University of Southampton
Faculty of Engineering and Applied Science
Department of Electronics and Computer Science
Southampton S17 1BJ

Power Minimisation Techniques for Testing Low Power VLSI Circuits

by *Nicola Nicolici*
a dissertation submitted
in partial fulfillment of the requirements
for the degree of Doctor of Philosophy
at University of Southampton
October 2000

UNIVERSITY OF SOUTHAMPTON

ABSTRACT

FACULTY OF ENGINEERING
DEPARTMENT OF ELECTRONICS AND COMPUTER SCIENCE

Doctor of Philosophy

Power Minimisation Techniques for Testing Low Power VLSI Circuits

by Nicola Nicolici

Testing low power very large scale integrated (VLSI) circuits has recently become an area of concern due to yield and reliability problems. This dissertation focuses on minimising power dissipation during test application at logic level and register-transfer level (RTL) of abstraction of the VLSI design flow.

The first part of this dissertation addresses power minimisation techniques in scan sequential circuits at the logic level of abstraction. A new best primary input change (BPIC) technique based on a novel test application strategy has been proposed. The technique increases the correlation between successive states during shifting in test vectors and shifting out test responses by changing the primary inputs such that the smallest number of transitions is achieved. The new technique is test set dependent and it is applicable to small to medium sized full and partial scan sequential circuits. Since the proposed test application strategy depends only on controlling primary input change time, power is minimised with no penalty in test area, performance, test efficiency, test application time or volume of test data. Furthermore, it is shown that partial scan does not provide only the commonly known benefits such as less test area overhead and test application time, but also less power dissipation during test application when compared to full scan. To achieve power savings in large scan sequential circuits a new test set independent multiple scan chain-based technique which employs a new design for test (DFT) architecture and a novel test application strategy, is presented. The technique has been validated using benchmark examples, and it has been shown that power is minimised with low computational time, low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance.

The second part of this dissertation addresses power minimisation techniques for testing low power VLSI circuits using built-in self-test (BIST) at RTL. First, it is important to overcome the shortcomings associated with traditional BIST methodologies. It is shown how a new BIST methodology for RTL data paths using a novel concept called test compatibility classes (TCC) overcomes high test application time, BIST area overhead, performance degradation, volume of test data, fault-escape probability, and complexity of the testable design space exploration. Second, power minimisation in BIST RTL data paths is achieved by analysing the effect of test synthesis and test scheduling on power dissipation during test application and by employing new power conscious test synthesis and test scheduling algorithms. Third, the new BIST methodology has been validated using benchmark examples. Further, it is shown that when the proposed power conscious test synthesis and test scheduling is combined with novel test compatibility classes simultaneous reduction in test application time and power dissipation is achieved with low overhead in computational time.

Contents

Acknowledgement

My supervisor, Dr. Bashir Al-Hashimi, has provided me with guidance and encouragement needed to complete the work presented in this dissertation. I am very grateful to him for teaching me how to conduct research, how to clearly and effectively present my work, and how to organise and write technical papers.

The members of the Electronic Systems Design Group in the Department of Electronics and Computer Science of University of Southampton have inspired me with their support. I wish to acknowledge Professor Andrew Brown and Dr. Mark Zwolinski for serving on my MPhil to PhD transfer examination, as well as Dr. Alan Williams. I am thankful to Theo Gonciari, Paul Rosinger, Marcus Schmitz, and Mauricio Varea, for their constant friendship and providing appreciated feedback on several occasions.

I want to acknowledge Staffordshire University, where I have started my graduate studies and made many friends, to whom I give thanks. I am grateful to Dr. Peter Kollig of Philips Semiconductors, for many technical and friendly discussions. Also, I would like to thank my former colleagues and mentors at the University of Timişoara, particularly Professor Mircea Vlăduţiu and Dr. Petru Eleş, as well as University of Illinois, Virginia Polytechnic, and University of Southern California for providing valuable references.

I am especially indebted to Tanti Dorica and many friends from my hometown Timişoara in Romania. Thank you for being a part of my life, for so many years, with your unlimited friendship: Crapu, Frigu, Ghiolbanu, Maşi, Ochelari, Părinte, Supi, Ţavă, and Ţrni.

Members of my family, my cousin Radoica, aunt Ţveta, uncle Emanoil, and many more than I can include here, have loved me far beyond what I could ever return. Special thanks to my aunt Mitra, uncles Nica and Cedomir, and great uncle Staniša for their steady support over the years. No words would be enough to thank my mother Liubinca, father Ilia, brother Miloş, and grandmother Liubiţa. Therefore, to them and to the memory of my grandparents Dara, Jarco, and Rada, I dedicate this dissertation as a token of gratitude.

Abbreviations

ALAP	As Late As Possible
ALU	Arithmetic Logic Unit
ASAP	As Soon As Possible
ATE	Automatic Test Equipment
ATPG	Automatic Test Pattern Generation
BAO	BIST Area Overhead
BILBO	Built-In Logic Block Observer
BIST	Built-in Self-Test
BPIC	Best Primary Input Change
CA	Cellular Automata
CAD	Computer-Aided Design
CBILBO	Concurrent Built-In Logic Block Observer
CMOS	Complementary Metal-Oxide Semiconductor
CUT	Circuit Under Test
DCT	Discrete Cosine Transform
DFT	Design For Test
EDA	Electronic Design Automation
ESC	Extra Scan Chain
EWF	Elliptic Wave Filter
G-TIG	Global Test Incompatibility Graph
FT-DP	Fully Testable Data Path
HDL	Hardware Description Language
HLS	High Level Synthesis
LFSR	Linear Feedback Shift Register
NTC	Node Transition Count

MISR	Multiple-Input Signature Analyser
MSC	Multiple Scan Chains
PC-TSS	Power Conscious Test Synthesis and Scheduling
PD	Performance Degradation
PT-DP	Partially Testable Data Path
RAM	Random Access Memory
RTL	Register-Transfer Level
SA	Signature Analyser
SP	Shifting Power dissipation
SR	Shift Register
TAP	Test Application Power dissipation
TAT	Test Application Time
TA-TSS	Time and Area Test Synthesis and Scheduling
TCC	Test Compatibility Classes
TCG	Test Compatibility Graph
TIG	Test Incompatibility Graph
TPG	Test Pattern Generator
VHDL	VHSIC Hardware Description Language
VLSI	Very Large Scale Integration
VTD	Volume of Test Data

Chapter 1

Introduction

The topic of this investigation is power minimisation during test application in low power digital very large scale integrated (VLSI) circuits. This is a sub-problem of the general goal of testing VLSI circuits. Testing VLSI circuits bridges the gap between the imperfection of integrated circuit (IC) manufacturing and consumer expectations of flawless products [94]. Manufacturers test their products and discard the defective components to ensure that only defect free chips make their way to the consumer. With the advent of deep sub-micron technology [138], the tight constraints on power dissipation of VLSI circuits have created new challenges for testing low power VLSI circuits which need to overcome the traditional test techniques that do not account for power dissipation during test application.

The aim of this chapter is to place the problem of testing low power VLSI circuits within the general context of the VLSI design flow. The rest of the chapter is organised as follows. Section 1.1 overviews the VLSI design flow and outlines the importance of testing integrated circuits. External testing using automatic test equipment and the need for design for test (DFT) methods are described in Section 1.2. Section 1.3 introduces built-in self-test (BIST) and provides the terminology used throughout the dissertation with the help of detailed examples. Section 1.4 describes the importance of power minimisation during test application for low power VLSI circuits which will lead to an improvement in both test efficiency and circuit yield and reliability. Finally Section 1.5 provides an overview of the dissertation and outlines the main contributions of the research programme.

1.1 VLSI Design Flow

In the complementary metal-oxide semiconductor (CMOS) technology, process technologies race to keep pace with Moore's law which observes that chip processing power doubles every 18 months [7]. While the increase in integration comes with numerous beneficial effects, the perception of the faulty behaviour is changing. This section introduces the design and test flow of CMOS integrated circuits, which is the dominant fabrication technology for implementation of VLSI circuits that contain more than 10^5 transistors [59].

As shown in Figure 1.1 the design flow of VLSI circuits is divided into three main steps: specification, implementation and manufacturing [148]. *Specification* is the step of describing the functionality of the VLSI circuit. The specification is done in hardware description languages (HDLs), such as VHDL [136] or Verilog [137] in two different design domains, the behavioural domain or the structural domain [59], at various levels of abstraction. For example the *logic level of abstraction* is represented by means of expressions in Boolean algebra in the behavioural domain, or interconnection of logic gates in the structural domain. Going up in abstraction level, one reaches the register-transfer level. *Register-transfer level (RTL)* is the abstraction level of the VLSI design flow where an integrated circuit is seen as sequential logic consisting of registers and functional units that compute the next state given the current memory state. The highest level for system specification is the *algorithmic level* where the specification consists of tasks that describe the abstract functionality of the system.

Implementation is the step of generating a structural netlist of components that perform the functions required by the specification. According to the design methodology, the implementation can be either full custom or semicustom [51]. In the full custom design methodology the design is handcrafted requiring an extensive effort of a design team to optimise each detailed feature of the circuit. In semicustom design methodology, which can be either library cell-based or gate array-based, a significant portion of the implementation is done automatically using computer-aided design (CAD) tools. CAD tools are used to capture the initial specification in hardware description languages, to translate the initial specification into internal representation, to translate the behaviour into structural implementation, to optimise the resulted netlist, to map the circuit into physical logic gates , and to route the connections between gates.

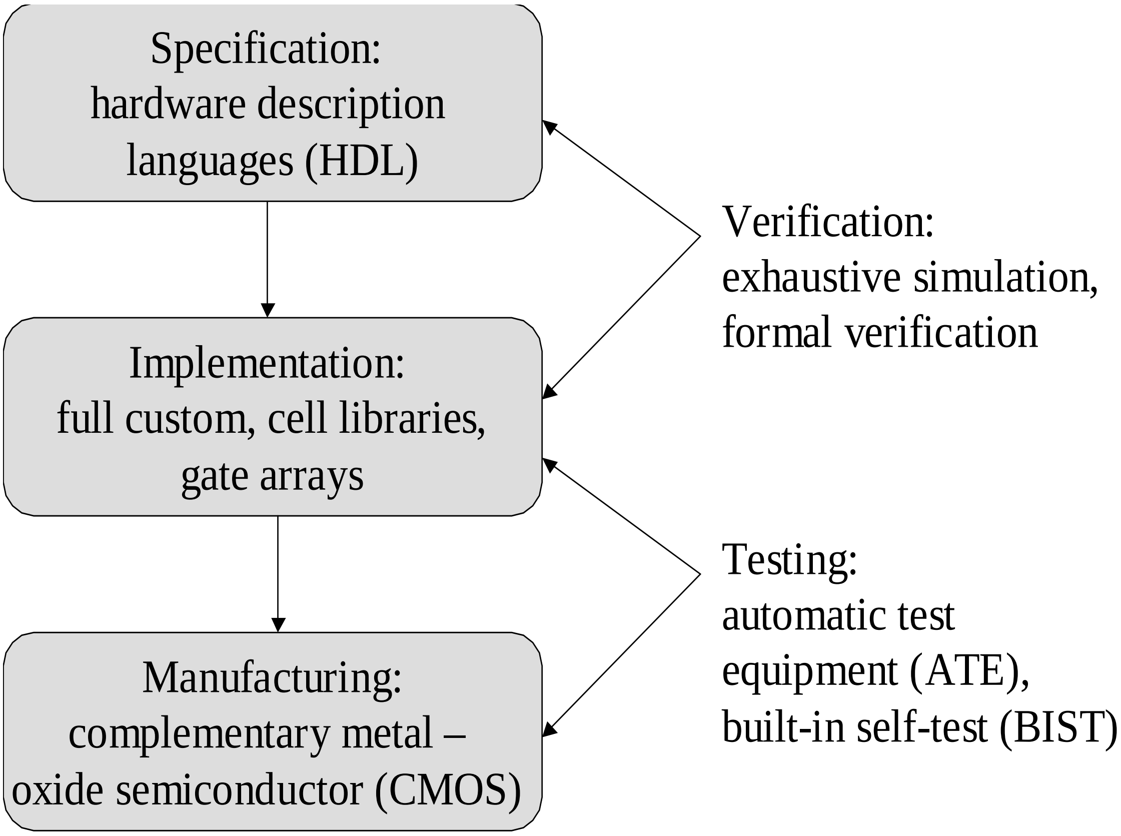


Figure 1.1: VLSI design flow.

Manufacturing is the final step of the VLSI design flow and it results in a physical circuit realised in a fabrication technology with transistors connected as specified by implementation. The term fabrication technology refers to the semiconductor process used to produce the circuit which can be characterised by the type of semiconductor (e.g. silicon), the type of transistors (e.g. CMOS), and the details of a certain transistor technology (e.g. 0.35 micron). CMOS technology is the dominant technology for manufacturing VLSI circuits and it is considered throughout this dissertation. From now onwards, unless explicitly specified, the term VLSI circuit refers to CMOS VLSI circuit. Due to significant improvement in the fabrication technology designers can place millions of transistors on a single piece of silicon that only accommodated thousands of transistors a few decades ago. However, complex designs are more failure-prone during the design flow. Therefore, to increase the reliability of the final manufactured product two more problems have to be addressed during the early stages of the VLSI design flow shown in Figure 1.1: *verification* and *testing*.

Verification involves comparing the implementation to the initial specification. If there are mismatches during verification, then the implementation may need to be modified to more closely match the specification [128]. In traditional VLSI design flow the comparison between specification and implementation is accomplished through exhaustive simulation. Because exhaustive simulation for complex designs is practically infeasible, simulation provides at best only a probabilistic assurance. Formal verification, in contrast to simulation uses rigorous mathematical reasoning to prove that an implementation meets all or parts of its specification [95].

Testing assures that the function of each manufactured circuit corresponds to the function of the implementation [2, 127]. Producing reliable VLSI circuits depends strongly on testing to eliminate various defects caused by the manufacturing process. Basic types of defects in VLSI circuits [138] are the following: particles (small bits of material that bridge two lines), incorrect spacing, incorrect implant value, misalignment, holes (exposed area that is unexpectedly etched), weak oxides, and contamination. The defects lead to faulty behaviour of the circuit which can be determined either by parametrical testing [163] or logic testing [2]. Testing circuits parametrically includes measuring the current flowing through the power supply in the quiescent or static state [163]. As CMOS technology scales down parametric testing is no longer practical due to an increase in sub-threshold leakage current. Logic testing involves modelling manufacturing defects at the logic level of abstraction of the VLSI design flow, where faulty behaviour is measured by the logic value of the primary outputs of the circuit [2]. The basic fault models for logic testing are stuck-at fault model, bridging fault model, open fault model, and timing related fault models such as gate delay and path delay fault models [7]. The earliest and most common fault model is the stuck-at fault model where single nodes in the structural netlist of logic gates are assumed to have taken a fixed logic value (and thus is stuck-at either 0 or 1). From now onwards throughout this dissertation, testing VLSI circuits refers to the most common and generally accepted logic testing for stuck-at fault model.

Having described manufacturing defects and their fault models, the following two sections describe how test patterns are applied to the circuit under test to distinguish the fault free and faulty circuits. The application of test patterns to detect faulty circuits can be done either externally using automatic test equipment (Section 1.2) or internally using built-in self-test (Section 1.3).

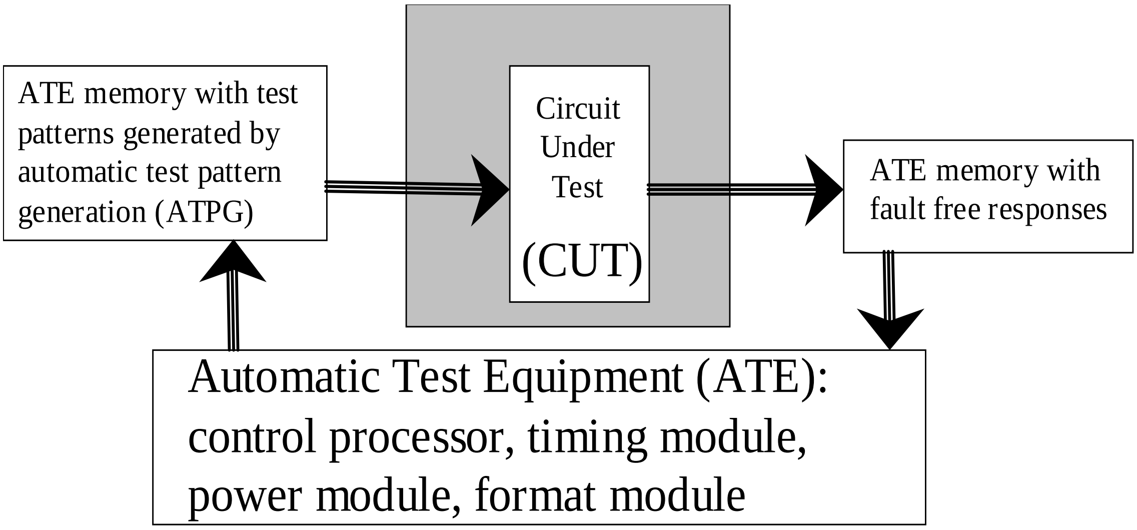

ATE memory with test patterns generated by automatic test pattern generation (ATPG)
Circuit Under Test (CUT)
ATE memory with fault free responses
Automatic Test Equipment (ATE): control processor, timing module, power module, format module

Figure 1.2: Basic principle of external testing using ATE

1.2 External Testing Using Automatic Test Equipment

Given the design complexity of state of the art VLSI circuits, the manufacturing test process relies heavily on automation. Figure 1.2 shows the basic principle of external testing using automatic test equipment with its three basic components: *circuit under test* is the integrated circuit part which is tested for manufacturing defects; *automatic test equipment (ATE)* including control processor, timing module, power module, and format module; *ATE memory* that supplies test patterns and measures test responses. In the following an overview [138] of each of the previously outlined components is presented.

The *circuit under test (CUT)* is the part of silicon wafer or packaged device to which tests are applied to detect manufacturing defects. The connections of the CUT pins and bond pads to ATE must be robust and easily changed since testing will connect and disconnect millions of parts to the ATE to individually test each part.

The *ATE* includes control processor, timing module, power module, and format module. Control processor is a host computer that controls the flow of the test process and communicates to the other ATE modules whether CUT is faulty or fault free. Timing module defines clock edges needed for each pin of the CUT. Format module extends test pattern information with timing and format information that specifies when the signal to a pin will go high or low, and power module provides power supply to CUT and is responsible for accurately measuring currents and voltages.

The *ATE memory* contains test patterns supplied to the CUT and the expected fault free responses which are compared with the actual responses during testing. State of the art ATE measures voltage response with millivolt accuracy at a timing accuracy of hundreds of picoseconds [138]. *Test patterns* or *test vectors* stored in ATE memory are obtained using automatic test pattern generation (ATPG) algorithms [35]. From now onwards throughout this dissertation, *test patterns* and *test vectors* are used interchangeably. ATPG algorithms can broadly be classified into random and deterministic algorithms. Random ATPG algorithms involve generation of random vectors and *test efficiency* (test quality quantified by fault coverage) is determined by fault simulation [2]. Deterministic ATPG algorithms generate tests by processing a structural netlist at the logic level of abstraction using a specified fault list from a fault universe (defined by an explicit fault model such as stuck-at fault model outlined in the previous Section 1.1). Compared to random ATPG algorithms, deterministic ATPG algorithms produce shorter and higher quality tests in terms of test efficiency, at the expense of longer computational time. High computational time associated with deterministic ATPG algorithms is caused by low controllability and observability of the internal nodes of the circuit. This problem is more severe for sequential circuits where despite recent advancements in ATPG [34] computational time is large, and test efficiency is not satisfactory. Further, the growing disparity between the number of transistors on a chip and the limited input/output pins makes the problem of achieving high test efficiency very complicated and time consuming.

Design for testability (DFT) is a methodology that improves the testability, in terms of controllability and observability, by adding test hardware and introducing specific test oriented decisions during the VLSI design flow shown in Figure 1.1. This often results in shorter *test application time*, higher fault coverage and hence test efficiency, and easier ATPG. The most common DFT methodology is scan based DFT where sequential elements are modified to scan cells and introduced into a serial shift register. This is done by having a scan mode for each scan cell where data is not loaded in parallel from the combinational part of the circuit, but it is shifted in serially from the previous scan cell in the shift register. Scan based DFT can further be divided into *full scan* and *partial scan*. The main advantage of full scan is that by modifying all the sequential elements to scan cells it reduces the ATPG problem for sequential circuits to the more computationally tractable ATPG for combinational circuits. On the other hand, partial scan mod-

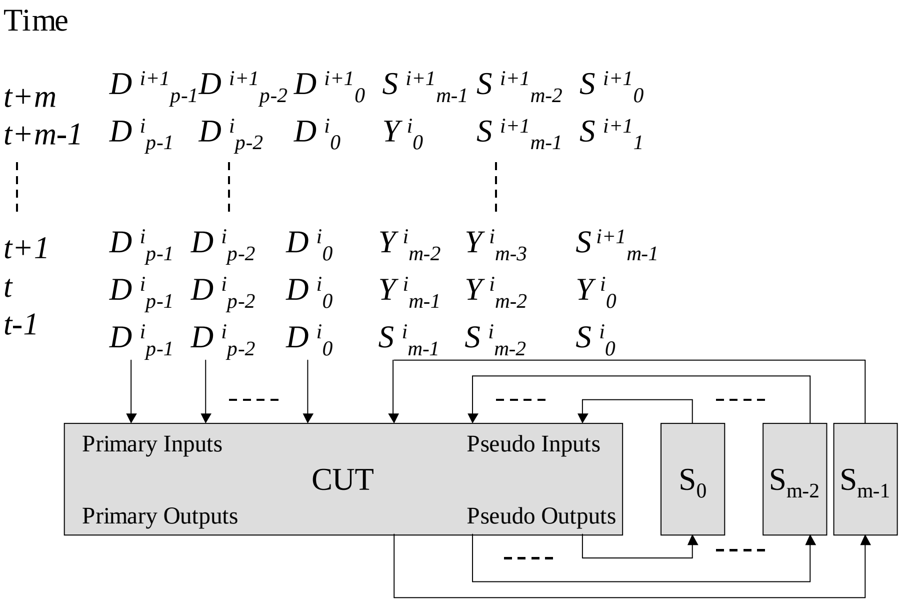

Time
t+m
t+m-1
t+1
t
t-1
D i+1 p-1 D i+1 p-2 D i+1 0 S i+1 m-1 S i+1 m-2 S i+1 0
D i p-1 D i p-2 D i 0 Y i 0 S i+1 m-1 S i+1 1
D i p-1 D i p-2 D i 0 Y i m-2 Y i m-3 S i+1 m-1
D i p-1 D i p-2 D i 0 Y i m-1 Y i m-2 Y i 0
D i p-1 D i p-2 D i 0 S i m-1 S i m-2 S i 0
Primary Inputs
Pseudo Inputs
CUT
Primary Outputs
Pseudo Outputs
S0
Sm-2
Sm-1

Figure 1.3: Scan based design

ifies only a small subset of sequential elements leading to lower test area overhead at the expense of more complex ATPG. The introduction of scan based DFT leads to the modification of the *test application strategy* which describes how test patterns are applied to the CUT. Unlike the case of combinational circuits or non-scan sequential circuits where a test pattern is applied every clock cycle, when scan based DFT is employed each test pattern is applied in a *scan cycle*. Figure 1.3 illustrates the application of a test pattern $V_{i+1} = D^{i+1}_{p-1}D^{i+1}_{p-2}\ldots D^{i+1}_{0}S^{i+1}_{m-1}S^{i+1}_{m-2}\ldots S^{i+1}_{0}$ at time $t+m$ after shifting out the test response of test pattern $V_i = D^i_{p-1}D^i_{p-2}\ldots D^i_0 S^i_{m-1}S^i_{m-2}\ldots S^i_0$ applied at $t-1$, where p is the number of primary inputs, and m is the number of memory elements modified to scan cells $S_0\ldots S_{m-1}$ that are pseudo inputs to the CUT. The scan cycle lasts for $m+1$ clock cycles of which m clock cycles are required to shift out the pseudo output part of the test response $Y^i_{m-1}Y^i_{m-2}\ldots Y^i_0$ for test vector V_i (time t to $t+m-1$) and one clock cycle is required to apply V_{i+1}.

This section has described the basic principles of external testing using ATE and concepts of scan based DFT method. Finally, it should be noted that five main test parameters which assess the quality of a scan DFT method when using external ATE are: *test area* required by extra DFT hardware, *performance*, *test efficiency*, *test application time* and *volume of test data*.

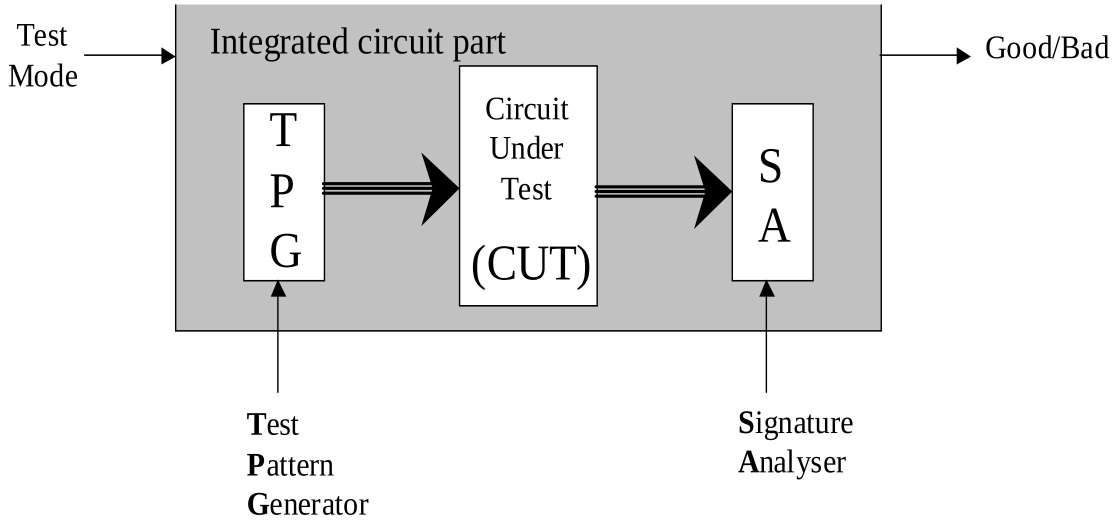


Figure 1.4: Basic principle of internal testing using BIST

1.3 Internal Testing Using Built-In Self-Test

Despite its benefits of detecting manufacturing defects, external testing using ATE has the following two problems. Firstly, ATE is extremely expensive and its cost is expected to grow in the future as the number of chip pins increases [138]. Secondly, when applying generally accepted scan based DFT, test patterns cannot be applied to the circuit under test in a single clock cycle since they need to be shifted through the scan chain in a scan cycle. This makes at-speed testing extremely difficult.

These problems have led to development of *built-in self-test (BIST)* [2, 4, 5, 6, 10] which is a DFT method where parts of the circuit are used to test the circuit itself. Therefore test patterns are not generated *externally* as in the case of ATE (Figure 1.2 of Section 1.2), but they are generated *internally* using BIST circuitry. To a great extent this alleviates the reliance on ATE and testing can be carried out at normal functional speed. In some cases this not only substantially reduces the cost of external ATE, but also enables the detection of timing related faults. The basic principle of BIST is illustrated in Figure 1.4. The heavy reliance on external ATE including ATE memory which stores the test patterns (Figure 1.2), is eliminated by BIST which employs on chip test pattern generator (TPG) and signature analyser (SA). When the circuit is in the test mode, TPG generates patterns that set the CUT lines to values that differentiate the faulty and fault-free circuits, and SA evaluates circuit responses.

The most relevant approach for exhaustive, pseudoexhaustive, and pseudorandom generation of test patterns is the use of a linear feedback shift register (LFSR) as TPG [4]. LFSR is widely employed by BIST methods mainly due to its simple and fairly regular structure, its pseudorandom properties which lead to high fault coverage and test efficiency, and its shift property that leads to easy integration with serial scan (Section 1.2). The typical components of an LFSR are memory elements (latches or flip flops) and exclusive OR (XOR) gates. Despite their simple appearance LFSRs are based on complex mathematical theory [13] that help explain their behaviour as test pattern generators and response analysers. While LFSR can be used to compact and analyse the test responses for single-output CUT, its simple extension to multiple-input signature analyser (MISR) compacts and analyses test sequences for multiple-output CUT. MISR can be extended to built-in logic block observer (BILBO) or to concurrent BILBO (CBILBO) [2] to perform both test pattern generation and signature analysis. An alternative to LFSR for test pattern generation are cellular automata (CA) [175] in which each cell consisting of a memory element is connected only to its neighbouring cells.

Based on the time of application and its relation to functional operation, BIST methods are classified in two broad categories, on-line BIST and off-line BIST [114]. On one hand, in on-line BIST testing occurs during functional operation. Despite its benefits of in-field testing and on-line fault detection for improving fault coverage, on-line BIST leads to excessive power dissipation which increases packaging cost and reduces circuit reliability. Moreover, on-line testing conflicts with power management policies implemented in the state of the art deep sub-micron VLSI circuits to reduce power dissipation. This makes on-line BIST highly inefficient for testing low power VLSI circuits. On the other hand, the same test efficiency is achieved by off-line BIST which deals with testing a circuit when it is not performing its normal functions. From now onwards throughout this dissertation, unless explicitly specified, the term BIST refers to off-line BIST.

Based on the trade-off between *test application time* required to achieve a satisfactory fault coverage and *BIST area overhead* associated with extra test hardware, BIST methods can broadly be classified into *scan BIST* and *parallel BIST*. *BIST embedding* is a particular case of the parallel BIST where functional registers are modified to test registers to generate test patterns and analyse test responses when the circuit is in the test mode. Scan BIST and BIST embedding methodologies are described in the following two subsections.

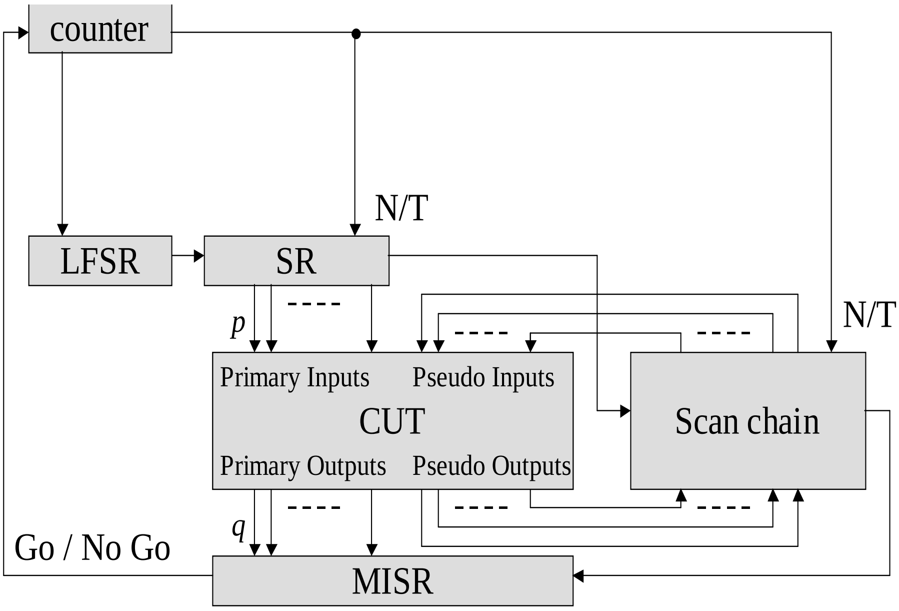


Figure 1.5: Basic principle of scan BIST methodology.

1.3.1 Scan BIST Methodology

Scan BIST methodology is an extension of scan DFT method introduced in Section 1.2, where test patterns are not shifted in the scan chain using external ATE, but they are generated on-chip using TPG. The basic principle of scan BIST is shown in Figure 1.5. In order to provide pseudorandom patterns, an LFSR is used as the TPG, and the serial output of the LFSR is connected to a shift register (SR) connected to the primary inputs of the CUT in order to supply the test pattern. The serial output of SR is connected to the serial input of the internal scan chain, and the serial output of the scan chain and primary outputs of the CUT are analysed using an MISR. A counter is employed to indicate when shifting is complete, so that the pattern stored in the SR and scan chain can be applied to the CUT by activating N/T. The extra hardware required by the counter, SR, LFSR, and MISR leads to a minor impact on the performance, however at the expense of long test application time to achieve satisfactory fault coverage [180, 181]. The long test application time is due to applying each test pattern in a scan cycle which comprises the time required to shift in the patterns into the SR and the internal scan chain. Therefore, the test application strategy for a scan BIST methodology is called *test-per-scan* [5]. This is unlike the *test-per-clock* test application strategy where a test pattern is applied every clock cycle as in the case of the *BIST embedding* methodology explained in the following section.

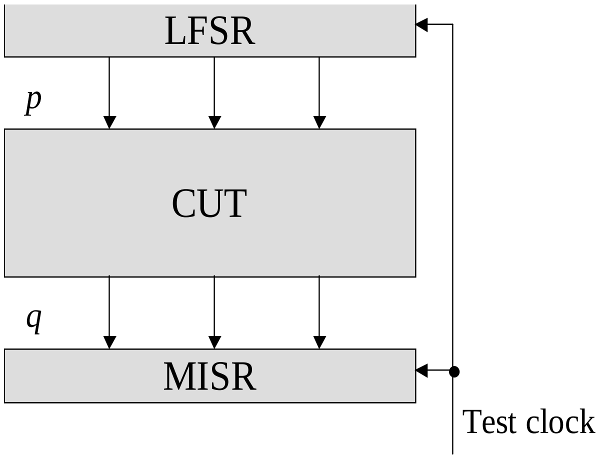


Figure 1.6: Parallel BIST

1.3.2 BIST Embedding Methodology

In a parallel BIST methodology, test patterns are applied to the CUT every clock cycle which leads to a substantial reduction in test application time when compared to the scan BIST methodology (Figure 1.5 from Section 1.3.1). Figure 1.6 shows a circuit under test having p inputs and q outputs which is tested as one entity using an LFSR for test pattern generation and an MISR for signature analysis. Since most practical circuits are too complex to be tested as one entity, a circuit is partitioned into modules [47]. *BIST embedding* is the parallel BIST methodology where each module is a test primitive in the sense that test patterns are generated and output responses are compressed using test registers for *each* module [114, 115, 155]. This methodology is particularly suitable for data path circuits described at register-transfer level of the VLSI design flow where modules are tested using test registers which are a subset of functional registers. The following example overviews the BIST embedding methodology for RTL data paths.

Example 1.1 Consider the data path shown in Figure 1.7 which was described initially in [47]. The data path consists of six modules $M_1 \ldots M_6$, and nine registers $R_1 \ldots R_9$ that are modified into test registers. To make a module testable, each input port is directly or indirectly (through a driving path as multiplexer network or a bus) fed by a TPG and every output port directly or indirectly feeds a SA. For example in the case of M_1, $LFSR_1$ acts as TPG and $MISR_5$ operates as SA. These TPGs and SAs are said to be associated with module M_1. TPGs and SAs are configured as one of the following: LFSRs, MISRs,

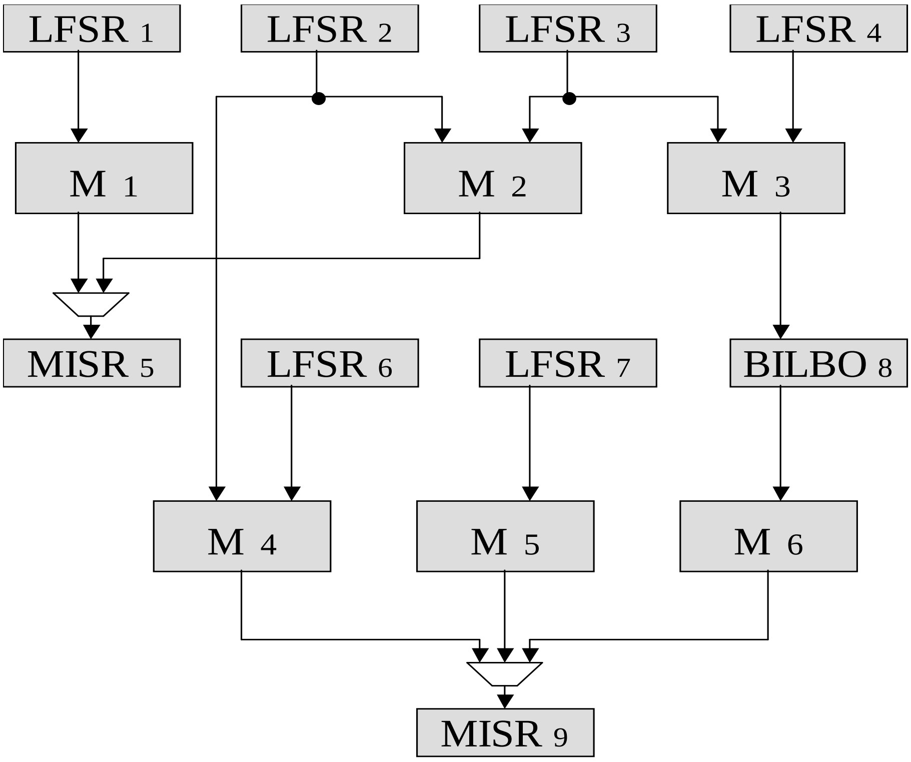
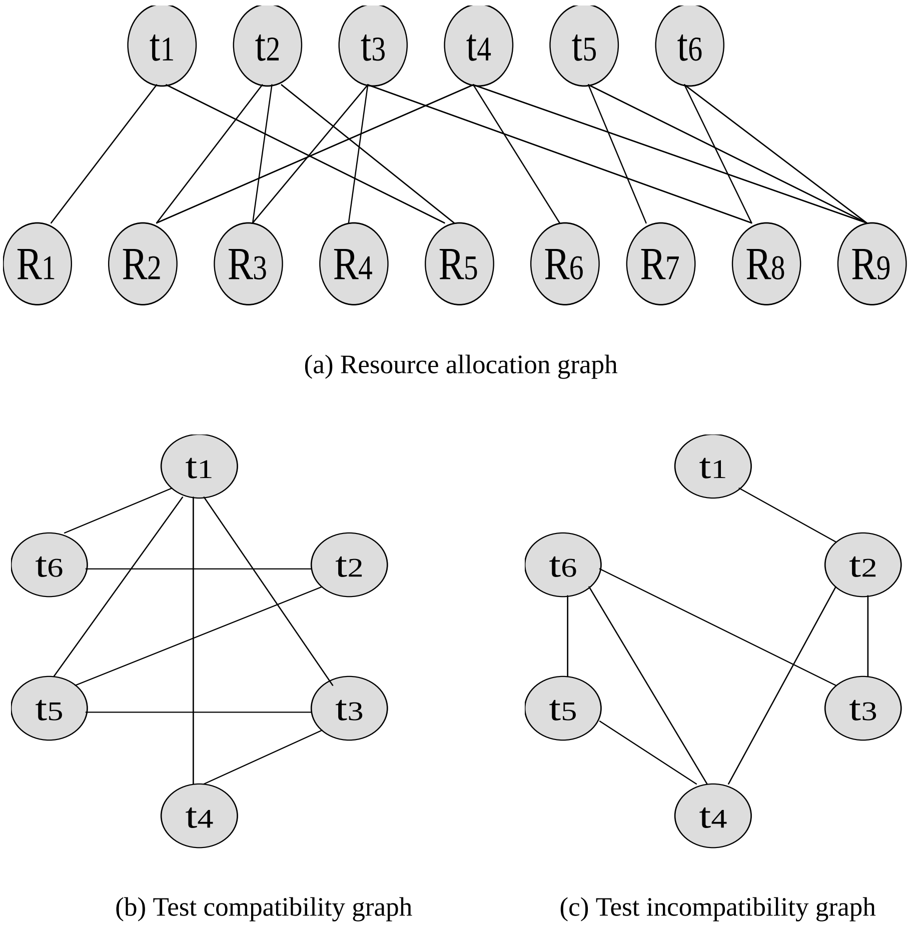


Figure 1.7: Example 1.1 data path

BILBOs, and CBILBOs [2]. During the test of modules M_k, $k = 1 \ldots 6$, the associated TPGs and SAs are first initialised to known states, then a sufficient number of test patterns are generated by the TPGs and applied to M_k. Outputs from M_k are compressed in SAs to form a signature. After all patterns are applied to M_k, the final signature is shifted out of the SAs and compared with the fault-free signature.

Test hardware is allocated such that each module receives test patterns and its output responses are observable during test. The process of allocating test hardware (test resources) to each module is referred to as *test synthesis*. Since test hardware is allocated for BIST, *test synthesis* and *BIST synthesis* are used interchangeably throughout this dissertation. Due to the test hardware required by TPGs and SAs, a BIST data path has a greater area than the original circuit. This extra area is referred to as *BIST area overhead*. Also, test hardware often increases circuit delays that may lead to *performance degradation*. Depending on test hardware allocation generated by *test synthesis*, some modules from the data path may be tested at the same time while others cannot. This is due to the conflicts which may arise between different modules that need to use the same test resources during testing. A *test schedule* specifies the order of testing all the modules by eliminating all the conflicts between modules. A test schedule is divided into several *test sessions*, where in each test session one or more modules are tested. Data paths with many modules in conflict have a higher number of test sessions and hence longer test application time. The *test application time* of a built-in self-testable data path is the time to complete the test schedule added to the shifting time required to shift in the seeds for test pattern generators and shift out signatures stored in signature analysers as described in Example 1.1. In the following the concepts defined in [47] are introduced based on the example data path shown in Figure 1.7. These concepts are necessary to understand how a test schedule is generated and serve as a basis for the technique proposed in Chapters 5 and 6.

