions by 63% compared to the design using the standard scan architecture. The last two rows show the average and worst relative reductions for all experiments.

Figure 5.12 shows the average values for the relative **mTC**, as a percentage of the values corresponding to the standard scan architecture, and the fraction of

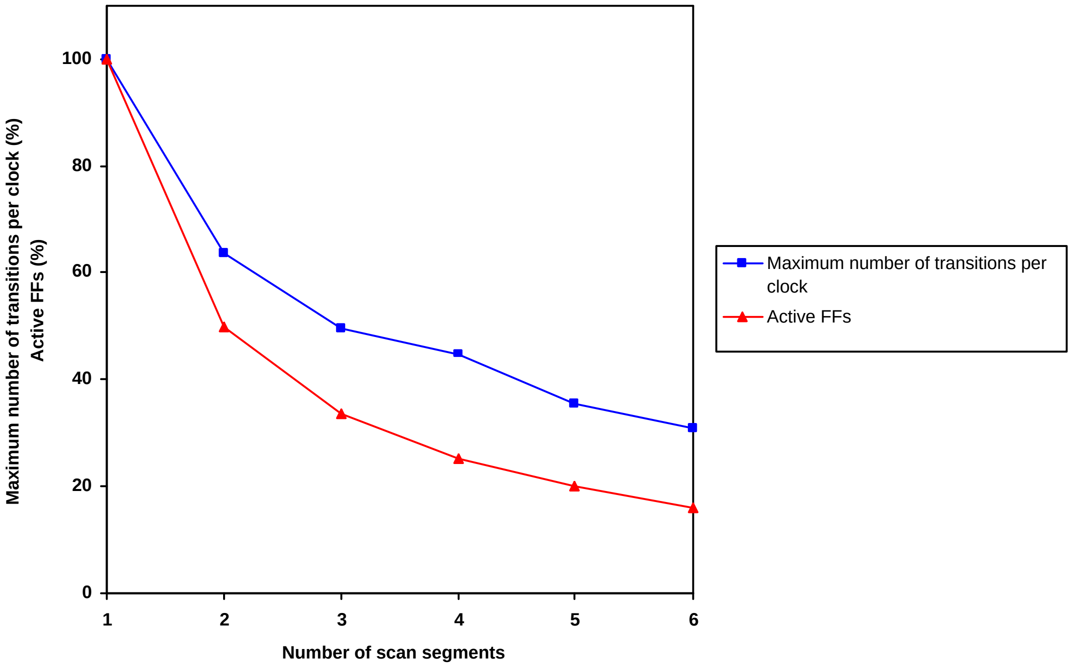


Figure 5.12: Average trends for maximum number of transitions per clock

simultaneously active flip-flops vs. the number of scan segments. Typical **mTC** values of 64% can be obtained with scan architectures with two scan segments, 50% for three scan segments, and 30% for six scan segments. The relative **mTC** is typically 15-20% higher than the percentage of active flip-flops.

5.4 Concluding Remarks

This Chapter presented a scan chain architecture using mutually exclusive scan segment activation. This architecture is capable of reducing the average power dissipation and it also eliminates peak power problems during capture cycles, which have not been addressed by previous approaches based on scan chain partitioning. The maximum number of flip-flops which can change their values simultaneously is limited to the scan segment length. Increasing the number of scan segments reduces the switching activity in the circuit under test and consequently power dissipation. The algorithmic procedure proposed for assigning flip-flops to scan segments enables full reuse of test vectors generated using standard ATPG tools without affecting their fault coverage. An implementation of the proposed method had been integrated into an automated design flow, using commercial synthesis and simulation tools, which was used for a set of experiments performed on 17 benchmark designs. These experiments showed that significant reductions in both peak and average power dissipation can be achieved using the proposed scan architecture, without affecting the performance of the designs and with minimal impact on area and test time.

Chapter 6

Conclusions and Further Work

The increasing demand for portable electronics with extended battery life and cheaper packaging and cooling made power consumption one of the major driving factors in today's semiconductor industry. As a result, most important commercial IC design tools feature power optimisation capabilities. Ensuring product quality is another key element for meeting the short time-to-market windows and surviving the aggressive competition on th IC market. Therefore, test is a necessary step of the manufacturing process which separates good devices from faulty ones, also providing diagnosis information for the faulty devices. Over the past decade, scan-based design-for-testability (DFT) has been broadly adopted as a methodology which enables automatic test pattern generation (ATPG) and provides high quality results. Methodologies such as scan-based logic built-in self-test (BIST) extend the applicability of scan-based DFT. With the increasing gate densities and rising on-chip frequencies, effects until recently ignored, such as power dissipation and resistive voltage drop, become critical issues in IC design and test. The isolation of test development from the rest of the design flow renders low power design techniques ineffective during test. DFT methodologies such as test scheduling increase the concurrency of on-chip test activities in order to reduce the overall test time. This increased level of switching activity is reflected into higher average power dissipation during test. The amount of power dissipated during test can exceed by two or three times the amount of power which is typically dissipated during the normal operation mode of the device. The most important effects of excessive power dissipation during test affects the reliability of, and, at the extreme, permanently damage , the circuit under test. Moreover, traditional scan insertion disables the effect of the clock gating logic in low power circuits, thus all flip-flops in the design are clocked in each test clock. This can cause large voltage drops, which may alter the test data loaded into the scan chains of the design, and hence make the test results unreliable. This thesis has investigated a set of power conscious DFT techniques which overcome the shortcomings of existing approaches and also solve a number of problems which have not been addressed before. An overview of the proposed solutions is presented in the following.

Chapter 2 has analysed the problem of power dissipation during test from the perspective of power constrained test scheduling (PCTS) algorithms. A common characteristic of existing PCTS algorithms is the global peak model used to describe the power dissipation during test of each of the embedded blocks (or cores) of a system. This power approximation model, being overly pessimistic, introduces a large approximation error which results in an under-optimal test concurrency, and consequently to unnecessarily long test times. A power profile manipulation technique, based on test set transformations, is proposed which allows more accurate descriptions of embedded blocks' power dissipation during test. This technique can be used in conjunction with existing test scheduling algorithms in order to enhance their performance, and hence reduce the overall test time without violating the imposed power constraint.

The work presented in Chapter 3 focuses on scan-based BIST environments. While several low power test pattern generators have been recently proposed, their common shortcoming is that they can not achieve complete fault coverage within reasonable test time. A new test pattern generator which combines the advantages of mixed-mode test generation, i.e. complete fault coverage with relatively short test sentences, with test set transformations for test power reduction is proposed. This test pattern generator combines the output sequences of two re-seedable multiple-polynomial LFSR structures in order to produce low transition test patterns with high fault coverage. An added advantage of this test pattern generator is that it does not require any changes on the circuit under test.

Chapter 4 has presented an efficient method for compressing low power test sets. In this case the reduction in power dissipation during test as well as the compression improvement are obtained simultaneously through test set transformations combined with modifications of the circuit under test. It is shown that asymmetric run-length coding schemes are inefficient in compressing low power test sets, hence leading to unnecessarily large tester memory requirements. A symmetric run-length coding scheme for compressing low power test sets is proposed. Traditional scan latch reordering algorithms, targeting only overall transition count reduction, may increase the scan-in transition count and hence adversely affect the compression efficiency of run-length coding schemes. To overcome this problem, a parametrised scan latch reordering algorithm which offers the possibility of controlling the distribution of transitions between the scan-in and scan-out sets in addition to reducing of the overall transition count is also proposed. This scan latch reordering algorithm allows the selection of the optimum solution, according to each application's specific needs, in terms of compression efficiency and test power savings.