A test t_k for a module M_k has an *allocation relation* with a test register R_i if the register generates test patterns for M_k or analyses test responses of M_k. In general, the allocation between modules and test registers can be represented by a bipartite graph with a node set consisting of tests and resources. The *resource allocation graph* for the data path example from Figure 1.7 is shown in Figure 1.8(a). If there is an allocation relation between t_k and R_i, then there is an edge between t_k and R_i in the resource allocation graph. For example in the case of M_2 from Figure 1.7 $LFSR_2$ and $LFSR_3$ generate test patterns, and $MISR_5$ analyses test responses. Therefore, in the resource allocation graph shown in Figure 1.8(a) there is an edge between t_2 and R_2, between t_2 and R_3, and between t_2 and R_5. If a resource node (register) is connected to more that one test this indicates a conflict between the tests that require that resource. A pair of tests that share a test resource cannot be run concurrently and are referred to as *incompatible*. Otherwise, they are *compatible*. Pairs of compatible tests form a relation on the set of tests which is a compatibility relation. Such a relation can be represented by a *test compatibility graph (TCG)* shown in Figure 1.8(b). In a TCG a node appears for each test and an edge exists between two nodes if the corresponding two tests are compatible. For example, in the case of TCG from Figure 1.8(b) there is an edge between t_1 and t_3 since the two tests do not share any resources in the resource allocation graph shown in Figure 1.8(a). The test compatibility graph indicates which tests can be run concurrently. The complement of the test compatibility graph is the *test incompatibility graph (TIG)* shown in Figure 1.8(c). Unlike the TCG where there is an edge between two compatible tests, an edge appears in the TIG if the corresponding two tests are incompatible, i.e. they share the



(a) Resource allocation graph

(b) Test compatibility graph

(c) Test incompatibility graph

Figure 1.8: Resource allocation, test compatibility and incompatibility graphs for data path shown in Figure 1.7.

same resources in the resource allocation graph shown in Figure 1.8(a). For example, since $LFSR_2$ (or simply R_2) generates test patterns for both M_2 and M_4 there is an edge between between R_2 and t_2 and another edge between R_2 and t_4 in the resource allocation graph. This will lead to a conflict between M_2 and M_4 and an edge between t_2 and t_4 will be introduced in the TIG shown in Figure 1.8(c). The TCG and the TIG can be used as a basis for scheduling the tests such that the number of test sessions and hence the total test application are minimised. A clique (a complete subgraph of a graph [51]) of the TCG represents a set of tests which can run concurrently. For example, in the case of the TCG shown in Figure 1.8(b), $\{t_1, t_3, t_4\}$ is a clique which means that modules M_1, M_3, and M_4 can be tested concurrently. Thus, the test scheduling problem reduces to finding all the cliques in the TCG and covering all the nodes in the TCG with the minimum number of cliques. This problem can also be thought of as finding the minimum number of colours required to colour the TIG. This is because the graph colouring problem aims to minimise the number of colours in a graph such that two adjacent nodes do not have the same colour. Since all the nodes with the same colour in the TIG belong to the same clique in the TCG, minimum number of colours in the TIG will indicate the minimum number of test sessions which leads to the lowest test application time. The test scheduling problem to minimise the test application time was shown to be NP-hard [47, 58] and therefore fast heuristics must be developed. Also, efficient algorithms for BIST synthesis for RTL data paths are required since it was formulated as 0-1 integer linear programming problem [155], which is, in general, NP-hard [58].

It should be noted that test scheduling differs fundamentally from traditional operation scheduling in high level synthesis (HLS) [51, 59]. Unlike operation scheduling which is based on a data dependency graph, test scheduling is based on the TCG and the TIG shown in Figures 1.8(b) and 1.8(c). Therefore, in test scheduling there is no concern with regard to the precedence and the order of execution. The main objective in test scheduling is to minimise the number of test sessions and hence test application time by increasing the test concurrency based on the conflict information derived from the resource allocation graph (Figure 1.8(a)).

It is desirable to allocate test hardware (*test synthesis*) such that both test application time and BIST area overhead are reduced. For each testable data path there are one or more test schedules according to the resource allocation and test incompatibility graphs (*test scheduling*). Test synthesis and test scheduling are strictly interrelated since each test resource allocation determines the number of conflicts between different tests. Example 1.2 outlines this interrelation between test synthesis and test scheduling.

Example 1.2 Figure 1.9 shows a data path with two registers, R_1 and R_2, and two modules, M_1 and M_2. If only register R_2 is modified to a $LFSR_2$ for generating test patterns for both M_2 and M_1 (through a multiplexer), then due to the test resource conflict it is necessary to schedule tests t_1 and t_2 at different test times. This leads to an increase in test application time. However, if both registers R_1 and R_2 are modified to $LFSR_1$ and $LFSR_2$ then no test resource conflict occurs and t_1 and t_2 may be scheduled at the same time. The use of two test registers leads to lower test application time at the expense of higher BIST area overhead.

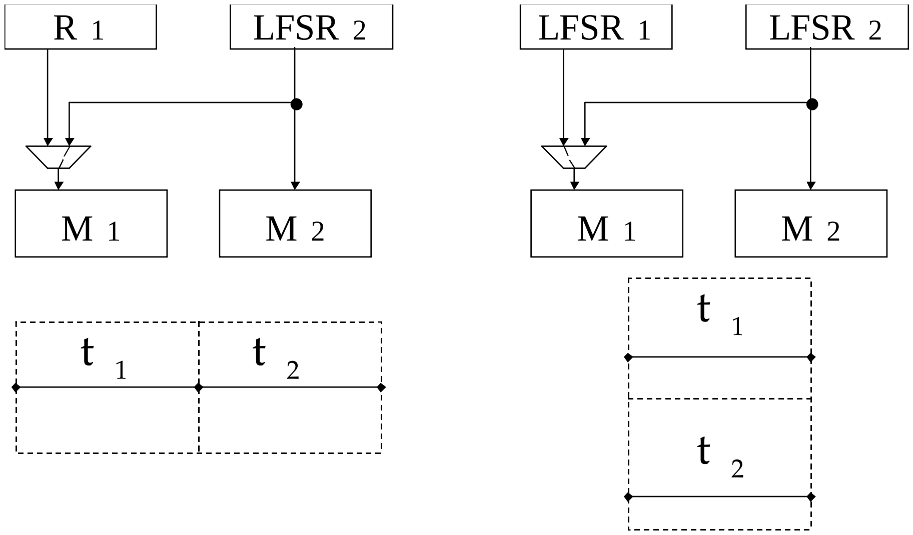
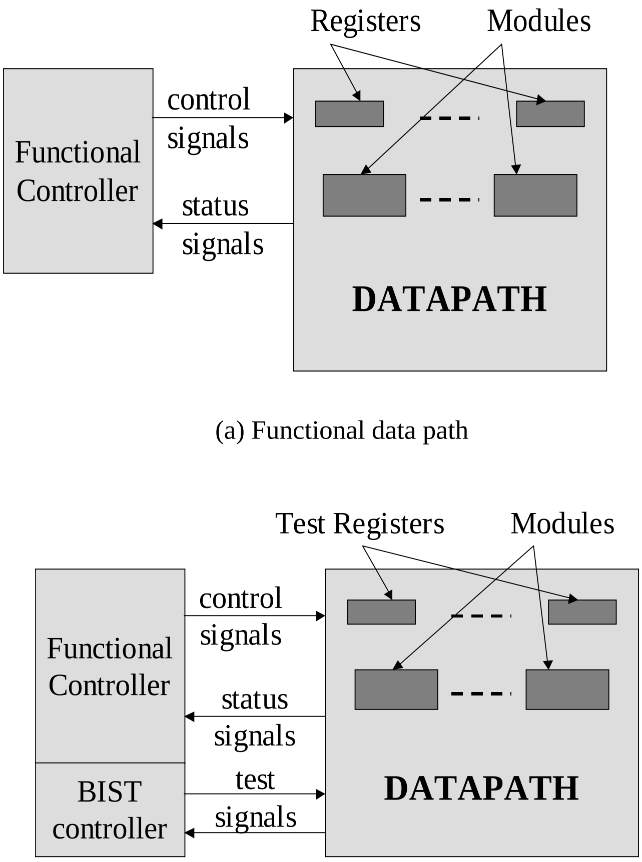


Figure 1.9: Interrelation between test synthesis and test scheduling.

The set of feasible test resource allocations and test schedules define a *testable design space*. For complex circuits, with a large number of registers and modules, the size of the testable design space is huge due to the high number of test resource allocations and test schedules. Exploring different alternatives in the design space in order to minimise one or more test parameters, such as test application time or BIST area overhead, is referred to as *testable design space exploration*. To achieve high quality solutions with both low test application time and low BIST area overhead, *efficient testable design space exploration* is required. Further, efficient testable design space exploration is also important from the computational time standpoint, since for complex circuits the size of the testable design space is huge.

After test resources are allocated (*test synthesis*) and the test schedule is generated (*test scheduling*) the final step is to synthesise a BIST controller that controls the execution of test sessions and shifts in the seeds for TPGs and shifts out the signatures stored in SAs. In order to achieve minimum area overhead, the BIST controller is merged with the functional controller into a single control unit for the data path. Figure 1.10 shows the extention of a functional data path (Figure 1.10(a)) to a self-testable data path (Figure 1.10(b)) with merged functional and BIST controllers. A particular advantage of specifying a circuit at RTL is that control and status signals during the functional specification are merged and optimised with the test signals that operate the data path during testing.



(a) Functional data path

(b) Testable data path

Figure 1.10: Functional and testable data paths.

In addition to test application time, BIST area overhead, and performance degradation, other BIST parameters include *volume of test data* and *fault-escape probability*. Volume of test data affects storage requirements and shifting time required to shift in the seeds for the TPGs and to shift out the signatures stored in SAs. Hence, volume of test data has an influence on test application time which is the sum of the shifting time and the time required to complete the test sessions. High aliasing probability in signature analysis registers [139, 173] leads to data paths with high fault-escape probability which lowers fault coverage and hence decreases test efficiency. Finally, it should be noted that the six main test parameters which assess the quality of BIST embedding methodology are: *test application time*, *BIST area overhead*, *performance degradation*, *volume of test data*, *fault-escape probability*, and *efficiency of testable design space exploration*.

1.4 Power Dissipation During Test Application

The ever increasing demand for portable computing devices and wireless communication systems requires low power VLSI circuits. Minimising power dissipation during the VLSI design flow increases lifetime and reliability of the circuit [29, 158, 169]. Numerous techniques for low power VLSI circuit design were reported [158] for CMOS technology where the dominant factor of power dissipation is dynamic power dissipation caused by switching activity [29, 169]. While these techniques have successfully reduced the circuit power dissipation during functional operation, testing of such low power circuits has recently become an area of concern mainly because of the following two reasons (detailed in Section 2.1). Firstly, it was reported in [199] that there is significantly higher switching activity during testing than during functional operation and hence higher power dissipation. This can decrease the reliability of the circuit under test due to excessive temperature and current density which cannot be tolerated by circuits designed using power minimisation techniques. Secondly, high switching activity during test application leads to manufacturing yield loss which can be explained as follows. High switching activity during test application causes a high rate of current flowing in power and ground lines leading to excessive power and ground noise. This noise can erroneously change the logic state of circuit lines leading to incorrect operation of circuit gates causing some good dies to fail the test [191]. Therefore, addressing the problems associated with testing low power VLSI circuits has become an important issue.

Most of the solutions reported for power minimisation during normal operation reduce *spurious transitions during functional operation (glitches)* which do not carry any useful functional information and cause useless power dissipation. Consequently, power can be minimised during test application by eliminating *spurious transitions during test application* which do not carry any *useful test operation*. Spurious transitions during test application will be defined later at different levels of abstraction in Chapters 3 and 6, and their minimisation is the aim of this investigation. Since dynamic power dissipation caused by switching activity is the dominant factor of power dissipation in CMOS VLSI circuits [29, 158, 169], from now onwards, unless explicitly specified, the terms *dynamic power dissipation* and *power dissipation* are used interchangeably throughout this dissertation.

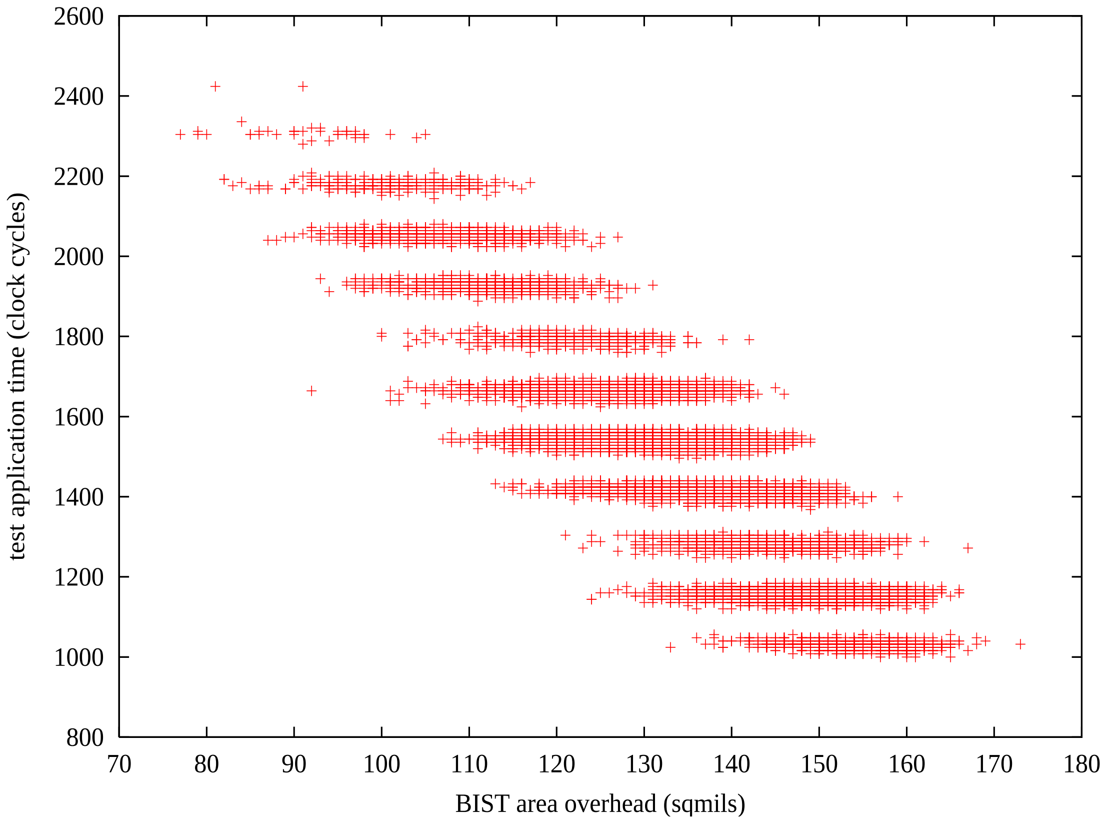


Figure 1.11: Test application time vs. BIST area overhead

1.4.1 Three Dimensional Testable Design Space

The testable design space exploration, as described in Section 1.3.2, involves a trade-off between test application time and BIST area overhead, as shown in Figure 1.11 for 32 point discrete cosine transform data path with 60 registers, 9 multipliers, 12 adders, and an execution time constraint of 30 control steps. The results were obtained by synthesising and technology mapping [51] into 0.35 micron AMS technology [9] 35,000 BIST data paths which is a large statistical sample of the entire design space of ≈10^{23} BIST data paths. The BIST data paths were specified in VHDL [136], and test application time (in terms of clock cycles) and BIST area overhead (in terms of square mils) were obtained using the experimental validation flow detailed in appendix A. BIST area overhead in terms of square mils reflects not only the additional test hardware required by test registers, but also the additional gates required to integrate the functional and test controller as outlined in Figure 1.10 from Section 1.3.2. Figure 1.11 shows that as test application

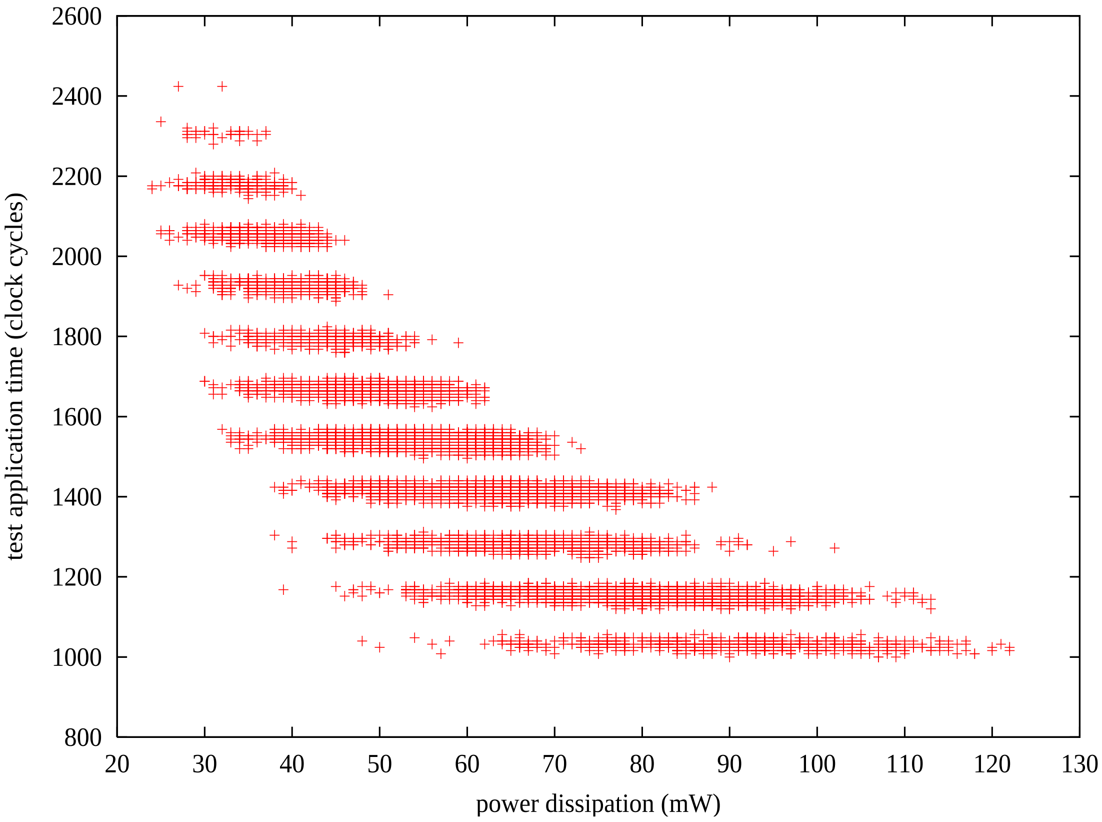


Figure 1.12: Test application time vs. power dissipation

time decreases there is an increase in BIST area overhead. However, there are many test resource allocations leading to identical values in test application time with significantly different values in BIST area overhead. For example, in the case of the lowest test application time equal to 1064 clock cycles, BIST area overhead varies from approximately 130 square mils to 180 square mils. This justifies the need for efficient BIST hardware synthesis algorithms which minimise both test application time and BIST area overhead. This problem is addressed in Chapter 5.

The main disadvantage of trading off *only* test application time and BIST area overhead is that testable data paths are selected without providing the flexibility of exploring alternative solutions in terms of power dissipation. Indeed, a large number of optimum or near-optimum solutions in terms of test application time and BIST area overhead may be found, but with *different* power dissipation. Thus, power dissipation is a new parameter which should be considered during testable design space exploration. Figure 1.12 shows the trade-off between test application time and power dissipation for the 32 point discrete

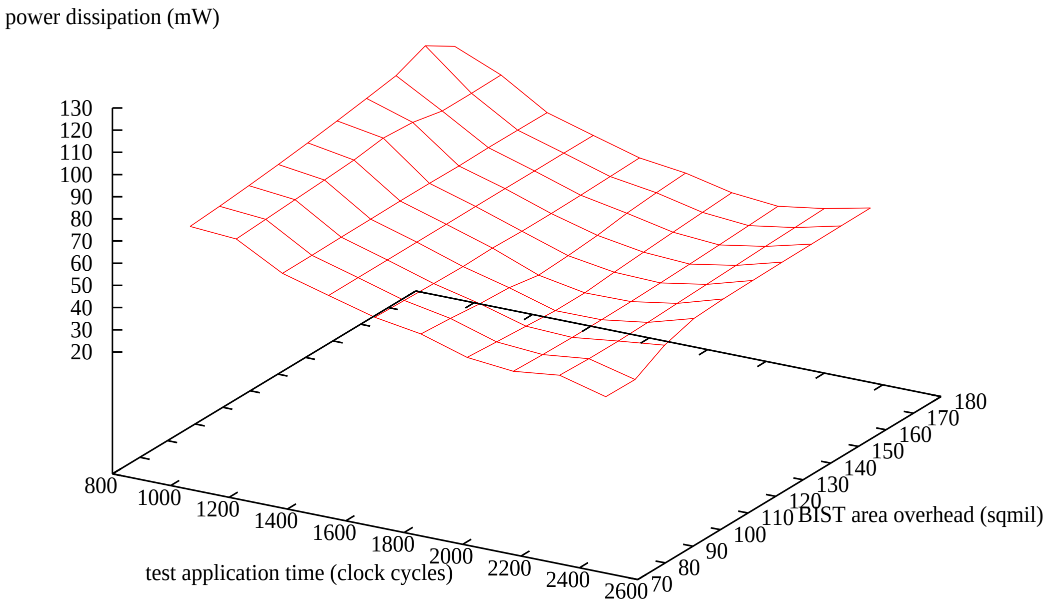


Figure 1.13: Three dimensional testable design space

cosine transform data path. In the case of the lowest test application time equal to 1064 clock cycles, power dissipation varies from approximately 40mW to 130mW. The different values in power dissipation during test application are not caused only by different values in BIST area overhead (Figure 1.11). Since power dissipation is dependent on switching activity of the active elements during each test session, the variation in power dissipation is also due to *useless power dissipation* defined in Chapter 6.

Finally, Figure 1.13 shows the three dimensional testable design space for the 32 point discrete cosine transform data path. Unlike the case of exploring *only* test application time and BIST area overhead (Figure 1.11) or *only* test application time and power dissipation (Figure 1.12), the exploration of the three dimensional design space accounts for *all* the three parameters: test application time, BIST area overhead and power dissipation (Figure 1.13). The aim of the techniques proposed in Chapter 6 is to efficiently explore the three dimensional design space and eliminate useless power dissipation without any effect on test application time or BIST area overhead.

1.5 Dissertation Organisation and Contributions

This dissertation presents new techniques in terms of algorithms and methodologies for testing low power VLSI circuits at the logic and register-transfer levels of abstraction of the VLSI design flow. The rest of the dissertation is organised as follows. Motivation for low power testing and a comprehensive review of previously reported approaches for minimising power dissipation during test application is provided in Chapter 2.

Chapter 3 introduces a new technique [142, 145] for power minimisation during test application in scan sequential circuits with no penalty in area overhead, test application time, test efficiency, performance, or volume of test data when compared to standard scan method described in Section 1.2. The technique is test set dependent and it is applicable to small to medium sized scan sequential circuits at the logic level of abstraction.

Chapter 4 introduces a new test set independent technique [143] applicable to large scan sequential circuits and shows how with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, considerable savings in power dissipation during test application in large scan sequential circuits is achieved with low computational time.

Prior to investigating power minimisation techniques for testing low power VLSI circuits at RTL, Chapter 5 addresses testability of RTL data paths using BIST [139, 140, 144]. A new BIST methodology based on test compatibility classes achieves an improvement in terms of test application time, BIST area overhead, performance degradation, volume of test data, and fault-escape probability over the traditional BIST embedding methodology described in Section 1.3.2.

Chapter 6 shows how power dissipation during test application is minimised at the register-transfer level of abstraction of the VLSI design flow [141]. The three dimensional testable design space described in Figure 1.13 from Section 1.4 is explored using novel power conscious test synthesis and test scheduling algorithms at the expense of low overhead in computational time.

Finally, conclusions and directions for future research are given in Chapter 7. The previously outlined contributions in Chapters 3, 4, 5, and 6, and summarised in the final Chapter 7 have resulted in original work published or submitted for publication [139, 140, 141, 142, 143, 144, 145].

Chapter 2

Motivation and Previous Work

Personal mobile communications and portable computing systems are the fastest growing sectors of the consumer electronics market. The electronic devices at the heart of such products need to dissipate low power, in order to conserve battery life and meet packaging reliability constraints. Low power design in terms of algorithms, architectures, and circuits has received significant attention and research input over the last decade [8, 116, 131, 158, 161, 179]. Although low power design methodologies will solve the problem of designing complex, low power digital very large scale integrated (VLSI) circuits, such circuits will still be subject to manufacturing defects. It was implicitly assumed that traditional design for testability (DFT) methodologies are suitable for digital circuits designed using low power methods. However, recent research has shown that this assumption is not valid and leads to lower circuit reliability and reduced manufacturing yield [66, 188, 191]. For example, it was reported in [199] that a VLSI chip can dissipate up to three times higher power during testing when compared to normal (functional) operation. While some overstressing of devices during a burn-in phase may be desirable, increasing the power dissipation by several times can be destructive. The additional power dissipation is caused by significantly higher switching activity during testing than in functional operation. This is due to the fact that there is a fundamental conflict between the very aims of low power design where the correlation between input patterns is increased and traditional DFT methodologies where any correlation between test vectors is destroyed in order to reduce test application time.

The aim of this chapter is to justify the need for low power testing in order to preserve high circuit yield and reliability (Section 2.1) and to provide an overview of the solutions recently proposed to reduce power dissipation during test application (Section 2.2).

2.1 Motivation for Low Power Testing

Considerable research on low power design and testability of VLSI circuits was done over the last decade [94, 116]. With the advent of deep sub-micron technology and tight yield and reliability constraints, in order to perform a non-destructive test for high performance VLSI circuits power dissipation during test application needs not to exceed the power constraint set by the power dissipated during functional operation of the circuit [27, 48, 66, 138, 174, 199]. This is because excessive power dissipation during test application caused by high switching activity may lead to the following two problems [66, 188, 191]:

i. ***Reliability decrease*** caused by the following two reasons: heat dissipation and electromigration. The use of special cooling equipment to remove excessive heat dissipation during test application caused by higher switching activity, is difficult and costly as tests are applied at higher levels of circuit integration such as BIST (Section 1.3). Therefore, excessive heat dissipation may lead to permanent damage of the circuit under test or affect the reliability by accelerating corrosion mechanisms [188]. Also electromigration rate increases with temperature and current density that is underestimated by state of the art approaches [91, 187] which assume signal correlations that are eliminated when DFT methodologies such as scan (Section 1.2) or scan BIST (Section 1.3.1) are employed.

ii. ***Manufacturing yield loss*** caused by high power/ground noise combined with large resistive voltage drop. On one hand, to test a bare dice during manufacturing test using automatic test equipment (Section 1.2), power must be supplied through probes which have higher inductance than the power and ground pins of the circuit package leading to significantly higher power/ground noise. On the other hand, resistive voltage drop caused by large maximum instantaneous current flowing in the power lines is underestimated by state of the art approaches [91, 187] since they assume signal correlations that are destroyed when employing scan based DFT methodologies. Therefore, high power/ground noise combined with large resistive voltage drop can erroneously change the logic state of circuit lines causing some good circuits to fail the test, leading to *unnecessary* loss of manufacturing yield.

2.2 Previous Work

The previous section has motivated the need to minimise power dissipation during test application in order to increase circuit yield and reliability. A review of the sources of higher power dissipation during test application and recently proposed low power testing techniques is given in Sections 2.2.1 and 2.2.2 respectively.

2.2.1 Sources of Higher Power Dissipation During Test Application

This section reviews *low power design* techniques and methodologies which lead to the conflict between low power dissipation during functional operation and achieving high testability of the circuit under test. Dynamic power dissipation in CMOS VLSI circuits depends on three parameters: supply voltage, clock frequency, and switching activity [29, 158, 169]. While the first two parameters reduce power dissipation at the expense of circuit performance, power reduction by minimising switching activity and hence switched capacitance does not introduce performance degradation and it is the main technique researched over the last decade [158]. Depending on the level of abstraction, sources of high power dissipation during test application due to increased switching activity can broadly be classified into *logic level* sources and *register-transfer level* sources:

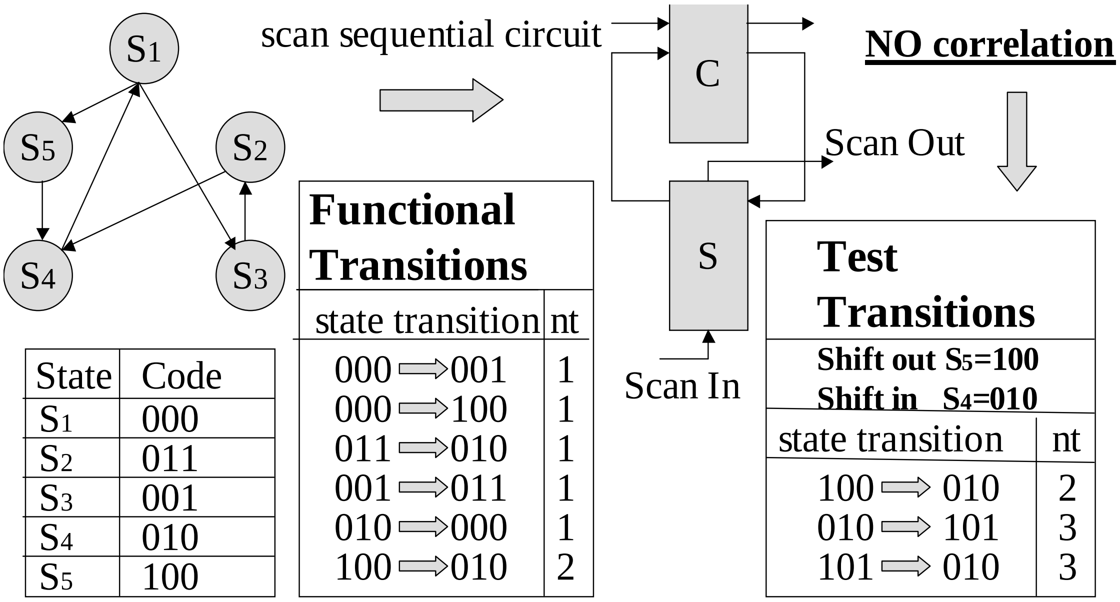
i. Sources of high power dissipation during test application caused by design techniques at the logic level of abstraction can further be classified:

   (a) Low power combinational circuits are synthesised by algorithms [11, 19, 90, 102, 168, 178, 183, 195] which seek to optimise the signal or transition probabilities of circuit nodes using the spatial dependencies inside the circuit (spatial correlation), and assuming the transition probabilities of primary inputs to be given (temporal correlation) [124]. The exploitation of spatial and temporal correlations during functional operation for low power synthesis of combinational circuits leads to high switching activity during test application since correlation between consecutive test patterns generated by automatic test pattern generation (ATPG) algorithms is very low [171]. This is because a test pattern is generated for a given target fault without any consideration of the previous test pattern in the test sequence. Therefore, lower correlation between consecutive test patterns during test application may lead to higher switching activity and hence power dissipation when compared to functional operation [191].

(b) Low power sequential circuits are synthesised by state assignment algorithms which use state transition probabilities [17, 18, 20, 42, 123, 168, 184]. The state transition probabilities are computed assuming the input probability distribution and the state transition graph which are valid during functional operation. These two assumptions are not valid during the test mode of operation when scan DFT technique is employed. While shifting out test responses, the scan cells are assigned uncorrelated values that destroy the correlation between successive functional states. Furthermore, in the case of data path circuits with large number of states that are synthesised for low power using the correlations between data transfers [30, 96, 103, 104, 105, 106, 107], in the test mode scan registers are assigned uncorrelated values that are never reached during functional operation, which may lead to higher power dissipation than during the functional operation.

ii. High power dissipation during test application caused by design techniques at the register-transfer level of abstraction is due to the following. Systems which comprise a high number of memory elements and multifunctional execution units employ power conscious architectural decisions such as power management where blocks are not simultaneously activated during functional operation [16, 107]. Hence, inactive blocks do not contribute to dissipation during the functional operation. The fundamental premise for power management is that systems and their components experience nonuniform workload during the functional operation [15]. However, such an assumption is not valid during test application. In order to minimise test application time when the system is in the test mode, concurrent execution of tests is required. Therefore, by concurrently executing tests many blocks will be active at the same time leading to a conflict with the power management policy. This will result in higher power dissipation during test application when compared to functional operation.

The following two examples illustrate the sources of higher switching activity during test application than during normal operation at two different levels of abstraction of the VLSI design flow: *logic level* (Example 2.1) and *register-transfer level* (Example 2.2).



State	Code
S1	000
S2	011
S3	001
S4	010
S5	100

Functional Transitions

state transition	nt
000⇒001	1
000⇒100	1
011⇒010	1
001⇒011	1
010⇒000	1
100⇒010	2

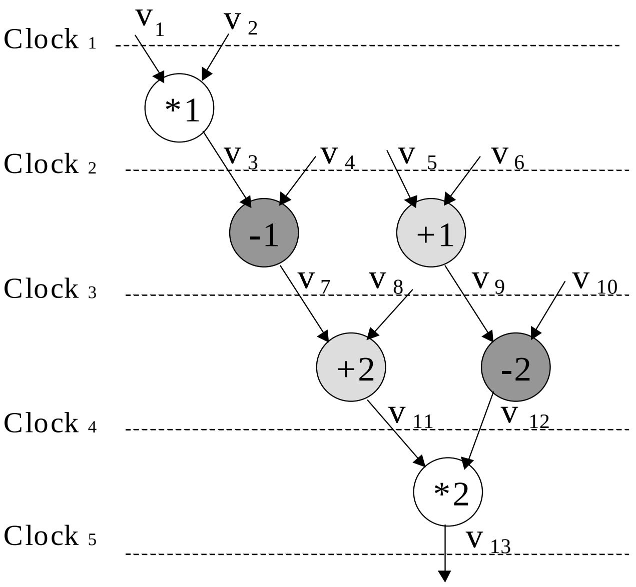
Test Transitions

Shift out S5=100
Shift in S4=010

state transition	nt
100⇒ 010	2
010⇒ 101	3
101⇒ 010	3

Figure 2.1: Sources of higher power dissipation during test application than during normal operation at logic level of abstraction when scan DFT is employed.

Example 2.1 Consider the state transition graph and its circuit implementation shown in Figure 2.1. The functional description of the state transition graph comprises five states S_1, S_2, S_3, S_4, S_5, and circuit implementation consists of the combinational part **C** and the sequential part **S**. In order to achieve high test efficiency scan based DFT is employed and sequential elements are transformed into scan cells with serial input, Scan In, and serial output, Scan Out. To reduce power dissipation during functional operation, state assignment algorithms for low power, outlined in problem (ib), allocate a code to each state such that the number of transitions (**nt**) is minimised. However, when scan based DFT is employed, state transition correlations that exist during functional operation are destroyed. This leads to greater number of transitions during testing, and *hence higher power dissipation*, as in the case when shifting out the test response with sequential part S_5 and shifting in the next test pattern with sequential part S_4. For example, during testing, the following state transitions 010→101 and 101→010 lead to **nt**=3 which is higher than any number of transitions during functional operation.



Register	Variables
R_1	v_1, v_{11}
R_2	v_2, v_{12}
R_3	v_7
R_4	v_5
R_5	v_6, v_8
R_6	v_3
R_7	v_9
R_8	v_4, v_{10}, v_{13}

Figure 2.2: Data flow graph to illustrate the implementation of low power RTL data path.

Example 2.2 In order to show that the high test concurrency required for low test application time aimed for by the BIST embedding methodology (Section 1.3.2) leads to higher power dissipation during test application, consider the data flow graph from Figure 2.2 and its low power implementation shown in Figure 2.3. The 13 variables in the data flow graph $\{v_1 \ldots v_{13}\}$ are mapped to 8 registers $\{R_1 \ldots R_8\}$, and the 6 operations $\{(*_1)(*_2)(+_1)(+_2)(-_1)(-_2)\}$ are mapped to 3 functional units (modules) $\{(*)(+)(-)\}$ (Figures 2.2 and 2.3). According to the variable assignment shown in Figure 2.2, the multiplier $(*)$ is active only in clock cycles 1 and 4, and the adder $(+)$ and the substractor $(-)$ are active only in clock cycles 2 and 3. Similarly registers $\{R_1,R_2\}$ are active only in clock cycles 1 and 4, $\{R_4,R_6\}$ only in clock cycle 2, $\{R_3,R_7\}$ only in clock cycle 3, R_5 only in clock cycles 2 and 3, and R_8 only in clock cycles 2,3, and 5. This implies that *not all* the data path elements are active at the same time which leads to low power dissipation during functional operation as shown in Figure 2.3. However, if tests for $\{(*)(+)\}$ are executed at the same time during test application by employing BIST embedding methodology, and modifying registers $\{R_1,R_2,R_3,R_5\}$ to LFSRs, and registers $\{R_7,R_8\}$ to MISRs, then modules $\{(*)(+)\}$ and registers $\{R_1,R_2,R_3,R_5,R_7,R_8\}$ are active at the same time. Higher switching activity caused by high test concurrency leads to higher power dissipation during test application than during the functional operation.

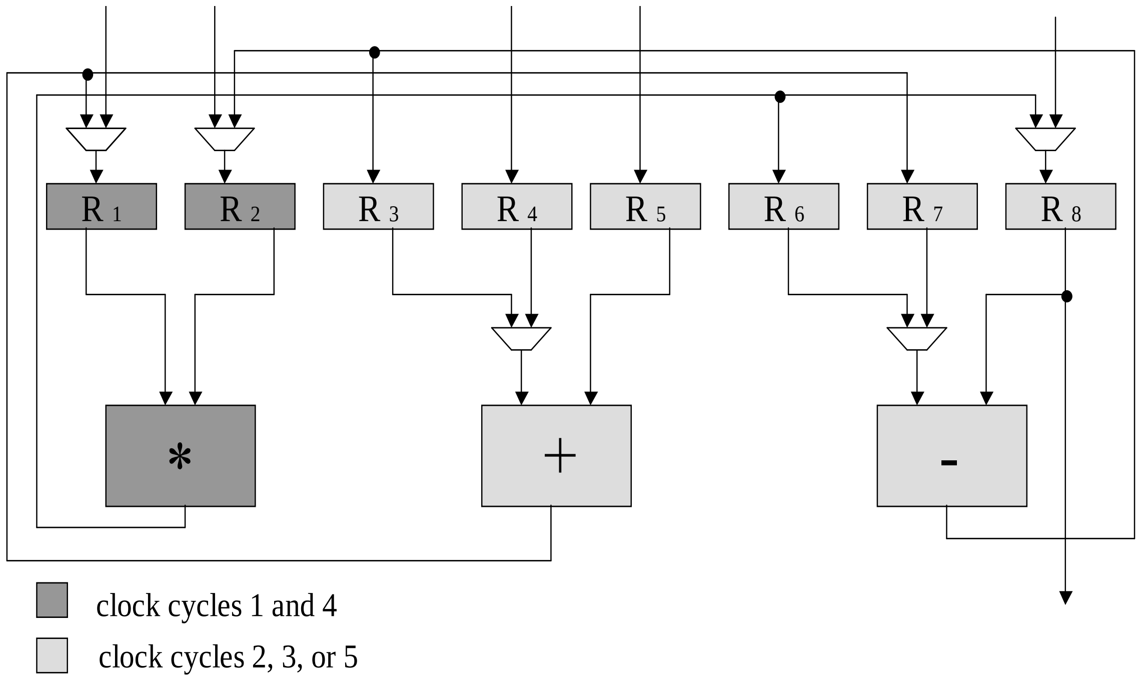
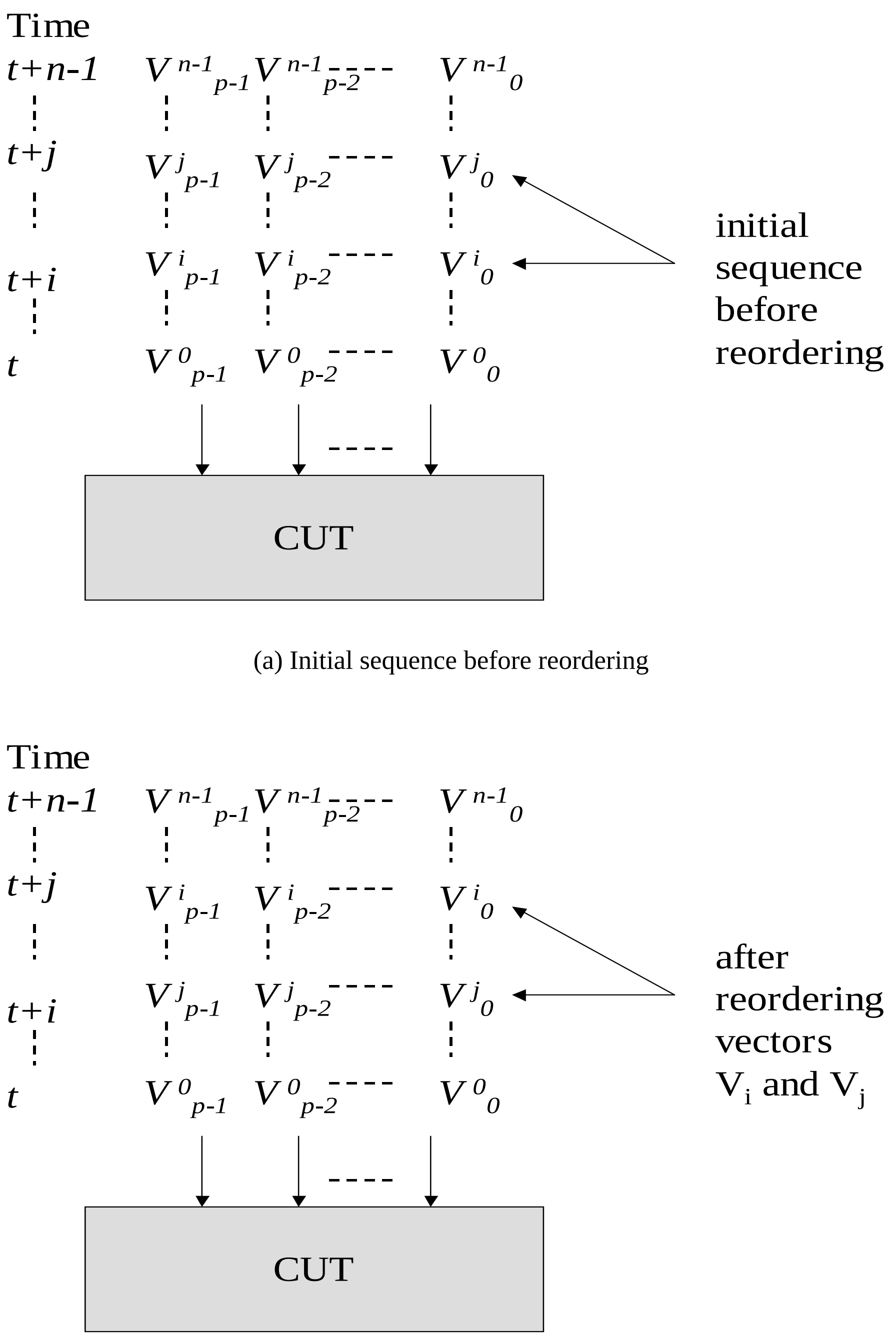


Figure 2.3: Low power RTL data path implementing data flow graph shown in Figure 2.2.