Chapter 5 has presented a test set independent approach for reducing power dissipation during test. The solution consists of a low power scan chain architecture with mutually exclusive scan segment activation. Through careful scan chain partitioning, the proposed scan architecture reduces both average and peak power dissipation during test. The novel feature of the proposed architecture compared to exiting low power scan architecture is the peak power reduction during capture cycles. Thus, the proposed scan architecture eliminates the risk of chip overheating (due to high average power) as well as the risks of large voltage drops, during both shift and capture cycles, which can alter the test data loaded into the scan chains, and hence render the test results unreliable. The impact of the proposed architecture on area overhead is negligible compared to the test power dissipation savings and the test reliability improvement it offers.

This thesis has proposed and demonstrated four novel power-conscious DFT approaches targeting different possible scenarios.

- The **power profile manipulation** is a **system-level technique** targeting the system integration design step. This technique does not require modifications of the circuit under test, hence it is suitable for **systems built from IP cores**, where the system integrator does not have structural information about the cores, and where test patterns are applied from an external tester as the method requires the possibility of changing the order in which the test patterns are applied. The computational complexity of this technique is $\mathcal{O}(nlog(n))$, where n is the number of test patterns.

- The **low power mixed mode BIST TPG** is a **core-level solution** which does not require modifications of the circuit under test, and therefore it is suitable to the **system integration step when using IP cores** or for designs where modifications of the embedded cores must be avoided for preserving their performance. The computational complexity of this soution is $\mathcal{O}(n^3)$, where n is the degree of the feedback polynomial.

- The **weighted scan latch reordering** provides an efficient **core-level solution** for reducing simultaneously the power dissipation during test and the volume of test data. This approach requires modifications of the circuit under test, thus it targets the **core development step** designed for complex systems where the volume of test data is an important issue. The computational complexity of this soution is $\mathcal{O}(nmlog(m))$, where n is the number of test patterns and m is the length of the scan chain.

- The **low power segmented scan architecture** is another **core-level solution** which requires modifications of the circuit under test, hence it is designed for the core development step. Unlike the weighted scan latch reordering approach, this solution is **test set independent**, hence changes of the test set do not require a design re-spin. In addition, as and added benefit, this solution reduces power dissipation during capture cycles, thus it is recomended for **systems with high test concurrency** where high capture power is likely to cause noise related test failures. The computational complexity of the core algorithm is $\mathcal{O}(n)$, where n is the number of nodes of the circuit.

The efficiency of the proposed solutions have been validated through extensive experiments using in-house developed tools as well as available commercial and academic synthesis, simulation and test generation software. The results of the experimental validation have shown that power dissipation during test can be reduced without affecting test quality and with minimum impact on the cost of test. All proposed solutions have computational complexities of at most $\mathcal{O}(n^3)$, thus they are all practical for real-life designs. The remaining challenge is to fully integrate these solutions into commercial design and test tools to fully exploit their potential.

6.1 Further Work

Based on the work presented in this thesis, number of potential directions for further research toward low power test solutions have been identified and are outlined in the following.

6.1.1 Low Power Delay Test

The advancements in process technologies made possible multi-million transistor chips operating at frequencies in the gigahertz range and supply voltages approaching 1 volt. However, the increased on-chip routing length and low supply voltage have a severe impact on the delay of these circuits. As critical path delays increase, the performance of the circuit degrades increasing the risk of not meeting timing requirements. Therefore, delay tests are becoming critical for ensuring the quality of digital VLSI chips. While several low power approaches, including those presented in this thesis, have been proposed for stuck-at tests, the problem of reducing power dissipation during delay tests has not been addressed yet. Due to the growing importance of delay tests, methodologies and architectures suitable for solving this problem should be investigated.

6.1.2 Low Power Test Response Compression

Test data compression methods have been successfully used to reduce the volume of low power test stimulus data and hence the tester memory requirements. As less data has to be transfered through the slow tester channels (compared to the on-chip bus frequencies), these methods also reduce the overall test time. Normally scan-out vectors, i.e. test responses, are compacted using single/multiple input shift register (SISR/MISR) structures before being sent to the tester. The SISR/MISR compaction is a lossy compression method. For diagnosis, however, it is desired that the entire content of the scan chain is shifted-out from the chip under test to the external tester for thorough analysis. In order to reduce tester memory requirements for storing test responses and the amount of chip-to-tester data traffic, it would be convenient to compress on-chip the test responses and send the compressed result to the tester. Test data compression methods could exploit specific features of low power test responses, such as their low transition count, in order to achieve high compression with simple hardware. Unified test data compression methods targeting both stimulus and response compression for low power test sets should be investigates as they would contribute towards lowering the overall cost of test.