2.2.2 Other Work on Low Power Testing

This section gives a review of recently proposed solutions for reducing switching activity and hence power dissipation during test application which leads to solving **problems (i)-(ii)** outlined in the previous Section 2.2.1.

Problem (ia): A new ATPG tool [191] was proposed to overcome the low correlation between consecutive test vectors during test application in combinational circuits. Despite achieving the objectives of safe and inexpensive testing of low power circuits the approach in [191] increased the test application time. A different approach for minimising power dissipation during test application in combinational circuits is based on test vector ordering [49, 56, 69, 72, 73]. The basic idea beyond test vector ordering is to find a new order of the set such that correlation between consecutive test patterns is increased as shown in Figure 2.4. For example by considering a p input combinational circuit with a test set of n test vectors (Figure 2.4(a)), by swapping the position of test vectors V_i and V_j will lead to a lower power dissipation (Figure 2.4(b)). Test vector ordering is done in a post-ATPG phase with no overhead in test application time since test vectors are reordered such that correlation between consecutive test vectors matches the assumed transition probabilities


Time
$t+n-1$
$t+j$
$t+i$
t
V^{n-1}_{p-1} V^{n-1}_{p-2} ---- V^{n-1}_{0}
V^{j}_{p-1} V^{j}_{p-2} ---- V^{j}_{0}
V^{i}_{p-1} V^{i}_{p-2} ---- V^{i}_{0}
V^{0}_{p-1} V^{0}_{p-2} ---- V^{0}_{0}
initial sequence before reordering
CUT
Time
$t+n-1$
$t+j$
$t+i$
t
V^{n-1}_{p-1} V^{n-1}_{p-2} ---- V^{n-1}_{0}
V^{i}_{p-1} V^{i}_{p-2} ---- V^{i}_{0}
V^{j}_{p-1} V^{j}_{p-2} ---- V^{j}_{0}
V^{0}_{p-1} V^{0}_{p-2} ---- V^{0}_{0}
after reordering vectors V_i and V_j
CUT

(a) Initial sequence before reordering

(b) After reordering

Figure 2.4: Reducing power dissipation during test application in combinational circuit by reordering test vectors [49, 56, 69, 72, 73].

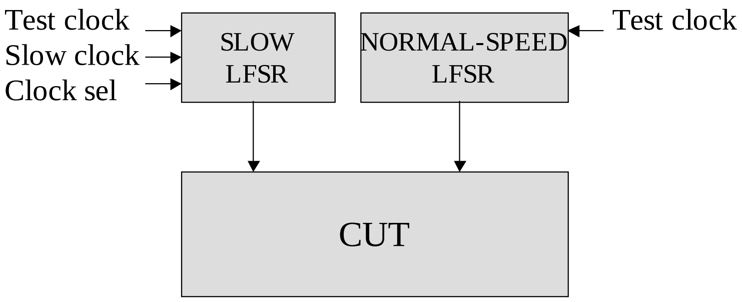
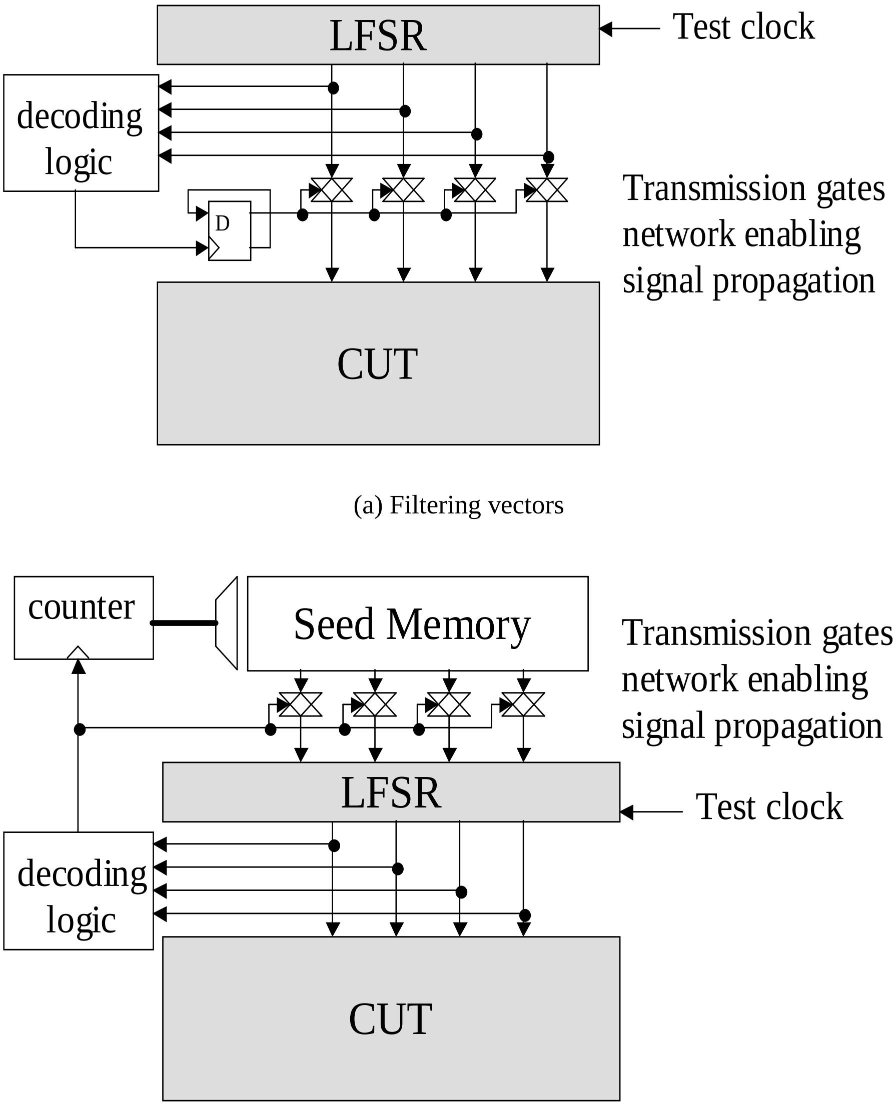


Figure 2.5: Dual speed LFSR for power minimisation during test application [190]

of primary inputs used for switching activity computation during low power logic synthesis. However the computational time in [49] is very high due to the complexity of the test vector ordering problem which is reduced to finding a minimum cost hamiltonian path in a complete, undirected, and weighted graph. The high computational time is overcome by the techniques proposed in [56, 69, 72] where test vector ordering assumes a high correlation between switching activity in the circuit under test and the hamming distance [56, 72] or transition density [69] at circuit primary inputs. For combinational circuits employing BIST several techniques for minimising power dissipation were proposed recently [22, 44, 45, 68, 70, 119, 120, 121, 190, 197, 198]. In [190] the use of dual speed linear feedback shift register (DS-LFSR) lowers the transition density at the circuit inputs leading to minimised power dissipation. The DS-LFSR operates with a slow and a normal speed LFSR, as shown in Figure 2.5, in order to increase the correlation between consecutive patterns. It should be noted that the slow LFSR has both a slow clock and a normal clock as inputs, as well as a control signal which selects the appropriate clock depending on the operation. It was shown in [190] that test efficiency of the DS-LFSR is higher than in the case of the LFSR based on a primitive polynomial with a reduction in power dissipation at the expense of more complex control and clocking. optimum weight sets for input signal distribution are determined in order to minimise average power [198], while the peak power is reduced by finding the best initial conditions in the cellular automata (CA) cells used for pattern generation [197]. It was proven in [22] that all the primitive polynomial LFSR of the same size, produce the same power dissipation in the circuit under test, thus advising the use of the LFSR with the smallest number of XOR gates since it yields the lowest power dissipation by itself. A mixed solution based on reseeding LFSRs and test vector inhibiting to filter a few non-detecting sub-sequences of a pseudorandom


LFSR
Test clock
decoding logic
D
Transmission gates network enabling signal propagation
CUT
counter
Seed Memory
Transmission gates network enabling signal propagation
LFSR
Test clock
decoding logic
CUT

(a) Filtering vectors

(b) Reseeding LFSR

Figure 2.6: Vector filtering and test pattern generator reseeding for power minimisation during test application [68].

test sequence was proposed in [68, 70]. A sub-sequence is non-detecting if all the faults found by it are also observed by other detecting sub-sequences from the pseudorandom test sequence. An enhancement of the test vector inhibiting technique was presented in [70, 119, 120, 121] where all the non-detecting sub-sequences are filtered. The basic principle of filtering non-detecting sequences is to use decoding logic to detect the first and the last vectors of each non-detecting sequence as shown in Figure 2.6(a). After the

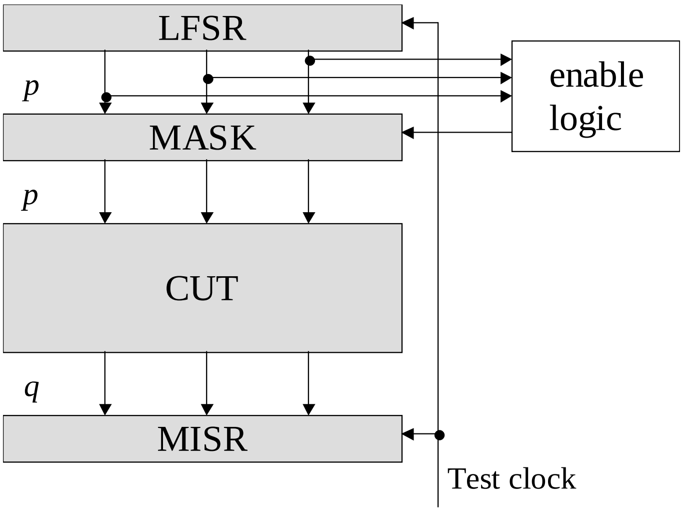


Figure 2.7: Precomputation-based architecture for power minimisation during test application [45]

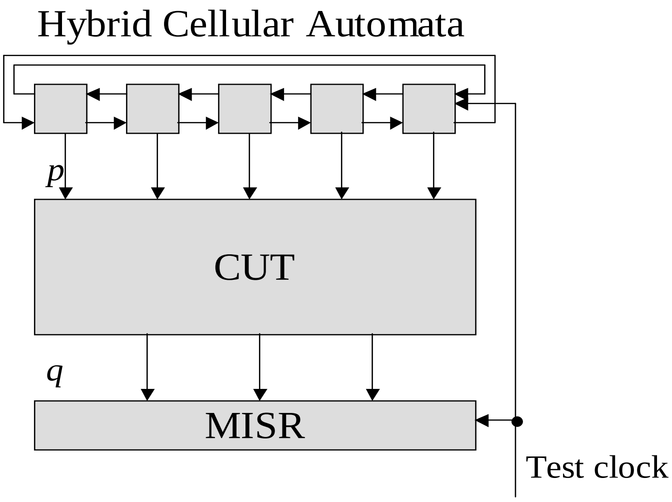


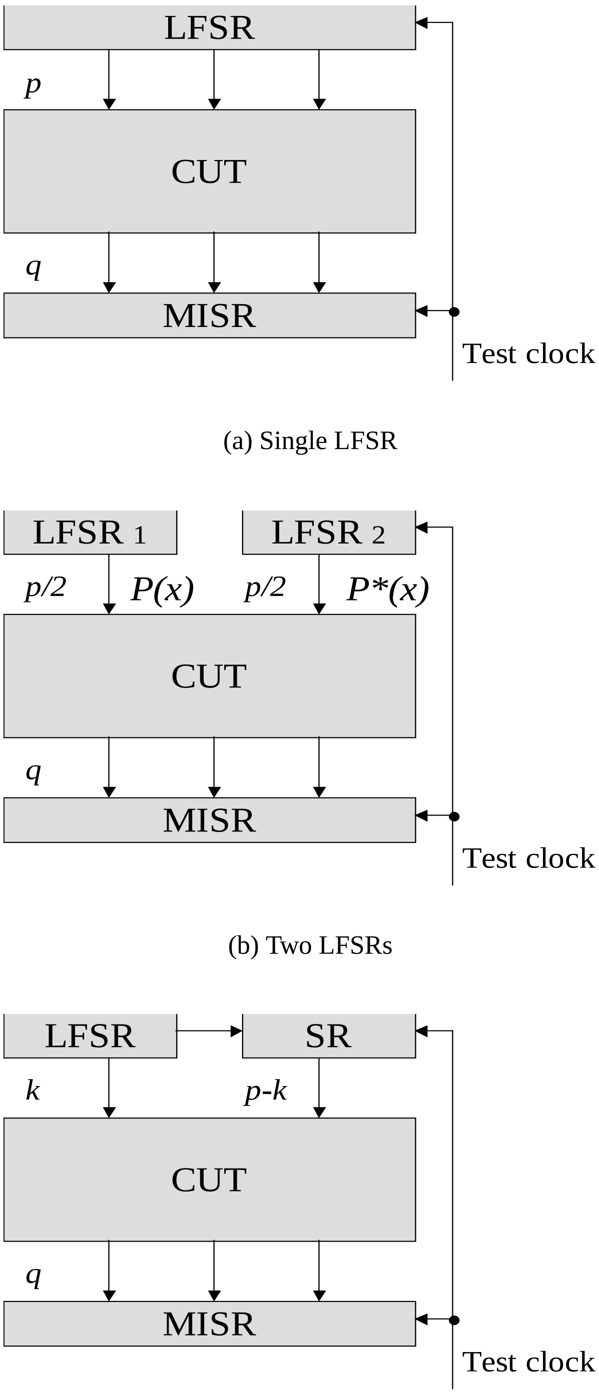
Figure 2.8: Hybrid cellular automata for power minimisation during test application [44]

detection of the first vector of a non-detecting sequence, the inhibiting structure using a transmission gates network enabling signal propagation [159], prevents the application of test vectors to the CUT. To increase the test efficiency by detecting random pattern resistant faults with a small test sequence, an enhanced BIST structure based on reseeding the LFSR is shown in Figure 2.6(b). The particular feature of the proposed BIST structure is that the seed memory is composed of two parts: the first part contains seeds for random pattern resistant faults and the second part contains seeds to inhibit the non-detecting sequences [68]. The seed memory combined with the decoding logic (Figure 2.6(b)) is better than only decoding logic (Figure 2.6(a)) in terms of low power dissipation and high fault coverage, at the expense of higher BIST area overhead.

A different approach for filtering non-detecting vectors inspired by the precomputation architecture [8] is presented in [45]. The MASK block shown in Figure 2.7 is a circuit with a latch-based architecture or AND-based architecture which either eliminates or keeps unaltered the vectors produced by the LFSR. The enable logic implements an incompletely specified boolean fuction whose on-set [51] is the set of the unaltered vectors and whose off-set is the set of the eliminated (non-detecting) vectors [45]. An improvement in area overhead associated with filtering non-detecting vectors without penalty in fault coverage or test application time was achieved using a non-linear hybrid cellular automata [44]. The hybrid cellular automata shown in Figure 2.8 generates test patterns for the CUT using cell configurations optimised for low power dissipation under given fault coverage and test application time constraints. The regularity of multiplier modules and linear sized test set required to achieve high fault coverage lead to efficient low power BIST implementations for data paths [12, 74, 75, 77, 89].

Regardless of the implementation type of the test pattern generator, BIST architectures significantly differ one from another in terms of power dissipation [166]. The three different architectures shown in Figures 2.9(a)-2.9(c) were evaluated for power dissipation, BIST area overhead and test application time. It was found in [166] that the architecture consisting of an LFSR and a shift register SR (Figure 2.9(c)) produces lower power dissipation, BIST area overhead and test application time when compared to a single LFSR (Figure 2.9(a)) and two LFSRs with reciprocal characteristic polynomials (Figure 2.9(b)). However, this is achieved at the expense of lower fault coverage and hence reduced test efficiency due to the modified sequence of patterns applied to the CUT which does not detect all the random pattern resistant faults.

Circuit partitioning into subcircuits and conscious subcircuit test planning have an important influence on power dissipation as described in [67]. The main justification for circuit partitioning is to obtain two different structural circuits of approximately the same size, so that each circuit can be successively tested in two diferent sessions as shown in Figures 2.10(a)-2.10(d). In order to minimise the BIST area overhead of the resulting BIST scheme (Figures 2.10(c)-2.10(d)), the number of connections between the two

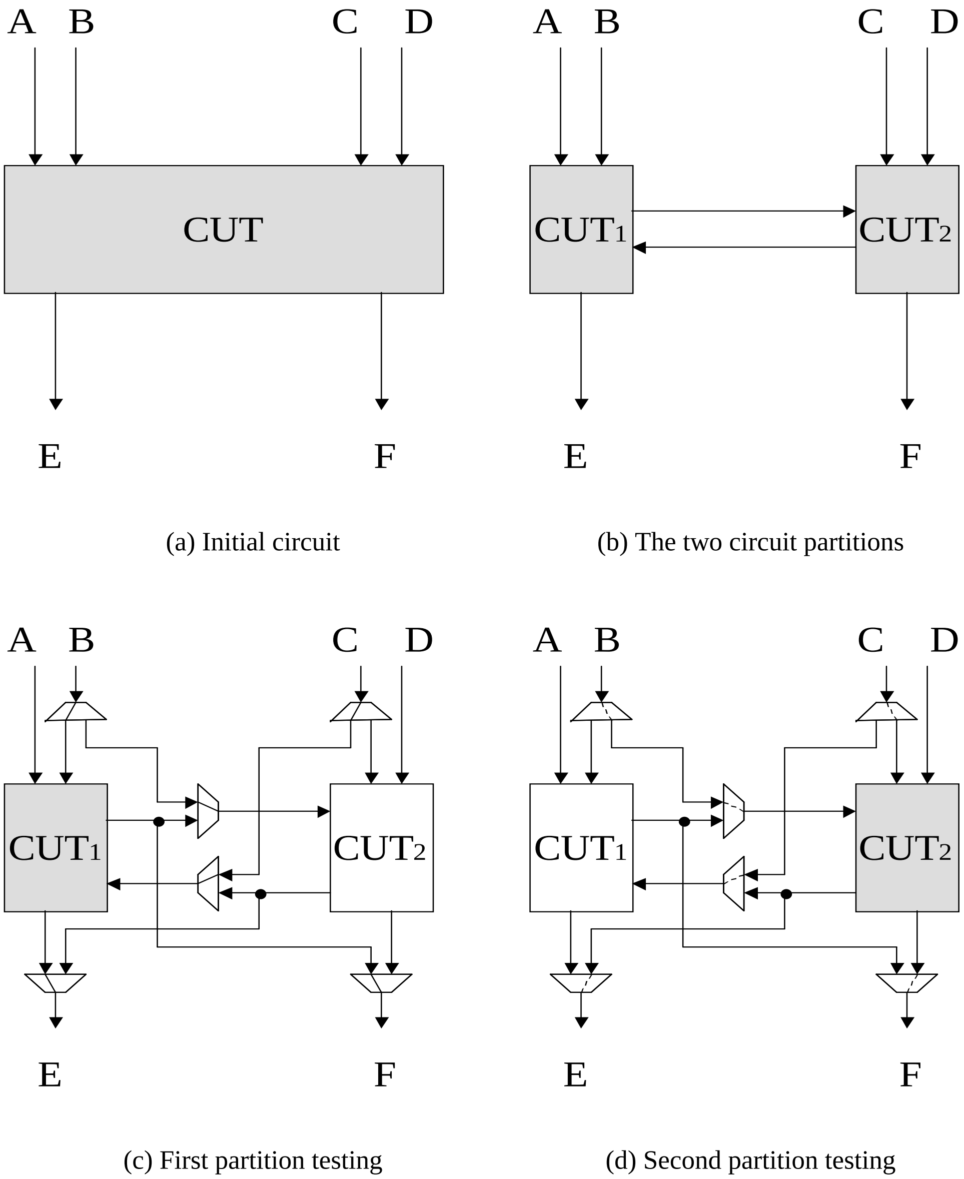


(a) Single LFSR

(b) Two LFSRs

(c) LFSR and SR

Figure 2.9: Evaluating different BIST architectures for power minimisation during test application [166].



(a) Initial circuit

(b) The two circuit partitions

(c) First partition testing

(d) Second partition testing

Figure 2.10: Circuit partitioning for power minimisation during test application [67].

subcircuits has to be minimum. It was shown in [67] that by partitioning a single circuit entity into two subcircuits and executing two successive tests as shown in Figures 2.10(c)-2.10(d), savings in power dissipation can be achieved with roughly the same test application time as in the case of a single circuit entity.

Although the techniques proposed for minimising power dissipation during test application in combinational circuits at the logic level of abstraction achieve good results, they can further be combined with the techniques proposed at register-transfer level.

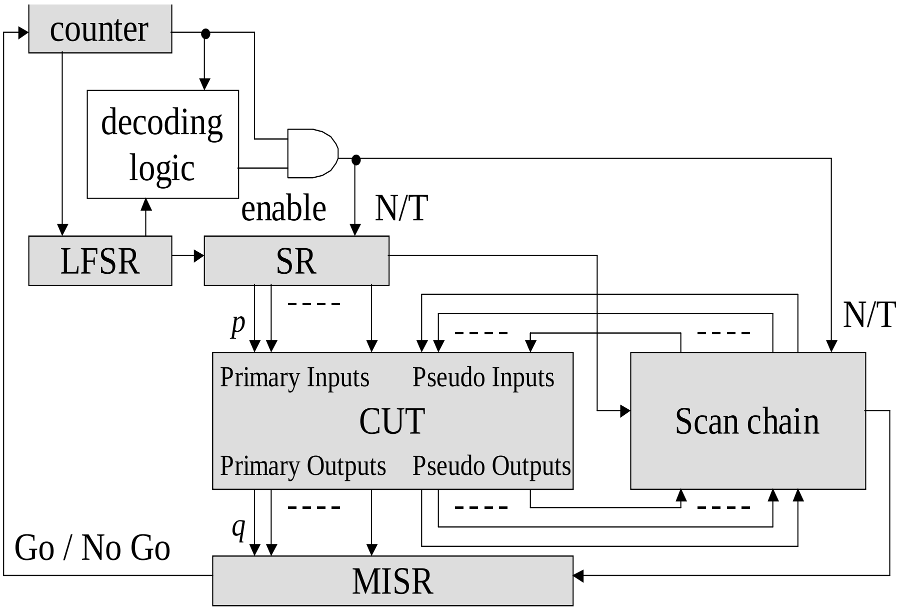


Figure 2.11: Precomputation architecture for power minimisation during test application in scan BIST methodology [46]

Problem (ib): To minimise power dissipation in non-scan sequential circuits during test application a test pattern generation methodology for low power dissipation was proposed in [43]. The methodology is based on three independent steps comprising redundant test pattern generation, power dissipation measurement and optimum test sequence selection. The solution proposed in [43], which is based on genetic algorithms, achieves considerable savings in power dissipation, however cannot be applied to scan sequential circuits where shifting power dissipation is the major contributor to the total power dissipation.

To minimise shifting power dissipation in scan sequential circuits, test vector inhibiting techniques proposed for combinational circuits are extended to scan sequential circuits [46]. The scan BIST methodology presented in Figure 1.5 from Section 1.3.1 requires to be extended with decoding logic and an AND gate to enable pattern shifting as shown in Figure 2.11. Based on the content of the LFSR the decoding logic detects whether the test pattern to be shifted belongs to the subset of detecting sequences. If the pattern is non-detecting the propagation through the SR and scan chain is stopped. In [61] the test vector inhibiting technique is extended where the modules and modes with the highest power dissipation are identified, and gating logic is introduced in order to reduce

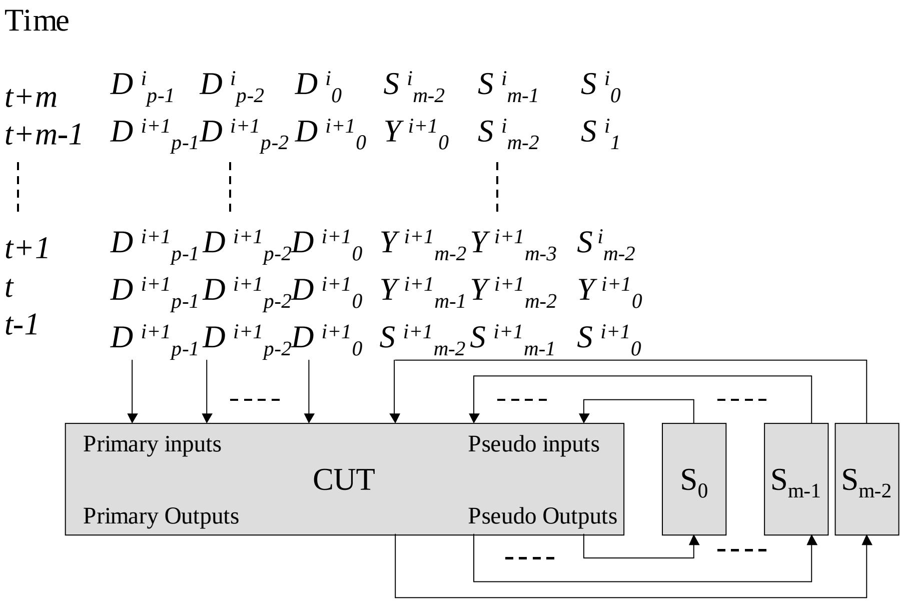


Figure 2.12: Test vector and scan cell reordering for power minimisation in full scan sequential circuits [49]

power dissipation. Despite substantial savings in power dissipation vector detection and gating logic introduce not only significant area overhead but also considerable performance degradation for modified scan cell design. In [182] a new scan BIST structure was proposed based on the experimental observation that a very high fault coverage can be obtained by a small number of clusters of test vectors. Although not targeted specifically for low power dissipation during test application the approach in [182], yields high fault coverage with correlated scan patterns which can also lead to lower power dissipation. A similar approach is employed in the low transition random test pattern generator (LT-RTPG) proposed in [192], where neighbouring bits of the test vectors are assigned identical values in most test vectors. A simple and fast procedure to compact scan vectors as much as possible without exceeding power dissipation was proposed in [172]. All the previous scan-based BIST methods [46, 61, 172, 182, 192] introduce test area overhead and/or further performance degradation when compared to scan DFT methodology.

A different technique [49] based on test vector and scan cell ordering minimises power dissipation in full scan sequential circuits without any overhead in test area or performance degradation as shown in Figure 2.12. The input sequence at the primary and pseudo inputs of the CUT while shifting out test response in the case of standard

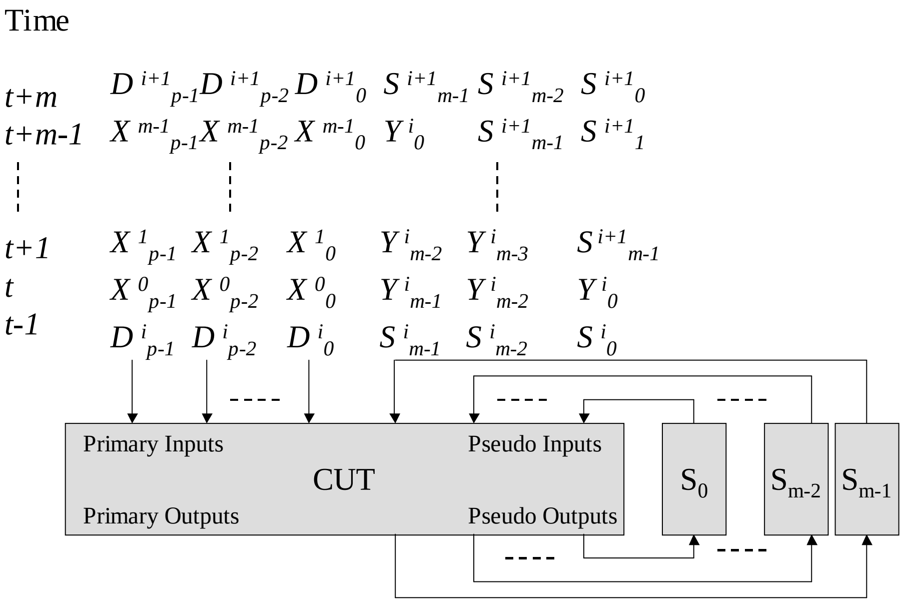

Time
$t+m$ D^{i+1}_{p-1} D^{i+1}_{p-2} D^{i+1}_{0} S^{i+1}_{m-1} S^{i+1}_{m-2} S^{i+1}_{0}
$t+m-1$ X^{m-1}_{p-1} X^{m-1}_{p-2} X^{m-1}_{0} Y^{i}_{0} S^{i+1}_{m-1} S^{i+1}_{1}
$t+1$ X^{1}_{p-1} X^{1}_{p-2} X^{1}_{0} Y^{i}_{m-2} Y^{i}_{m-3} S^{i+1}_{m-1}
t X^{0}_{p-1} X^{0}_{p-2} X^{0}_{0} Y^{i}_{m-1} Y^{i}_{m-2} Y^{i}_{0}
$t-1$ D^{i}_{p-1} D^{i}_{p-2} D^{i}_{0} S^{i}_{m-1} S^{i}_{m-2} S^{i}_{0}
Primary Inputs
Pseudo Inputs
CUT
Primary Outputs
Pseudo Outputs
S_0
S_{m-2}
S_{m-1}

Figure 2.13: Power minimisation in scan sequential circuits using extra primary input vectors for each clock cycle of the scan cycle of every test pattern presented in [189]

scan design (Figure 1.3 from Section 1.2) is significantly modified when reordering scan cells S_{m-1} and S_{m-2} and test vectors $V_{i+1} = D^{i+1}_{p-1}D^{i+1}_{p-2}\ldots D^{i+1}_{0}S^{i+1}_{m-1}S^{i+1}_{m-2}\ldots S^{i+1}_{0}$ and $V_i = D^{i}_{p-1}D^{i}_{p-2}\ldots D^{i}_{0}S^{i}_{m-1}S^{i}_{m-2}\ldots S^{i}_{0}$. The new sequence obtained after reordering will lead to lower switching activity and hence lower power dissipation due to higher correlation between consecutive patterns at the primary and pseudo inputs of the CUT. Further benefit of the post-ATPG technique proposed in [49] is that minimisation of power dissipation during test application is achieved without any decrease in fault coverage and/or increase in test application time. The technique is test set dependent which means that power minimisation depends on the size and the value of the test vectors in the test set. Due to its test set dependence, the technique proposed in [49] is computationally infeasible due to large computational time required to explore the large design space.

A different approach to achieve power savings is the use of extra primary input vectors, which leads to supplementary volume of test data [88, 189]. The technique proposed in [189] exploits the redundant information that occurs during scan shifting to minimise switching activity in the CUT as shown in Figure 2.13. While shifting out the pseudo output part of the test response $Y^{i}_{m-1}Y^{i}_{m-2}\ldots Y^{i}_{0}$ during the clock cycles $t\ldots t_{m-1}$ the value of the primary inputs is redundant. Therefore this redundant information can

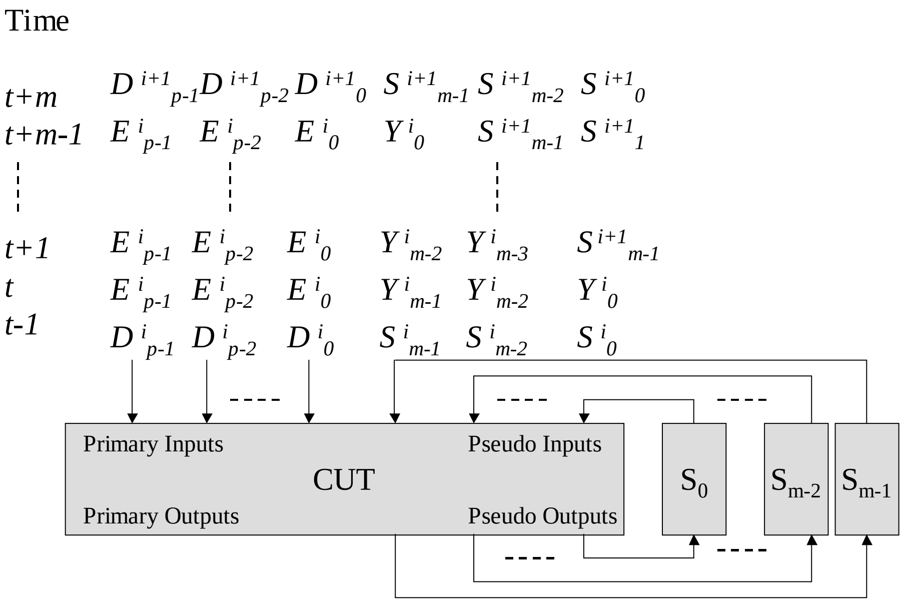
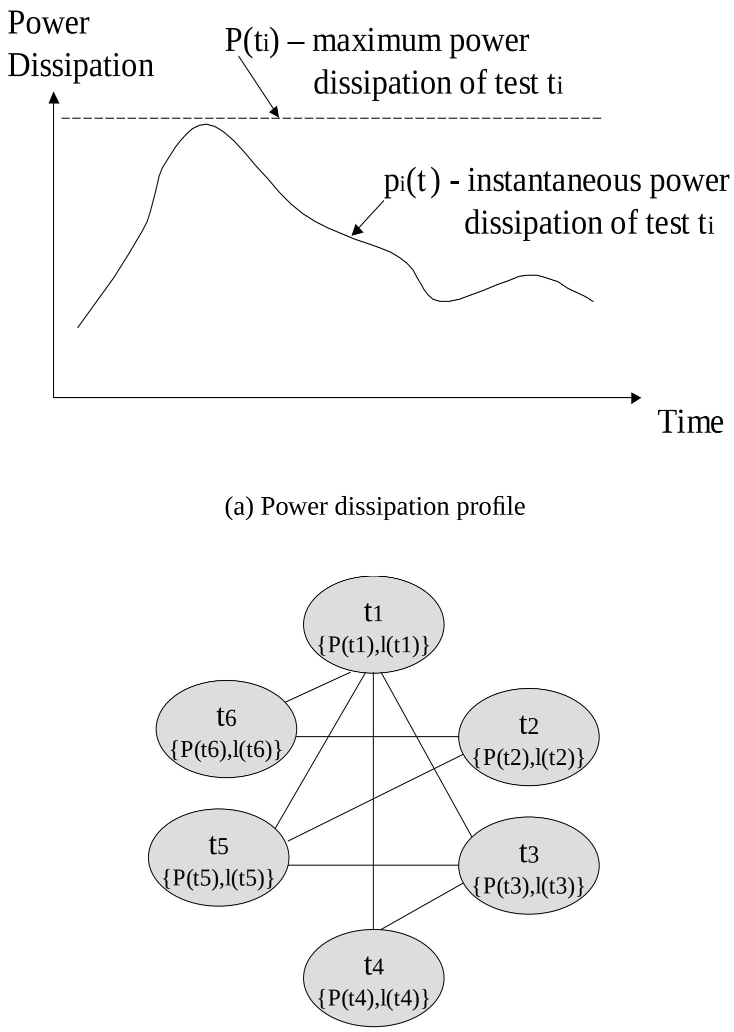

Time
t+m
t+m-1
t+1
t
t-1
D i+1 p-1 D i+1 p-2 D i+1 0 S i+1 m-1 S i+1 m-2 S i+1 0
E i p-1 E i p-2 E i 0 Y i 0 S i+1 m-1 S i+1 1
E i p-1 E i p-2 E i 0 Y i m-2 Y i m-3 S i+1 m-1
E i p-1 E i p-2 E i 0 Y i m-1 Y i m-2 Y i 0
D i p-1 D i p-2 D i 0 S i m-1 S i m-2 S i 0
Primary Inputs
Pseudo Inputs
CUT
Primary Outputs
Pseudo Outputs
S0
Sm-2
Sm-1

Figure 2.14: Power minimisation in scan sequential circuits using only a single extra primary input vector for all the clock cycles of the scan cycle of every test pattern [88]

be exploited by computing an extra primary input vector $X^k_{p-1}X^k_{p-2}\dots X^k_0$ for *each* clock cycle $t+k$, with $k = 0\dots m-1$, of the scan cycle $t\dots t_{m-1}$ of every test pattern $V_i = D^i_{p-1}D^i_{p-2}\dots D^i_0 S^i_{m-1}S^i_{m-2}\dots S^i_0$. However, despite achieving considerable power savings the technique requires large test application time which is related to a long computational time, and a large volume of test data. The volume of test data is reduced in [88] where a D-algorithm like ATPG [2] is developed to generate a single control vector to mask the circuit activity while shifting out the test responses. Unlike the technique proposed in [189] based on a large number of extra primary input vectors, the solution presented in [88] employs a *single* extra primary input vector $E^i_{p-1}E^i_{p-2}\dots E^i_0$ for *all* the clock cycles of the scan cycle of every test pattern $V_i = D^i_{p-1}D^i_{p-2}\dots D^i_0 S^i_{m-1}S^i_{m-2}\dots S^i_0$ (Figure 2.14). The input control technique proposed in [88] can further be combined with previously proposed scan cell and test vector ordering [49] to achieve, however, modest savings in power dissipation despite a substantial reduction in volume of test data when compared to [189]. However, both approaches based on extra primary input vectors [88, 189] require high computational time and hence are infeasible for large sequential circuits.

Despite their efficiency for minimising power dissipation in scan sequential circuits, the previous approaches trade off one test parameter at the benefit of another test parame-



(a) Power dissipation profile

(b) Power annotated test compatibility graph from Figure 1.8(b)

Figure 2.15: Power profile and power annotated test compatibility graph for power constrained test scheduling

ter of the scan based DFT method described in Section 1.2. Therefore new techniques are required for small to medium sized and large scan sequential circuits. These problems are addressed and solved using the new techniques proposed in Chapters 3 and 4.

Problem (ii): The motivation for considering testability at the register-transfer level and the representative previous work is presented in Section 5.1.1 from Chapter 5. This paragraph overviews only the relevant previous approaches for minimising power dissipation during test application in RTL data paths.

To overcome the problem of high power dissipation during test application at RTL motivated by Example 2.2, numerous power-constrained test scheduling algorithms were proposed under a BIST environment [25, 41, 108, 109, 110, 132, 133, 134, 165, 167, 199]. The approach in [199] schedules the tests under power constraints by grouping and ordering based on floorplan information. A further exploration in the solution space of the scheduling problem is provided in [41] where a resource allocation graph formulation (Figure 1.8(a) from Section 1.3.2) for the test scheduling problem is given and tests are scheduled concurrently without exceeding their power constraint during test application. To simplify the scheduling problem the worst case power dissipation (maximum instantenous power dissipation) is used to characterise the power constraint of each test as shown in Figure 2.15(a). The test compatibility graph introduced in Figure 1.8(b) is annotated with power and test application time information as shown in Figure 2.15(b). The power rating $P(t_i)$ characterised by maximum power dissipation (Figure 2.15(a)) and test application time $l(t_i)$ are used for scheduling unequal length tests under a power constraint. To overcome the identification of all the cliques in a graph and the covering table minimisation problem applied in [41], which are well known NP-hard problems, the solutions proposed in [132, 133, 134] use list scheduling, left edge algorithm and a tree growing technique as an heuristic for the block test scheduling problem. Power constrained test scheduling is extended to system on a chip in [25, 165, 167]. A test infrastructure and power constrained test scheduling algorithms for a scan-based architecture are presented in [108, 109, 110].

All the previous approaches for power constrained test scheduling have assumed a fixed amount of power dissipation associated with each test. This is an optimistic assumption which is not valid when employing BIST for RTL data paths designed for low power due to useless power dissipation introduced in Section 6.2. This problem is addressed and solved using the new techniques proposed in Chapter 6.