Appendix A

Experimental Setup for Low Power Mixed-Mode BIST

This appendix describes the experimental setup for the work presented in Chapter 3. An in-house developed tool, referred to as “dualmplfsr” in the following listing, is used computes the feedback polynomials and the initial seeds for the dual MP-LFSR structure for a given set of deterministic cubes. The tool also generates a Verilog description of the experimental configuration shown in Figure 3.3 in Section 3.6, Chapter 3, and the ModelSim simulation script. The UNIX script used to perform the experiments is shown in the following listing.

```
log=finalresults.txt
echo > $log

for circ in s526 s641 s713 s820 s832 s838 s953 s1196 s1238 s1423 s5378 \
s9234 s13207 s15850 s38417 s38584 ; do
    # $i is the current circuit under test

    echo ${circ}.scan
    echo >> $log
    echo +++++++++++++++++++++++++++++++++++++++ >> $log
    echo >> $log
    echo ${circ}.scan >>  $log

    # compute an initial set of deterministic cubes
    # in order to derive the lengths of the MP-LFSR structures
    echo atalanta ...
    atalanta -s 1 -N -X -t ${circ}.scan.test ${circ}.scan > /dev/null
    echo parse_atalanta_tests ...
    parse_atalanta_tests < ${circ}.scan.test > ${circ}.scan.test.txt1


for rs in 1000 2000 4000 8000 16000; do
    # $rs is the length of the pseudorandom sequence

    echo >> $log
    echo --------------------------------------- >> $log
    echo $rs pseudorandom vectors
    echo $rs pseudorandom vectors >> $log

    # generate $rs psudorandom vectors with the default LFSRs
    # for the previously computed plynomial degrees
    echo dualmplfsr ...
    dualmplfsr -testfile ${circ}.scan.test.txt -cmax 1 -extrabits 2 \
-randvectors 0 -randseq $rs
    echo fsim ...

    # estimate the fault coverage for the single and dual MP-LFSR TPGs
    fsim -t oos.test -l ${circ}.scan.log ${circ}.scan > /dev/null
    fsim -t rand_os.test -l ${circ}.scan.rnd.log ${circ}.scan > /dev/null
    fsim -t prand_os.test -l ${circ}.scan.prnd.log ${circ}.scan > /dev/null
    echo pseudorandom FC >> $log
    ./extractFC < ${circ}.scan.prnd.log >> $log
    echo masked pseudorandom FC >> $log
    ./extractFC < ${circ}.scan.rnd.log >> $log


    # compute uncovered faults for both TPGs
    echo extract_faults ...
    extract_faults < ${circ}.scan.log > ${circ}.scan.flt
    extract_faults < ${circ}.scan.prnd.log > ${circ}.scan.rnd.flt

    # compute deterministic cubes for the single MP-LFSR TPG
    cat ${circ}.scan.test.txt1 > ${circ}.scan.test.txt
    echo >> $log
    echo single MP-LFSR TPG: >> $log
    echo atalanta[2] ...
    atalanta -s 1 -r 0 -N -X -t ${circ}.scan.test -f ${circ}.scan.rnd.flt \
${circ}.scan > /dev/null
    echo parse_atalanta_tests[2] ...
    parse_atalanta_tests < ${circ}.scan.test >> ${circ}.scan.test.txt

    # compute feedback polynomials and initial seeds for the
    # deterministic test cubes
    echo dualmplfsr[2] ...
    dualmplfsr -testfile ${circ}.scan.test.txt -cmax 1 -extrabits 2 \
-randvectors 0 -randseq $rs >> $log
    echo fsim[2] ...


    # compute deterministic cubes for the dual MP-LFSR TPG
    cat ${circ}.scan.test.txt1 > ${circ}.scan.test.txt
    echo >> $log
    echo proposed approach: >> $log
    echo atalanta[2] ...
    atalanta -s 1 -r 0 -N -X -t ${circ}.scan.test
-f ${circ}.scan.flt ${circ}.scan > /dev/null
    echo parse_atalanta_tests[2] ...
    parse_atalanta_tests < ${circ}.scan.test >> ${circ}.scan.test.txt

    # compute feedback polynomials and initial seeds for the
    # deterministic test cubes
    echo dualmplfsr[3] ...
    dualmplfsr -testfile ${circ}.scan.test.txt -cmax 1 -extrabits 2 \
-randvectors 0 -randseq $rs -vhdlfile ${circ}_syn.vhd >> $log

    # generate Verilog files for simulation
    cat tb_common_begin.v > tb.v
    cat masklfsr.v >> tb.v
    cat lfsr.v >> tb.v
    cat tb_middle.v >> tb.v
    cat tb_common_end.v >> tb.v

    # perform simulation for capturing the switching activity
    # in the circuit
    echo Simulation ...
    vsim -do sim.do

    # estimate power dissipation based on the captured
    # switching activity
    echo Power estimation ...

    echo Single MP-LFSR >> $log
    power_estimate -hdl ${circ}_syn.vhd -f vhdl -lib MTC35000.db \
-s orig.saif testbench/cut1 -t cut | ./filterpower >> $log

    echo Dual MP-LFSR >> $log
    power_estimate -hdl ${circ}_syn.vhd -f vhdl -lib MTC35000.db \
-s proposed.saif testbench/cut2 -t cut | ./filterpower >> $log

done; # for $rs
done; # for $circ
```

The following listing shows an example of automatically-generated Verilog testbench description used for switching activity capture.

```
////////////////////////////////////////////////////////////

module clkgen(clk);

output clk;
reg clk;
initial begin: init1
        clk = 1;
        while(1)
        begin
```

```
                #25 clk = ~clk;
        end
end

endmodule

////////////////////////////////////////////////////////////

module masklfsr( clk, lfsr_out );

// LFSR of degree 21 with default initial seed

input clk;
output lfsr_out;
reg lfsr_out;
reg [0:20] seed;

initial begin: init1
        seed = 1;
        lfsr_out = 0;
end

always @(posedge clk)
begin
        seed = {(seed[20]^seed[19]^seed[17]^seed[16]^
             seed[14]^seed[13]^seed[8]^seed[3]^seed[2]^
             seed[1]^seed[0]),seed[0:19]};
        lfsr_out = seed[0];
end

endmodule

/////////////////////////////////////////////////////////////

module lfsr( clk, lfsr_out );

// LFSR of degree 41 with default initial seed

```

```
input clk;
output lfsr_out;
reg lfsr_out;
reg [0:40] seed;

initial begin: init1
        seed = 1;
        lfsr_out = 0;
end

always @(posedge clk)
begin
        seed = {(seed[40]^seed[33]^seed[32]^seed[31]^
             seed[29]^seed[27]^seed[26]^seed[25]^
             seed[24]^seed[20]^seed[19]^seed[17]^
             seed[16]^seed[14]^seed[13]^seed[12]^
             seed[10]^seed[8]^seed[5]^seed[4]^
             seed[0]),seed[0:39]};
        lfsr_out = seed[0];
end

endmodule

////////////////////////////////////////////////////////////

module testbench( reset );

parameter scanLength = 1000; // # of FFs in the scan chain

input reset;
integer bitcnt;  // # of bits shifted into the scan chain
wire clk;
wire o1,o2;      // the outputs of the two LFSRs

reg  scanin_cut1, scanin_cut2, scan_enable;
wire scanout_cut1,scanout_cut2;

clkgen ck(clk);         // instantiate clock generator
```

```
lfsr lfsr1(clk,o1);     // instantiate main LFSR
masklfsr lfsr2(clk,o2); // instantiate secondary LFSR

// instantiate two copies of the circuit
cut cut1(scanin_cut1, scanout_cut1, clk, scan_enable, reset);
cut cut2(scanin_cut2, scanout_cut2, clk, scan_enable, reset);

initial begin: init1
        bitcnt = 0;
        scanin_cut1 = 0;
        scanin_cut2 = 0;
        scan_enable = 0;
end

always @(posedge clk)
begin
        if ( bitcnt == scanLength )
        begin
            scan_enable = 0;
            bitcnt = 0;
        end
        else
        begin
             scan_enable = 1;
             bitcnt = bitcnt + 1;
        end

        // connect the main LFSR to the scan-in line of the
        //    first copy of the circuit
        scanin_cut1 = o1;

        // connect the AND composition of the outputs of the
        //   two LFSRs to the scan-in line of the second copy
        //   of the circuit
        scanin_cut2 = o1 & o2;
end

endmodule
```

The following listing shows an example of automatically-generated ModelSim simulation script used to capture the switching activity.

```
vlib work

# compile circuit under test
vcom -reportprogress 300 -work alcatel {s15850_syn.vhd}

# compile simulation testbench
vlog -reportprogress 300 -work alcatel {tb.v}

# start simulator linked with the Synopsys DPFLI external library
# for capturing switching activity
vsim -foreign "dpfli_init \
/usr/synopsys/auxx/syn/power/dpfli/lib-linux/dpfli.so"\
-L /home/paul/Tools/DesignKits/alcatel/mti5.4d/MTC35000/vit3.0/lib_sim\
alcatel.testbench

# initialise the reset signal and reset the CUT
force -freeze /testbench/reset 1 0, 0 {2ns}
run 4ns

# define as toggle capture region the first copy of the CUT
# and start capture
set_toggle_region /testbench/cut1
toggle_start

# run simulation
run 152750

# stop toggle capture and report to file
toggle_stop
toggle_report orig.saif 1e-9 /testbench/cut1

# initialise the reset signal and reset the CUT
restart -force
force -freeze /testbench/reset 1 0, 0 {2ns}
run 4ns
```

```

# define as toggle capture region the second copy of the CUT
# and start capture
set_toggle_region /testbench/cut2
toggle_start

# run simulation
run 152750

# stop toggle capture and report to file
toggle_stop
toggle_report proposed.saif 1e-9 /testbench/cut2

# exit simulation
exit -force
```

Appendix B

Decoding Unit Implementation for Symmetric Golomb Codes

This appendix provides more information on the implementation of the decoding unit for the symmetreic Golomb coding scheme presented in Chapter 4. The first listing describes the finite state machine used for decoding symmetric Golomb codes for a Golomb group size of 4.

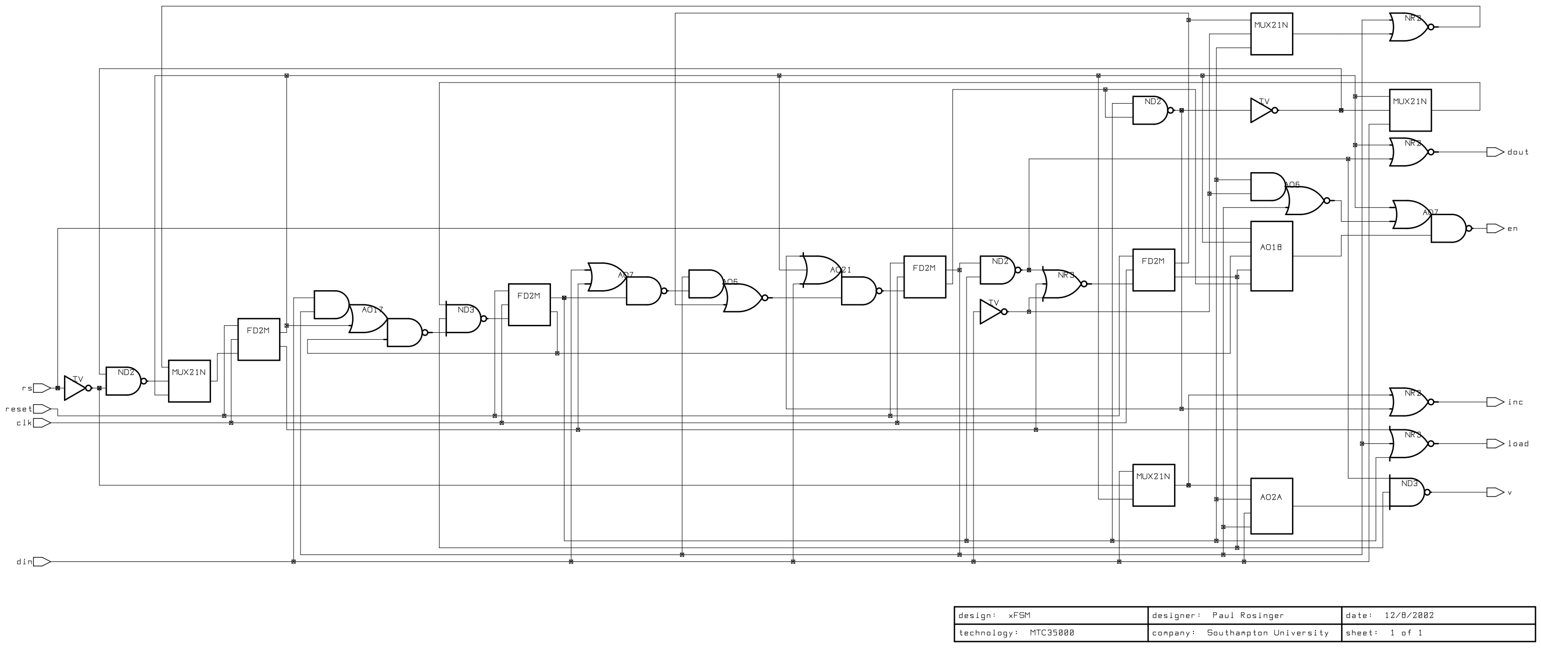
```
# Extended Golomb state machine

.design xFSM
.inputnames clk reset din rs
.outputnames en dout inc v load
.clock clk rising_edge
.asynchronous_reset reset rising S0

--      S0      S00     1-000
--      S00     S1      1-001

1-      S1      S2      00110
-0      S2      S2      00110
-1      S2      S3      1-000
1-      S3      S2      00110
0-      S3      S4      1-000
```

```

0-      S1      S4      1-000

0-      S4      S5      1-000
0-      S5      S8      0-000
1-      S5      S8      00010

1-      S4      S6      10010
0-      S6      S8      00010
1-      S6      S7      00010
--      S7      S8      00010

--      S8      S00     11010
```

The following listing is the VHDL description of the test-bench used to simulate the decoding unit for the symmetric coding scheme.

```
library ieee;
use ieee.std_logic_1164.all;
use ieee.numeric_std.all;
use work.all;

entity test_dec is
  generic ( testbits : integer := 20 );
end;

architecture testbench of test_dec is

  signal clk             : std_logic;
  signal reset           : std_logic;
  signal rs              : std_logic;
  signal dout            : std_logic;
  signal din             : std_logic;
  signal en              : std_logic;
  signal inc             : std_logic;
  signal v               : std_logic;
  signal loadSignBit     : std_logic;
  signal signBit        : std_logic;
  signal scanInData     : std_logic;

  type input_array is array (1 to testbits) of std_logic;

  -- test input data
  constant input_data : input_array := ( (
    ’1’, ’1’, ’1’, ’0’, ’0’, ’0’,       -- code for 111111110
    ’0’, ’0’, ’0’, ’1’,                 -- code for 01
    ’1’, ’1’, ’0’, ’0’, ’0’,            -- code for 11110
    ’1’, ’1’, ’0’, ’1’, ’1’             -- code for 1111110
    ) );

  component xFSM
    port (clk   : in  std_logic;
          reset : in  std_logic;
          din   : in  std_logic;
          rs    : in  std_logic;
          en    : out std_logic;
          dout  : out std_logic;
          inc   : out std_logic;
          v     : out std_logic;
          load  : out std_logic);

  end component;

  signal cnt : integer := 0;

begin

  -- instantiate extended FSM
  dec : xFSM
    port map (clk, reset, din, rs, en, dout, inc, v, loadSignBit);

  CLOCK_GENERATION : process
  begin
    CLK <= ’1’, ’0’ after 25 ns;
    wait for 50 ns;
  end process CLOCK_GENERATION;

  INITIAL_RESET : process
  begin
    reset <= '1', '0' after 120 ns;
    wait;
  end process INITIAL_RESET;

  MAIN                  : process (clk, reset)
    variable currentBit : integer := 1;
  begin

    if reset = '1' then
      -- initialisation
      din <= '0';

    elsif clk'event and clk = '0' then
      -- negative clock edge

      if loadSignBit = '1' then
        signBit <= din;
      end if;

    elsif clk'event and clk = '1' then
      -- positive clock edge

      if v = '1' then
        -- data shifted into the scan chain
        scanInData <= dout xor signBit;
      else
        -- scan chain is disabled
        scanInData <= 'X';
      end if;

      -- emulate a modulo 4 counter
      if inc = '1' then
        cnt <= cnt + 1;
      end if;
      if cnt = 3 then
        rs  <= ’1’ after 3 ns;
        cnt <= 0;
      else
        rs  <= ’0’ after 3 ns;
      end if;

      -- emulate ATE
      if currentBit <= testBits then
        if en = ’1’ then
          din       <= input_data(currentBit);
          currentBit := currentBit + 1;
        end if;
      else
        din         <= ’0’;
      end if;
    end if;
  end process MAIN;
end testbench;
```

Figure B.1 shows the synthesised finite state machine for symmetric Golomb codes and a Golomb group size 4.



design: xFSM	designer: Paul Rosinger	date: 12/8/2002
technology: MTC35000	company: Southampton University	sheet: 1 of 1

Figure B.1: Schematic of the synthesised FSM for symmetric Golomb codes and Golomb group size 4

Appendix C

Control Unit and Experimental Setup for Low Power Scan Test

This appendix provides more information on the experimental setup for the work presented in Chapter 5.