Summary: This chapter motivated the need for low power testing in order to maintain high circuit yield and reliability. A review of the sources of higher power dissipation during test application when compared to the functional operation was presented. Recently proposed low power testing techniques at logic level and register-transfer level of abstraction were overviewed.

Chapter 3

Power Minimisation in Scan Sequential Circuits Based On Best Primary Input Change Time

The most important design for testability (DFT) method, at the logic level of abstraction, employed for increasing the testability of VLSI circuits is the scan-based DFT method [2, 186]. The scan-based DFT method makes sequential elements (latches or flip flops) controllable and observable by chaining them into a shift register (scan chain). Early test automation approaches have inserted scan cells after the preliminary stages of gate placement and routing were completed [59]. However, due to the increasing complexity of very deep sub-micron VLSI circuits scan chains need to be inserted in a structural network of logic gates at the logic level of abstraction of the VLSI design flow. Therefore, the best exploration of alternative solutions for power minimisation in scan sequential circuits, is most effectively done at the logic level of abstraction. This is illustrated in Figure 3.1 where scan cells can be inserted either prior to or after the logic optimisation phase. The design is specified in a hardware description language (HDL) (either VHDL [136] or Verilog [137]) at the register-transfer level (RTL) of abstraction of the VLSI design flow and RTL synthesis translates the initial design into a network of logic gates before logic optimisation satisfies the area and delay constraints, and prepares the design for the physical design automation tools.

This chapter addresses power minimisation during test application in small to medium sized scan sequential circuits by analysing and exploiting the influence of primary input change time on the minimisation of power dissipation during test application. A new test application strategy based on best primary input change (BPIC) time in scan sequential

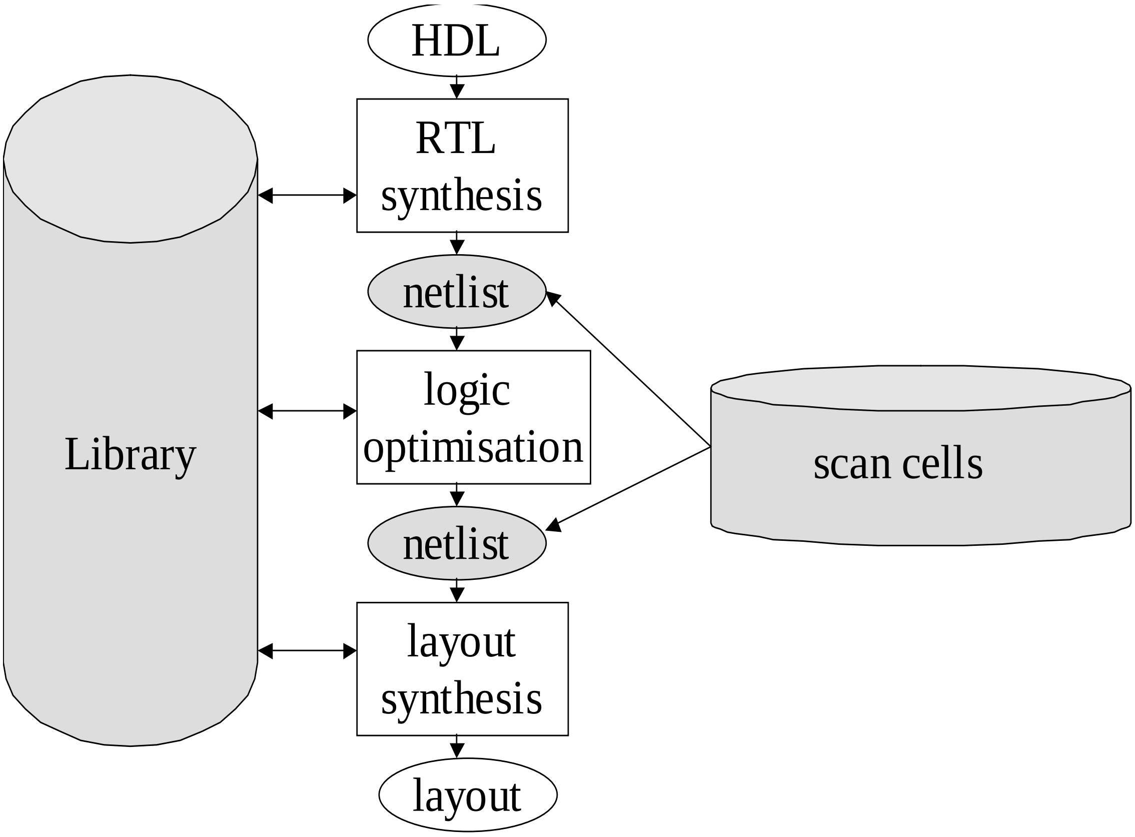


Figure 3.1: Logic level scan insertion.

circuits is introduced. Furthermore, the effect of combining the primary input change time with test vector and scan cell ordering [49] on power dissipation is investigated. The proposed test application strategy depends only on controlling the primary input change time, and hence minimises power dissipation during test application with no penalty in test area, performance, test efficiency, test application time or volume of test data.

The rest of the chapter is organised as follows. Section 3.1 gives the motivation and objectives of the proposed research. In Section 3.2, the power dissipation model and the parameters which are accountable for power dissipation in scan circuits during test application are described. Section 3.3 explains why the primary input change time has strong impact on reducing spurious transitions during test application. New algorithms for exploiting all the parameters which lead to considerable savings in power dissipation are introduced in Section 3.4. Experimental results and a comparative study of full scan and partial scan from the power dissipation standpoint are presented in Section 3.5. Finally, concluding remarks are given in Section 3.6.

3.1 Motivation and Objectives

To reduce the complexity of ATPG for sequential circuits structured DFT is required. When all the sequential elements are chained into a shift register, the full scan DFT method is employed [2]. Design teams use an existing family of scan cells from a standard cell library developed by a semiconductor manufacturer or third-party library vendor. Although full scan reduces the complexity of ATPG for sequential circuits to ATPG for combinational circuits, which is more tractable, there are three main shortcomings associated with full scan design [97]: increase in critical path delays which leads to performance degradation; increase in test area due to extra hardware; long test application due to serial shifting of test patterns and responses. To reduce performance degradation, test area overhead and test application time associated with full scan, partial scan was proposed [3]. The main attribute of partial scan DFT method [26, 31, 93, 113, 126, 152, 153, 154, 177] is to select a small number of scan cells which allows ATPG to achieve a high fault coverage in a low computational time. Most of the previous approaches [46, 49, 88, 172, 189, 192] proposed to reduce power dissipation in scan sequential circuits introduce further overhead in performance, area or test application time. The only technique for power minimisation in full scan sequential circuits with no penalty in test area, performance, test efficiency, test application time or volume of test data was proposed in [49]. This technique is based on test vector ordering and scan cell ordering. On the one hand, test vector ordering proposed in [49] is efficient for full scan, but it is prohibited for partial scan. This is due to the fact that testing partial scan sequential circuits is a combination of testing full scan and non-scan sequential circuits where fixed test vector order fault activation and fault-effect propagation sequences through non-scan cells are required. On the other hand, scan cell ordering was previously used to improve coverage of delay faults in skew-load delay fault testing [122], to reduce test application time [135], and to minimise routing area overhead [32]. However, scan cell ordering proposed in [49] is test set dependent and targets minimisation of power dissipation during test application. A test set dependent approach for power minimisation depends on the size and the value of the test vectors in the test set. This is unlike the test set independent approaches, where power minimisation depends only on the circuit structure and savings are guaranteed regardless of the size and and the value of the test vectors in the test set. Finally, the technique proposed in [49] is applicable only to full scan sequential circuits, due to test vectoring and did not consider the effect of the timing of the primary input part of the test vector on the power dissipation during test application.

This chapter proposes a new test set dependent test application strategy which is applicable to both full scan and partial scan sequential circuits with no penalty in test area, performance, test efficiency, test application time or volume of test data. It is also shown that the smaller number of scan cells in partial scan sequential circuits leads not only to commonly known less test area overhead and test application time, but also to less power dissipation during test application and computational time required for design space exploration when compared to full scan sequential circuits.

3.2 Power Dissipation During Test Application

Section 3.2.1 introduces the power dissipation model used by the techniques and algorithms presented in Sections 3.3 and 3.4 respectively. Section 3.2.2 reviews scan cell and test vector ordering proposed by previous research [49] for full scan sequential circuits, and Section 3.2.3 investigates the applicability of scan cell and test vector ordering for partial scan sequential circuits.

3.2.1 Power Dissipation Model

Total power dissipation in CMOS circuits can be divided into static, short circuit, leakage and dynamic power dissipation. The static power dissipation is negligible for correctly designed circuits. Short circuit power dissipation caused by short circuit current during switching and power dissipated by leakage currents contribute up to 20% of the total power dissipation. The remaining 80% is attributed to dynamic power dissipation caused by switching of the gate outputs [29]. If the gate is part of a synchronous digital circuit controlled by a global clock, it follows that the dynamic power P_d required to charge and discharge the output capacitance load of every gate is:

$$P_d = 0.5 \times C_{load} \times (V_{DD}^2 / T_{cyc}) \times N_G \tag{3.1}$$

where C_{load} is the load capacitance, V_{DD} is the supply voltage, T_{cyc} is the global clock period, and N_G is the total number of gate output transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$). The vast majority of power reduction techniques concentrate on minimising the dynamic power

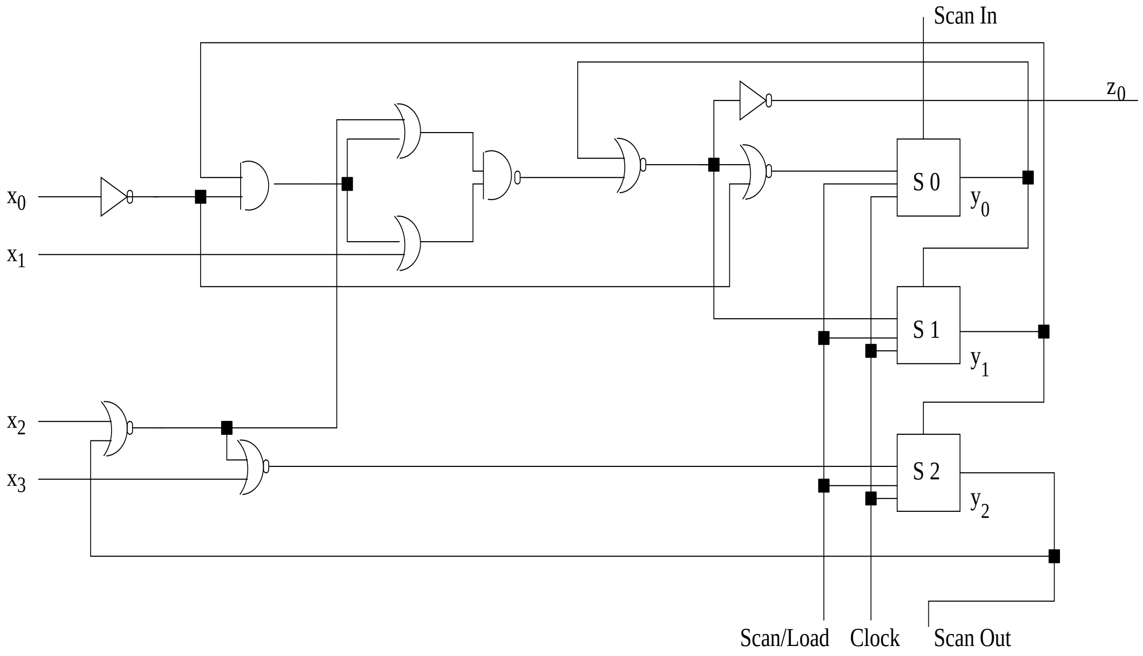


Figure 3.2: Example 3.1 circuit (*s27* from [23]).

dissipation P_d by minimising switching activity. Thus, *node transition count*

$$NTC = \sum_{for\ all\ gates\ G} N_G \times C_{load} \tag{3.2}$$

is used as quantitative measure for power dissipation throughout this chapter. It is assumed that the load capacitance for each gate is equal to the number of fanouts. The node transition count in scan cells, N_{SC}, is considered as in [49], where it was shown that for input changes 0 → 0 and 1 → 1, $N_{SC_{min}} = 2$, whilst for input changes 0 → 1 and 1 → 0, $N_{SC_{max}} = 6$. Similarly, the node transition count in non-scan cells, N_{NSC}, is considered $N_{NSC_{min}} = 1$ and $N_{NSC_{max}} = 4$. It should be noted that non-scan cells are not clocked while shifting out test responses which leads to zero value in NTC.

3.2.2 The Influence of Test Vector and Scan Cell Ordering on Power Minimisation in Full Scan Sequential Circuits

Previous research has established that the node transition count in full scan sequential circuits depends on two factors, test vector ordering and scan cell ordering, when the circuit is in the test mode [49]. The following example shows how test vector and scan cell ordering affect the circuit activity during test application in full scan sequential circuits.

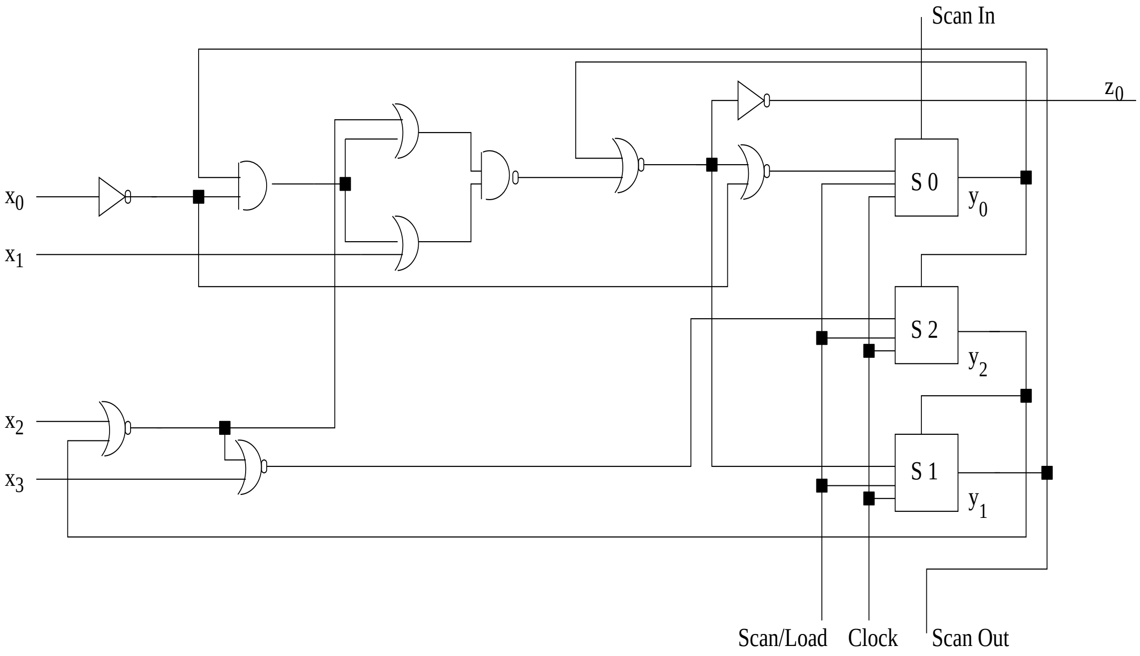


Figure 3.3: Example 3.1 circuit (*s27* from [23]) after permuting the order of S_1 and S_2.

Example 3.1 To illustrate the factors accountable for power dissipation consider the *s27* circuit (Figure 3.2) from the commonly accepted ISCAS89 benchmark set [23]. The primary inputs are $\{x_0, x_1, x_2, x_3\}$, $\{S_0, S_1, S_2\}$ are the scan cells, $\{y_0, y_1, y_2\}$ are the present state lines, and $\{z_0\}$ is the circuit output. Using the GATEST [170] ATPG tool, it was shown that 5 test vectors are needed to achieve 100% fault coverage. The test vectors are $\{1101011, 0000000, 0010010, 0111111, 1100010\}$. For easy reference they are labelled as $\{V_0, V_1, V_2, V_3, V_4\}$. Each test vector consists of primary inputs and pseudo inputs (present state lines) in the following order $x_0 x_1 x_2 x_3 y_0 y_1 y_2$. Assuming that initially all the primary and pseudo inputs are set to 0 and using Equation 3.2 the node transition count is calculated as $NTC = 372$. A detailed description for calculating NTC over the entire test application period is outlined in Section 3.3. By reordering the test vectors as such $\{V_0, V_2, V_4, V_3, V_1\}$ a new lower value for node transition count is obtained $NTC = 352$. This shows that reordering of test vectors reduces power dissipation during test application by increasing the correlation between consecutive test vectors. Note that the NTC is computed over the entire test application period of $n \times (m+1) + m$ clock cycles, where n is the number of test vectors and m is the number of scan cells. Now the effect of scan cell ordering on power savings is examined. Consider the reordered test vector set

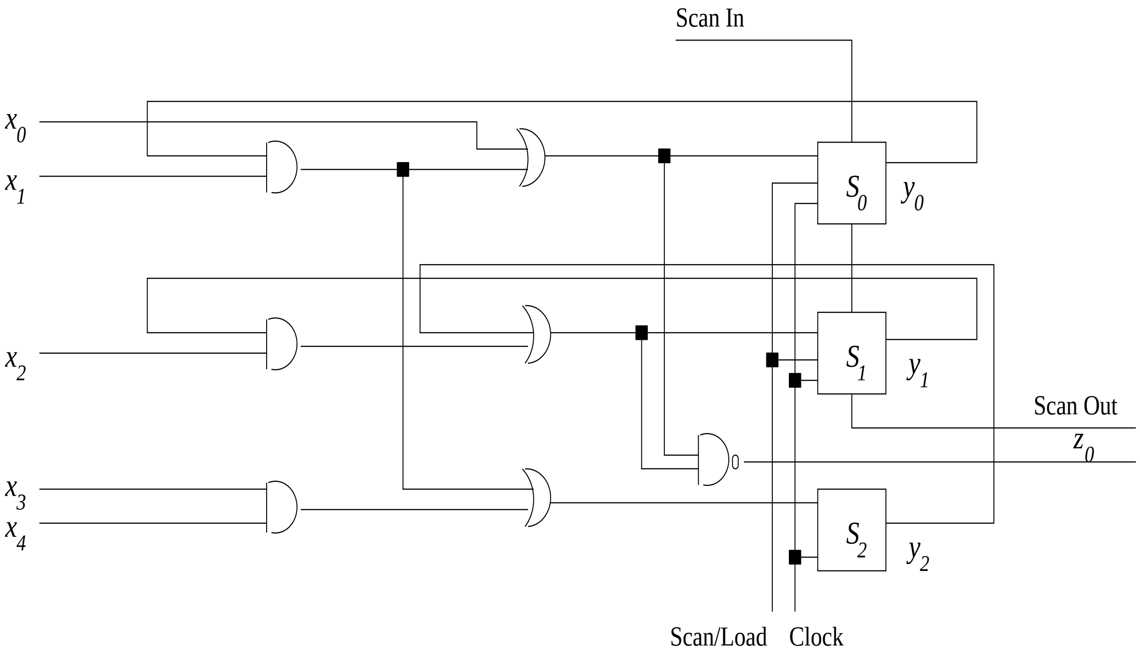


Figure 3.4: Example 3.2 partial scan circuit.

$\{V_0, V_2, V_4, V_3, V_1\}$ and reordering scan cells to $\{S_0, S_2, S_1\}$ as shown in Figure 3.3 the value of node transition count is reduced further to $NTC = 328$. This reduction is due to the higher correlation between succesive states during shifting in test vectors and shifting out test responses. If test vector ordering and scan cell ordering are done simultaneously a further reduction in node transition count is achieved $NTC = 296$, for the following test vector order $\{V_0, V_2, V_3, V_4, V_1\}$ and scan cell order $\{S_2, S_1, S_0\}$. This shows that scan cell ordering and test vector ordering are interrelated which leads to higher savings than when either scan cell ordering or test vector ordering are considered separetely.

3.2.3 The Influence of Scan Cell Ordering on Power Minimisation in Partial Scan Sequential Circuits

It was shown in Example 3.1 how test vector ordering affects circuit activity and hence power dissipation in full scan sequential circuits. However, test vector ordering proposed in [49] is prohibited for partial scan due to the fixed test vector order fault activation and fault-effect propagation sequences through non-scan cells [3]. On the other hand, scan cell ordering can be applied for partial scan sequential circuits as shown in the following example.

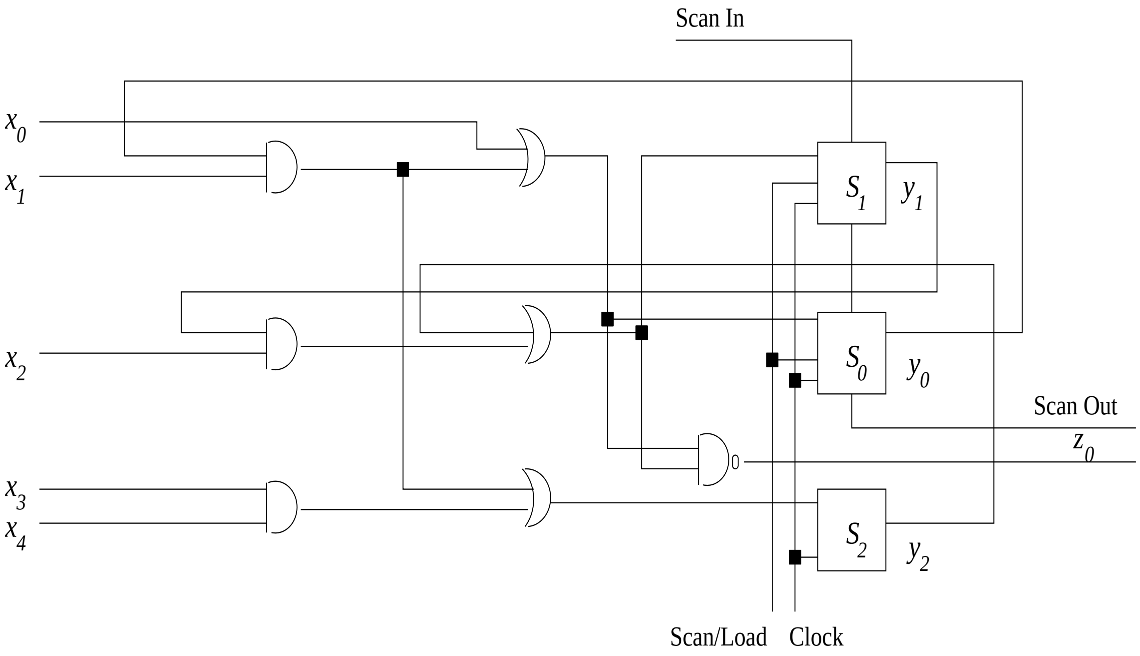


Figure 3.5: Example 3.2 partial scan circuit after permuting scan cells S_0 and S_1.

Example 3.2 To investigate the influence of scan cell ordering on power dissipation during test application in partial scan sequential circuits consider the simple circuit shown in Figure 3.4. The primary inputs are $\{x_0,x_1,x_2,x_3,x_4\}$, $\{S_0,S_1\}$ are the scan cells, $\{S_2\}$ is the non-scan cell, $\{y_0,y_1,y_2\}$ are the present state lines, and $\{z_0\}$ is the circuit output. The scan cells are selected using the logic level partial scan tool OPUS [37, 38]. Using the logic level ATPG tool GATEST [170], 6 test vectors are generated to achieve 100% fault coverage. The test vectors are $\{1011110, 0001010, 0111010, 0110100, 1010111, 0100101\}$. For easy reference they are labelled as $\{V_0,V_1,V_2,V_3,V_4,V_5\}$. Each test vector consists of a primary input part and a present state part in the following order $x_0x_1x_2x_3x_4y_0y_1$. Initially all the primary inputs and present state lines are considered 0 and using Equation 3.2 the node transition count is calculated as $NTC = 224$. By reordering the scan cells to $\{S_1,S_0\}$ as shown in Figure 3.5 the value of the node transition count is reduced to $NTC = 216$.

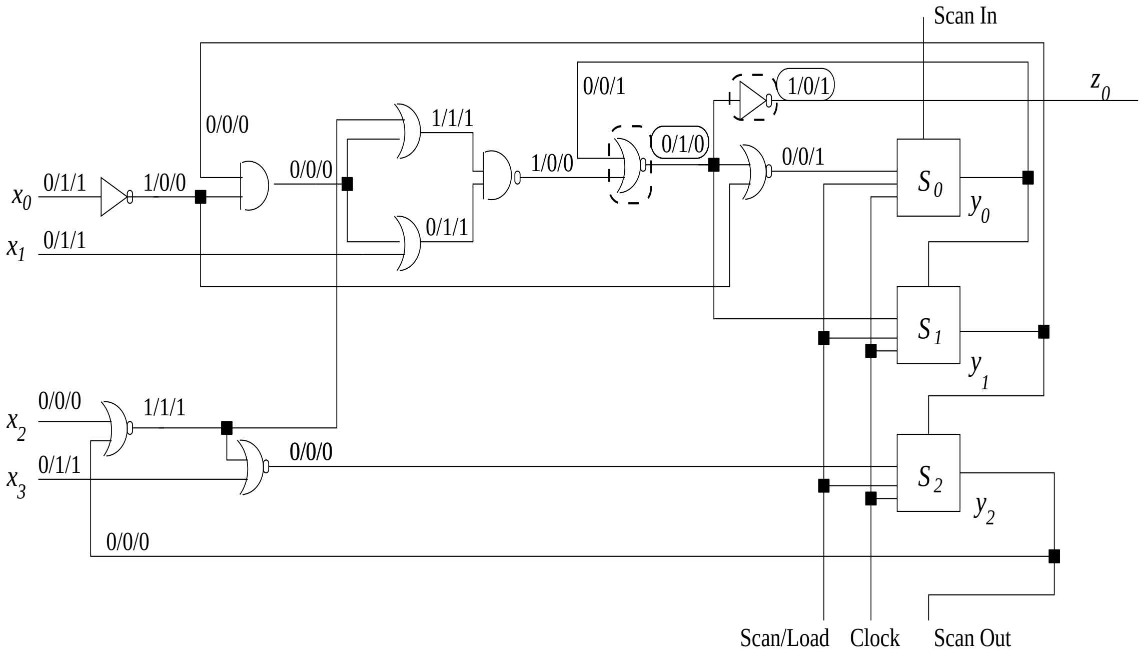
The techniques shown in the previous Examples 3.1 and 3.2 yield modest savings in NTC, and hence in power dissipation. To further reduce power dissipation during test application in the circuit under test, a new test application strategy is described in the following Section 3.3.

3.3 New Technique for Minimisation of Power Dissipation During Test Application By Controlling Primary Input Change Time

In this section the key ideas of the proposed technique are presented. The influence of primary input change time on the reduction of spurious transitions, and hence savings in the total number of transitions, is demonstrated through detailed examples. Section 3.3.1 introduces the new test application strategy for small to medium sized full scan sequential circuits with no penalty in test area, performance, test efficiency, test application time or volume of test data. Section 3.3.2 illustrates the applicability of the proposed test application strategy to partial scan sequential circuits, and Section 3.3.3 describes how the proposed new test application strategy can be extended to scan BIST methodology introduced in Figure 1.5 from Section 1.3.1.

3.3.1 New Test Application Strategy for Full Scan Sequential Circuits

To motivate the need for a new test application strategy for power minimisation, an overview of testing scan sequential circuits is provided. For a scan sequential circuit, each test vector $V_i = x_i@y_i$ applied to the circuit under test is composed of primary input part x_i and pseudo input (present state part) y_i, where @ denotes concatenation. Given m scan cells, for each test vector $V_i = x_i@y_i$ the present state part y_i is shifted in m clock cycles t_0 to t_{m-1}. In the case of partial scan sequential circuits, the non-scan cells preserve their value during clock cycles t_0 to t_{m-1}. In the next clock cycle t_m the entire test vector $V_i = x_i@y_i$ is applied to the circuit under test. A scan cycle represents the $m+1$ clock cycles t_0 to t_m required to shift in the present state part of the test vector and apply the entire test vector to the circuit under test. In the following m clock cycles of the next scan cycle the test response y_i' is shifted out simultaneously with shifting in the present state part of the next test vector $V_j = x_j@y_j$. The values of the primary inputs are important only at t_m when the entire test vector is applied. Therefore the primary inputs can be changed at clock cycles t_0 to t_{m-1} without affecting test efficiency. The transitions which occur in the circuit combinational part, without any influence on test efficiency or test data, are defined as follows.



(a) Primary inputs change as soon as possible (ASAP) at t_0

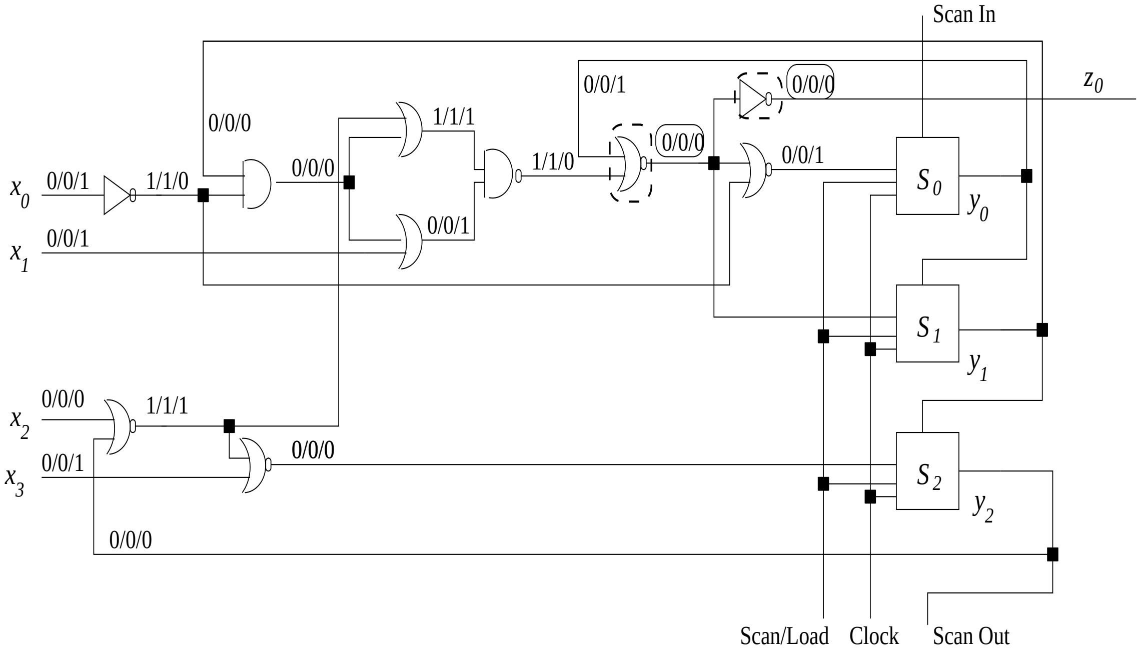
Figure 3.6: Example circuit (*s27* from [23]) illustrating factors which lead to spurious transitions during test application

Definition 3.1 A spurious transition during test application in scan sequential circuits is a transition which occurs in the combinational part of the circuit under test while shifting out the test response and shifting in the present state part of the next test vector. These transitions do not have any influence on test efficiency since the values at the input and output of the combinational part are not useful test data.

It was assumed in Example 3.1 circuit (Section 3.2.2) that changing of the primary inputs $x_0x_1x_2x_3$ occurs at time t_0 The following two definitions introduce two test application strategies that will be used throughout this dissertation.

Definition 3.2 The test application strategy where primary inputs change at t_0 is called *as soon as possible* (ASAP).

Definition 3.3 The test application strategy where primary inputs change at t_m is called *as late as possible* (ALAP), where m is the number of sequential elements converted to scan cells.

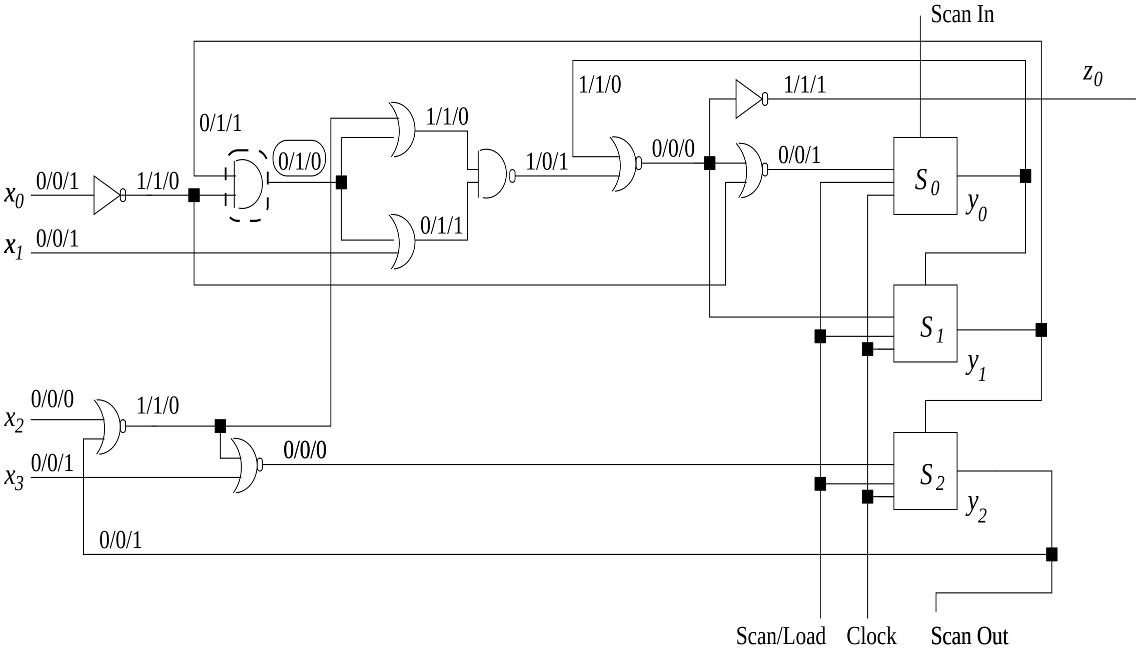


(b) Primary inputs change at t_1

Figure 3.6: Example circuit (*s27* from [23]) illustrating factors which lead to spurious transitions during test application

Having introduced ASAP and ALAP test application strategies the following example shows their shortcomings and the need of a new test application strategy.

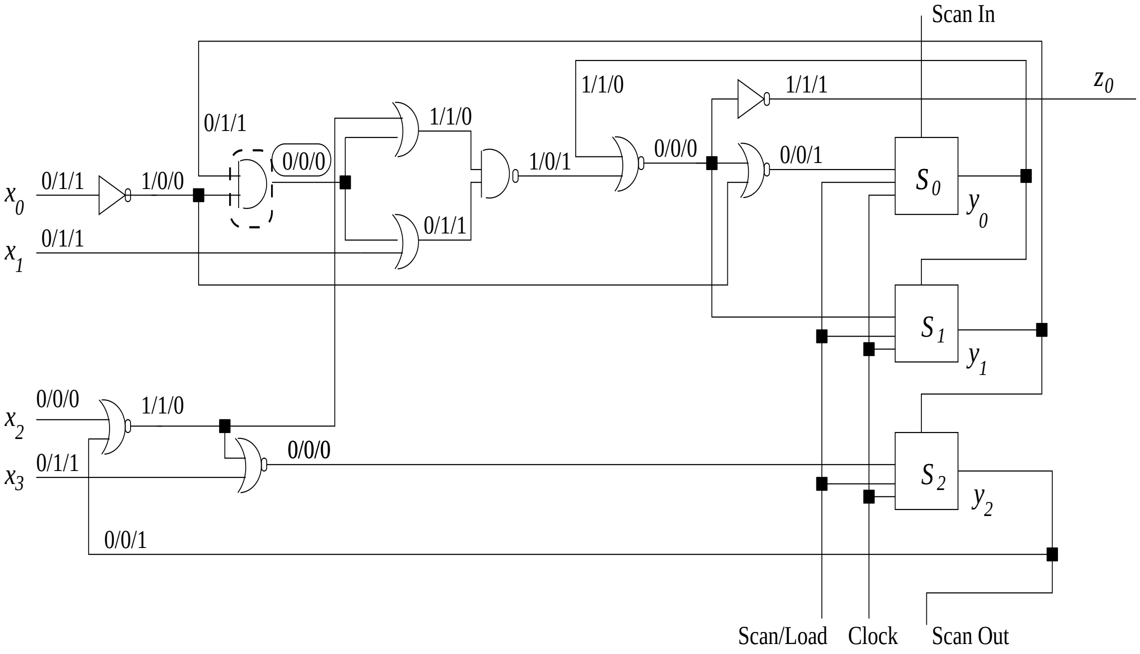
Example 3.3 For the particular example in Figure 3.6, where the number of scan cells is 3, at times t_0, t_1, and t_2 the scan cells are in the shift mode and the values on the input lines of the combinational part of the circuit are irrelevant. The value of primary inputs is important only at t_3 when the entire test vector is applied to the combinational part of the circuit. Therefore, the primary inputs can keep the value of the previous test vector during t_0, t_1, and t_2 without affecting the testing process. To illustrate the importance of primary input change time consider the application of test vector $\{0000000\}$ followed by test vector $\{1101011\}$. The circuit lines are described in terms of three values. For example in Figure 3.6(a), in the case of primary input x_0 the value $0/1/1$ denotes value 0 at t_3 when applying $\{0000000\}$ and value 1 at t_0 and t_1 when shifting in the second test vector $\{1101011\}$. When primary inputs $x_0x_1x_2x_3$ change at t_0 (ASAP test application strategy introduced in Definition 3.2) as shown in Figure 3.6(a) the two marked boxes



(c) Primary inputs change as late as possible (ALAP) at t_3

Figure 3.6: Example circuit (*s27* from [23]) illustrating factors which lead to spurious transitions during test application

illustrate spurious transitions $0/1/0$ and $1/0/1$ at the output of the marked NOR and NOT gate respectively. Since the value of primary inputs is irrelevant during shifting out the test response, if the primary inputs are changed at t_1 the controlling value 1 at the input of the marked NOR gate is preserved at t_1 and no spurious transitions at the output of the marked NOR and NOT gates will occur, as shown in Figure 3.6(b). The primary inputs can keep their value until t_3 when test vector $\{1101011\}$ is applied to the circuit (ALAP test application strategy introduced in Definition 3.3). However, changing the primary inputs at t_3 will not yield the minimum number of transitions as demonstrated in Figures 3.6(c) and 3.6(d) using the same test vectors. In Figure 3.6(c), in the case of primary input x_0 the value $0/0/1$ denotes value 0 at t_1 and t_2 when shifting in $\{1101011\}$ and value 1 at t_3 when applying $\{1101011\}$. When primary inputs $x_0x_1x_2x_3$ are changed at t_3 as shown in Figure 3.6(c) the marked box illustrates a spurious transition $0/1/0$ at the output of the marked AND gate. However if the primary inputs are changed earlier at t_2 the controlling value 0 at the input of the marked AND gate is preserved at t_2 and no spurious transitions at the output of the marked AND gate will occur as shown in Figure 3.6(d).



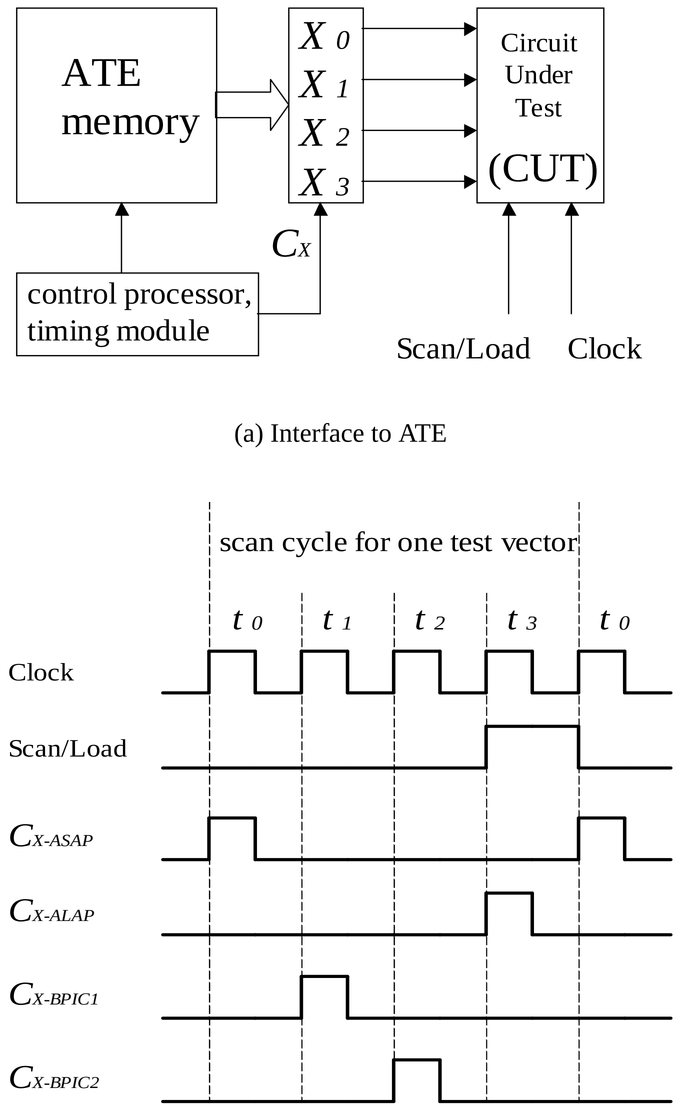
(d) Primary inputs change at t_2

Figure 3.6: Example circuit (*s27* from [23]) illustrating factors which lead to spurious transitions during test application

So far it was shown in Example 3.3 that both ASAP or ALAP test application strategies lead to spurious transitions during shifting in test vectors and shifting out test responses. Now the question is when should the primary inputs change such that the smallest number of spurious transitions occur which leads to lower power dissipation? Before introducing the new test application strategy which reduces spurious transitions during test application the following necessary definition is given.

Definition 3.4 The best primary input change time of test vector V_j is the time when the primary input part x_i of the previous test vector V_i changes to the primary input part x_j of the actual test vector V_j, leading to the smallest value of node transition count during the scan cycle when test vector V_j is applied after test vector V_i.

Finding the best primary input change time will lead to higher correlation between consecutive values on the input lines of the combinational part of the circuit. This leads to minimum value of *NTC* during the scan cycle, and yields savings in power dissipation. Defintion 3.5 is used to introduce the new test application strategy.



(a) Interface to ATE

(b) Different primary input change times

Figure 3.7: Interface to ATE for different test application strategies.

Definition 3.5 The test application strategy where best primary input change time for each test vector V_i, with $i = 0 \ldots n-1$, is determined such that the minimum value of node transition count over the entire test application period is achieved, is referred to as *best primary input change* (BPIC) test application strategy.

To clarify the notation used throughout this chapter, Figure 3.7 shows how different test application strategies are interfaced to ATE (Figure 1.2 in Section 1.2) under the zero

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	y'_2	y'_1	y'_0	NTC
0 (t_0)	V_0	S	0	1	1	0	1	0	0	0	14
1 (t_1)	V_0	S	1	1	1	0	1	1	0	0	10
2 (t_2)	V_0	S	1	1	1	0	1	1	1	0	19
3 (t_3)	V_0	L	-	1	1	0	1	0	0	1	18
4 (t_0)	V_2	S	0	0	0	1	0	0	0	0	15
5 (t_1)	V_2	S	1	0	0	1	0	1	0	0	10
6 (t_2)	V_2	S	0	0	0	1	0	0	1	0	14
7 (t_3)	V_2	L	-	0	0	1	0	1	1	0	19
8 (t_0)	V_3	S	1	0	1	1	1	1	1	1	11
9 (t_1)	V_3	S	1	0	1	1	1	1	1	1	10
10 (t_2)	V_3	S	1	0	1	1	1	1	1	1	6
11 (t_3)	V_3	L	-	0	1	1	1	0	0	0	18
12 (t_0)	V_4	S	0	1	1	0	0	0	0	0	16
13 (t_1)	V_4	S	1	1	1	0	0	1	0	0	10
14 (t_2)	V_4	S	0	1	1	0	0	0	1	0	24
15 (t_3)	V_4	L	-	1	1	0	0	0	1	0	16
16 (t_0)	V_1	S	0	0	0	0	0	0	0	1	18
17 (t_1)	V_1	S	0	0	0	0	0	0	0	0	18
18 (t_2)	V_1	S	0	0	0	0	0	0	0	0	6
19 (t_3)	V_1	L	-	0	0	0	0	0	0	0	6
20 (t_0)	-	S	0	0	0	0	0	0	0	0	6
21 (t_1)	-	S	0	0	0	0	0	0	0	0	6
22 (t_2)	-	S	0	0	0	0	0	0	0	0	6
TOTAL											296

(a) As Soon As Possible (ASAP) test application strategy

Table 3.1: The flow of test data for the circuit in Figure 3.6 during the entire test application period

delay model. The control processor and the timing module initialise the primary input values $x_0x_1x_2x_3$ using the C_X signal, as shown in Figure 3.7(a). Depending on the value of the Scan/Load signal, different primary input change times are chosen by the activation of C_X, as illustrated in Figure 3.7(b). For example, in the case of the ASAP test application strategy, C_{X-ASAP} is active only at t_0. Similarly, the primary inputs change at t_1 by the activation of $C_{X-BPIC1}$ at t_1.

Figures 3.6(a) to 3.6(d) have illustrated the reduction of spurious transitions over a three clock cycles period. The following example gives insight of the proposed technique

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	y'_2	y'_1	y'_0	NTC
0 (t_0)	V_0	S	0	0	0	0	0	0	0	0	6
1 (t_1)	V_0	S	1	0	0	0	0	1	0	0	10
2 (t_2)	V_0	S	1	**1**	**1**	**0**	**1**	1	1	0	17
3 (t_3)	V_0	L	-	1	1	0	1	0	0	1	18
4 (t_0)	V_2	S	0	**0**	**0**	**1**	**0**	0	0	0	15
5 (t_1)	V_2	S	1	0	0	1	0	1	0	0	10
6 (t_2)	V_2	S	0	0	0	1	0	0	1	0	14
7 (t_3)	V_2	L	-	0	0	1	0	1	1	0	19
8 (t_0)	V_3	S	1	**0**	**1**	**1**	**1**	1	1	1	11
9 (t_1)	V_3	S	1	0	1	1	1	1	1	1	10
10 (t_2)	V_3	S	1	0	1	1	1	1	1	1	6
11 (t_3)	V_3	L	-	0	1	1	1	0	0	0	18
12 (t_0)	V_4	S	0	0	1	1	1	0	0	0	10
13 (t_1)	V_4	S	1	0	1	1	1	1	0	0	10
14 (t_2)	V_4	S	0	0	1	1	1	0	1	0	14
15 (t_3)	V_4	L	-	**1**	**1**	**0**	**0**	0	1	0	16
16 (t_0)	V_1	S	0	1	1	0	0	0	0	1	14
17 (t_1)	V_1	S	0	**0**	**0**	**0**	**0**	0	0	0	18
18 (t_2)	V_1	S	0	0	0	0	0	0	0	0	6
19 (t_3)	V_1	L	-	0	0	0	0	0	0	0	6
20 (t_0)	-	S	0	0	0	0	0	0	0	0	6
21 (t_1)	-	S	0	0	0	0	0	0	0	0	6
22 (t_2)	-	S	0	0	0	0	0	0	0	0	6
TOTAL											266

(b) Proposed Best Primary Input Change (BPIC) test application strategy

Table 3.1: The flow of test data for the circuit in Figure 3.6 during the entire test application period

for power dissipation minimisation during the entire test application period when applied to full scan sequential circuits.

Example 3.4 Tables 3.1(a) and 3.1(b) show the flow of test data for the benchmark circuit *s27* of Figure 3.6 for ASAP and the proposed test application strategy respectively. In Table 3.1(a) consider the scan cell order $\{S_2, S_1, S_0\}$ and test vector order $\{V_0, V_2, V_3, V_4, V_1\}$ after simultaneous test vector ordering and scan cell ordering was carried out as shown in Section 3.2. The first column shows the clock cycle index and the second column outlines the test vector which is scanned in during t_0, t_1, t_2 and applied at t_3. The operation type

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	y'_1	y'_2	y'_0	NTC
0 (t_0)	V_1	S	0	**0**	**0**	**0**	**0**	0	0	0	6
1 (t_1)	V_1	S	0	0	0	0	0	0	0	0	6
2 (t_2)	V_1	S	0	0	0	0	0	0	0	0	6
3 (t_3)	V_1	L	-	0	0	0	0	0	0	0	6
4 (t_0)	V_0	S	0	**1**	**1**	**0**	**1**	0	0	0	14
5 (t_1)	V_0	S	1	1	1	0	1	1	0	0	10
6 (t_2)	V_0	S	1	1	1	0	1	1	1	0	10
7 (t_3)	V_0	L	-	1	1	0	1	0	0	1	27
8 (t_0)	V_4	S	0	1	1	0	1	0	0	0	14
9 (t_1)	V_4	S	0	**1**	**1**	**0**	**0**	0	0	0	11
10 (t_2)	V_4	S	1	1	1	0	0	1	0	0	10
11 (t_3)	V_4	L	-	1	1	0	0	1	0	0	6
12 (t_0)	V_3	S	1	1	1	0	0	1	1	0	10
13 (t_1)	V_3	S	1	**0**	**1**	**1**	**1**	1	1	1	16
14 (t_2)	V_3	S	1	0	1	1	1	1	1	1	10
15 (t_3)	V_3	L	-	0	1	1	1	0	0	0	18
16 (t_0)	V_2	S	0	0	1	1	1	0	0	0	10
17 (t_1)	V_2	S	0	0	1	1	1	0	0	0	6
18 (t_2)	V_2	S	1	0	1	1	1	1	0	0	10
19 (t_3)	V_2	L	-	**0**	**0**	**1**	**0**	1	1	0	19
20 (t_0)	-	S	1	0	0	1	0	1	1	1	10
21 (t_1)	-	S	1	0	0	1	0	1	1	1	10
22 (t_2)	-	S	1	0	0	1	0	1	1	1	6
TOTAL											251

(c) Proposed Best Primary Input Change (BPIC) test application strategy combined with simultaneous scan cell and test vector ordering

Table 3.1: The flow of test data for the circuit in Figure 3.6 during the entire test application period

(**S**can or **L**oad) is shown in column 3. In the case of a **S**can operation the fourth column gives the *ScanIn* value. Columns 5-8 show the values of primary inputs $x_0x_1x_2x_3$ and the columns 9-11 show the next state values $y'_2y'_1y'_0$. The last column shows the value of the node transition count NTC for each clock cycle. The NTC is calculated as follows. In clock cycle (i) the NTC in the combinational part is computed by considering the primary inputs of clock cycle (i) and the next state values of clock cycle ($i-1$), which are present state values at clock cycle (i). The NTC in the sequential part is the sum of NTC of each scan cell by scanning/loading the next state values $y'_2y'_1y'_0$ in clock cycle (i), using the values of $N_{SC_{min}}$ and $N_{SC_{max}}$ outlined in Section 3.2. Initially all the primary and scan inputs are set to 0 and the node transition count over the entire test application period under the ASAP test application strategy is $NTC = 296$. This value can be reduced if spurious transitions during shifting in test vectors and shifting out responses are avoided by modifying the primary input change time. The change of primary input part of test vector V_i at time t_j is indicated by $t_{V_i} = t_j$. If the primary input change times are set to $t_{V_0} = t_2$, $t_{V_2} = t_0$, $t_{V_3} = t_0$, $t_{V_4} = t_3$, and $t_{V_1} = t_1$ as shown in the marked boxes of Table 3.1(b), the node transition count reduces to $NTC = 266$. The reason for reducing the number of transitions is the increased correlation between consecutive values of primary and pseudo inputs during t_0, t_1, t_2, when test vectors are scanned in and test responses are scanned out. For example by changing the primary inputs of t_{V_4} at t_3 the NTC in clock cycles 12 and 14 reduces from 16 and 24 respectively in the case of ASAP (Table 3.1(a)) to 10 and 14 respectively in the case of BPIC (Table 3.1(b)). Note that in clock cycles when test responses are loaded in scan cells (**L** in column 3), the correct test response values from Table 3.1(a) are preserved. When combining primary input change time with simultaneous scan cell ordering and test vector ordering further improvements are achieved. For test vector order $\{V_1, V_0, V_4, V_3, V_2\}$, scan cell order $\{S_1, S_2, S_0\}$ and primary input change times set at $t_{V_1} = t_0$, $t_{V_0} = t_0$, $t_{V_4} = t_1$, $t_{V_3} = t_1$, and $t_{V_2} = t_3$, it is shown that the new value of node transition count is reduced further to $NTC = 251$ (Table 3.1(c)). This highlights the importance of combining the best primary input change time with simultaneous scan cell and test vector ordering for NTC reduction.

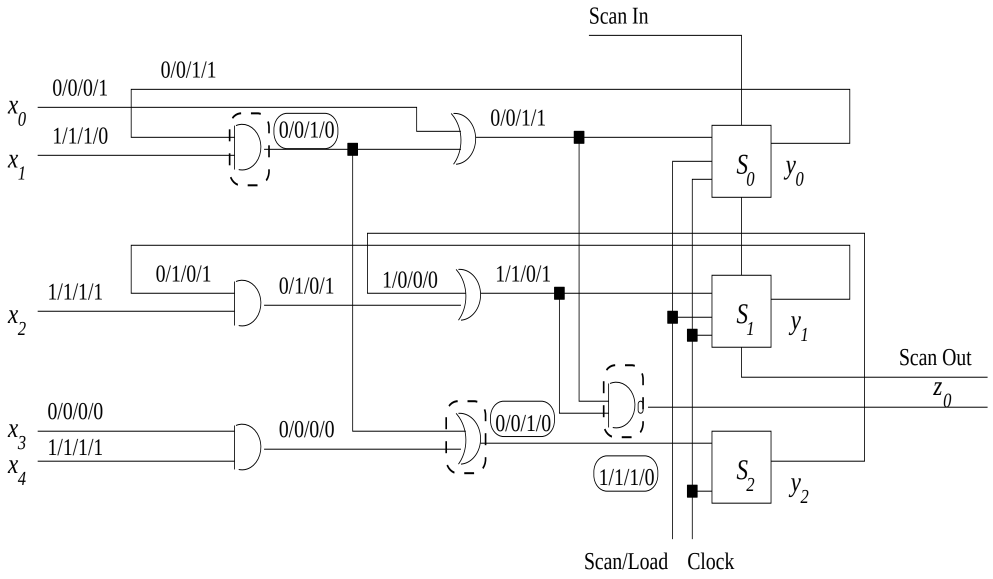
Summary: The previous Example 3.4 has highlighted the importance of combining primary input change time with scan cell and test vector ordering for NTC reduction in full scan sequential circuits. In the case of the ASAP test application strategy, the node transition count is $NTC = 296$ over the entire test application period (Table 3.1(a)). When applying the proposed BPIC test application strategy node transition count is reduced to $NTC = 266$ (Table 3.1(b)). To achieve maximum reduction in node transition count, the proposed BPIC test application strategy is combined with scan cell and test vector ordering leading to $NTC = 251$ (Table 3.1(c)). Computing the best primary input change time for every test vector is described in the algorithm *BPIC-ALG* from Section 3.4.1.

3.3.2 The Applicability of the New Test Application Strategy to Partial Scan Sequential Circuits

Having introduced the new BPIC test application strategy for full scan sequential circuits, this section shows through two detailed examples that the BPIC test application strategy is applicable to partial scan sequential circuits. The importance of combining the proposed test application strategy with scan cell ordering is outlined.

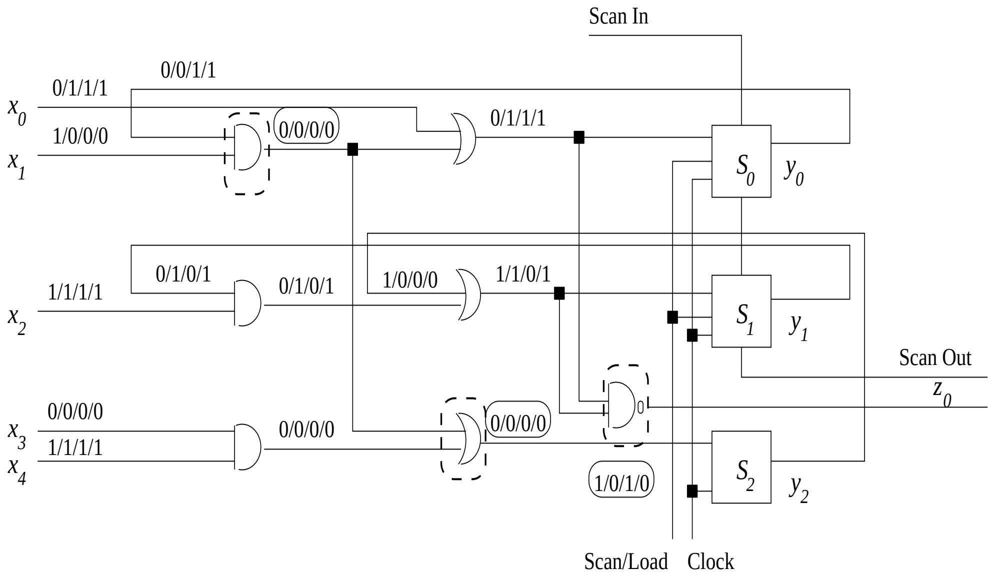
So far it was assumed that the changing time of the primary inputs $x_0x_1x_2x_3x_4$ of circuit shown in Figure 3.4 (Example 3.2 from Section 3.2.3) occurs at clock cycle t_2. To illustrate the importance of primary input change time on the number of spurious transitions in *partial* scan sequential circuits consider the following example.

Example 3.5 Consider the application of test vector V_3={0110100} followed by test vector V_4={1010111} to the circuit of Figure 3.8. The circuit data lines are described in terms of four values. For example in Figure 3.8(a), in the case of primary input x_0 the value $0/0/0/1$ denotes value 0 at t_2 when applying V_3, value 0 at t_0 and t_1 respectively when shifting in the present state part of the second test vector V_4, and value 1 at t_0 when apply-



(a) Primary inputs change as late as possible (ALAP) at t_2

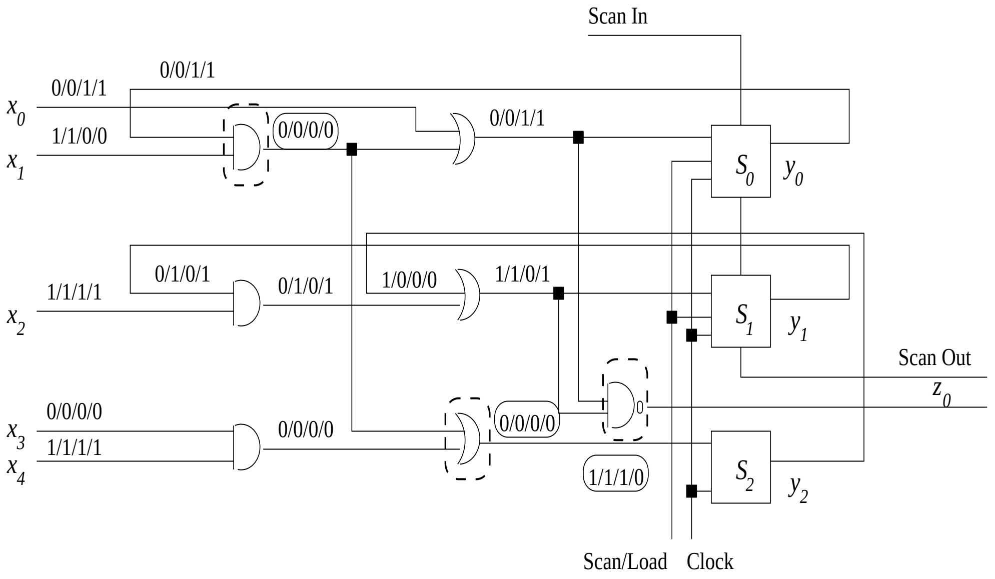
Figure 3.8: Example partial scan sequential circuit illustrating the effect of primary input change time on the reduction of spurious transitions during test application



(b) Primary inputs change as soon as possible (ASAP) at t_0

Figure 3.8: Example partial scan sequential circuit illustrating the effect of primary input change time on the reduction of spurious transitions during test application

ing V_4. When primary inputs $x_0x_1x_2x_3x_4$ change at t_2 as shown in Figure 3.8(a) the two marked boxes illustrate the spurious transition $0/0/1/0$ at the output of the marked AND gate which further propagates at the output of the marked OR gate. However the value of primary inputs is irrelevant during shifting out the test response. Thus, the primary inputs can be changed as early as t_0 after test vector V_3 is applied to the circuit under test. When primary inputs $x_0x_1x_2x_3x_4$ change at t_0 as shown in Figure 3.8(b) the controlling value 0 at the input of the marked AND gate is preserved at t_1 and no spurious transitions at the output of the marked AND and OR gates will occur. However, changing the primary inputs at t_0 does not yield the minimum value of node transition count. The marked box in Figure 3.8(b) illustrates a spurious transition $1/0/1/0$ at the output of the marked NAND gate. The value of $NTC = 41$ over the scan cycle period t_0, t_1 and t_2 in the case of ALAP test application strategy is reduced to $NTC = 37$ in the case ASAP test application strategy. However, both ALAP and ASAP test application strategies fail to achieve the minimum NTC. If the primary inputs change at clock cycle t_1 the controlling value 0 at the input of the marked NAND gate is preserved at t_0 and no spurious transitions at the output of the marked NAND gate will occur, as shown in Figure 3.8(c). Furthermore, the



(c) Primary inputs change at t_1

Figure 3.8: Example partial scan sequential circuit illustrating the effect of primary input change time on the reduction of spurious transitions during test application

controlling value 0 at the input of the marked AND gate is preserved at t_1 and no spurious transitions at the output of the marked AND and OR gates will occur, as shown in the marked boxes in Figure 3.8(c). Thus, the minimum value of $NTC = 35$ is achieved when primary input change time is set to t_1.

Figures 3.8(a)-3.8(c) have illustrated the reduction of spurious transitions during a four clock cycles period. Now, to give insight of the proposed BPIC test application strategy during the entire test application period in partial sequential circuits, consider the following example.

Example 3.6 To outline the advantage of controlling primary input change time of each test vector, Tables 3.2(a) and 3.2(b) show the flow of test data for the circuit of Figure 3.8 for ALAP and BPIC test application strategy respectively, during the entire test application period. The first column shows the clock cycle index and the second column outlines the test vector which is scanned in during t_0, t_1 and applied at t_2. The operation type (**S**can or **L**oad) is shown in the third column. In the case of a **S**can operation the fourth column gives the value on scan input line $ScanIn$. Columns 5-9 show the values of primary inputs

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	x_4	y'_0	y'_1	y'_2	NTC
0 (t_0)	V_0	S	0	0	0	0	0	0	0	0	0	4
1 (t_1)	V_0	S	1	0	0	0	0	0	1	0	0	8
2 (t_2)	V_0	L	-	1	0	1	1	1	1	0	1	12
3 (t_0)	V_1	S	0	1	0	1	1	1	0	1	1	15
4 (t_1)	V_1	S	1	1	0	1	1	1	1	0	1	13
5 (t_2)	V_1	L	-	0	0	0	1	0	0	1	0	22
6 (t_0)	V_2	S	0	0	0	0	1	0	0	0	0	10
7 (t_1)	V_2	S	1	0	0	0	1	0	1	0	0	8
8 (t_2)	V_2	L	-	0	1	1	1	0	1	0	1	13
9 (t_0)	V_3	S	0	0	1	1	1	0	0	1	1	15
10 (t_1)	V_3	S	0	0	1	1	1	0	0	0	1	15
11 (t_2)	V_3	L	-	0	1	1	0	1	0	1	0	13
12 (t_0)	V_4	S	1	0	1	1	0	1	1	0	0	13
13 (t_1)	V_4	S	1	0	1	1	0	1	1	1	0	16
14 (t_2)	V_4	L	-	1	0	1	0	1	1	1	0	12
15 (t_0)	V_5	S	1	1	0	1	0	1	1	1	0	4
16 (t_1)	V_5	S	0	1	0	1	0	1	0	1	0	8
17 (t_2)	V_5	L	-	0	1	0	0	1	0	0	0	15
18 (t_0)	-	S	0	0	1	0	0	1	0	0	0	4
19 (t_1)	-	S	0	0	1	0	0	1	0	0	0	4
TOTAL												224

(a) As Late As Possible (ALAP) test application strategy

Table 3.2: Flow of test data for the circuit in Figure 3.8 during the entire test application period for ALAP and BPIC test application strategies and its effect on NTC

$x_0x_1x_2x_3x_4$ and the columns 10-12 show the next state values $y'_0y'_1y'_2$. The last column 13 shows the value of NTC for each clock cycle. The NTC is calculated as follows. In clock cycle (i) the NTC in the combinational part is computed by considering the primary inputs of clock cycle (i) and the next state values at clock cycle ($i-1$), which are present state values at clock cycle (i). The NTC in the sequential part is the sum of NTC of each cell (scan cells S_0 and S_1 and non-scan cell S_2) by scanning/loading the next state values $y'_0y'_1y'_2$ in clock cycle (i), using the values of $N_{SC_{min}}$, $N_{SC_{max}}$, $N_{NSC_{min}}$, and $N_{NSC_{max}}$ outlined in Section 3.2. Note that when shifting out test responses the non-scan cell S_2 is not clocked and therefore no transitions occur. Initially all the primary inputs and present state lines are considered 0 and the node transition count over the entire test application

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	x_4	y'_0	y'_1	y'_2	NTC
0 (t_0)	V_0	S	0	**1**	**0**	**1**	**1**	**1**	0	0	0	8
1 (t_1)	V_0	S	1	1	0	1	1	1	1	0	0	8
2 (t_2)	V_0	L	-	1	0	1	1	1	1	0	1	8
3 (t_0)	V_1	S	0	**0**	**0**	**0**	**1**	**0**	0	1	1	18
4 (t_1)	V_1	S	1	0	0	0	1	0	1	0	1	12
5 (t_2)	V_1	L	-	0	0	0	1	0	0	1	0	16
6 (t_0)	V_2	S	0	0	0	0	1	0	0	0	0	10
7 (t_1)	V_2	S	1	**0**	**1**	**1**	**1**	**0**	1	0	0	8
8 (t_2)	V_2	L	-	0	1	1	1	0	1	0	1	13
9 (t_0)	V_3	S	0	**0**	**1**	**1**	**0**	**1**	0	1	1	15
10 (t_1)	V_3	S	0	0	1	1	0	1	0	0	1	15
11 (t_2)	V_3	L	-	0	1	1	0	1	0	1	0	13
12 (t_0)	V_4	S	1	0	1	1	0	1	1	0	0	13
13 (t_1)	V_4	S	1	**1**	**0**	**1**	**0**	**1**	1	1	0	13
14 (t_2)	V_4	L	-	1	0	1	0	1	1	1	0	9
15 (t_0)	V_5	S	1	1	0	1	0	1	1	1	0	4
16 (t_1)	V_5	S	0	1	0	1	0	1	0	1	0	8
17 (t_2)	V_5	L	-	**0**	**1**	**0**	**0**	**1**	0	0	0	15
18 (t_0)	-	S	0	0	1	0	0	1	0	0	0	4
19 (t_1)	-	S	0	0	1	0	0	1	0	0	0	4
TOTAL												214

(b) Proposed Best Primary Input Change (BPIC) test application strategy

Table 3.2: Flow of test data for the circuit in Figure 3.8 during the entire test application period for ALAP and BPIC test application strategies and its effect on NTC

period under the ALAP test application strategy is $NTC = 224$. This value can be reduced if spurious transitions are avoided by determining best primary input change time for each test vector. If the primary input change times are set to $t_{V_0} = t_0$, $t_{V_1} = t_0$, $t_{V_2} = t_1$, $t_{V_3} = t_0$, $t_{V_4} = t_1$, and $t_{V_5} = t_2$ as shown in the marked boxes of Table 3.2(b), the node transition count reduces to $NTC = 214$. The reason for reducing NTC is the increased correlation between consecutive values of primary inputs and present state lines. For example by changing test vector V_4 at t_1 ($t_{V_4} = t_1$) the NTC in clock cycles 13 and 14 reduces from 16 and 12 respectively in the case of ALAP (Table 3.2(a)) to 13 and 9 respectively in the case of BPIC (Table 3.2(b)). It should be noted that for the particular circuit of Figure 3.8 the value of $NTC = 214$ when applying the proposed BPIC test application strategy

Cycle	Vector	Op	SI	x_0	x_1	x_2	x_3	x_4	y'_1	y'_0	y'_2	NTC
0 (t_0)	V_0	S	1	0	0	0	0	0	1	0	0	8
1 (t_1)	V_0	S	0	0	0	0	0	0	0	1	0	12
2 (t_2)	V_0	L	-	**1**	**0**	**1**	**1**	**1**	0	1	1	12
3 (t_0)	V_1	S	1	**0**	**0**	**0**	**1**	**0**	1	0	1	18
4 (t_1)	V_1	S	0	0	0	0	1	0	0	1	1	12
5 (t_2)	V_1	L	-	0	0	0	1	0	1	0	0	16
6 (t_0)	V_2	S	1	0	0	0	1	0	1	1	0	10
7 (t_1)	V_2	S	0	0	0	0	1	0	0	1	0	8
8 (t_2)	V_2	L	-	**0**	**1**	**1**	**1**	**0**	0	1	1	13
9 (t_0)	V_3	S	0	**0**	**1**	**1**	**0**	**1**	0	0	1	11
10 (t_1)	V_3	S	0	0	1	1	0	1	0	0	1	10
11 (t_2)	V_3	L	-	0	1	1	0	1	1	0	0	12
12 (t_0)	V_4	S	1	**1**	**0**	**1**	**0**	**1**	1	1	0	12
13 (t_1)	V_4	S	1	1	0	1	0	1	1	1	0	4
14 (t_2)	V_4	L	-	1	0	1	0	1	1	1	0	5
15 (t_0)	V_5	S	0	1	0	1	0	1	0	1	0	8
16 (t_1)	V_5	S	1	1	0	1	0	1	1	0	0	16
17 (t_2)	V_5	L	-	**0**	**1**	**0**	**0**	**1**	0	0	0	11
18 (t_0)	-	S	0	0	1	0	0	1	0	0	0	4
19 (t_1)	-	S	0	0	1	0	0	1	0	0	0	4
TOTAL												206

(c) Proposed Best Primary Input Change (BPIC) test application strategy combined with scan cell ordering

Table 3.2: Flow of test data for the circuit in Figure 3.8 during the entire test application period for ALAP and BPIC test application strategies and its effect on NTC

by itself is better than when applying scan cell ordering by itself ($NTC = 216$ as shown in Example 3.2). When combining BPIC test application strategy with scan cell ordering further improvements are achieved. For scan cell order $\{S_1, S_0\}$ and primary input change times set at $t_{V_0} = t_2$, $t_{V_1} = t_0$, $t_{V_2} = t_2$, $t_{V_3} = t_0$, $t_{V_4} = t_0$, and $t_{V_5} = t_2$, it is shown that the new value of node transition count is further reduced to $NTC = 206$ (Table 3.2(c).

Summary: The previous Example 3.6 has highlighted the importance of combining primary input change time with scan cell ordering for NTC reduction in partial scan sequential circuits. To achieve maximum reduction in node transition count the proposed BPIC test application strategy is combined with scan cell ordering leading to $NTC = 206$ (Table

3.2(c) from Example 3.6). Note that the proposed BPIC test application strategy which is equally applicable to both full and partial scan sequential circuits depends only on controlling primary input change time, and hence does not require extra DFT hardware. This means that power dissipation is minimised without an increase in test area or performance. Furthermore, since no extra test data is necessary and the complete test vector computed by the ATPG tool is applied to the circuit under test irrespective of the primary input change time, the proposed test application strategy does not decrease test efficiency and no penalty in test application time or volume of test data is added. Unlike the case of extra primary input vectors [88, 189], the proposed BPIC test application strategy minimises power dissipation with no penalty in test area, performance, test efficiency, test application time or volume of test data. Furthermore, in the case of the proposed BPIC test application strategy the computational time is low since the algorithm for finding the best primary input change is polynomial and can be used for computing the cost function in the design space exploration as explained in Section 3.4.

3.3.3 Extention of the New BPIC Test Application Strategy to Scan BIST Methodology

So far the proposed BPIC test application strategy (Definition 3.5) was applied to full and partial scan sequential circuits using external automatic test equipment ATE (Figure 1.2 from Section 1.2). This can be summarised in Figure 3.9 where the best primary input change time k is highlighted. However, the proposed BPIC test application strategy is not applicable only to standard full and partial scan sequential circuits using external ATE. In the following the minor modifications which need to be considered when using scan BIST methodology (Figure 1.5 from Section 1.3.1) are outlined. Figure 3.10 shows that the serial output of the linear feedback shift register (LFSR) is fed directly into the scan chain and the primary inputs are directly controllable. Therefore, primary inputs can be changed at the best primary input change time which is calculated in the same way as for full scan and partial scan sequential circuits as described in the following Section 3.4. This will lead to a lower area overhead associated with scan BIST methodology (Figure 1.5 of Section 1.3.1) at the expense of higher interference from ATE which needs to store the primary input part of each test vector.

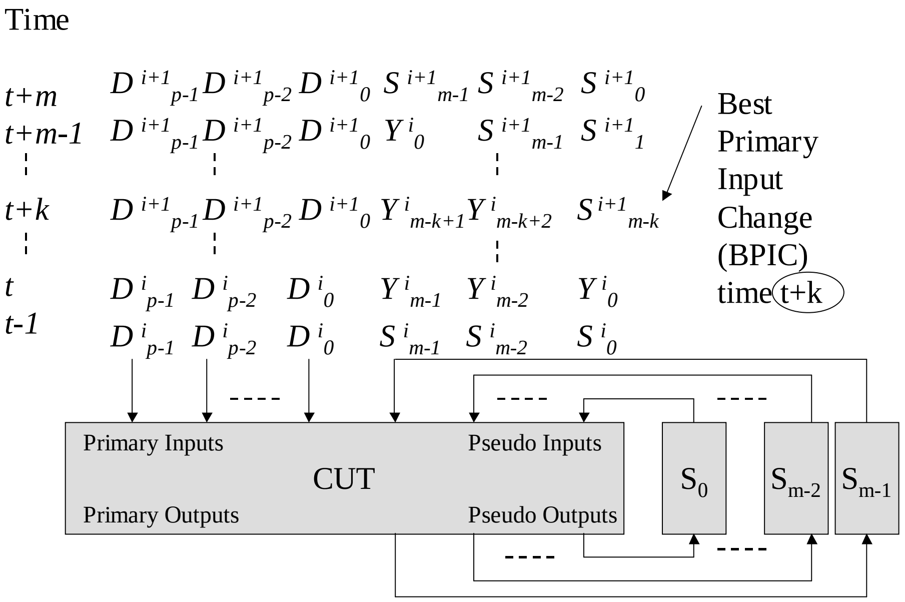

Time
t+m
t+m-1
t+k
t
t-1
D^{i+1}_{p-1} D^{i+1}_{p-2} D^{i+1}_{0} S^{i+1}_{m-1} S^{i+1}_{m-2} S^{i+1}_{0}
D^{i+1}_{p-1} D^{i+1}_{p-2} D^{i+1}_{0} Y^{i}_{0} S^{i+1}_{m-1} S^{i+1}_{1}
D^{i+1}_{p-1} D^{i+1}_{p-2} D^{i+1}_{0} Y^{i}_{m-k+1} Y^{i}_{m-k+2} S^{i+1}_{m-k}
D^{i}_{p-1} D^{i}_{p-2} D^{i}_{0} Y^{i}_{m-1} Y^{i}_{m-2} Y^{i}_{0}
D^{i}_{p-1} D^{i}_{p-2} D^{i}_{0} S^{i}_{m-1} S^{i}_{m-2} S^{i}_{0}
Best Primary Input Change (BPIC) time t+k
Primary Inputs
Pseudo Inputs
CUT
Primary Outputs
Pseudo Outputs
S_0
S_{m-2}
S_{m-1}

Figure 3.9: Summary of the proposed BPIC test application strategy when employing standard scan DFT using external ATE

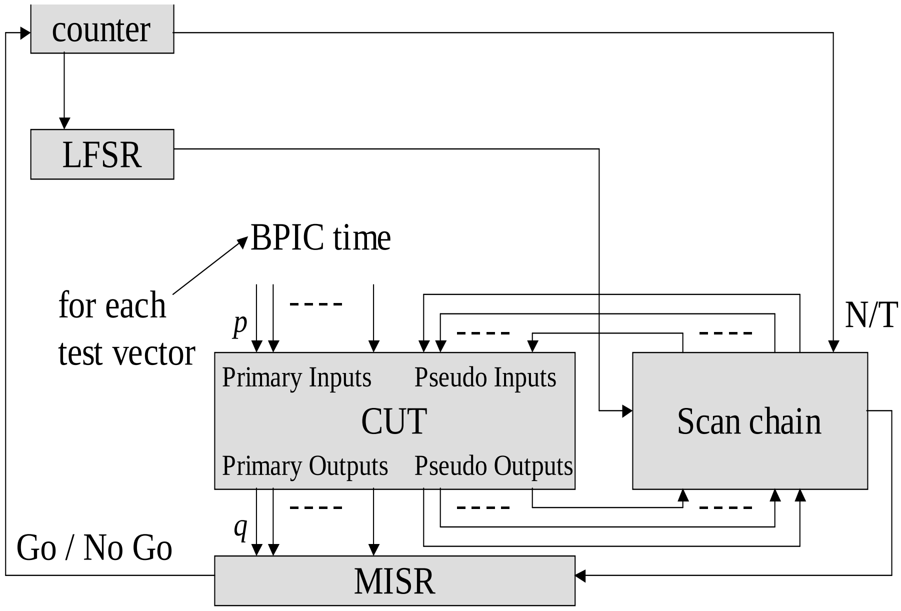


Figure 3.10: Extention of the proposed BPIC test application strategy to scan BIST methodology

3.4 Novel Algorithms for Minimising Power Dissipation During Test Application

Having described in Section 3.3 the new best primary input change (BPIC) test application strategy (Definition 3.5), now algorithms which compute best primary input change times used by BPIC test application strategy, are considered. Section 3.4.1 introduces a new and exact algorithm which computes best primary input change time for each test vector with respect to a given test vector and scan cell order. Section 3.4.2 shows how combining the proposed test application strategy with the recently introduced scan cell and test vector ordering using a simulated annealing-based design space exploration leads to further reductions in power dissipation during test application.

3.4.1 Best Primary Input Change (*BPIC*) Algorithm

Spurious transitions (Definition 4.1) induced by fixed primary input changes as outlined in Section 3.3 are solved by changing the primary inputs of each test vector such that the minimum number of transitions is achieved. For a given scan cell order with m scan cells, the total number of primary input change times is $(m+1)$. Considering n test vectors, in a given test vector order, the total number of configurations of primary input changing for all the test vectors is $(m+1)^n$. Best Primary Input Change Algorithm (*BPIC-ALG*) computes the best primary input change time for each test vector for a given scan cell order and test vector order. Figure 3.11 illustrates the pseudocode of the proposed *BPIC-ALG* algorithm. The function accepts as input, a test set **S** and a circuit **C**. The outer loop represents the traversal of all the test vectors from test set **S**. All the $m+1$ primary input change times for test vector V_i are then considered in the inner loop. For each primary input change time t_j, circuit **C** is simulated and the node transition count $NTC_{i,j}$ is registered. After the completion of the inner loop the best primary input change time t_{B_i}, for which NTC_{i,B_i} is minimum, is retained and the outer loop continues until the entire test set is examined. The algorithm computes the best solution in a computational time which is polynomial in the number of test vectors n, the number of scan cells m, and the circuit size $|\mathbf{C}|$. It should be noted that *BPIC-ALG* is *test set dependent* and hence it is applicable only to *to small to medium sized* sequential circuits as outlined in Section 3.1. Despite the reductions which are achieved by the *BPIC-ALG* algorithm as shown in

```
ALGORITHM: BPIC-ALG
INPUT: Test Set S, Circuit C
OUTPUT: Best primary input change times $\{\mathbf{t_{B_0}}, \mathbf{t_{B_1}}, \ldots, \mathbf{t_{B_{n-1}}}\}$
        Node transition count over the entire test application period NTC

NTC $\leftarrow$ 0
for every test vector $V_i$ from S with $i = 0, \ldots, n-1$ {
    for every primary change time $t_{V_i} = t_j$ with $j = 0, \ldots, m$
        compute $NTC_{i,j}$ by simulating C during the scan cycle when
        applying $V_i$ using the scan cell order $\{\mathbf{S_0}, \ldots, \mathbf{S_{m-1}}\}$
    get best primary input change time $\mathbf{t_{B_i}}$ for test vector $V_i$
    such that $NTC_{i,B_i}$ is minimum
    NTC $\leftarrow$ NTC $+ NTC_{i,B_i}$
}
return $\{\mathbf{t_{B_0}}, \mathbf{t_{B_1}}, \ldots, \mathbf{t_{B_{n-1}}}\}$, NTC
```

Figure 3.11: Proposed *BPIC-ALG* algorithm for determining the Best Primary Input Change time for each test vector

Section 3.5, all the factors accountable for power dissipation during test application must be combined for achieving best results, as described in the following section.

3.4.2 Simulated Annealing-Based Design Space Exploration

High power dissipation problems caused by an inadequate test vector ordering and scan cell ordering for full scan sequential circuits are solved by an simulated annealing algorithm which can escape local minima. Since test vector ordering is not applicable to partial scan as outlined in Section 3.2.3 the simulated annealing algorithm for partial scan sequential circuits uses only scan cell ordering. For a test set which consists of n test vectors there are $n!$ test vector orderings. Furthermore for each test vector ordering there are $m!$ scan cell orderings, where m is the number of scan cells. Finding the optimum test vector and scan cell order is NP-hard [49, 58]. The total complexity of the design space, defined by the set of scan cell and test vector orderings, is $n! \times m!$ which even for small design problems with 15 test vectors and 15 scan cells is computationally expensive.

Figure 3.12 illustrates the basic steps of simulated annealing-based optimisation. The optimisation function accepts as input a test set **S** and a circuit **C** which are set to initial configuration S_{INIT} and C_{INIT} respectively. The calculation of the initial control parameter value [52] is based on the assumption that a sufficiently large number of generated solutions (for example 95%) should be accepted at the beginning of the annealing process. The outer loop modifies the control parameter of the simulated annealing algorithm which is gradually lowered as the annealing process proceeds. Within the inner loop a new sequence of solutions is generated at a constant control parameter value. The length of a sequence of solutions is set to 20. The control parameter is decreased in such a way that the stationary distributions at the end of the sequences of solutions are close to each other. By evaluating information about the cost distribution within each sequence of solutions, a fast decrease of the control parameter is given according to the cooling schedule from [52]. Each new solution is generated using one of the following:

- randomly choose two scan cells S_i and S_j from the actual scan cell order $\{S_0,\ldots,S_i,\ldots,S_j,\ldots,S_{m-1}\}$ and exchange their position generating a new scan cell order $\{S_0,\ldots,S_j,\ldots,S_i,\ldots,S_{m-1}\}$, where m is the number of scan cells in the circuit.
- randomly choose two test vectors V_a and V_b from the actual test vector order $\{V_0,\ldots,V_a,\ldots,V_b,\ldots,V_{n-1}\}$ and exchange their position generating a new test vector order $\{V_0,\ldots,V_b,\ldots,V_a,\ldots,V_{n-1}\}$, where n is the number of test vectors in the test set.