The following UNIX script was used to perform the experiments necessary to validate the proposed scan architecture with mutually exclusive scan segment activation.

```
export result=results_iscas89.txt
rm -fr $result

#echo > $result
echo > dc_shell.output

for circ in s5378 s9234 s13207 s15850 s35932 s38417 s38584 ; do
    # $circ is the current CUT

    echo
    echo XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX >> $result
    echo circuit $circ >> $result

    for nseg in 2 3 4 5; do
    # $nseg is the current number of scan segments

    echo $circ $nseg
    echo xxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxxx >> $result
    echo nsegments: $nseg >> $result

    date >> $result

    # perform scan insertion, scan chain partition and generate
    # simulation and synthesis scripts, and the pre-synthesis
    # description of the CUT with inserted scan chain

    echo Scan insertion ... >> $result
    ./netlist -net ${circ}.bench.net -verilog_pre_synth \
${circ}.v.presyn -npartitions $nseg -scan_chains ${circ}.sc \
-test_bench ${circ}_tb.v -sim_script ${circ}_sim.scr \
-vhdl_synth ${circ}_syn.vhd -synth_scr ${circ}_synth.scr \
>> $result

    date >> $result

    # synthesise the CUT
    echo Synthesis ... >> $result
    dc_shell < ${circ}_synth.scr >> dc_shell.output

    # run simulation for capturing switching activity
    date >> $result
    echo Simulation ... >> $result
    vsim -do ${circ}_sim.scr

    # estimate average power dissipation based on the
    # captured switching activity for the CUTs with
    # standard scan chain and with the low power scan
    # chain respectively
    date >> $result
    echo Power estimation ... >> $result

    echo >> $result
    if test $nseg == "2"  ; then
        echo Standard scan >> $result
        power_estimate -hdl ${circ}_syn.vhd -f vhdl -lib MTC35000.db \
-s back_full.saif testbench/cut1 -t cut | ./filterpower >> $result
    fi
    echo Low power scan >> $result
    power_estimate -hdl ${circ}_syn.vhd -f vhdl -lib MTC35000.db \
-s back_partitioned.saif testbench/cut1 -t cut | ./filterpower >> $result

done
done
```

An example of synthesis script for Synopsys' Design Compiler is shown in the following script.

```
analyze -f verilog s5378.v.presyn
elaborate cut
compile -area_effort low -exact_map -incremental
change_names -rule vhdl
write -f vhdl -hierarchy -output s5378_syn.vhd
report_area
quit
```

The following listing shows an example of automatically-generated Verilog test bench, including the scan control unit, used for simulations.

```
module scan_adapter(test_mode,clk, scan_clk, scan_segment_select,mode);

input clk, test_mode, mode;
parameter nsegments = 1, segmentlength = 1;

integer counter,i,bitcnt;

output [nsegments:0] scan_clk;
output [nsegments:0] scan_segment_select;
reg [nsegments:0] scan_clk;
reg [nsegments:0] scan_segment_select;

initial begin: init1
        i = 0;
        counter = 0;
        scan_clk = 0;
        scan_segment_select = 0;
        bitcnt = -1;

end

always @(clk)
begin

        if(mode == 1) //low power
        begin
        if(clk == 1)
        begin
                bitcnt = (bitcnt + 1 ) % ((nsegments+1)
                                 *(segmentlength+1)
                           + nsegments + 1);
                counter = (counter + 1) % (nsegments + 1);
        end
        for(i = 0; i<= nsegments; i = i + 1)
        begin
                if(bitcnt >= (nsegments+1) * (segmentlength+1))
                          scan_segment_select[i] = 0;
                else
                        scan_segment_select[i] = 1;

        end
        #10 for(i = 0; i<= nsegments; i = i + 1)
        begin
                if (test_mode == 1)
                   scan_clk[i] = clk & (counter == i);
                else
                   scan_clk[i] = clk;
        end
        end
        else //standard
        begin
                if(clk == 1)
                begin
                        bitcnt = (bitcnt + 1 ) % ((nsegments+1) *
                        (segmentlength+1) + 1);
                end
                for(i = 0; i<= nsegments; i = i + 1)
                begin

                   if(bitcnt >= (nsegments+1) * (segmentlength+1))
                          scan_segment_select[i] = 0;
                   else
                         scan_segment_select[i] = 1;
                end
                #10 for(i = 0; i<= nsegments; i = i + 1)
                begin
                   scan_clk[i] = clk;
                end
        end
end

endmodule

////////////////////////////////////////////////////////////

module clkgen(clk);
output clk;
reg clk;
initial begin: init1
        clk = 1;
while(1)
begin
        #25 clk = ~clk;
end
end
endmodule
```

```

///////////////////////////////////////////////////////////

module lfsr(clk,lfsr_out);

input clk;
output lfsr_out;
reg lfsr_out;
reg [0:8] seed;


initial begin: init1
        seed = 1;
        lfsr_out = 0;
end

always @(posedge clk)
begin
        seed = {(seed[8]^seed[3]),seed[0:7]};
        lfsr_out = seed[0];
end

endmodule

////////////////////////////////////////////////////////////

module testbench(mode,reset);
parameter nsegments = 1, segmentlength = 163;


// mode = 0 for standard scan
// mode  = 1 for low power scan

input mode,reset;
integer i;
wire clk;
wire scanin;
reg test_mode;
```

```
wire [nsegments+1:0] scan_clk;
wire [nsegments+1:0] scan_segment_select;
reg [nsegments+1:0] scanin_cut, scanclk_cut, scanena_cut;
wire [nsegments+1:0]  scanout_cut;

clkgen ck(clk);
scan_adapter #(nsegments,segmentlength) sa(test_mode,
                                        clk,
                                        scan_clk[nsegments:0],
                                        scan_segment_select[nsegments:0],
                                        mode);
lfsr lfsr1(clk,scanin);
cut     cut1(   scanin_cut,
                scanout_cut,
                scan_clk,
                scan_segment_select,
                reset);

initial begin: init1

        i = 0;
        test_mode = 1;

        scanin_cut = 0;

        scanclk_cut = 0;
        scanena_cut = 0;

end

always @clk
begin
        if(mode == 0) // standard scan
        begin
                scanin_cut[nsegments] = scanin;
                for(i = nsegments; i > 0 ;i = i-1)
                begin
                    scanin_cut[i-1] = scanin;
                end
        end
        else // low power scan
        begin

                for(i = 0;i<=nsegments; i = i+1)
                begin
                    scanin_cut[i] = scanin;
                end
        end
end

endmodule
```