The alternative application of exchanges between randomly chosen test vectors and scan cells proves to be efficient in exploring the discrete design space. It should be noted that for partial scan sequential circuits the exchanges between test vectors are prohibited and *SA-Optimisation* from Figure 3.12 is not executing line 6. For each new solution the proposed best primary input change algorithm *BPIC-ALG*(S_{NEW},C_{NEW}) is called to determine the best primary input change times for all the test vectors from S_{NEW} according to the scan cell order in C_{NEW}. New solutions are either accepted or rejected depending on the acceptance criterion defined in the simulated annealing algorithm [98]. If the best solution so far is reached then it is saved in $\{S_{BEST},C_{BEST}\}$ which is returned together with best primary input change times at the end of the optimisation process. The optimisation process is terminated after the variation in the average cost over a selected number of sequences of solutions falls bellow a given value as described in [52]. It should be noted that all the factors accountable for power dissipation during test application as described in Sections 3.2 and 3.3 are included in the optimisation process.

```
ALGORITHM: SA-Optimisation
INPUT: Test Set S, Circuit C
OUTPUT: Best found test vector order $\{\mathbf{V_{B_0}}, \ldots, \mathbf{V_{B_{m-1}}}\}$
        Best found scan cell order $\{\mathbf{S_{B_0}}, \ldots, \mathbf{S_{B_{m-1}}}\}$
        Best primary input change times $\{\mathbf{t_{B_0}}, \mathbf{t_{B_1}}, \ldots, \mathbf{t_{B_{n-1}}}\}$

Current test vector order is set to the initial test vector order $S_{INIT}$ of S
$\{V_{C_0}, \ldots, V_{C_{n-1}}\} \leftarrow \{V_{I_0}, \ldots, V_{I_{n-1}}\}$
Current scan cell order is set to the initial scan cell order $C_{INIT}$ of C
$\{S_{C_0}, \ldots, S_{C_{m-1}}\} \leftarrow \{S_{I_0}, \ldots, S_{I_{m-1}}\}$
repeat
    repeat
        Generate a new scan cell order $C_{NEW} = \{S_{N_0}, \ldots, S_{N_{m-1}}\}$ by swapping
        the positions of randomly chosen scan cells $S_{C_i}$ and $S_{C_j}$
        Generate a new test vector order $S_{NEW} = \{t_{N_0}, \ldots, t_{N_{n-1}}\}$ by swapping
        the positions of randomly chosen test vectors $t_{C_a}$ and $t_{C_b}$
        compute $NTC$ of the new solution and best primary input change
        times $\{t_{N_0}, \ldots, t_{N_{n-1}}\}$ using BPIC-ALG(S,C,$\{S_{N_0}, \ldots, S_{N_{m-1}}\}$)
        if accepted according to the SA acceptance criterion [98] then
                $\{S_{C_0}, \ldots, S_{C_{m-1}}\} \leftarrow \{S_{N_0}, \ldots, S_{N_{m-1}}\}$
        if the best solution so far is reached then {
                update $S_{BEST}$ and $C_{BEST}$ with $S_{NEW}$ and $C_{NEW}$
                $\{\mathbf{S_{B_0}}, \ldots, \mathbf{S_{B_{m-1}}}\} \leftarrow \{S_{N_0}, \ldots, S_{N_{m-1}}\}$
                $\{\mathbf{V_{B_0}}, \ldots, \mathbf{V_{B_{m-1}}}\} \leftarrow \{V_{N_0}, \ldots, V_{N_{m-1}}\}$
                $\{\mathbf{t_{B_0}}, \ldots, \mathbf{t_{B_{n-1}}}\} \leftarrow \{t_{N_0}, \ldots, t_{N_{n-1}}\}$
        }
    until the number of solutions equals the solution sequence length
    decrease the control parameter value according to the
    cooling schedule from [52]
until the system is frozen
return $\{\mathbf{V_{B_0}}, \ldots, \mathbf{V_{B_{m-1}}}\}$ $\{\mathbf{S_{B_0}}, \ldots, \mathbf{S_{B_{m-1}}}\}$ $\{\mathbf{t_{B_0}}, \ldots, \mathbf{t_{B_{n-1}}}\}$
```

Figure 3.12: Proposed simulated annealing-based design space exploration for combining test vector and scan cell ordering and the proposed best primary input change (BPIC) test application strategy

It is important to note that the size of the design space for partial scan sequential circuits is $m!$, and it is significantly smaller than the size of the design space for for full scan sequential circuits $m! \times n!$, where m is the number of scan cells and n is the number of test vectors. This is due to the fault activation and the fault-effect propagation sequences through non-scan cells which prohibit test vector ordering in the case of partial scan sequential circuits. While previous algorithms with no penalty in test area, performance, test efficiency, test application time or volume of test data, presented in [49] use a *simple greedy heuristic* for asymmetric traveling salesman problem to find a *sub-optimal* test vector ordering for a given scan cell ordering, which is NP-hard, the proposed *BPIC-ALG* is an *exact* algorithm which always returns the *best (optimal)* primary input change times in *polynomial time* for a given scan cell ordering. However, when considering test vector and scan cell ordering the optimisation process for the discrete, degenerate and highly irregular design space makes the problem tractable only for small to medium sized scan sequential circuits as shown in Section 3.5.

3.5 Experimental Results

Experimental results are divided in three separate sections. Section 3.5.1 gives the results for a number of full scan sequential circuits. Section 3.5.2 outlines the experimental results for partial scan sequential circuits, and Section 3.5.3 highlights further benefits of partial scan in minimising power dissipation during test application.

3.5.1 Experimental Results for Full Scan Sequential Circuits

This section demonstrates through a set of benchmark examples that the proposed BPIC test application strategy yields savings in power dissipation during test application in **full** scan sequential circuits. Furthermore, the savings can be substantially improved when BPIC is integrated with test vector ordering and scan cell ordering as described in the algorithm in Section 3.4.2. The *BPIC-ALG* and *SA-Optimisation* algorithms described in Section 3.4 were implemented within the framework of a low power testing system on a 350 MHz Pentium II PC with 64 MB RAM running Linux and using GNU CC version 2.7.

The average value of node transition count (NTC) reported throughout this section, is calculated under the assumption of the zero delay model using Equation 3.1 from Section 3.2.1. The use of zero delay model is motivated by very rapid computation of NTC required by the algorithms from Section 3.4, and by the observation that power dissipation under the zero delay model has a high correlation with power dissipation under the real delay model [176]. Besides, the aim of this chapter is not to give exact values of power dissipation during test application, but to *validate* the new BPIC test application strategy (Definition 3.5) for power minimisation that equally applies to every delay model. Further, although the power due to glitches is neglected in Equation 3.1, the zero delay model provides *reliable relative power information* that is reported throughout this experimental section. *Reliable relative power information* provided by NTC using the zero delay model means that savings in NTC, and savings in power dissipation obtained after technology mapping the circuit and accounting for glitching activity during test application, are within the same range. Therefore, to validate that experimental results using NTC reported throughout this section provide *reliable relative power information* appendix B shows that the savings in the case of the real delay model that accounts for glitching activity are even higher than the savings in the case of the zero delay model. This conclusion was reached after technology mapping circuit *s344* [23] in AMS 0.35 micron technology [9], and using state of the art real delay model simulator [130] with power and timing information [9] (appendix B). Furthermore, this result can also be explained by the fact that by eliminating spurious transitions (Definition 3.1) the propagation of hazards and glitches is also eliminated leading to even greater reductions in power dissipation in the case of the real delay model [87].

Table 3.3 shows the results when the BPIC test application strategy is applied by itself (i.e. without scan cell and test vector ordering) for 24 commonly accepted ISCAS89 benchmark circuits [23]. The first and second columns give the circuit name and the number of scan cells (SC) respectively. The third column gives the number of test vectors (TV) generated by the ATPG tool ATOM [83]. The average value of node transition count (NTC), which is the total value of NTC divided by the total number of clock cycles, for ASAP, ALAP, and the proposed BPIC test application strategies outlined in Section 3.3, are given in columns 4, 5 and 6 respectively. It can be clearly seen from Table 3.3 that BPIC test application strategy has the least average value of NTC for all the benchmark circuits when compared to ASAP and ALAP test application strategies. To give an indication of the reductions in average value of NTC, columns 7 and 8 show the percentage reduction of BPIC over ASAP and ALAP test application strategies. The reduction varies from approximately 10% as in the case of *s641* down to under 1% as in the case of *s526*. Table 3.3 has shown the reductions in node transition count using a non-compact test set. In order to reduce test application time while maintaining the same test quality, compact test sets are used. Compact test sets may lead to higher power dissipation because of an increased number of sensitised paths by each test vector. However, using the proposed BPIC test application strategy similar average values of NTC are achieved for all the benchmark circuits when comparing compact test sets to non-compact test sets. Table 3.4 shows experimental results for compact test set generated by MINTEST [84] when applying the proposed BPIC test application strategy without scan cell and test vector ordering. For example, in the case of *s298* the average value of NTC for non compact test set is 114.73 (Table 3.3), whereas for compact test set the average value of NTC is 107.85 (Table 3.4) despite a considerable reduction in the number of test vectors from 52 as in the case of non-compact test set to 23 as in the case of the compact test set. The similar values of NTC are due to finding the best primary input change time for reducing spurious transitions during shifting in test vectors and shifting out responses. This clearly shows that using compact test sets and hence decreasing the test application time will *not* increase the power dissipation during test application in full scan sequential circuits.

While the application of the BPIC test application strategy reduces average value of NTC when compared to ASAP and ALAP test application strategies, as shown in Tables 3.3 and 3.4 for non-compact and compact test sets respectively, further savings can be achieved when BPIC is combined with test vector ordering and scan cell ordering. The results of the proposed simulated annealing-based design space exploration (Section 3.4.2) for solving simultaneous test vector ordering, scan cell ordering and primary input change time for non-compact test set is shown in Table 3.5. In order to asses the impact of all the factors accountable for power dissipation the following three experiments were carried out. In the first experiment the average value of NTC is computed using scan cell and test vector ordering under ASAP test application strategy. The results are given in columns 2 and 3. The reduction in column 3 is computed over the node transition count

circuit	SC	TV	ASAP NTC	ALAP NTC	proposed BPIC NTC	**%reduction over ASAP**	**%reduction over ALAP**	CPU time (s)
s208	8	65	55.39	55.73	53.48	**3.45**	**4.03**	0.17
s298	14	52	115.39	115.59	114.73	**0.56**	**0.74**	0.28
s344	15	62	131.43	131.45	130.54	**0.67**	**0.69**	0.48
s349	15	65	132.46	132.45	131.47	**0.75**	**0.74**	0.50
s382	21	72	145.12	145.54	143.91	**0.83**	**1.12**	0.80
s386	6	109	86.61	86.20	84.22	**2.76**	**2.29**	0.38
s400	21	71	146.32	146.35	144.78	**1.05**	**1.07**	0.81
s420	16	98	107.19	107.17	104.46	**2.53**	**2.52**	1.00
s444	21	77	149.08	149.93	148.05	**0.69**	**1.25**	0.97
s510	6	90	115.50	115.04	114.13	**1.17**	**0.79**	0.40
s526	21	107	185.83	186.17	184.79	**0.55**	**0.73**	1.45
s641	19	99	186.74	184.03	168.71	**9.65**	**8.32**	2.19
s713	19	100	198.88	196.83	180.42	**9.28**	**8.33**	2.29
s820	5	190	139.22	138.89	136.79	**1.74**	**1.51**	1.02
s832	5	200	138.46	137.96	136.05	**1.73**	**1.37**	1.07
s838	32	183	199.81	199.84	195.00	**2.40**	**2.42**	14.72
s953	29	138	169.93	169.66	167.20	**1.60**	**1.44**	4.71
s1196	18	227	105.18	105.35	100.43	**4.51**	**4.67**	6.38
s1238	18	240	106.85	107.43	102.06	**4.48**	**5.00**	6.69
s1423	74	135	508.09	509.91	502.78	**1.04**	**1.39**	56.48
s1488	6	196	346.62	346.92	342.76	**1.11**	**1.19**	2.68
s1494	6	191	351.92	352.85	348.54	**0.95**	**1.22**	2.62
s5378	179	358	1786.44	1786.58	1774.36	**0.67**	**0.68**	3113.87
s9234	211	660	3123.16	3123.33	3098.91	**0.77**	**0.78**	16395.37

Table 3.3: Experimental results for non-compact test set generated by ATOM [83] when applying the proposed BPIC test application strategy without scan cell and test vector ordering.

of the initial scan cell and test vector. In the second experiment the test application strategy was changed to ALAP, and the results are shown in columns 4 and 5. In the third experiment the proposed BPIC test application strategy is combined with scan cell and test vector ordering and the results are given in columns 6 and 7. Note that BPIC always produces better results than ASAP and ALAP due to higher correlation between successive states during shifting in test vectors and shifting out test responses. This clearly shows the importance of integrating *all* the factors accountable for power dissipation in the optimisation process. The reduction value depends on the type of the circuit and the

circuit	SC	TV	ASAP NTC	ALAP NTC	proposed BPIC NTC	**%reduction over ASAP**	**%reduction over ALAP**	CPU time (s)
s208	8	27	54.80	54.72	52.42	**4.35**	**4.20**	0.07
s298	14	23	108.25	108.57	107.85	**0.36**	**0.66**	0.12
s344	15	13	124.36	124.21	123.48	**0.70**	**0.58**	0.10
s349	15	13	128.35	128.33	127.62	**0.57**	**0.55**	0.10
s382	21	25	147.91	148.14	146.75	**0.78**	**0.93**	0.28
s386	6	63	85.77	85.26	83.38	**2.78**	**2.19**	0.21
s400	21	24	154.04	154.44	152.76	**0.83**	**1.08**	0.28
s420	16	43	100.73	100.46	98.12	**2.59**	**2.33**	0.48
s444	21	24	155.46	156.30	154.19	**0.81**	**1.35**	0.32
s510	6	54	114.03	113.75	112.53	**1.31**	**1.06**	0.24
s526	21	49	182.74	182.98	182.17	**0.31**	**0.44**	0.68
s641	19	21	172.88	176.55	161.08	**6.82**	**8.75**	0.47
s713	19	21	195.45	192.38	181.10	**7.34**	**5.86**	0.49
s820	5	93	138.03	137.65	135.55	**1.79**	**1.51**	0.51
s832	5	94	138.83	138.37	136.33	**1.79**	**1.47**	0.50
s838	32	75	187.95	187.56	185.14	**1.49**	**1.29**	6.35
s953	29	76	169.64	169.02	166.64	**1.76**	**1.40**	2.66
s1196	18	113	105.67	105.42	100.43	**4.95**	**4.73**	3.39
s1238	18	121	103.92	103.83	98.91	**4.82**	**4.74**	3.40
s1423	74	20	506.10	506.89	501.14	**0.98**	**1.13**	8.73
s1488	6	101	365.47	365.66	361.40	**1.11**	**1.16**	1.45
s1494	6	100	369.59	370.63	365.89	**1.00**	**1.28**	1.40
s5378	179	97	1808.88	1809.32	1791.68	**0.95**	**0.97**	826.89
s9234	211	105	3045.30	3045.51	3014.90	**0.99**	**1.01**	2637.05

Table 3.4: Experimental results for compact test set generated by MINTEST [84] when applying the proposed BPIC test application strategy without scan cell and test vector ordering.

average value of NTC for the initial scan cell and test vector order. For example in the case of *s713* the reduction is 34% and it goes down to 4% as in the case of *s838*. However, this still presents an improvement when compared to ASAP and ALAP which yield reductions only of 3%. Table 3.6 shows the results for ASAP, ALAP and BPIC test application strategies when using compact test sets. Again the proposed BPIC provides better results than ASAP and ALAP, for all the circuits from the benchmark set. It is interesting to note that NTC values for the BPIC test application strategy when using non-compact and compact test sets are similar. This indicates that the test set size has no influence on

circuit	**Test application strategy targeted during optimisation**						CPU time (s)
	ASAP NTC	%reduction	ALAP NTC	%reduction	**BPIC** NTC	**%reduction**	
s208	48.39	12.64	49.19	11.19	**45.35**	**18.13**	24390
s298	97.97	15.09	92.49	19.84	**92.09**	**20.19**	33960
s344	115.55	12.08	119.68	8.93	**114.85**	**12.61**	34270
s349	118.62	10.45	121.01	8.64	**117.87**	**11.01**	35040
s382	125.93	13.22	125.98	13.18	**124.47**	**14.23**	40430
s386	70.50	18.60	71.86	17.03	**69.74**	**19.48**	41550
s400	128.97	11.85	132.12	9.70	**125.30**	**14.36**	34550
s420	93.94	12.35	94.51	11.82	**92.41**	**13.78**	43410
s444	130.27	12.61	134.33	9.89	**129.27**	**13.28**	42940
s510	98.91	14.35	101.60	12.02	**97.98**	**15.16**	36920
s526	162.13	12.75	160.96	13.37	**160.52**	**13.61**	37340
s641	142.19	23.85	141.44	24.25	**132.42**	**29.08**	42470
s713	146.91	26.13	144.98	27.09	**130.34**	**34.46**	44630
s820	111.71	19.75	112.10	19.47	**110.83**	**20.39**	43710
s832	115.19	16.80	114.07	17.61	**113.19**	**18.24**	36520
s838	193.50	3.16	193.80	3.01	**190.29**	**4.76**	45970
s953	128.24	24.53	128.61	24.31	**127.26**	**25.10**	46470

Table 3.5: Experimental results for non-compact test set generated by ATOM [83] when applying the proposed BPIC test application strategy integrated with test vector ordering and scan cell ordering.

the average value of NTC confirming that the only three accountable factors for power dissipation during test application in full scan circuits are test vector ordering, scan cell ordering and primary input change time. For some of the examples the computational time for completing the optimisation may increase over 40,000s using a Pentium II processor at 350 MHz, as shown in Table 3.5. This is due to the huge size of the design space and the low number of solutions with identical NTC which clearly leads to longer times for exploration and convergence of the simulated annealing algorithm. However, when using compact tests as shown in Table 3.6, due to smaller number of test vectors and consequently the size design space, lower time for completion is required which leads to the conclusion that compact test sets have benefits in both test application time as well as in computational time with similar reductions in power dissipation.

circuit	**Test application strategy targeted during optimisation**						CPU time (s)
	ASAP NTC	%reduction	ALAP NTC	%reduction	**BPIC** NTC	**%reduction**	
s208	47.85	12.68	47.90	12.59	**44.02**	**19.67**	8399
s298	91.48	15.49	96.22	11.11	**91.13**	**15.81**	12980
s344	94.92	23.66	100.27	19.37	**94.02**	**24.39**	15550
s349	104.62	18.48	110.23	14.11	**104.14**	**18.85**	13560
s382	120.37	18.61	124.09	16.10	**118.49**	**19.88**	29020
s386	70.33	18.00	69.72	18.71	**68.54**	**20.09**	24500
s400	120.51	21.76	123.36	19.92	**119.51**	**22.41**	29380
s420	90.97	9.69	87.61	13.02	**83.13**	**17.47**	28940
s444	129.44	16.73	132.04	15.06	**120.07**	**22.76**	29040
s510	95.79	15.99	97.68	14.33	**94.80**	**16.86**	22940
s526	151.31	17.19	158.35	13.34	**150.87**	**17.44**	29310
s641	143.36	17.07	130.56	24.48	**120.16**	**30.49**	29210
s713	152.77	21.83	144.71	25.96	**133.49**	**31.69**	24920
s820	110.61	19.86	111.23	19.41	**109.71**	**20.51**	28870
s832	112.39	19.04	112.30	19.10	**111.40**	**19.75**	28950
s838	170.01	9.54	170.75	9.15	**168.36**	**10.42**	30220
s953	128.15	24.45	128.70	24.13	**127.06**	**25.09**	30040

Table 3.6: Experimental results for compact test set generated by MINTEST [84] when applying the proposed BPIC test application strategy integrated with test vector ordering and scan cell ordering.

3.5.2 Experimental Results for Partial Scan Sequential Circuits

This section demonstrates through a set of benchmark examples that the proposed BPIC test application strategy outlined in Section 3.3 yields savings in power dissipation during test application in **partial** *scan sequential circuits*. Furthermore the savings can be substantially improved when the proposed BPIC test application strategy is combined with scan cell ordering.

Table 3.7 shows circuit and test set characteristic for 17 circuits from ISCAS89 benchmark set [23]. The first and second columns give the circuit name and the number of primary inputs respectively. The third and fourth columns give the number of scan cells and non-scan cells respectively. The scan cells are selected using the logic level partial scan tool OPUS [37, 38] by cutting all the cycles in the circuit. Column 5 gives the number of test vectors generated by the logic level ATPG tool GATEST [170] to achieve the fault

circuit	primary inputs	scan cells	non-scan cells	test vectors	fault coverage(%)
s344	9	5	10	41	99.41
s349	9	5	10	41	98.85
s382	3	9	12	112	98.74
s386	7	5	1	198	99.73
s400	3	9	12	128	97.40
s444	3	9	12	128	95.99
s510	19	5	1	70	100.00
s526	3	3	18	321	85.22
s641	35	7	12	104	94.64
s713	35	7	12	106	88.46
s820	18	4	1	283	99.88
s832	18	4	1	459	98.39
s953	16	5	24	206	100.00
s1423	17	22	52	257	95.31
s1488	8	5	1	221	100.00
s1494	8	5	1	232	99.20
s5378	35	30	149	1465	92.43

Table 3.7: Circuit and test set characteristic for 17 benchmark circuits from ISCAS89 benchmark set [23] used in experimental results.

coverage shown in the last column. Table 3.8 shows the results when the proposed BPIC test application strategy is applied by itself (i.e. without scan cell ordering) for the 17 benchmark circuits described in Table 3.7. The average value of NTC, which is the total value of NTC divided by the total number of clock cycles over the entire test application period, for ALAP and the proposed BPIC test application strategies are given in columns 2 and 3 respectively. It can be clearly seen from Table 3.8 that BPIC test application strategy has smaller average value of NTC for all the benchmark circuits when compared to ALAP test application strategy [2]. To give an indication of the reductions in average value of NTC, column 4 shows the percentage reduction of BPIC over ALAP test application strategy. The reduction value depends on the type of the circuit and the average value of NTC for the initial scan cell order. The reduction varies from approximately 15% as in the case of *s713* down to under 1% as in the case of *s349*. The last column gives the computational time for the exact *BPIC-ALG* algorithm, which computes best primary input change times used by BPIC test application time strategy. For most of the circuits it took approximately $<$ 1s to find the best primary input change times for all the

circuit	initial ALAP *NTC*	proposed BPIC *NTC*	**%reduction over initial ALAP**	CPU time (s)
s344	85.39	84.14	**1.45**	0.10
s349	81.21	80.42	**0.98**	0.09
s382	78.92	77.48	**1.82**	0.44
s386	76.83	74.48	**3.06**	0.43
s400	81.22	79.87	**1.66**	0.52
s444	88.58	87.36	**1.36**	0.61
s510	104.13	102.54	**1.52**	0.23
s526	55.44	54.93	**0.91**	0.60
s641	98.55	85.47	**13.26**	0.80
s713	110.61	94.68	**14.40**	0.83
s820	120.26	117.78	**2.06**	1.01
s832	128.81	126.40	**1.87**	1.62
s953	95.85	93.00	**2.97**	1.16
s1423	186.82	183.60	**1.72**	10.78
s1488	318.00	311.72	**1.97**	2.14
s1494	326.87	321.91	**1.51**	2.23
s5378	523.49	498.88	**4.70**	380.11

Table 3.8: Reduction in average value of *NTC* when applying the proposed BPIC test application strategy (Definition 3.5) compared to ALAP test application strategy [2].

test vectors. This indicates that the proposed *BPIC-ALG* can be used for fast computation of the cost function in the optimisation process when combined with scan cell ordering as explained in Section 3.4.2. There are exceptional circuits with very high number of test vectors (1465 in the case of *s5378*) where the computational time is up to 380s (last row from Table 3.8). However, this is still acceptable low computational time for the calculation of the cost function.

While the application of the proposed BPIC test application strategy reduces average value of *NTC* when compared to ALAP test application strategy, as shown in Table 3.8 further savings can be achieved when the proposed BPIC test application strategy is combined with scan cell ordering. Before combining scan cell ordering and the proposed the proposed BPIC test application strategy, the influence of scan cell ordering under the ALAP test application strategy [2] is examined as shown in Table 3.9. For circuits with small number of scan cells, the exploration of the entire design space is computationally inexpensive. In the case of *s713*, where for 7 scan cells there are $7! = 5040$ possible scan

circuit	initial ALAP *NTC*	Optimised ALAP *NTC*	**%reduction over initial ALAP**	CPU time (s)
s344	85.39	80.88	**5.28**	10
s349	81.21	79.77	**1.78**	10
s382	78.92	71.85	**8.95**	705
s386	76.83	70.81	**7.83**	45
s400	81.22	71.65	**11.78**	1258
s444	88.58	79.75	**9.96**	1618
s510	104.13	100.73	**3.62**	22
s526	55.44	55.30	**0.26**	3
s641	98.55	86.99	**11.72**	3039
s713	110.61	97.95	**11.42**	3191
s820	120.26	115.28	**4.14**	21
s832	128.81	120.34	**6.57**	33
s953	95.85	91.54	**4.49**	111
s1423	186.82	171.37	**8.26**	5989
s1488	318.00	305.49	**3.93**	214
s1494	326.87	310.46	**5.02**	224
s5378	523.49	394.88	**24.56**	44830

Table 3.9: Reduction in average value of *NTC* when applying scan cell ordering leading to optimised scan cell order under the ALAP test application strategy [2].

cell orderings as outlined in Section 3.4.2, it took 3191s to find the optimum scan cell order which yields 11.42% reduction in average value of *NTC*. However for larger circuits as in the case of *s1423* and *s5378* where the size of the design space is $22! \approx 10^{21}$ and $30! \approx 2.5 \times 10^{32}$ respectively, *SA-Optimisation* algorithm from Section 3.4.2 is required for efficient design space exploration of the discrete, degenerate and highly irregular design space. For example it takes up to 44830s to find an sub-optimum scan cell order as in the case of *s5378*. This is due to the huge size of the design space and the low number of solutions with identical *NTC* which clearly leads to long computational times for the convergence of the simulated annealing algorithm. It should be noted that for most of the benchmark circuits scan cell ordering under ALAP test application strategy (Table 3.9) yields higher reductions in average value of *NTC* than when the proposed BPIC test application strategy is applied by itself (Table 3.8), at the expense of significantly greater computational time. Furthermore, there are circuits (*s641* and *s713*) where the proposed BPIC test application strategy applied by itself generates higher reductions in *NTC* with

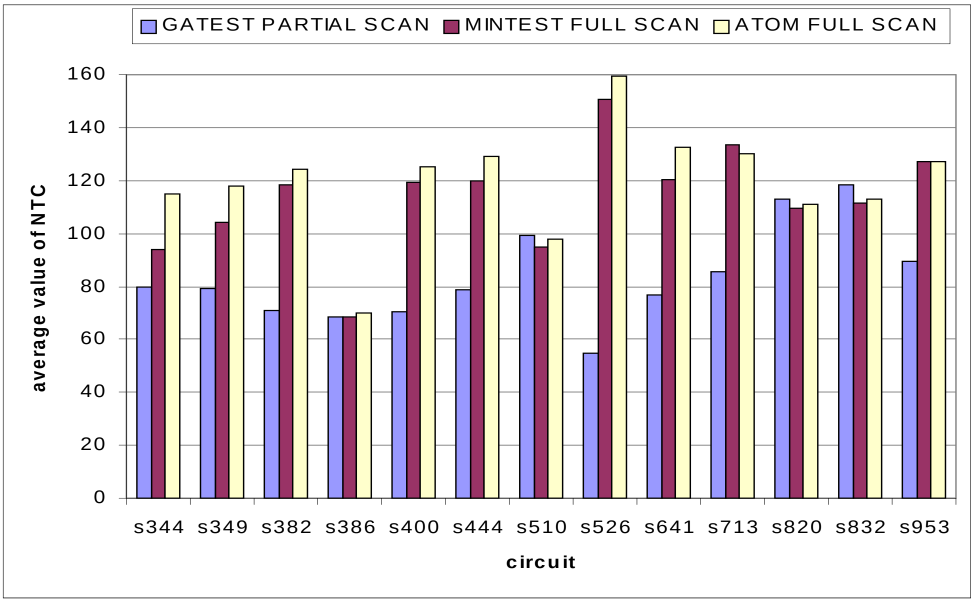
circuit	initial ALAP *NTC*	Optimised BPIC *NTC*	**%reduction over initial ALAP**	CPU time (s)
s344	85.39	79.93	**6.39**	57
s349	81.21	79.18	**2.50**	57
s382	78.92	71.17	**9.81**	4895
s386	76.83	68.65	**10.64**	265
s400	81.22	70.66	**13.01**	3117
s444	88.58	78.98	**10.83**	5570
s510	104.13	99.57	**4.37**	127
s526	55.44	54.80	**1.15**	12
s641	98.55	76.99	**21.87**	25390
s713	110.61	85.58	**22.62**	25170
s820	120.26	113.19	**5.87**	100
s832	128.81	118.37	**8.10**	162
s953	95.85	89.49	**6.63**	683
s1423	186.82	169.61	**9.21**	26270
s1488	318.00	299.81	**5.71**	1257
s1494	326.87	306.02	**6.37**	1321
s5378	523.49	373.26	**28.69**	121700

Table 3.10: Reduction in average value of *NTC* when combining the proposed BPIC test application strategy with scan cell ordering leading to optimised scan cell order under the BPIC test application strategy (Definition 3.5).

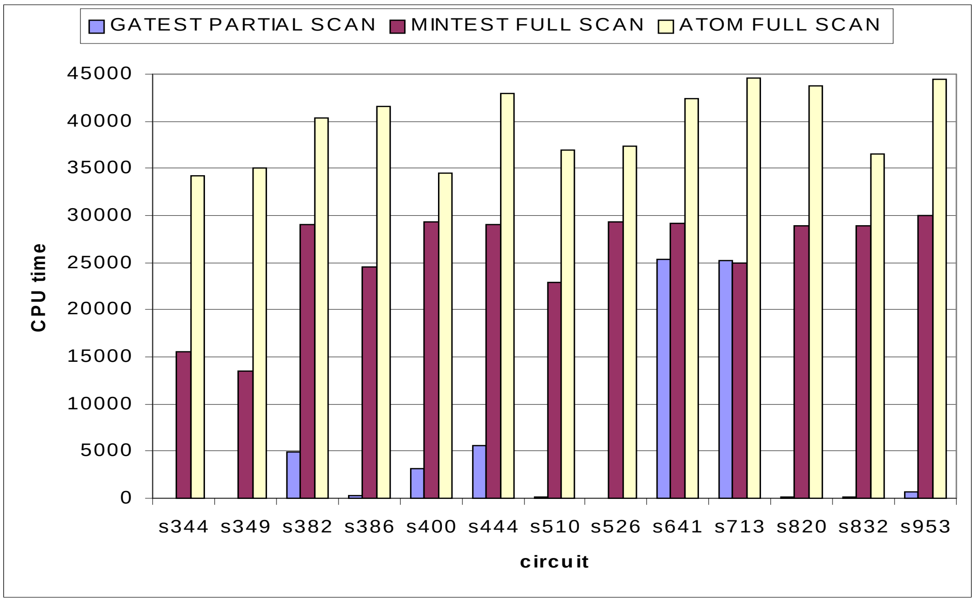
computational time which is three orders of magnitude lower when compared to scan cell ordering under the ALAP test application strategy. For example, in the case of *s713* it took 3191s to achieve 11.42% reduction in average value of *NTC* by scan cell ordering (last two columns of Table 3.9) when compared to only 0.80s of computational time to achieve 14.40% reduction in average value of *NTC* by the proposed BPIC test application strategy (last two columns of Table 3.8). To achieve maximum reductions in average value of *NTC*, the proposed BPIC test application strategy and scan cell ordering are combined as shown in Table 3.10. For all the benchmark circuits the combination of the BPIC test application strategy and scan cell ordering leads to higher reductions than when any parameter is considered by itself. For example in the case of *s5378* the reduction in average value of *NTC* is 28.69% at the expense of high computational time which is due to high number of scan cells and hence large design space and low convergence of the simulated annealing algorithm.

3.5.3 Further Benefits of Partial Scan in Minimising Power Dissipation in Scan Based Sequential Circuits

It is known that partial scan has advantages in terms of test area overhead and test application time when compared to full scan [3]. This section shows how the proposed BPIC test application strategy for partial scan provides further benefits in terms of power dissipation and computational time required for design space exploration when compared to full scan. Figures 3.13(a) and 3.13(b) show a comparison of average value of NTC and computational time for partial and full scan sequential circuits for various benchmark circuits. The results for full scan sequential circuits were outlined in Table 3.5 (ATOM [83]), and Table 3.6 (MINTEST [84]). The average value of NTC for partial scan is significantly smaller when compared to full scan, as shown in Figure 3.13(a). The reduction is due to partial scan DFT methodology, which in the test mode of operation does not clock the non-scan cells while test responses are shifted out, leading to significant savings in power dissipation. It is interesting to note that for benchmark circuits *s820* and *s832* the average value of NTC is lower for full scan sequential circuits. This is due to the fact that for both circuits 4 out of 5 sequential elements are modified to scan cells and full scan sequential circuits allow test vector ordering which gives higher degree of freedom during the optimisation process. However, the processing time is significantly lower for partial scan sequential circuits as shown in Figure 3.13(b) which gives a comparison of computational overhead. It should be noted that large circuits (*s1423* and *s5378*) are not handled in the case of the full scan sequential circuits due to the huge size of the design space where both scan cell and test vector ordering are considered. Furthermore, for all the benchmark circuits shown in Figure 3.13(b) the computational time required for exploring the design space of partial scan is substantially smaller (orders of magnitude) than the the computational time required for exploring the design space of full scan. This is caused by the reduction in the size of the design space to be explored due to smaller number of scan cells and by the exact and polynomial time *BPIC-ALG* algorithm. Finally, based on the results shown in Figures 3.13(a) and 3.13(b), it may be concluded that partial scan has advantages not only in less test area overhead and test application time, but also in less power dissipation during test application (i.e. average value of NTC in Figure 3.13(a)) and computational time required for design space exploration (CPU time in Figure 3.13(b)) when compared to full scan.



(a) Average value of *NTC*



(b) Computational time

Test set for partial scan is generated by GATEST [170]
Compact and non-compact test sets for full scan are generated by ATOM [83] and MINTEST [84]

Figure 3.13: Comparison of average value of *NTC* and computational time for partial and full scan sequential circuits when using the proposed BPIC test application strategy

3.6 Concluding Remarks

This chapter has proposed a new technique for minimising power dissipation in full scan sequential circuits during test application. The technique is based on increasing the correlation between successive states during shifting in test vectors and shifting out test responses by changing the primary inputs such that the smallest number of transitions is achieved. A new algorithm, which is test set dependent, computes best primary input change (BPIC) time for each test vector was presented. It was shown that combining the described technique with scan cell and test vector ordering, using a simulated annealing-based design space exploration, reductions in power dissipation during test application in small to medium sized full scan sequential circuits are achieved. Exhaustive experimental results using both compact and non-compact test sets have shown that compact test sets have similar power dissipation during test application with reduction in test application time and computational time when compared to non-compact test sets.

The new BPIC test application strategy introduced in Definition 3.5 of Section 3.3 is equally applicable to minimising power dissipation in partial scan sequential circuits. Since test vector ordering proposed for power reduction in full scan sequential circuits [49] is prohibited for partial scan sequential circuits, the proposed test application strategy yields power reduction as shown in the experimental results (Section 3.5). Since the proposed test application strategy depends only on controlling primary input change time, power is minimised with no penalty in test area, performance, test efficiency, test application time or volume of test data. It was shown that combining the proposed BPIC test application strategy and scan cell ordering using a simulated annealing-based design space exploration algorithm yields reductions in power dissipation during test application in partial scan sequential circuits.

This chapter has shown that partial scan does not provide only the commonly known benefits such as less test area overhead and test application time, but also less power dissipation during test application and computational time required for design space exploration, when compared to full scan. This reinforces that partial scan should be the preferred choice as design for test methodology for sequential circuits when low power dissipation during test application is of prime importance for high yield and reliability.

Chapter 4

Multiple Scan Chains-Based Power Minimisation in Scan Sequential Circuits

Chapter 3 has shown how power dissipation during test application can be minimised in scan sequential circuits with no penalty in area overhead, test application time, test efficiency, performance, or volume of test data. However, the computation of the best primary input change (BPIC) time introduced in Definitions 3.4 and 3.5 in Section 3.3 from Chapter 3, is dependent on the size and the value of the test vectors in the test set. Therefore, integrating the proposed best primary input change time with scan cell and test vector ordering leads to discrete, degenerate and highly irregular design space, and hence high computational time, which limits the applicability of the new BPIC test application strategy only to small to medium sized scan sequential circuits. This chapter introduces a new test set independent technique based on multiple scan chains and shows how with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, considerable savings in power dissipation during test application in large scan sequential circuits can be achieved in low computational time. Further, the extra test hardware required by the proposed technique employing multiple scan chains can be specified at the logic level and synthesised with the rest of the circuit. This makes the proposed multiple scan chains-based power minimisation technique easily embeddable in the existing VLSI design flow as described previously in Figure 3.1 from Chapter 3.

The rest of the chapter is organised as follows. Section 4.1 outlines the shortcomings of the previously proposed approaches for power minimisation in large scan sequential circuits and gives the motivation and the objectives of the proposed technique. In Section 4.2, a brief review of standard test terminology and power dissipation concepts used throughout the chapter are presented. Section 4.3 introduces a new technique for power minimisation in large scan sequential circuits based on multiple scan chains including the proposed DFT architecture, scan cell classification, clock tree power minimisation and extention to scan BIST methodology. Partitioning scan cells in multiple scan chains based on their classification, and a new test application strategy based on the DFT architecture described in the previous section are introduced in Section 4.4. Experimental results and a comparative study of the proposed BPIC in Chapter 3 and the multiple scan chain-based technique are presented in Section 4.5. The chapter concludes in Section 4.6.

4.1 Motivation and Objectives

The aim of this chapter is to reduce power dissipation in large scan sequential circuits. Despite their benefits in lowering power dissipation during test application, the previously described techniques [46, 49, 61, 88, 172, 189, 192] in Section 2.2 and the new BPIC test application strategy, also described in [145], are inefficient for large scan sequential circuits due to one or more of the following problems:

a. test area overhead associated with detection logic [46, 61] required to find nonessential vectors (i.e. vectors which do not contribute to an increase in fault coverage).

b. performance degradation associated with modified scan cell design [61].

c. large test application time required to achieve significant power savings [61, 172, 189, 192].

d. clock tree power dissipation is tackled by clock gating only for nonessential test vectors [61].

e. high number of extra test vectors [189] emerges as a problem to testers which need to change to support the large volume of test data [94].

f. computational time may be prohibitively large hindering the exploration for large sequential circuits [49, 145, 189, 88].

The previous techniques [46, 49, 61, 88, 145, 172, 189, 192] proposed separate solutions for solving one of the problems (a) - (f) at the expense of the other problems. For example while test vector inhibiting techniques [46, 61] achieve good savings in power dissipation, considerable area overhead for detection logic is introduced (problem a) or further performance degradation is incurred (problem b). On the other hand techniques based on adjacent patterns [172, 192] require considerable test application time (problem c) to achieve the required fault coverage. Therefore, savings in power dissipation are achieved at the expense of very high test application time. Furthermore, clock tree power dissipation (problem d) which can be up to one third of total power dissipation [185] is tackled only in [61] where the clock is gated only for non-essential test vectors. This implies that for essential vectors there are no savings in clock tree power dissipation which can lead to extremely high power dissipation over limited time intervals. The technique proposed in [189] necessitates an increase of $(m \times p)/(m+p)$ in the volume of test data where m is the number of scan cells and p is the number of primary inputs. This is due to computing an extra test vector of p bits for every single clock cycle while shifting out test responses of each test vector. Therefore instead of having only $(m+p) \times n$ bits of test data, the technique proposed in [189] employs $((m+p) \times n) + (m \times p \times n)$ bits of test data, and hence a large increase of $(m \times p)/(m+p)$. While volume of test data (problem e) was not a concern in the past for small to medium sized circuits it is recently emerging as a problem for testers which need to change to support the large volume of test data [94]. The technique proposed [88] overcomes the problem with large volume of test data by computing a single extra vector for every test vector. However, it yields modest savings in power dissipation due to inability to fully mask the activity in the combinational part of the circuit. Furthermore, to achieve good fault coverage both techniques based on extra vectors [88, 189] require longer test sequences and hence both higher test application time (problem c) and computational time (problem f). Finally techniques which operate in a post-ATPG phase [49, 145] using compact test sets for high fault coverage require huge computational time (problem f) since they are strongly test set dependent and require probabilistic optimisation. Despite achieving good results with no penalty in area overhead, test application time, test efficiency, performance, or volume of test data, the large computational overhead of the techniques proposed [49, 145] prohibits their applicability to large scan sequential circuits.

The aim of this chapter is to introduce a new technique for power minimisation during test application in large scan sequential circuits based on a novel DFT architecture which eliminates all the above mentioned problems (a) - (f). The proposed DFT architecture is based on partitioning scan cells into multiple scan chains which reduces the clock tree power dissipation and does not have performance penalty. A new test application strategy for the proposed DFT architecture which applies a single extra test vector while shifting out test responses for each scan chain is presented. The multiple scan chain-based approach for power minimisation which is test set independent, is applicable to both non-compact and compact test sets leading to low test application time. It is shown that with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, high savings in power dissipation during test application in large scan sequential circuits are achieved in low computational time.

4.2 Background and Definitions

A brief review of standard test terminology used throughout this chapter are presented. The *controlling* value for a gate is a single input value that uniquely determines the output to a known value independent of the other inputs to the gate. For example, the controlling value for OR gate is 1, and for AND gate is 0. If the value of an input is the complement of the controlling value, then the input has a *noncontrolling* value. A path is a set of connected gates and wires. A path is defined by a single input wire and a single output wire per gate. A signal is an *on-input* if it is on the target path. A signal is an *off-input* (*side input*) if it is an input to a gate which is on a target path but is not an on-input. If two faults can be detected by a single test vector, they are *compatible faults*. Consequently, two faults are *incompatible*, if they cannot be detected by a single test vector. A test vector from a given test set is an *essential* test vector, if it detects at least one fault that is not detected by any other test vector in this test set. A test vector is *non-essential* with respect to a given test set if all the faults detected by it are also detected by other test vectors in the given test set. A test set independent approach for power minimisation depends only on circuit structure and savings are guaranteed regardless of the size and and the value of the test vectors in the test set. This is unlike the test set dependent approaches, where power minimisation depends on the size and the value of the test vectors in the test set, as introduced in Section 3.1.

4.3 Power Minimisation in Scan Sequential Circuits Based on Multiple Scan Chains

In this section a new technique for power minimisation in large scan sequential circuits based on multiple scan chains is introduced. Section 4.3.1 overviews the proposed DFT architecture for power minimisation. Section 4.3.2 defines compatible, incompatible, and independent scan cells and their importance for partitioning scan cells into multiple scan chains. Section 4.3.3 gives an important theoretical result showing the advantage of the proposed DFT architecture from the clock tree power dissipation standpoint, and Section 4.3.4 describes how the proposed multiple scan chains can be extended to scan BIST methodology.

4.3.1 Proposed Design for Testability Architecture Using Multiple Scan Chains

The proposed DFT architecture using multiple scan chains $SC_0 \ldots SC_{k-1}$ is illustrated in Figure 4.1. The scan input *ScanIn* is routed to all scan chains while the scan output *ScanOut* is selected from the output of each scan chain. Scan chains $SC_0 \ldots SC_{k-1}$ are operated using non-overlapping enable signals for clocks $CLK_0 \ldots CLK_{k-1}$. Non-overlapping enable signals gate the system clock *CLK* using a scan control register where the number of cells equals the number of scan chains. This implies that scan chains $SC_0 \ldots SC_{k-1}$ are enabled one by one during each scan cycle. While shifting out test responses through scan chain SC_i, only the bit position i of scan control register is set to 1 while the other positions are set 0. This is easily implemented by shifting the value of 1 through scan control register using the extra scan clock *SCLK*. Before starting the first scan cycle, the initial vector $10 \ldots 00$ is set up in the scan control register using the scan input *ScanIn*. Thereafter, for each scan cycle, the $10 \ldots 00$ value is propagated circularly through the scan control register. It should be noted that when the circuit under test is in the test mode all the faults in the extra logic are observable through *ScanOut* line using the test data which is shifted through the k scan chains and control data shifted through the scan control register. Therefore, the extra test hardware, including the selection logic shown in Figure 4.1, has no penalty on test efficiency. During the normal operation of the circuit $CLK_0 \ldots CLK_{k-1}$ are active at the same time, since when normal/test signal N/T is 1 the outputs of the extra OR gates are 1 and *CLK* is not gated by the scan control register.

To provide a brief overview of the test application strategy for the proposed DFT architecture, while shifting out test responses present in scan cells within scan chain SC_i, primary inputs are set to extra test vector EV_i which eliminates the spurious transitions (Definition 4.1 from Section 4.3.2) that originate from scan cells within scan chain SC_i. The number of clock cycles required to shift in the present state (pseudo input) part of each test vector equals the number of scan cells and the test response is loaded in a single clock cycle. This implies that there are no extra clock cycles for each test vector applied to the circuit under test and hence no penalty on test application time. Note that the proposed DFT architecture has no penalty on performance, since extra test hardware is not inserted on critical paths [51]. Further, the extra test hardware required by the scan control register and selection logic can be specified at the logic level and synthesised with the rest of the circuit which makes the proposed DFT architecture easily embeddable in the existing VLSI design flow. It should be noted that the proposed DFT architecture introduced for full scan sequential circuits is equally applicable to partial scan sequential circuits. Since the number of scan cells in a partial scan sequential circuit is approximately 10% of the total number of state elements [3] it is more appropriate to illustrate the applicability of the proposed architecture on large full scan sequential circuits. However, as the complexity of state of the art digital circuits increases it is expected that in the foreseeable future partial scan sequential circuits with very high number of scan cells will appear. In such cases, the technique proposed in this chapter *is applicable with no modifications to partial scan sequential circuits*. What makes the proposed multiple scan chain-based DFT architecture particularly suitable for large scan sequential circuits is that partitioning scan cells into multiple scan chains is test set independent (Section 4.4), and it depends only on the circuit size and structure unlike the solutions described in the previous Chapter 3 which strongly rely on the size of the test set and hence are applicable only to small to medium sized scan sequential circuits.

The algorithm for partitioning scan cells into multiple scan chains and the new test application strategy using multiple scan chains and extra test vectors shown in Figure 4.1, are described later in Section 4.4. Before describing generation of multiple scan chains, scan cells need to be classified into three broad classes as described in the following Section 4.3.2.

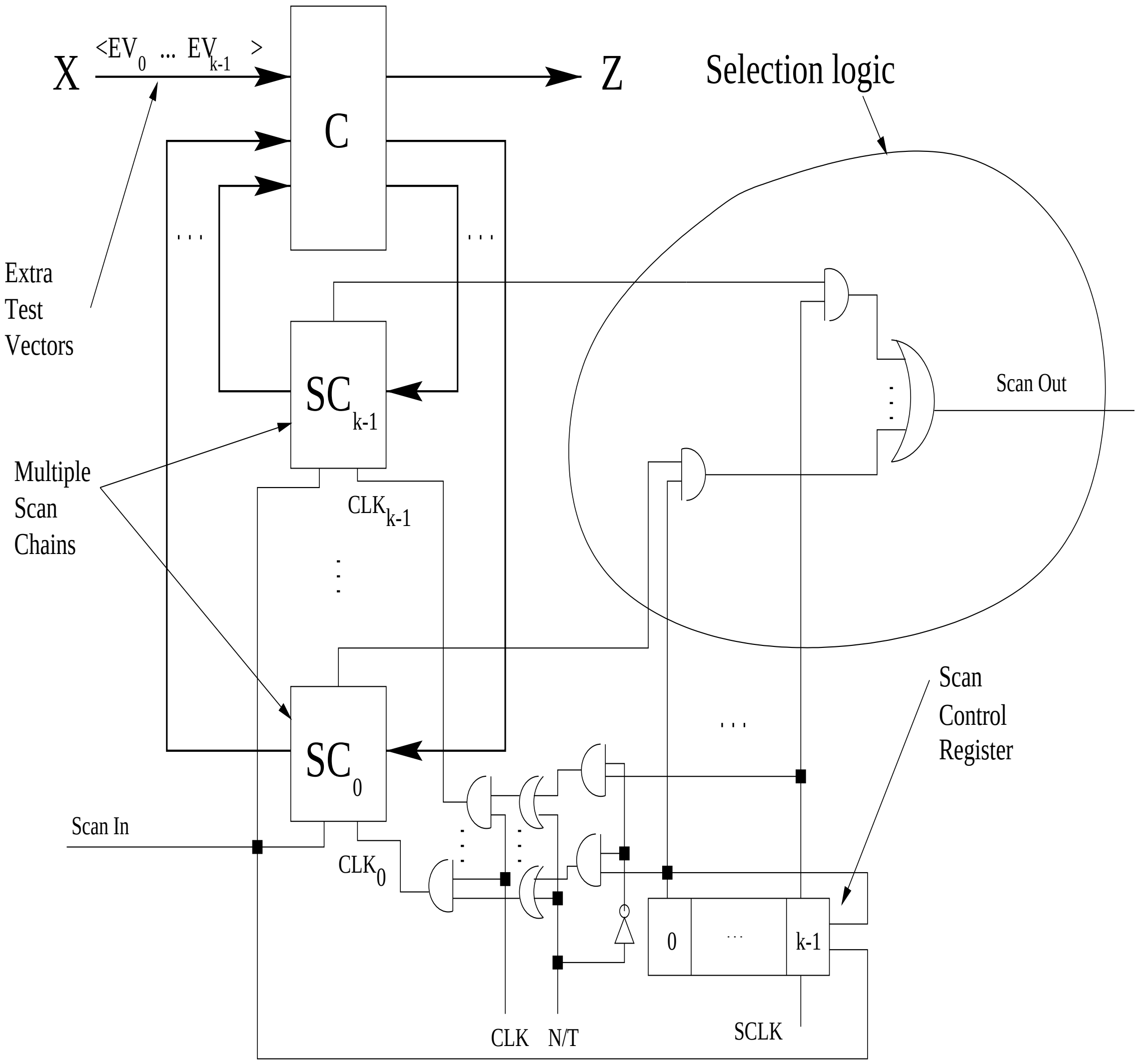


Figure 4.1: Proposed DFT architecture based on multiple scan chains.

4.3.2 Compatible, Incompatible, and Independent Scan Cells

In order to partition scan cells into multiple scan chains, they need first to be classified into three broad classes: compatible, incompatible, and independent scan cells. It should be noted that scan cell classification is not done explicitly by enumeration or exhaustive search, but it is done implicitly by the multiple scan chains partitioning algorithm as explained later in Figure 4.9 from Section 4.4.1. The proposed classification is important for computing extra test vectors associated with each scan chain that eliminate spurious transitions which were defined in Chapter 3. For the sake of completeness, Definition 3.1 from Section 3.3 in Chapter 3 is given below.

Definition 4.1 A spurious transition during test application in scan sequential circuits is a transition which occurs in the combinational part of the circuit under test while shifting out the test response and shifting in the present state part of the next test vector. These transitions do not have any influence on test efficiency since the values at the input and output of the combinational part are not useful test data.

Now the compatible and incompatible scan cells are introduced.

Definition 4.2 Two scan cells S_i and S_j are *compatible* if all primary inputs x_k are assigned values c_k that eliminate the spurious transitions which originate from both S_i and S_j. The values c_k of primary inputs x_k constitute the *extra test vector* which eliminates spurious transitions originating from both S_i and S_j.

Note that the sole purpose of extra test vectors is to reduce the spurious transitions during test application and has no effect on fault coverage which is determined by the original test set. The application of extra test vectors defines a novel test application strategy for power minimisation which is detailed in Section 4.4.2. Further, since a single extra test vector is used for each scan chain regardless of values loaded in scan cells then the volume of extra test data is dependent only on the number of scan chains and not on the number of scan cells and/or the size of the original test set.

Definition 4.3 Two scan cells S_i and S_j are *incompatible* if at least one primary input x_k that is assigned value i_k to eliminate the spurious transitions which originate from S_i will propagate the transitions which originate from S_j. Two incompatible scan cells cannot be assigned to the same scan chain since there is no extra test vector that can eliminate spurious transitions which originate from both of them.

The following example illustrates compatible and incompatible scan cells.

Example 4.1 Consider the circuit shown in Figure 4.2. The $\{x_0, x_1, x_2\}$ are primary inputs, $\{S_0, S_1, S_2, S_3, S_4, S_5\}$ are scan cells, $\{y_0, y_1, y_2, y_3, y_4, y_5\}$ are present state lines, and $\{z_0, z_1, z_2, z_3, z_4, z_5\}$ are circuit outputs. To eliminate spurious transitions at gate z_0 while shifting out test responses through scan cell S_0, primary input x_0 must be assigned the controlling value 0 of gate z_0. Similarly, to eliminate spurious transitions that originate from scan cell S_1, primary input x_0 must be assigned the controlling value 1 of gate z_1. Different values must be assigned to x_0 to eliminate spurious transitions which originate from scan cells S_0 and S_1. Therefore scan cells S_0 and S_1 are incompatible and are assigned to different scan chains $SC_0 = \{S_0\}$ and $SC_1 = \{S_1\}$. On the other hand, by assigning x_1 to the controlling value 0 of gates z_2 and z_3 the spurious transitions which originate from both scan cells S_2 and S_3 are eliminated. Thus, by introducing S_2 and S_3 into SC_0 and applying for example extra test vector $x_0x_1x_2 = \{000\}$ while shifting out test responses from $SC_0 = \{S_0, S_2, S_3\}$ no spurious transitions will occur at gates z_0, z_2 and z_3. Similarly, scan cells S_4 and S_5 are compatible since assigning 1 to the primary input x_2 eliminates spurious transitions at gates z_4 and z_5. By introducing S_4 and S_5 into SC_1 and applying extra test vector $x_0x_1x_2 = \{111\}$ while shifting out test responses from $SC_1 = \{S_1, S_4, S_5\}$ no spurious transitions will occur at gates z_1, z_4 and z_5. It should be noted that there is a strict interrelation between extra test vector value $x_0x_1x_2 = \{000\}$ and scan chain $SC_0 = \{S_0, S_2, S_3\}$, and $x_0x_1x_2 = \{111\}$ and scan chain $SC_1 = \{S_1, S_4, S_5\}$. While for the sake of simplicity, the extra test vectors $x_0x_1x_2 = \{000\}$ and $x_0x_1x_2 = \{111\}$ were described explicitly in this particular example, the extra test vectors and hence the multiple scan chains, are derived implicitly by the partitioning algorithm as explained later in Figure 4.9 from Section 4.4.1. Finally, note that output signals z_3 of scan chain SC_0 and z_5 of SC_1 are fed into the selection logic of the proposed DFT architecture from Figure 4.1.

The previous example has assumed a simple circuit where *all* the spurious transitions are eliminated by partitioning scan cells in two scan chains SC_0 and SC_1. However, some of the spurious transitions cannot be eliminated as described in the following example.

Example 4.2 Consider the circuit shown in Figure 4.3. The spurious transitions which originate in scan cells S_0 and S_1 cannot be eliminated at gate t_0 since both inputs are present state lines. However, by assigning x_0 and/or x_1 to the controlling value 0 of gate t_1 the spurious transitions will be eliminated at gate t_1. Scan cells S_0 and S_1 are compatible since same primary input values eliminate the spurious transitions of gate t_1.

Example 4.2 has illustrated that some of the spurious transitions cannot be eliminated since all the gate inputs depend on present state lines. Computing primary input values that eliminate spurious transitions (extra test vectors introduced in Definition 4.2) can be viewed as an ATPG problem to a *reduced circuit* with a *specified fault list* which are detailed in the algorithms presented in Section 4.4.1. The following Example 4.3 briefly illustrates the generation of the *reduced circuit* required to compute extra test vectors.

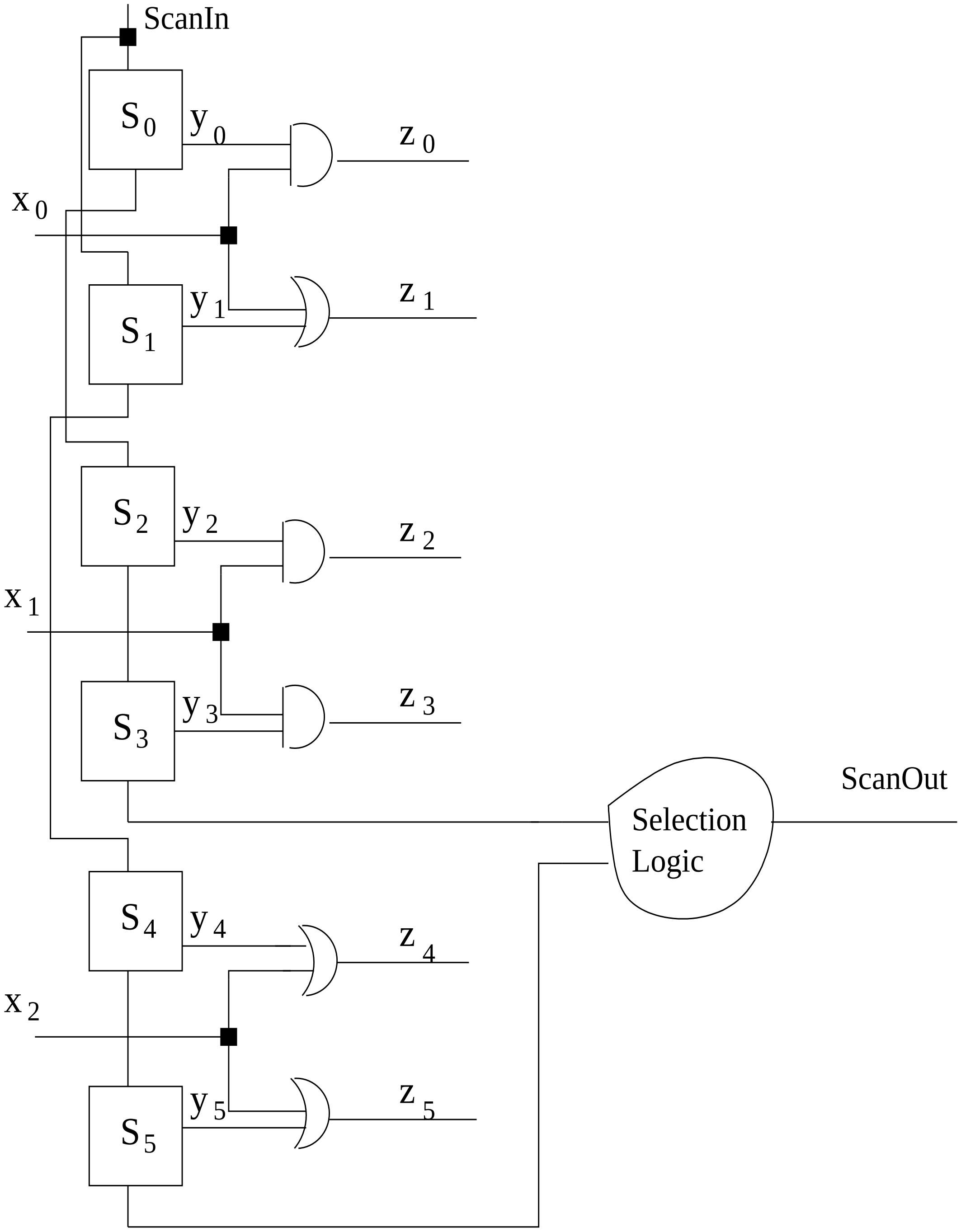


Figure 4.2: Example 4.1 circuit illustrating compatible and incompatible scan cells.

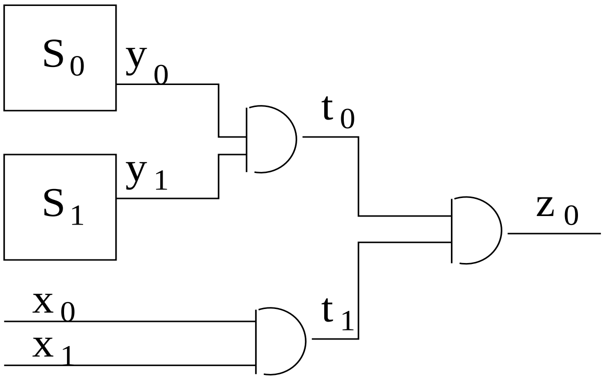


Figure 4.3: Example 4.2 circuit illustrating spurious transitions which cannot be eliminated on t_0.

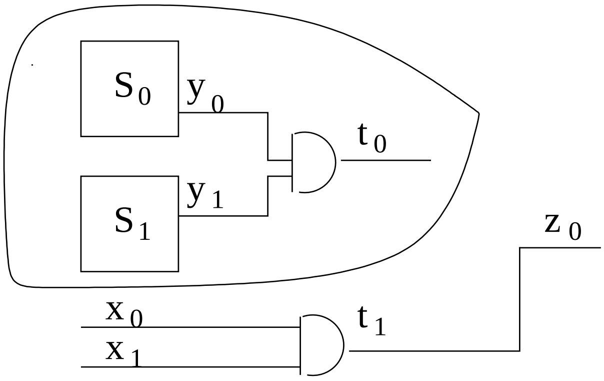


Figure 4.4: Reduced circuit of the example circuit from Figure 4.3 illustrating the steps required to compute extra test vectors.

Example 4.3 For the circuit shown in Figure 4.3 the reduced circuit is generated as follows using the part (a) of the algorithm shown in Figure 4.9 from Section 4.4.1. Initially the signal t_1 at the input of gate z_0 is identified to eliminate spurious transitions that originate from scan cells S_0 and S_1. Then scan cells S_0 and S_1, and the AND gate t_0 are excluded from the reduced circuit as shown in Figure 4.4. Furthermore, gate z_0 is modified to a buffer (signals t_1 and z_0 are identical). The targeted fault in the reduced circuit is t_1 stuck-at 1 ($sa-1$) which eliminates the spurious transitions at gate z_0 in the original circuit. Finally, the extra test vectors (Definition 4.2) that eliminate the spurious transitions during test application are computed $x_0x_1 = \{0X, X0\}$.

A particular case of scan cells are self-incompatible scan cells which are defined as follows.

Definition 4.4 A scan cell S_i is *self-incompatible* if at least one primary input x_k that is assigned value i_k to eliminate the spurious transitions which originate from S_i on one fanout path will propagate the transitions which originate from S_j on a different fanout path.

Now a new question which arises is whether the spurious transitions which originate from self-incompatible scan cells can be eliminated? In order to provide an answer consider the following example.

Example 4.4 Consider the circuit of Figure 4.5 where $\{x_0,x_1\}$ are primary inputs, S_0 is scan cell, y_0 is present state line, and $\{t_0,t_1,t_2\}$ are circuit lines. To eliminate spurious transitions at gate t_0 while shifting out test responses through scan cell S_0, primary input x_0 must be assigned the controlling value 1 of gate t_0. However, to eliminate spurious at gate t_1, primary input x_0 must be assigned the controlling value 0 of gate t_1. Different values must be assigned to x_0 to eliminate spurious transitions which originate from the same scan cell S_0 and hence scan cell S_0 is self-incompatible. However if primary input x_1 is assigned the controlling value 0 of gate t_2 the spurious transitions which originate in S_0 and propagate on path $\{S_0,t_1,t_2\}$ will be eliminated. Therefore by assigning extra test vector $x_0x_1 = \{10\}$, the spurious transitions propagating on both paths $\{S_0,t_0\}$ and $\{S_0,t_1,t_2\}$ will be eliminated. This leads to the conclusion that most of the spurious transitions originating in self-incompatible scan cells can be eliminated by examining the fanout paths of self-incompatible scan cells and assigning a single extra test vector while shifting out the test responses. However, the single extra test vector is at the expense of a small number of spurious transitions that cannot be eliminated as in the case of transitions on line t_1 in the simple circuit of Figure 4.5.

The previous example has shown that following a careful examination of fanout branches of self-incompatible scan cells, most of the spurious transitions originating in self- incompatible scan cells can be eliminated using a single value for the extra test vector. Finally, independent scan cells are introduced.

Definition 4.5 A scan cells S_i is *independent* if all the gates on all the paths which originate from S_i do not have at least one side input which can be justified by primary inputs.

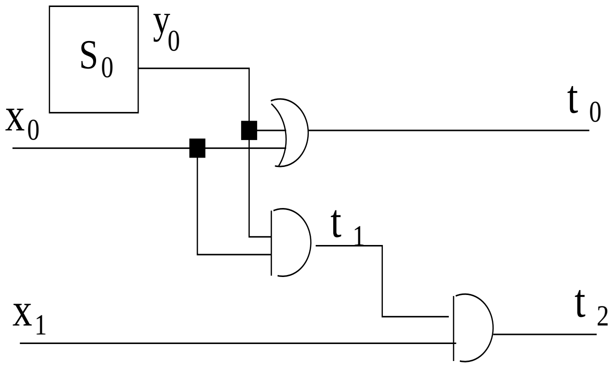


Figure 4.5: Example 4.4 circuit illustrating self-incompatible scan cells.

The independent scan cells are grouped in the extra scan chain (ESC) for which no extra test vector can be computed and hence the spurious transitions cannot eliminated. The following example illustrates independent scan cells.

Example 4.5 Consider the circuit shown in Figure 4.6. Output z_0 depends only on scan cells S_0 and S_1, and the next state y_4' of scan cell S_4 depends only on scan cells S_0, S_1, S_2 and S_3. There are no side inputs of gates t_0 and t_1 that can be justified by primary inputs such that spurious transitions originated from S_0, S_1, S_2 and S_3 are eliminated. Therefore scan cells S_0, S_1, S_2 and S_3 are independent.

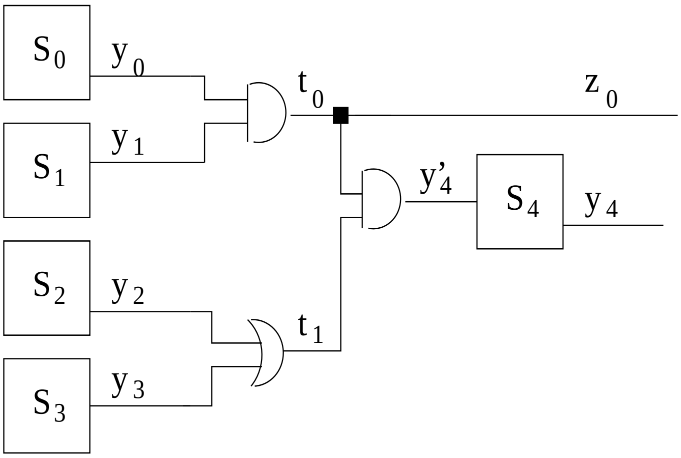


Figure 4.6: Example 4.5 circuit illustrating independent scan cells.

4.3.3 Power Dissipated by the Buffered Clock Tree

Previous research has established that power dissipated in the clock tree is typically one third of the total power dissipation [185] and hence it is necessary to minimise power dissipated in the clock tree not only during functional operation but also during test application. Unlike previous approaches which do not consider power dissipated by the buffered clock tree [46, 49, 88, 145, 172, 189, 192] or gate the clock tree *only* for non-essential test vectors from a large test set [61], the proposed DFT architecture using multiple scan chains (Figure 4.1 from Section 4.3.1) reduces clock tree power for *all* the test vectors of a very small test set where each test vector is essential (i.e. detects at least one fault). This is explained by the following theorem which gives an upper bound on power reduction.

Theorem 4.1 Consider k scan chains in the DFT architecture of Figure 4.1 then the power reduction of the buffered clock tree over the standard full scan architecture is upper bounded by $(k-1)/k$.

Proof: Let $\{m_0, \ldots, m_{k-1}\}$ be the size of each scan chain and $\sum\limits_{i=0}^{k-1} m_i = m$, where m is the total number of scan cells. Since for large dies the clock power dissipation changes from square-root dependence on the number of scan cells to a linear dependence [185] power dissipated by each scan chain SC_i can be approximated to $\lambda \times m_i$ where λ is dependent on clock frequency, supply voltage and wire lengths. The power dissipated while shifting test responses over an entire scan cycle (m clock cycles) for the proposed architecture is $P_{MSC} = \lambda \times \sum\limits_{i=0}^{k-1} m_i^2$ since over m_i clock cycles only the buffered clock tree feeding SC_i is active. On the other hand power dissipated in the traditional full scan architecture is $P_{FULL} = \lambda \times m^2 = \lambda \times (\sum\limits_{i=0}^{k-1} m_i) \times (\sum\limits_{i=0}^{k-1} m_i)$. Therefore the reduction in power dissipation is

$$Red = (P_{FULL} - P_{MSC})/P_{FULL} = 1 - (\lambda \times \sum_{i=0}^{k-1} m_i^2)/(\lambda \times (\sum_{i=0}^{k-1} m_i) \times (\sum_{i=0}^{k-1} m_i)).$$

Following Cauchy-Schwarz inequality [160] where

$$(\sum_{i=0}^{k-1} m_i) \times (\sum_{i=0}^{k-1} m_i) \leq k \times (\sum_{i=0}^{k-1} m_i^2)$$

the power reduction is upper bounded by $Red \leq 1 - 1/k = (k-1)/k$ □.

The previous theorem shows that power reduction of up to $(k-1)/k$ can be achieved in the buffered clock tree, with maximum reduction achieved when scan chains have an equal number of scan cells. It should be noted that by gating the clock of each scan chain not only average power reduction is achieved but also savings in peak power are guaranteed since while shifting out test responses only a single buffered clock tree is active.

4.3.4 Extention of the Proposed DFT Architecture Based on Multiple Scan Chains to Scan BIST Methodology

So far the proposed DFT architecture based on multiple scan chains introduced in Section 4.3.1 was applied to full scan sequential circuits using external automatic test equipment ATE (Figure 1.2 from Section 1.2). This can be summarised in Figure 4.7 where the extra test vectors for scan chains SC_0 and SC_{k-1} are highlighted. Further, it is shown that the test response Y^i_{m-1}, which is the test response in the first scan cell from scan chain SC_{k-1}, has yet to be shifted after the test responses from scan chains $SC_0, \ldots, SC_{k-2}$ were already shifted out.

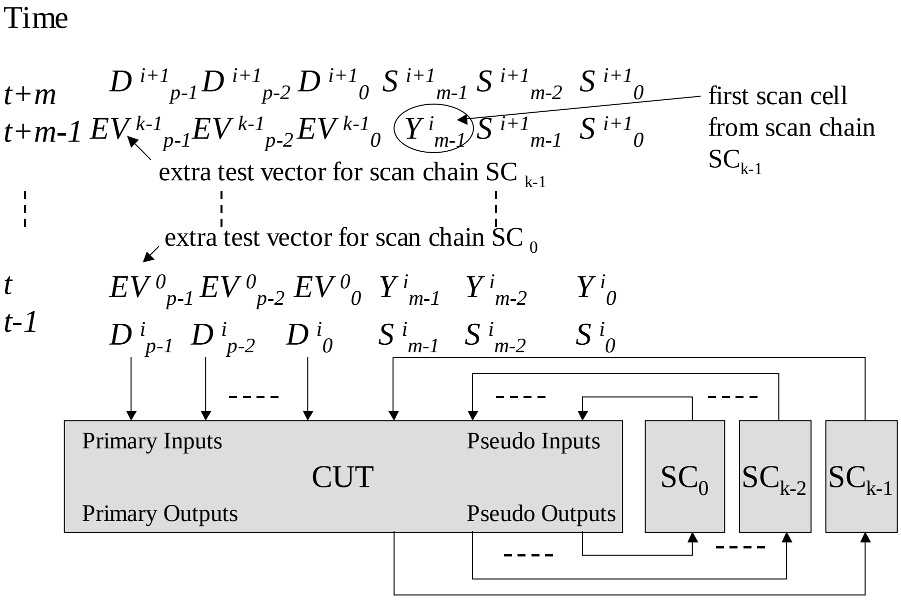

Time
t+m
t+m-1
t
t-1
D^{i+1}_{p-1} D^{i+1}_{p-2} D^{i+1}_{0} S^{i+1}_{m-1} S^{i+1}_{m-2} S^{i+1}_{0}
EV^{k-1}_{p-1} EV^{k-1}_{p-2} EV^{k-1}_{0} Y^{i}_{m-1} S^{i+1}_{m-1} S^{i+1}_{0}
first scan cell from scan chain SC_{k-1}
extra test vector for scan chain SC_{k-1}
extra test vector for scan chain SC_0
EV^{0}_{p-1} EV^{0}_{p-2} EV^{0}_{0} Y^{i}_{m-1} Y^{i}_{m-2} Y^{i}_{0}
D^{i}_{p-1} D^{i}_{p-2} D^{i}_{0} S^{i}_{m-1} S^{i}_{m-2} S^{i}_{0}
Primary Inputs
Pseudo Inputs
CUT
Primary Outputs
Pseudo Outputs
SC_0
SC_{k-2}
SC_{k-1}

Figure 4.7: Summary of the proposed DFT architecture based on multiple scan chains when employing standard scan DFT using external ATE

However, the proposed DFT architecture based on multiple scan chains is not applicable only to standard scan sequential circuits using external ATE. In the following the minor modifications which need to be considered when using scan BIST methodology (Figure 1.5 of Section 1.3.1) are given. Figure 4.8 shows that the serial output of the linear feedback shift register (LFSR) is fed directly into the scan chain which makes the primary inputs directly controllable while shifting out test responses from each scan chain. Therefore, extra test vectors associated with each scan chain can be applied to primary inputs while shifting in the present state part of the next test vector associated with each scan chain. Scan cells are partitioned into multiple scan chains and extra test vectors are calculated in the same way as for scan sequential circuits as described in the following Section 4.4. This will lead to a lower area overhead associated with scan BIST methodology (Figure 1.5 of Section 1.3.1) at the expense of higher interference from ATE which needs to store the primary input part of each test vector and the extra test vector associated with each scan chain.

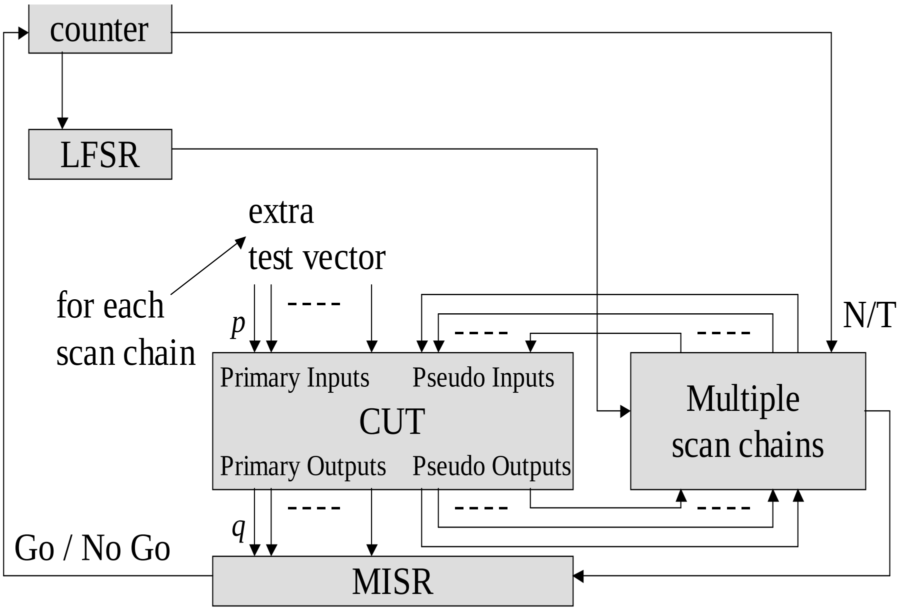


Figure 4.8: Extention of the proposed DFT architecture based on multiple scan chains to scan BIST methodology

4.4 Multiple Scan Chains Generation and New Test Application Strategy

Having described the new DFT architecture based on multiple scan chains and scan cell classification from power dissipation standpoint, now algorithms for multiple scan chain generation are introduced. In Section 4.4.1, partitioning scan cells into multiple scan chains based on their classification is given. Then, in Section 4.4.2 a new test application strategy based on the DFT architecture described in Section 4.3.1, is introduced.