The following listing shows the ModelSim script used for capturing the switching activity.

```
vlib work

# compile de CUT
vcom -reportprogress 300 -work alcatel {s5378_syn.vhd}

# compile the test-bench
vlog -reportprogress 300 -work alcatel {s5378_tb.v}

# start simulator linked with Synopsys’ DPFLI external library for
# capturing switching activity
vsim -foreign "dpfli_init\
/usr/synopsys/auxx/syn/power/dpfli/lib-linux/dpfli.so" \
-L /home/paul/Tools/DesignKits/alcatel/mti5.4d/MTC35000/vit3.0/lib_sim
\alcatel.testbench

# initialise the reset and mode signals
# and rested circuit
force -freeze /testbench/reset 1 0, 0 {2ns}

# mode = 1 stands for the low power scan chain
```

```
force -freeze /testbench/mode 1 0
run 4ns

# define the CUT as the toggle capture region
set_toggle_region /testbench/cut1

# start toggle capture
toggle_start

# run simulation
run 82500

# stop toggle capture and report to file
toggle_stop
toggle_report back_partitioned.saif 1e-9 /testbench/cut1


restart -force

# reinitialise design and select standard scan mode (mode = 0)
force -freeze /testbench/reset 1 0, 0 {2ns}
force -freeze /testbench/mode 0 0
run 4ns

# define the CUT as the toggle capture region
set_toggle_region /testbench/cut1

#start toggle capture
toggle_start

#run simulation
run 82000

# stop toggle capture and report to file
toggle_stop
toggle_report back_full.saif 1e-9 /testbench/cut1

#exit ModelSim
```

```
exit -force
```

Figure C.1 shows the gate-level schematic of the scan control unit for a design with three scan segments.

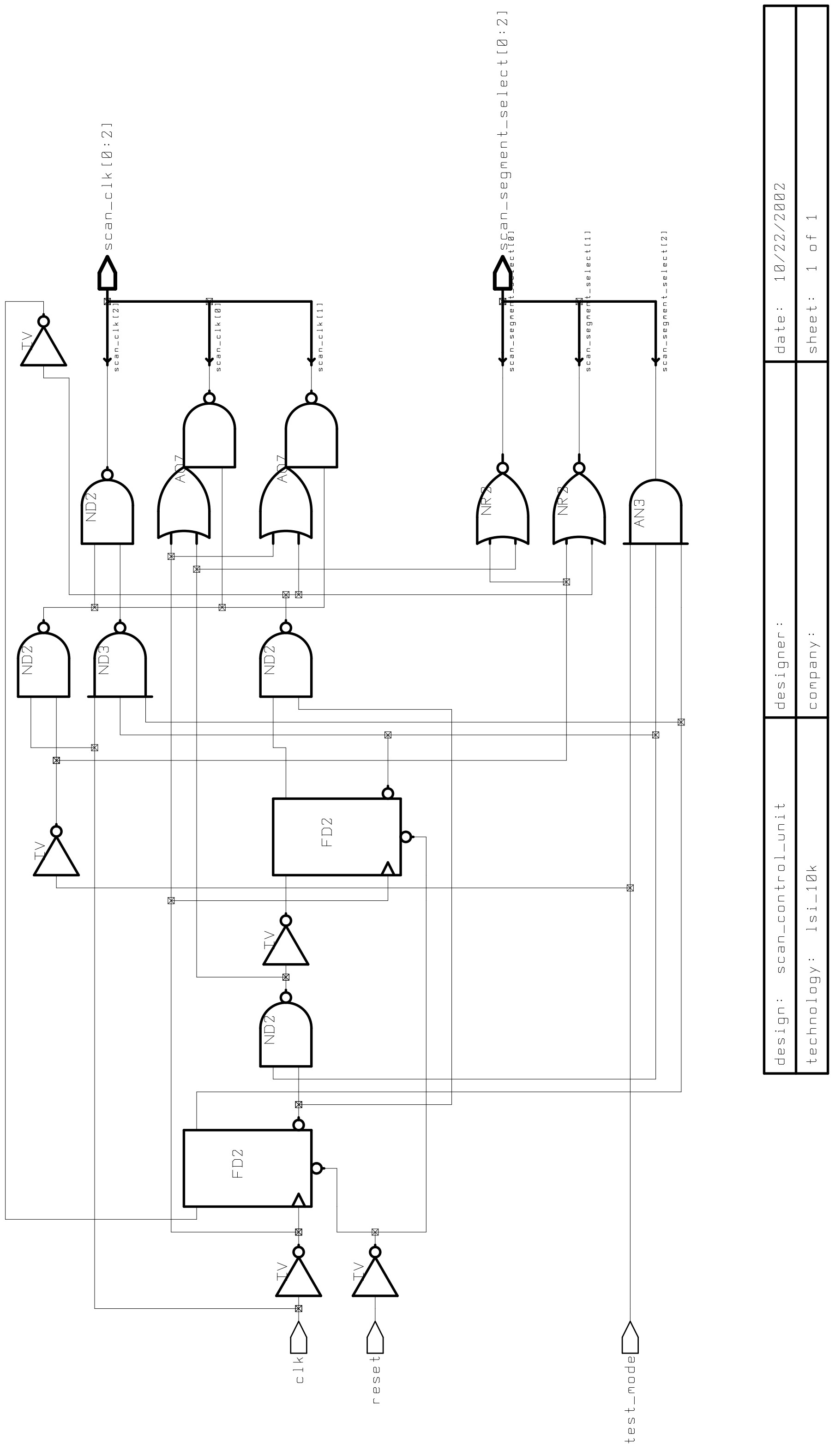


Figure C.1: Schematic of the synthesised scan control unit for three scan segments

Appendix D

Tools and Benchmark Circuits

This appendix provides brief descriptions of the benchmark circuits and tools used in the experiments referred throughout the thesis.

D.1 Academic and Commercial Software Tools

- **Academic software**
  - **ATALANTA** [Teca] is an ATPG tool for combinational circuits based on the single stuck-at fault model.
  - **FSIM** [Tecb] is a fault simulator for combinational circuits based on the single stuck-at fault model.
  - **MinTest** [IGA] is an ATPG tool for the single stuck-at fault model.
- **Commercial software**
  - **Design Compiler (Synopsys)** [Syn01a] is a behavioural to gate-level synthesis tool.
  - **Power Compiler (Synopsys)** [Syn01c] is a RT-level and gate-level power estimation tool.

- **ModelSim (Mentor Graphics)** [Gra00b] is a Verilog/VHDL simulation tool.

D.2 Benchmark Circuits

The ISCAS85 benchmark circuits [bl] are purely combinational designs with the number of gates, inputs and outputs listed in Table D.1.

Circuit	Gate count	Inputs	Outputs
c432	160	36	7
c499	202	41	32
c880	383	60	26
c1355	546	41	32
c1908	880	33	25
c2670	1193	233	140
c3540	1669	50	22
c5315	2307	178	123
c6288	2406	32	32
c7552	3512	207	108

Table D.1: The ISCAS85 benchmark suite

The ISCAS89 benchmark circuits [bl] are sequential designs with the number of gates, inputs, outputs and flip-flops listed in Table D.2.

Circuit	Gate count	Inputs	Outputs	Flip-flops
s526	141	3	6	21
s641	107	35	24	19
s713	139	35	23	19
s820	256	18	19	5
s832	262	18	19	5
s838	288	34	1	32
s953	311	16	23	29
s1196	388	14	14	18
s1238	428	14	14	18
s1423	490	17	5	74
s5378	1004	35	49	179
s9234	2027	19	22	228
s13207	2573	31	121	669
s15850	3448	14	87	597
s35932	12204	35	320	1728
s38417	8709	28	106	1636
s38584	11448	12	278	1452

Table D.2: The ISCAS89 benchmark suite

Bibliography

[ABF90] M. Abramovici, M. A. Breuer, and A. D. Friedman. *Digital Systems Testing and Testable Design*. IEEE Press, 1990.

[AJ89] S. B. Akers and W. Jansz. Test set embedding in a built-in self-test environment. In *Proc. of the IEEE International Test Conference (ITC)*, pages 257–263, 1989.

[AKS93a] V. D. Agrawal, C. R. Kime, and K. K. Saluja. A tutorial on built-in self test - part 1: Principles. *IEEE Design and Test of Computers*, 10(1):73–82, March 1993.

[AKS93b] V. D. 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):69–77, June 1993.

[AN98] A.Jas and N.A.Touba. Test vector decompression via cyclical scan chains and its application to core-based design. In *International Test Conference*, pages 458–464, 1998.

[BA00] M. L. Bushnell and V. D. Agrawal. *Essentials of Electronic Testing for Digital, Memory, and Mixed-Signal VLSI Circuits*. Kluwer Academic Publishers, 2000.

[BBK89] F. Brglez, D. Bryan, and K. Kozminski. Combinational profiles of sequential benchmark circuits. In *Proc. International Symposium on Circuits and Systems*, pages 1929–1934, 1989.

[BBM00] L. Benini, A. Bogliolo, and G. De Micheli. A survey of design techniques for system-level dynamic power management. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):299–316, June 2000.

[BCR83] Z. Barsilai, D. Coppersmith, and A. L. Rosenberg. Exhaustive generation of bit pattens with applications to vlsi self-testing. *IEEE Transactions on Computers*, 32(2):190–194, February 1983.

[BGG+01] 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. In *Proc. of the IEEE Asian Test Symposium (ATS)*, pages 253–258, November 2001.

[Bis02] B. Biswas. Crosstalk, power haunt UDSM designs. In *EE Times*, June 2002. http://www.eedesign.com/story/OEG20020604S0041.

[bl] Collaborative benchmarking laboratory. *ISCAS benchmark circuits.* North Carolina State University. http://www.cbl.ncsu.edu/CBL_Docs/Bench.html.

[Bly01] John Blyler. Dft drives quality, cost, and time to market. In *Wireless System Design*, pages 91–93, 2001.

[BMS86] P. H. Bardell, W. H. McAnney, and J. Savir. *Built-In Self Test - Pseudorandom Techniques.* John Wiley & Sons, 1986.

[BRCA01] J. Bedsole, R. Raina, A. Crouch, and M. S. Abadir. Very low cost testers: opportunities and challenges. *IEEE Design and Test of Computers*, 18(5):60–69, September-October 2001.

[Cad02] Cadence. *Physically Knowledgeable Synthesis (PKS) Datasheet*, 2002. http://www.cadence.com.

[CB95] A. P. Chandrakasan and R. W. Brodersen. *Low Power Digital CMOS Design.* Kluwer Academic Publishers, 1995.

[CC01a] A. Chandra and K. Chakrabarty. Combining low-power scan testing and test data compression for system-on-a-chip. In *Proc. of the IEEE/ACM Design Automation Conference (DAC)*, pages 166–169, 2001.

[CC01b] A. Chandra and K. Chakrabarty. Frequency-directed run-length (fdr) codes with application to system-on-a-chip test data compression. In *Proc. of the IEEE VLSI Test Symposium (VTS)*, pages 114–121, 2001.

[CC01c] A. Chandra and K. Chakrabarty. System-on-a-chip test data compression and decompression based on golomb codes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(3):355–368, March 2001.

[CC01d] A. Chandra and K. Chakrabarty. A unified approach to reduce SOC test data volume, scan power and testing time. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(3):355–368, March 2001.

[Cha00] K. Chakrabarty. Design of system-on-a-chip test access architectures under place-and-route and power constraints. In *Proc. IEEE/ACM Design Automation Conference (DAC)*, pages 432–437, 2000.

[CKS88] G. L. Craig, C. R. Kime, and K. K. Saluja. Test scheduling and control for VLSI built-in self-test. *IEEE Transactions on Computers*, 37(9):1099–1109, September 1988.

[CRRV99] F. Corno, M. Rebaudengo, M. S. Reorda, and M. Violante. Optimal vector selection for low power BIST. In *IEEE International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 219–226, 1999.

[CSA97] R. M. Chou, K. K. Saluja, and V. D. Agrawal. Scheduling tests for VLSI systems under power constraints. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 5(2):175–184, June 1997.

[DCPR98] V. Dabholkar, S. Chakravarty, I. Pomeranz, and S.M. Reddy. Techniques for minimizing power dissipation in scan and combinational circuits during test application. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(12):1325–1333, December 1998.

[DG91] W. Dufaza and G. Gambon. LFSR-based deterministic and pseudo-random test patterns gererators structures. In *European Test Conference*, pages 27–34, 1991.

[DM81] W. Daehn and J. Mucha. Hardware test pattern generators for built-in test. In *Proc. of the IEEE International Test Conference (ITC)*, pages 110–113, 1981.

[DPA84] R. Dandapani, J. H. Patel, and J. A. Abraham. Design of test pattern generators for built-in test. In *Proc. of the IEEE International Test Conference (ITC)*, pages 315–319, 1984.

[EB00] F. Emnett and M. Biegel. Power reduction through RTL clock gating. In *Synopsys Users Group (SNUG), San Jose*, 2000.

[FCN$^+$99] P. Flores, J. Costa, H. Neto, J. Monteiro, and J. Marques-Silva. Assignment and reordering of incompletely specified pattern sequences targeting minimum power dissipation. In *12th International Conference on VLSI Design*, pages 37–41, 1999.

[Gar01] R. Garcia. Rethink fault models for submicron-IC test. In *Test and Measurement World*, October 2001.

[Gib99] A. Gibbons. *Algorithmic Graph Theory.* Cambridge University Press, 1999.

[Gir00] P. Girard. Low power testing of VLSI circuits: Problems and solutions. In *First International Symposium on Quality of Electronic Design (ISQED)*, pages 173–180, 2000.

[GLPS97] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac. Reduction of power consumption during test application by test vector ordering. *IEE Electronics Letters*, 33(21):1752–1754, 1997.

[Gol66] S. W. Golomb. Run length encodings. *IEEE Transaction on Information Theory*, 18(3):172–179, July 1966.

[Gra00a] Mentor Graphics. *Mach TA and Mach PA Datasheet*, 2000. http://www.mentor.com.

[Gra00b] Mentor Graphics. *ModelSim Reference Manual*, 2000. http://www.mentor.com.

[Gre99] L. Green. The effects of signal integrity in sub-micron chip design. *Insight*, 4(3), 1999. Chips and Circuits section.

[Gro99] Technology Working Groups. http://public.itrs.net. In *International Technology Roadmap for Semiconductors*, 1999.

[GW99] S. Gerstendorfer and H. J. Wunderlich. Minimized power consumption for scan-based BIST. In *Proc. IEEE International Test Conference*, pages 77–84, 1999.

[HRT$^+$95] S. Hellebrand, J. Rajski, S. Tarnick, S. Venkataraman, and B. Courtois. Built-in test for circuits with scan based on reseeding of multiple-polynomial linear feedback shift registers. *IEEE Transactions on Computers*, 44(2):223–233, February 1995.

[HWH96] S. Hellebrand, H. J. Wunderlich, and A. Hertwig. Mixed mode BIST using embedded processors. In *Proc. of the IEEE International Test Conference (ITC)*, pages 195–204, 1996.

[IC01] V. Iyengar and K. Chakrabarty. Precedence-based, preemptive, and power-constrained test scheduling for system-on-a-chip. In *VLSI Test Symposium (VTS)*, pages 368–374, 2001.

[IGA] IGATE - University of Illinois. http://www.crhc.uiuc.edu/igate.

[Koe91] B. Koenemann. LFSR-coded test patterns for scan designs. In *European Test Conference*, pages 237–242, 1991.

[LP01] E. Larsson and Z. Peng. An integrated system-on-a-chip test framework. In *Proc. of the IEEE/ACM Design Automation Conference (DATE)*, pages 138–144, 2001.

[MGL+99] 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. In *IEEE European Test Workshop (ETW99)*, pages 165–170, 1999.

[MWMV00] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu. A comparison of classical scheduling approaches in power-constrained block-test scheduling. In *Proc. IEEE International Test Conference (ITC 2000)*, pages 882–891, 2000.

[NAH00] N. Nicolici and B. M. Al-Hashimi. Power conscious test synthesis and scheduling for BIST RTL data paths. In *Proc. of the IEEE International Test Conference (ITC)*, 2000.

[NAH02] 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):721–734, June 2002.

[NAHW00] 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 - Computers and Digital Techniques*, 147(5):313–322, September 2000.

[Nic00] N. Nicolici. *Power Minimisation Techniques for Testing Low Power VLSI Circuits.* PhD thesis, University of Southampton, UK, http://www.bib.ecs.soton.ac.uk/records/4937/, October 2000.

[Ped96] M. Pedram. Power minimization in IC design: Principles and applications. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(1):3–56, January 1996.

[RCV00] C. P. Ravikumar, G. Chandra, and A. Verma. Simultaneous module selection and scheduling for power-constrained testing of core based systems. In *13th International Conference on VLSI Design*, pages 462–467, 2000.

[Rot61] Roth. Techniques for the diagnosis of switching circuit failures. In *Proc. of Symposyum on Switching Circuit Theory and Logical Design*, 1961.

[RP96] J. M. Rabaey and M. Pedram. *Low power design methodologies.* Kluwer Academics, 1996.

[RP00] K. Roy and S. Prasad. *Low-Power CMOS VLSI Circuit Design.* John Wiley & Sons, 2000.

[RTK$^+$02] J. Rajski, J. Tyszer, M. Kassab, N. Mukherjee, R. Thompson, K. H. Tsai, A. Hertwig, N. Tamarapalli, G. Mrugalski, G. Eide, and J. Qian. Embedded deterministic test for low cost manufacturing test. In *Proc. of the IEEE International Test Conference (ITC)*, pages 301–310, 2002.

[RTZ98] J. Rajski, J. Tyszer, and N. Zacharia. Test data decompression for multiple scan designs with boundary scan. *IEEE Transactions on Computers*, 47(11):1188–1200, November 1998.

[SBW01] J. Saxena, K. M. Butler, and L. Whetsel. An analysis of power reduction techniques in scan testing. In *Proc. of the IEEE International Test Conference (ITC)*, pages 670–677, 2001.

[Sch97] H. Schwab. LP Solve. In *http://elib.zib.de/pub/Packages/mathprog/linprog/lp-solve*, 1997.

[SM00] T. Seng and E. J. McCluskey. Stuck-fault tests vs. actual defects. In *IEEE International Test Conference (ITC)*, pages 336–343, 2000.

[SOT00] R. Sankaralingam, R. R. Oruganti, and N. A. Touba. Static compaction techniques to control scan vector power dissipation. In *Proc. of the IEEE VLSI Test Symposium (VTS)*, pages 35–40, 2000.

[ST02a] R. Sankaralingam and N. A. Touba. Controlling peak power during scan testing. In *Proc. of the IEEE VLSI Test Symposium (VTS)*, pages 153–159, 2002.

[ST02b] R. Sankaralingam and N. A. Touba. Inserting test points to control peak power during scan testing. In *Proc. of the IEEE Symposium on Defect and Fault Tolerance (DFT)*, pages 138–146, 2002.

[Syn01a] Synopsys. *Design Compiler User Guide*, 2001. http://www.synopsys.com.

[Syn01b] Synopsys. *DFT Compiler User Guide*, 2001. http://www.synopsys.com.

[Syn01c] Synopsys. *Power Compiler Reference Manual*, 2001. http://www.synopsys.com.

[TCLB98] H. C. Tsai, K. T. Cheng, C. J. Lin, and S. Bhawmik. Efficient test-point selection for scan-based BIST. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 6(4):667–676, December 1998.

[Teca] Virginia Tech. ATALANTA. In *http://www.ee.vt.edu/ ”ha/ cadtools/ cadtools.html.*

[Tecb] Virginia Tech. FSIM. In *http://www.ee.vt.edu/ ”ha/ cadtools/ cadtools.html.*

[TM96] N. A. Touba and E. J. McCluskey. Altering a pseudo-random bit sequence for scan-based BIST. In *Proc. of the IEEE International Test Conference (ITC)*, pages 167–175, 1996.

[TM99] N. A. Touba and E. J. McCluskey. Rp-syn: Synthesis of random-pattern testable circuits with test point insertion. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(8):1202–1213, August 1999.

[TM01] N. A. Touba and E. J. McCluskey. Bit-fixing in pseudo-random sequences for scan bist. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(4):545–555, April 2001.

[WG98] S. Wang and S. K. Gupta. ATPG for heat dissipation minimization during test application. *IEEE Transactions on Computers*, 47(2):256–262, February 1998.

[WG99] S. Wang and S. K. Gupta. LT-RTPG: A new test-per-scan BIST TPG for low heat dissipation. In *Proc. of the IEEE International Test Conference (ITC)*, pages 85–94, 1999.

[WH92] H. J. Wunderlich and S. Hellebrand. The pseudo-exhaustive test of sequential circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 11(1):26–33, 1992.

[Whe00] L. Whetsel. Adapting scan architectures for low power operation. In *Proc. of the IEEE International Test Conference (ITC)*, pages 863–872, 2000.

[WK96] H. J. Wunderlich and G. Kiefer. Bit-flipping BIST. In *IProc of the IEEE International Conference on Computer Aided Design (ICCAD)*, 1996.

[WM86] L. T. Wang and E. J. McCluskey. Circuits for pseudo-exhaustive test pattern generation. In *Proc. of the IEEE International Test Conference (ITC)*, pages 25–37, 1986.

[Yea98] G. Yeap. *Practical low-power digital VLSI design.* Kluwer Academics, 1998.

[YTIH97] T. Yamaguchi, M. Tilgner, M. Ishida, and D. S. Ha. An efficient method for compressing test data to reduce the test data download time. In *Proc. of the IEEE International Test Conference (ITC)*, pages 79–88, 1997.

[ZDR00a] Y. Zorian, S. Dey, and M. Rodgers. Test of future system-on-chips. In *IProc of the IEEE International Conference on Computer Aided Design (ICCAD)*, pages 392–399, 2000.

[ZDR00b] Y. Zorian, S. Dey, and M. J. Rodgers. Test of Future System-on-Chips. In *IEEE International Conference on Computer Aided Design (ICCAD)*, pages 392–400, San Jose,CA, November 2000.

[Zor93] Y. Zorian. A distributed BIST control scheme for complex VLSI devices. In *Proc. of the IEEE VLSI Test Symposium (VTS)*, pages 4–9, 1993.