4.4.1 Partitioning Scan Cells into Multiple Scan Chains

Multiple Scan Chain Partitioning (*MSC-PARTITIONING*) algorithm identifies compatible scan cells introduced by Definition 4.2 of Section 4.3.2, groups them in scan chains and computes an extra test vector for each scan chain. Figure 4.9 gives the flow of the proposed *MSC-PARTITIONING* algorithm which is divided in five parts identified in boxes marked from (a) to (e). In order to facilitate the elimination of spurious transitions by computing an extra test vector for each scan chain the initial circuit **C** needs to be transformed to a reduced circuit **C'** (box (a)). A by-product of the reduction procedure is a specified fault list **L** (box (b)) which is targeted by an ATPG process [35] on the reduced circuit **C'** (box (c)). It is interesting to note that within the context of this section, ATPG is *not used* to detect the stuck-at faults of the initial circuit **C** (Section 1.2 from Chapter 1), but it is *used* to compute extra test vectors which when applied to primary inputs while shifting out test responses will mask the circuit activity and hence lead to reduction in power dissipation (Figure 4.1 from Section 4.3.1). Associated with each fault stuck-at non-controlling value nc_i on wire FS_i ($FS_i\ sa-nc_i$) in the specified fault list **L** is a set of scan cells whose spurious transitions will be eliminated in the original circuit **C** by applying extra test vector EV_i which detects $FS_i\ sa-nc_i$ in the reduced circuit **C'**. Therefore based on fault compatibility in the reduced circuit **C'**, scan cell classification in the original circuit **C** is done implicitly which leads to several partitions of the initial single scan chain (box (d)). However, some scan cells may be self-incompatible (Definition 4.4) which leads to iterations through the ATPG process with a respecified fault list (box (e)) until no self-incompatible scan cells are left. At the end of the *MSC-PARTITIONING* algorithm the multiple scan chains $\{SC_0,\ldots,SC_{k-1},ESC\}$ and extra test $\{EV_0,\ldots,EV_{k-1}\}$

will be used by the novel test application strategy described in Section 4.4.2. In the following each part of the *MSC-PARTITIONING* algorithm is explained in detail.

a. In the first part of the *MSC-PARTITIONING* of Figure 4.9 the initial circuit **C** is transformed into a reduced circuit **C'** as described in *CIRCUIT-REDUCTION* algorithm of Figure 4.10. The algorithm also identifies the *freezing signals* which are the signals that depend on primary inputs and should be set to the controlling value as side inputs to the gates which eliminate transitions that originate from scan cells as described in the following parts. Two lists of *eliminated_gates* and *modified_gates* contain the gates which ought to be eliminated and modified respectively in the reduced circuit **C'**. Initially *eliminated_gates* is set to all the scan cells whereas the *modified_gates* is void (lines 1-2). The circuit is traversed in breadth first search order using two lists *current_frontier* and *new_frontier*. While *current_frontier* is set initially to all the scan cells of **C** (line 3), the *new_frontier* initially is void (line 4). In the inner loop (lines 6-13) for all the gates neighbours of the current frontier it is checked where input gates already belong to *eliminated_gates* (i.e. depend on scan cells). If this is the case then the currently evaluated gate is introduced into *eliminated_gates*, removed from *modified_gates* (if applicable) and introduced to *new_frontier*. If at least one input does not belong to *eliminated_gates* then the currently evaluated gate is introduced to *modified_gates*. In the outer loop (lines 5-16) while current frontier is not void (i.e. no more gates need to be eliminated) the inner loop proceeds further. At the end of each iteration of the outer loop *current_frontier* and *new_frontier* are updated (lines 14 and 15). Finally, using the *eliminated_gates* and *modified_gates* the initial circuit **C** is modified to the reduced circuit **C'** (lines 16 and 17) as follows: gates that belong to *eliminated_gates* (depend only on scan cells) are excluded; gates that belong to *modified_gates* (depend on both scan cells and primary inputs) are modified to gates with input signals dependent only on primary inputs (in the case of gates with two inputs of which one is a freezing signal, the gate is modified to a buffer); all the freezing signals identified in the first step are set as primary outputs in the reduced circuit. Freezing signals $\{FS_0, \ldots, FS_{p-1}\}$, which are the outputs of the gates present in the $modified\_gates$, are determined simultaneously with identifying independent scan cells (Definition 4.5). The independent scan cells are grouped into the extra scan chain (ESC) which consists of

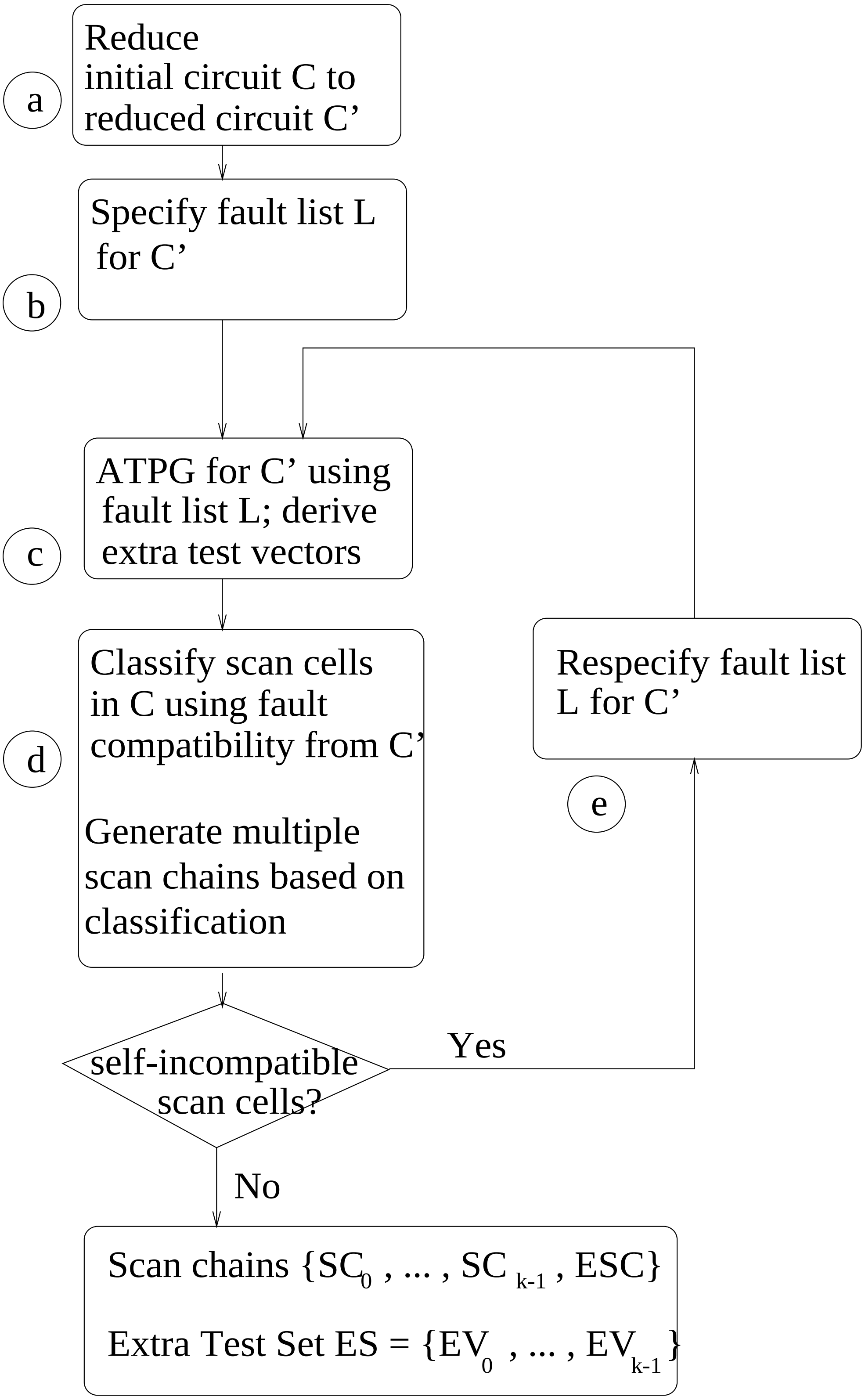

a
Reduce initial circuit C to reduced circuit C'
b
Specify fault list L for C'
c
ATPG for C' using fault list L; derive extra test vectors
d
Classify scan cells in C using fault compatibility from C'
Generate multiple scan chains based on classification
e
Respecify fault list L for C'
self-incompatible scan cells?
Yes
No
Scan chains {SC_0 , ... , SC $_{k-1}$, ESC}
Extra Test Set ES = {EV_0 , ... , EV_{k-1} }

Figure 4.9: Proposed algorithm for partitioning scan cells in multiple scan chains.

scan cells whose spurious transitions cannot be eliminated by computing an extra test vector. The algorithm returns not only the reduced circuit **C'** but also the list of freezing signals that will be used in the following part of the *MSC-PARTITIONING* of Figure 4.9.

b. In the second part a specified fault list **L** is created which will be provided together with the reduced circuit **C'** to an ATPG tool. Specified fault list **L** comprises freezing signals FS_i targeting the stuck-at the non-controlling value $sa-nc_i$ of the gate g_i from *modified_gates* list of algorithm *CIRCUIT-REDUCTION* of Figure 4.10. It is important to note that each fault FS_i $sa-nc_i$ has attached a list of scan cells $\{S_{i_0},\ldots,S_{i_{m-1}}\}$ whose spurious transitions in the initial circuit **C** are eliminated when setting gate FS_i to its controlling value. The list of scan cells is required during generation of scan chains in part (d) of the *MSC-PARTITIONING* algorithm.

c. In the third part, having generated the reduced circuit **C'** and the specified fault list **L**, any state of the art combinational ATPG tool can be used to generate test vectors for the faults from **L** for **C'**. Test vectors for the faults from **L** are the extra test vectors required to eliminate spurious transitions while shifting test responses in the initial circuit **C** as described in part (d). Since the freezing signals are primary outputs in **C'** as described in part (a) then **L** contains faults only on primary outputs. This will clearly speed up the the ATPG process since only backward justification and no forward propagation is required. Moreover, the specified fault list is significantly smaller than the entire fault set which will further reduce ATPG computational time for computing extra test vectors. It should be noted that some faults from **L** are redundant which implies that no extra test vector can be computed to stop the propagation of the spurious transitions from scan cells associated with the respective fault. However, this scan cells are treated as self-incompatible and handled by re-specifying the fault list as described in the last part (e) of the *MSC-PARTITIONING* of Figure 4.9.

d. Given the extra test set with a list of faults from **L** detected by each extra test vector EV_i, scan cell classification according to definitions from Section 4.3.2 is done as follows. If two faults FS_i $sa-nc_i$ and FS_j $sa-nc_j$ from **L** are incompatible (i.e. they are not detected by the same extra test vector) then each element of the two lists

```
ALGORITHM: CIRCUIT-REDUCTION
INPUT: Circuit C
OUTPUT: Reduced Circuit C'
          Freezing Signals $\{FS_0, \ldots, FS_{p-1}\}$

eliminated_gates = $\{S_0, \ldots, S_{m-1}\}$
modified_gates = Ø
current_frontier = $\{S_0, \ldots, S_{m-1}\}$
new_frontier = Ø
while (current_frontier $\neq$ Ø) {
    for all $g_x \in$ neighbours(current_frontier)
        if (all inputs of $g_x \in$ eliminated_gates) {
          add $g_x$ to eliminated_gates
          remove $g_x$ from modified_gates
          add $g_x$ to new_frontier
        }
        else
          add $g_x$ to modified_gates
    current_frontier = new_frontier
    new_frontier = Ø
}
generate reduced circuit C' as follows {
    eliminate all the gates $g_x \in$ eliminated_gates
    modify all the gates $g_y \in$ modified_gates
}
freezing signals $\{FS_0, \ldots, FS_{p-1}\}$ are
output signals of $\{g_0, \ldots, g_{p-1}\}$ = modified_gates
$\{S_{e_0}, \ldots, S_{e_{m-1}}\}$ for which no freezing signal
exists are introduced in the extra scan chain ESC
return Reduced Circuit C'
          Freezing signals $\{FS_0, \ldots, FS_{p-1}\}$
```

Figure 4.10: Proposed algorithm for circuit reduction for extra test vector computation

of scan cells associated with the two faults $\{S_{i_0},\ldots,S_{i_{m-1}}\}$ and $\{S_{j_0},\ldots,S_{j_{q-1}}\}$ respectively, are incompatible (otherwise they are compatible). This leads to grouping all the scan cells, associated with faults detectable by *single* extra test vector, into a *single* scan chain. However, this may lead to self-incompatible scan cells (Definition 4.4 of Section 4.3.2) when different extra test vectors eliminate spurious transitions from the same scan cell. Consequently, while there are self-incompatible the *MSC-PARTITIONING* algorithm will iterate through parts (e), (c), (d) as explained next.

e. In the case that there are self-incompatible scan cells after the generation of multiple scan chains then the problem needs to be addressed as it was briefly explained in example 4.4 of Section 4.3.2. The faults FS_i $sa-nc_i$ which have attached self-incompatible scan cells are removed from fault list **L** and new faults are specified on the lines in the fanout paths of FS_i. Thus, the respecified fault list **L** will be provided back to the ATPG process for computing extra test vectors (part (c)) which will be followed by new multiple scan chain generation based on fault compatibility (part (d)). This iterative process continues until there are no self-incompatible scan cells left.

The *MSC-PARTITIONING* algorithm of Figure 4.9 returns the scan chains of compatible scan cells, the extra scan chain ESC and the extra test set of extra test vectors used to define a new test application strategy, as explained in the following Section 4.4.2. It should be noted that scan cells are partitioned into multiple scan chains and extra test vectors are computed without any knowledge of the test set to be applied to achieve the required fault coverage. Therefore, computational time for circuit reduction depends only on the circuit size and structure (number of scan cells and circuit depth) and not on the size of the test set, which makes the proposed solution test set independent and applicable to large scan sequential circuits within low computational time. This is also due to the fact that although ATPG, used for detecting compatible faults in the reduced circuit, is NP-hard [2, 58], efficient heuristics [83, 84, 111] have been developed that could easily be integrated in the *MSC-PARTITIONING* algorithm. Note that the low computational time for large scan sequential circuits is achieved with small overhead in test area and test data which is also dependent only on the number of scan chains determined by the proposed *MSC-PARTITIONING* algorithm.

4.4.2 New Test Application Strategy Using Multiple Scan Chains and Extra Test Vectors

Having partitioned the scan cells into multiple scan chains with an extra test vector for each scan chain (Section 4.4.1), this section introduces a new test application strategy for power minimisation during test application in scan sequential circuits. *Node transition count* $NTC = \sum_{for\ all\ gates\ G} N_G \times C_{load}$ is used as quantitative measure for power dissipation throughout the section. N_G is the total number of gate output transitions ($0 \rightarrow 1$ and $1 \rightarrow 0$) and it is assumed that load capacitance for each gate is equal to the number of fanouts (Equation 3.1 from Section 3.2.1).

Multiple Scan Chain Test Application (*MSC-TEST APPLICATION*) algorithm computes the NTC during the entire test application period for the given test set **S**, circuit **C**, multiple scan chains $\{\mathbf{SC_0}, \ldots, \mathbf{SC_{k-1}}, \mathbf{ESC}\}$, and extra test set **ES** = $\{EV_0, \ldots, EV_{k-1}\}$. Figure 4.11 gives the pseudocode of the proposed *MSC-TEST APPLICATION* algorithm. The value of **NTC** is 0 at the beginning of the algorithm and it is gradually increased as the entire test set is traversed. The outer loop represents the traversal of all the test vectors V_i, with $i = 0, \ldots, n-1$, from test set **S**. Shifting out test responses through all the scan chains are then considered in the inner loop. For each scan chain SC_j, circuit **C** is simulated by applying the extra test vector EV_j to primary inputs and $NTC_{i,j}$ is added to **NTC**. $NTC_{i,j}$ stands for node transition count while shifting in present state part of test vector V_i through scan chain SC_j and applying extra test vector EV_j to the primary inputs. After shifting out the test responses though each scan chain SC_j the primary input part of test vector V_i is applied to primary inputs and $NTC_{i,ESC}$ is computed while shifting out test response through the extra scan chain ESC. Finally the entire test vector V_i is applied to the circuit under test and $NTC_{i,LOAD}$ required to load the test response in the scan cells, is added to **NTC**. After the completion of the inner loop, the outer loop continues until the entire test set is examined. The algorithm returns the value of **NTC** over the entire test application period. It should be noted that algorithms presented in this section are independent of test vector and scan cell order. Unlike the algorithms from [49, 88, 145, 189] whose computational time is prohibitively large hindering the exploration for large sequential circuits, the proposed *MSC-PARTITIONING* and *MSC-TEST APPLICATION* algorithms have low computational time and can handle large circuits as shown in the following Section 4.5.

```
ALGORITHM: MSC-TEST APPLICATION
INPUT: Test Set $\mathbf{S}=\{V_0,\ldots,V_{n-1}\}$, Circuit $\mathbf{C}$
       Scan Chains $\{\mathbf{SC_0},\ldots,\mathbf{SC_{k-1}},\mathbf{ESC}\}$
       Extra Test Set $\mathbf{ES} = \{EV_0,\ldots,EV_{k-1}\}$
OUTPUT: Node transition count NTC

NTC ← 0
for every $V_i$ from S with $i = 0,\ldots,n-1$ {
    for every $SC_j$ with $j = 0,\ldots,k-1$ {
        apply $EV_j$ to primary inputs;
        $EV_j$ is uniquely determined for
        every scan chain $SC_j$ by the
        MSC-PARTITIONING algorithm of
        Figure 4.10
        compute $NTC_{i,j}$ by simulating C when
        shifting in the present state part of test
        vector $V_i$ through scan cells from $SC_j$
        NTC ← NTC + $NTC_{i,j}$
    }
    compute $NTC_{i,ESC}$ by simulating C when
    shifting in the present state part of test
    vector $V_i$ through independent scan cells
    from ESC
    NTC ← NTC + $NTC_{i,ESC}$
    apply primary part of $V_i$ to primary inputs
    to get the circuit test response and
    compute $NTC_{i,LOAD}$
    NTC ← NTC + $NTC_{i,LOAD}$
}
return NTC
```

Figure 4.11: Proposed test application strategy using multiple scan chains and extra test vectors

4.5 Experimental Results

This section demonstrates through a set of benchmark examples that multiple scan chains combined with extra test vectors, as outlined in Section 4.3, yield savings in power dissipation during test application. The algorithms described in Section 4.4 were implemented on a 500 MHz Pentium III PC with 128 MB RAM running Linux and using GNU CC version 2.91. Section 4.5.1 shows the reduction in power dissipation at the expense of low overhead in test area and volume of test data, when the proposed multiple scan chains-based technique is employed for power minimisation. Section 4.5.2 provides a comparison with the BPIC test application strategy presented in Chapter 3.

4.5.1 Experimental Results for Multiple Scan Chains-Based Power Minimisation

The average value of *NTC* reported throughout this section is calculated using the Equation 3.1 from Chapter 3 under the assumption of the zero delay model. The use of the zero delay model is motivated by the observation that power dissipation under the zero delay model has a high correlation with power dissipation under the real delay model [176]. Furthermore, due to elimination of spurious transitions the propagation of hazards and glitches is also eliminated leading to even greater reductions for power dissipation in the case of real delay model, as explained in Section 3.5 from Chapter 3.

First column of Table 4.1 gives the number of scan cells of all full scan sequential circuits from ISCAS89 benchmark set [23]. Second and third columns give the number of scan chains (SC) and the length of the extra scan chain (ESC) respectively computed using the *MSC-PARTITIONING* algorithm outlined in Section 4.4.1. The number of scan chains varies from 2 as in the case of *s208* up to 7 as in the case of *s38584*. The small number of scan chains implies that both area overhead required to control multiple scan chains and volume of test data overhead caused by extra test vectors are very low since they are proportional to the number of scan chains. For most of the examples the size of the extra scan chain (ESC length) is nil or very low. However, there are two extreme cases as in the case of *s13207* and *s38417* where the number of independent scan cells (Definition 4.5 from Section 4.3.1) is very high leading to an increase in ESC length and hence insignificant penalty in power reduction. Furthermore, the computational time is

circuit	Scan Cells	Scan Chains (SC)	ESC length	CPU time (s)
s208	8	2	0	1
s298	14	3	6	1
s344	15	4	4	1
s349	15	4	1	1
s382	21	3	6	1
s386	6	3	0	1
s400	21	3	6	1
s420	16	2	0	1
s444	21	4	6	1
s510	6	4	0	1
s526	21	4	6	1
s641	19	3	0	1
s713	19	3	0	1
s820	5	5	0	1
s832	5	3	0	1
s838	32	2	0	1
s953	29	3	23	1
s1196	18	4	2	1
s1238	18	4	2	1
s1423	74	5	3	2
s1488	6	3	0	3
s1494	6	4	0	3
s5378	179	5	33	49
s9234	211	6	20	201
s13207	638	5	330	472
s15850	534	6	62	596
s35932	1728	2	0	1903
s38417	1636	5	1079	8151
s38584	1426	7	7	3543

Table 4.1: Experimental results for ISCAS89 benchmark circuits [23] in terms of number of scan chains, extra scan chain (ESC) length, and CPU time, when applying *MSC-PARTITIONING* algorithm from section 4.4.1.

very low ($< 1s$) for small circuits. Moreover, for large circuits which are not handled by previous approaches [49, 88, 189] and the best primary input change (BPIC) time test application strategy proposed in previous Chapter 3, as in the case of *s38584*, it takes $< 3600s$ to achieve substantial reduction in average value of NTC.

Table 4.2 shows the experimental results for all the circuits from ISCAS89 bench-

circuit	Test Vectors	traditional	proposed
	(TV)	*NTC*	*NTC*
s208	34	54.54	26.67
s298	33	103.56	39.23
s344	24	130.36	42.54
s349	22	131.90	52.59
s382	32	133.91	50.99
s386	74	81.31	63.75
s400	33	135.97	51.88
s420	73	111.69	54.46
s444	33	139.92	47.68
s510	60	123.89	64.38
s526	60	170.61	63.05
s641	58	166.32	60.03
s713	58	173.34	57.15
s820	110	137.52	111.08
s832	115	139.83	115.50
s838	148	227.46	108.24
s953	90	158.50	76.43
s1196	140	101.31	68.12
s1238	151	101.50	65.15
s1423	70	453.58	137.63
s1488	119	340.75	225.81
s1494	125	329.98	266.05
s5378	259	1772.07	527.87
s9234	366	3160.16	760.35
s13207	461	5949.81	2051.55
s15850	436	5260.90	942.07
s35932	65	11067.50	5440.19
s38417	904	15920.00	7159.88
s38584	658	12766.30	3914.41

(a) ATALANTA test set [111]

Table 4.2: Comparison in *NTC* when using the proposed multiple scan chains and the traditional single scan chain [2].

mark set [23] using three different ATPG test tools [83, 84, 111], to further validate the advantages of the proposed test set independent technique. The first and second columns of Table 4.2(a) give the circuit name and the number of test vectors (TV) respectively generated using the ATALANTA test tool [111]. Third column shows the initial average

circuit	Test Vectors (TV)	traditional *NTC*	proposed *NTC*
s208	65	55.82	25.93
s298	52	115.74	46.36
s344	62	131.58	48.03
s349	65	132.58	53.63
s382	72	145.63	51.81
s386	109	86.31	58.92
s400	98	107.24	43.82
s420	98	107.24	43.82
s444	77	150.01	49.74
s510	90	115.23	65.86
s526	107	186.24	67.38
s641	99	184.13	60.31
s713	100	196.92	59.92
s820	190	139.01	110.31
s832	200	138.07	114.29
s838	183	199.88	81.15
s953	138	169.70	76.37
s1196	227	105.37	68.37
s1238	240	107.46	66.11
s1423	135	509.96	150.41
s1488	196	347.17	227.33
s1494	191	353.12	237.43
s5378	358	1786.60	531.44
s9234	660	3123.35	754.58
s13207	709	5972.92	2056.18
s15850	643	5487.29	952.92
s35932	129	13039.30	6291.21
s38417	1458	15849.20	7136.23
s38584	989	12871.30	3912.55

(b) ATOM test set [83]

Table 4.2: Comparison in NTC when using the proposed multiple scan chains and the traditional single scan chain [2].

value of NTC (traditional NTC), which is the total value of NTC using the traditional single scan chain design [2] and ALAP test application strategy (Definition 3.3 in Chapter 3) divided by the total number of clock cycles over the entire test application period. The next column 4 shows the final average value of NTC (proposed NTC) when using

circuit	Test Vectors (TV)	traditional *NTC*	proposed *NTC*
s208	27	54.94	24.07
s298	23	108.88	43.81
s344	13	124.77	46.69
s349	13	128.91	55.86
s382	25	148.40	55.54
s386	63	85.45	59.09
s400	43	100.60	40.11
s420	43	100.60	40.11
s444	24	156.59	51.87
s510	54	114.04	66.14
s526	49	183.15	67.95
s641	21	176.95	62.92
s713	21	192.82	63.50
s820	93	137.89	112.44
s832	94	138.61	115.29
s838	75	187.63	70.41
s953	76	169.09	76.02
s1196	113	105.47	68.83
s1238	121	103.88	65.24
s1423	20	507.21	150.82
s1488	101	366.18	234.11
s1494	100	371.16	235.81
s5378	97	1809.42	537.51
s9234	105	3045.64	751.093
s13207	233	5977.48	2047.51
s15850	94	5481.82	947.82
s35932	12	10860.50	5374.45
s38417	68	14199.90	6486.23
s38584	110	12901.50	3896.92

(c) MINTEST test set [84]

Table 4.2: Comparison in *NTC* when using the proposed multiple scan chains and the traditional single scan chain [2].

the proposed multiple scan chains and extra test vectors (*MSC-TEST APPLICATION* algorithm from Section 4.4.2). The same experiment was completed for non-compact test sets generated by ATOM test tool [83] (Table 4.2(b)) and compact test sets generated by MINTEST compaction tool [84] (Table 4.2(c)) respectively. It should be noted that all the

circuit	power reduction (%)		
	ATALANTA	ATOM	MINTEST
s208	51.09	53.54	56.18
s298	62.10	59.94	59.75
s344	67.36	63.49	62.57
s349	60.12	59.54	56.66
s382	61.91	64.42	62.57
s386	21.59	31.73	30.84
s400	61.84	59.13	60.12
s420	51.23	59.13	60.12
s444	65.91	66.84	66.87
s510	48.03	42.84	42.01
s526	63.04	63.82	62.89
s641	63.90	67.24	64.43
s713	67.02	69.57	67.06
s820	19.22	20.64	18.45
s832	17.39	17.22	16.82
s838	52.41	59.39	62.47
s953	51.77	55.03	54.99
s1196	32.75	35.11	34.74
s1238	35.81	38.47	37.19
s1423	69.65	70.50	70.26
s1488	33.73	34.51	36.06
s1494	19.37	32.76	36.46
s5378	70.21	70.25	70.29
s9234	75.93	75.33	75.84
s13207	65.51	65.57	65.74
s15850	82.09	82.63	82.70
s35932	50.84	51.75	50.51
s38417	55.02	54.97	54.32
s38584	69.33	69.60	69.79

(a) Power reduction

Table 4.3: Comparison of experimental data for ATALANTA [111], ATOM [83], and MINTEST [84] test sets.

three test sets [83, 84, 111] achieve 100% fault coverage. It can be clearly seen that the proposed test application strategy (*MSC-TEST APPLICATION* from Section 4.4.2) has significantly smaller average value of NTC for all the benchmark circuits when compared

circuit	volume of test data overhead (%)			test area overhead (%)
	ATALANTA	ATOM	MINTEST	
s208	3.26	1.70	4.11	10.00
s298	1.06	0.67	1.53	11.33
s344	4.68	1.81	8.65	11.11
s349	5.11	1.73	8.65	11.11
s382	0.78	0.34	1.00	9.09
s386	2.18	1.48	2.56	10.33
s400	0.75	1.08	2.46	10.63
s420	1.45	1.08	2.46	9.52
s444	1.13	0.48	1.56	11.04
s510	5.06	3.37	5.62	10.52
s526	0.62	0.35	0.76	11.11
s641	3.35	1.96	9.25	8.00
s713	3.35	1.94	9.25	8.00
s820	3.55	2.05	4.20	12.50
s832	2.04	1.17	2.49	10.71
s838	0.69	0.56	1.37	4.65
s953	0.51	0.79	0.93	6.81
s1196	0.93	0.57	1.16	6.81
s1238	0.86	0.54	1.08	4.16
s1423	1.06	0.55	3.73	3.79
s1488	1.44	0.87	1.69	4.16
s1494	1.82	1.19	2.28	6.12
s5378	0.25	0.18	0.67	2.60
s9234	0.11	0.19	0.69	2.06
s13207	0.07	0.04	0.15	0.91
s15850	0.14	0.09	0.67	0.92
s35932	0.06	0.03	0.33	0.25
s38417	0.01	0.01	0.09	0.38
s38584	0.02	0.01	0.14	0.42

(b) Overhead in volume of test data and test area

Table 4.3: Comparison of experimental data for ATALANTA [111], ATOM [83], and MINTEST [84] test sets.

to initial value of NTC computed using the traditional test application strategy from [2] which employs a single scan chain.

To give an indication of the reductions in power dissipation, Table 4.3 shows the per-

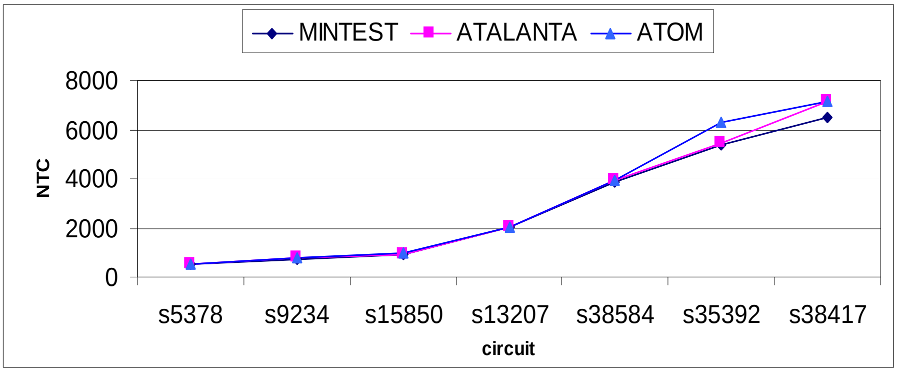


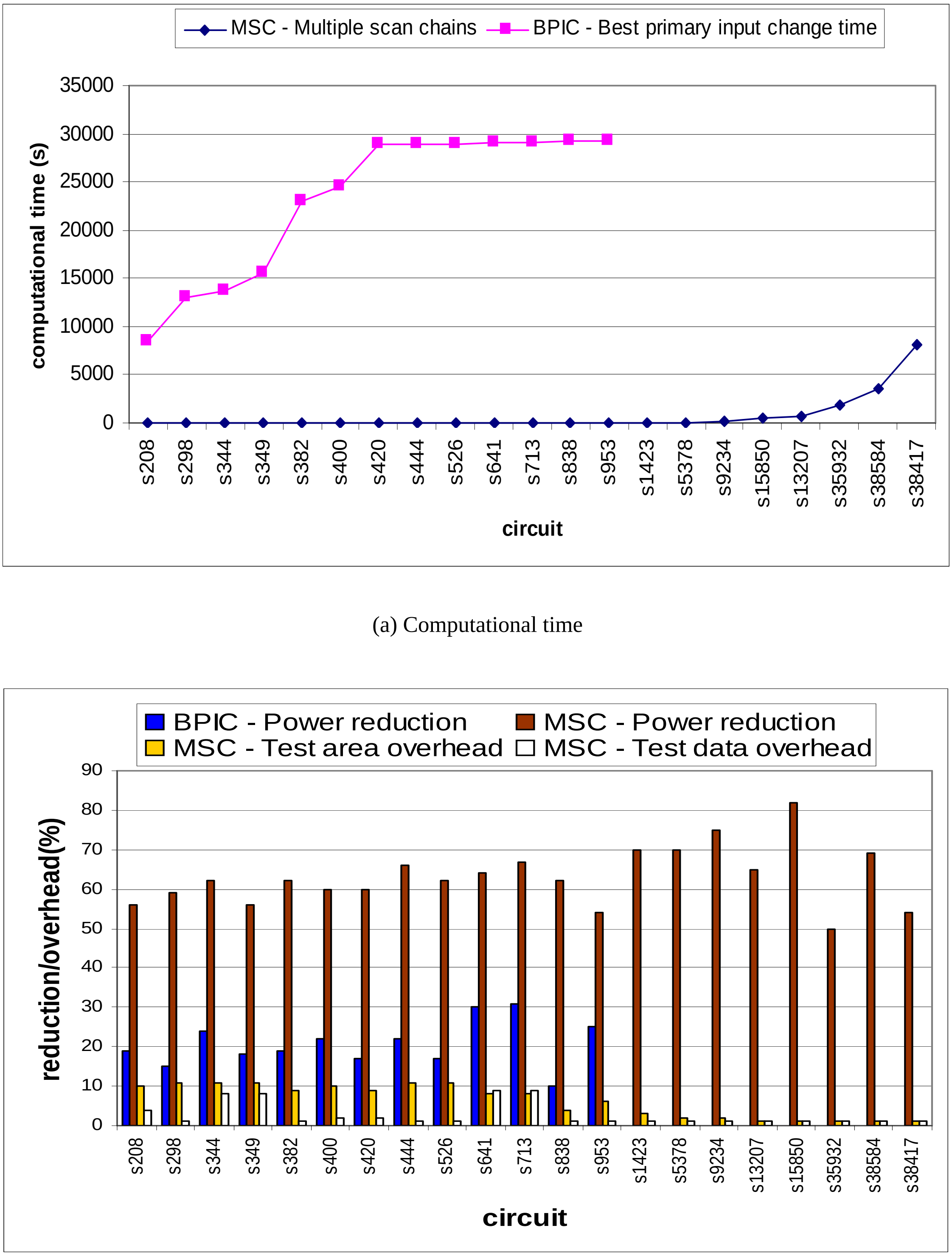
Figure 4.12: Curve illustrating the test set independent final value of NTC.

centage reduction in power dissipation (Table 4.3(a)) and percentage overhead in volume of test data (columns 2-4 of Table 4.3(b)) and test area (column 5 of Table 4.3(b)). The power dissipation is considered directly proportional to the average value of NTC. The test area overhead represents the extra logic required to multiplex the scan output signal (Figure 4.1) and it is computed accurately by synthesising and technology mapping the ISCAS89 circuits to AMS 0.35 micron technology [9]. The volume of test data overhead represents the number of extra bits required for the extra test vectors (the number of scan chains multiplied by the number of primary inputs). Note that test area overhead decreases as the complexity of the circuit increases. This is due to the fact that extra area occupied by scan control register and selection logic (Figure 4.1 from Section 4.3.1) required to control multiple scan chains is very small when compared to the size of large sequential circuits. The power reduction varies from approximately 82% as in the case of *15850* down to under 17% as in the case of *s832* when employing MINTEST [84]. It should be noted that moderate power reduction as in the case of *s386*, *s510*, *s820*, *s832*, *s1488*, *s1494* is due to very small number of scan cells (5 to 6 scan cells only as shown in Table 4.1) which are difficult to be partitioned in multiple scan chains. However, for modern complex digital circuits where the number of scan cells is significantly higher (1426 as in the case of *s38584*) the power reduction is up to 69% at the expense of insignificant $<$ 1% volume of test data and test area overhead. This clearly shows the advantage of the proposed power minimisation technique for large scan sequential circuits.

A further advantage of the proposed technique is that due to its test set independence the final average value of NTC is predictable within a given range of values regardless of test vectors applied to the circuit. This is justified by the fact that the proposed low overhead area multiple scan chain architecture introduced in Section 4.1 is not overly sensitive to the values of test vectors since only a single chain is active at a time and the spurious transitions within the combinational circuit are eliminated by the extra primary input vector *regardless* of the value loaded in non-active scan chains. This is shown in Figure 4.12 where the graphs for average value of NTC for for 7 largest ISCAS89 benchmark under three different size test sets are given. For all three test sets MINTEST [84], ATALANTA [111] and ATOM [83] the average values of NTC are are approximately equal. This implies that the proposed technique can further be applied to more DFT methodologies such as scan-based BIST [2] where regardless of the value of the pseudorandom test set the savings in power dissipation are guaranteed and final values of NTC are predictable.

4.5.2 Comparing Multiple Scan Chains-Based Power Minimisation and Best Primary Input Change Test Application Strategy

In order to show the suitability of the new multiple scan chain-based technique for large scan sequential circuits, a comparison, in terms of power reduction, overhead in volume of test data, test area, and computational time, for BPIC test application strategy proposed in Chapter 3 and multiple scan chains-based technique, is given in Figure 4.13. It can be concluded from Figure 4.13(a), in the case of large sequential circuits which are infeasible for computing the best primary input change time using the test set dependent approach from Chapter 3, the test set independent solution presented in Section 4.3 is applicable with low computational time. For example, BPIC is computationally intractable for all the large scan sequential circuits, while multiple scan chains-based power minimisation yields considerable savings in power dissipation with low overhead in volume of test data and test area. For example circuit *s953*, the 25% savings in power dissipation in the case of BPIC (Chapter 3) are significantly smaller when compared to 54% savings in the case of multiple scan chains-based technique. This is achieved at the expense of additional overhead of 6% in test area, and 1% in volume of test data, as shown in Figure 4.13(b). However, for small to medium sized circuits where design space exploration of a small number of scan cell and test vector orderings is feasible within reasonable computational limits, BPIC is applicable since it has no penalty in test area, performance, test efficiency, test application time or volume of test data.



(a) Computational time

(b) Power reduction and overhead in test area and test data

Figure 4.13: Comparison in power reduction, overhead in volume of test data and test area, and computational time for BPIC test application strategy introduced in chapter 3 and the newly proposed multiple scan chains (MSC) technique, when using MINTEST [84].

4.6 Concluding Remarks

This chapter has presented a new technique based on multiple scan chains and it has shown how with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, considerable savings in power dissipation during test application in large scan sequential circuits can be achieved in very low computational time. The technique is based on a new DFT architecture and a novel test application strategy. When compared to traditional approach which consists of a single scan chain [2], the proposed technique reduces spurious transitions (Definition 4.1 from Section 4.3.2) in circuit under test leading to substantial reduction in power dissipation. The proposed technique is test set independent with no penalty in test application time. Substantial power savings are achieved for both compact and non-compact test sets, as shown in Section 4.5, with no penalty in test efficiency. The newly introduced DFT architecture (Figure 4.1 from Section 4.3.1) does not introduce any penalty in performance, and it yields savings in clock tree power dissipation for *all* the test vectors of a very small test set where each test vector is essential, as described in Section 4.3.3. The proposed DFT architecture requires low overhead in test area to control multiple scan chains, which are successfully combined with extra test vectors in the newly introduced test application strategy in Section 4.4.2. Since a high number of extra test vectors [189] emerges as a problem to testers which need to change to support the large volume of test data [94], the proposed technique based on a small number of extra test vectors introduces low overhead in volume of test data as shown in Section 4.5. Moreover, due to the efficient algorithms described in Section 4.4 the proposed technique is computationally inexpensive, which makes it suitable for large sequential circuits. Finally, the synthesisable extra hardware required by the new DFT architecture introduced in Section 4.3.1, the efficient algorithms given in Section 4.4.1, and the novel test application strategy described in Section 4.4.2, make the technique proposed in this chapter easily embeddable in the existing VLSI design flow using state of the art third party electronic design automation tools.

Chapter 5

New BIST Methodology for RTL Data Paths Based on Test Compatibility Classes

Chapters 3 and 4 considered power minimisation during test application in scan sequential circuits at the logic level of abstraction. Following a natural trend the interest of researchers has shifted to the investigation of testing [54] and power minimisation techniques that account for power dissipation at higher levels of abstraction during the early stages of the VLSI design flow [116]. This is to avoid unnecessary iterations in the VLSI design flow. Further, it was shown in [158] that greatest power savings are achieved at higher levels of abstraction. However, to maintain the benefits of low power design at RTL and to achieve reliable in field testing, new techniques for power minimisation during test application for RTL data paths synthesised using low power high level synthesis algorithms [30, 96, 103, 104, 105, 106, 107] are needed.

Prior to investigating power minimisation techniques for testing low power circuits at RTL it is important to provide a novel BIST methodology which will overcome the problems with traditional BIST embedding methodologies [114, 115]. Unlike scan cells which are inserted at logic level of abstraction as shown in Figure 3.1 of Chapter 3, the complexity of designs that employ BIST methodology makes BIST hardware insertion particularly suitable at RTL. This is illustrated in Figure 5.1 where the initial design is specified in a hardware description language (HDL) at register-transfer level of the VLSI design flow. BIST hardware, which includes test registers and a BIST controller as described in Section 1.3.2 (Figure 1.10(b)), is inserted at RTL. This makes it possible for RTL synthesis to translate the initial design into a *BIST network of logic gates prior* to

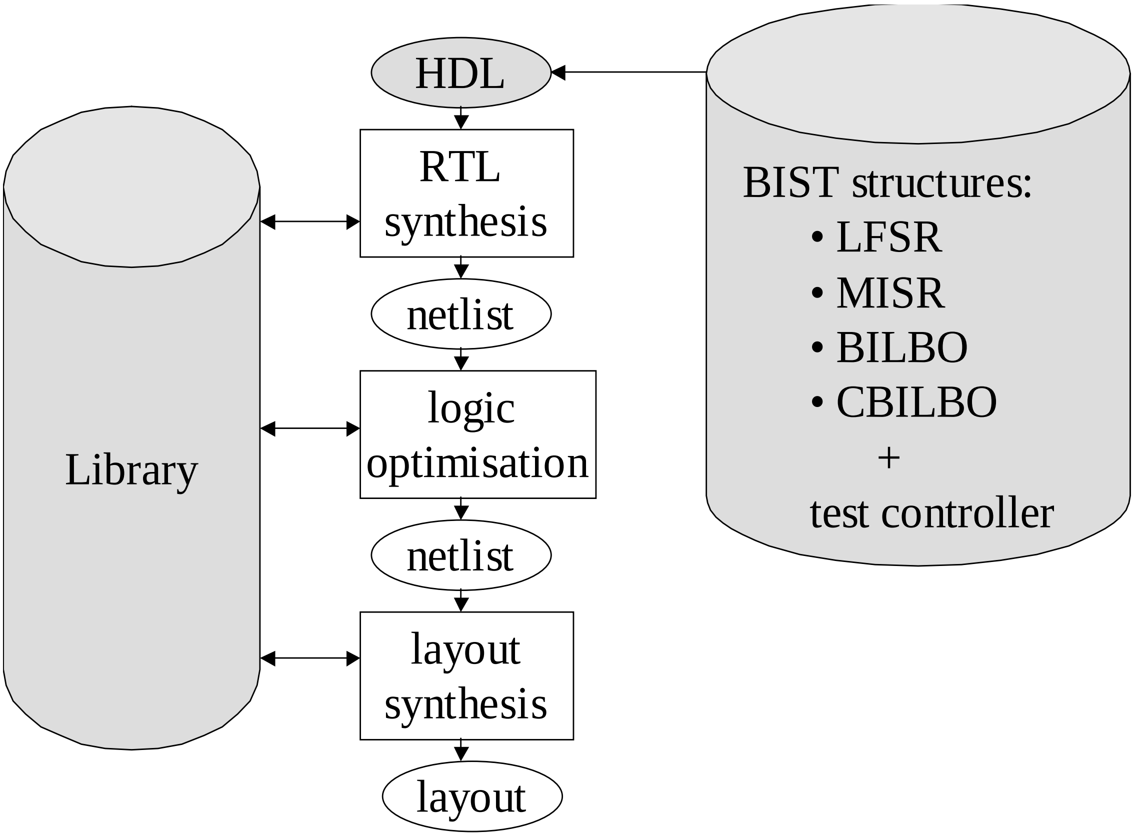


Figure 5.1: Register-transfer level (RTL) built-in self-test (BIST) hardware insertion.

logic optimisation which satisfies the area and delay constraints and prepares the design for the physical design automation tools. The aim of this chapter is to introduce a new BIST methodology for RTL data paths using a novel concept called test compatibility classes (TCC) which reduces the problems associated with traditional BIST embedding methodology such as high test application time, BIST area overhead, performance degradation, volume of test data, fault-escape probability and large computational time required for testable design space exploration. The rest of the chapter is organised as follows. Considering testability at RTL, motivation, and objectives of the proposed research are given in Section 5.1. Section 5.2 introduces the TCC grouping methodology. BIST hardware synthesis for TCC grouping is given in Section 5.3. Experimental results of benchmark and complex hypothetical data paths are presented in Section 5.4. Finally, concluding remarks are given in Section 5.5.

5.1 Considering Testability at Register-Transfer Level

RTL is the abstraction level of the VLSI design flow where an integrated circuit is seen as sequential logic consisting of registers and functional units that compute the next state given the current memory state. The functional units which compute the next state logic are arithmetic logic units (ALU), multipliers or complex multi-functional library modules. The complexity of modern digital circuits requires automated synthesis and optimisation techniques that can explore a wide class of implementations choices using computer-aided design (CAD) tools [51]. High level synthesis is the process of generating RTL structure from a behavioural description [129]. The modules (functional units) allocated by high level synthesis algorithms are generated by module generators which are able to synthesise the layout of modules with high performance and device density. The modules are placed in module libraries and have identical physical information. Given the complexity of modern digital circuits it is necessary that testability is addressed at RTL due to fewer elements than at the logic level which makes test synthesis and test scheduling problems more tractable.

5.1.1 Previous Work

Two main approaches were proposed to enhance the testability of digital circuits at RTL. The first approach is aimed at minimising the complexity of automatic test pattern generation (ATPG). In [36] the complexity of ATPG for scan-based design for testability (DFT) method is reduced by an efficient selection of scan flip-flops using RTL information. The high test application time associated with scan-based technique is overcome by using scan chain reconfiguration to reduce shifting time [135] and use of partial scan design of RTL circuits [80]. However, a significant disadvantage of the scan-based technique is that at-speed testing with the complete test set is not possible (i.e., all test patterns cannot be applied at the operational speed of the circuit). To solve at-speed testability, nonscan DFT techniques applicable to RTL data paths were proposed in [53]. Instead of selecting flip-flops to make controllable/observable as the conventional scan-based techniques, execution units are selected using an execution unit graph. Performing at-speed testability and reducing test area overhead is achieved at the cost of expensive test pattern generation phase. To reduce the costs of test pattern generation, an algorithm that adds minimum test hardware in order to ensure that all the embedded modules in the circuit are hierarchically testable was presented in [64]. In [65] a technique for extracting functional (control/data flow) information from RTL controller/data path is presented, thus avoiding the use of high level information [64]. Recently in [118] a testability analysis methodology for modular designs is introduced which extracts a set of justification and propagation requirements based on the cone-of-logic of each input and output. However, despite reducing both area overhead and ATPG complexity the test application time and the volume of test data are still high.

The second approach to enhance testability of RTL circuits is built-in self-test (BIST) which overcomes the problems of ATPG, test application time and volume of test data [4]. BIST hardware synthesis at RTL can be further subdivided to functional-based and structural-based BIST hardware synthesis. Functional-based BIST hardware synthesis based on algorithmic and deterministic BIST scheme was presented in [76].This algorithm uses a high level cell fault model, and data paths are assumed to be composed of only specific adders/subtracters and multipliers. Combination of different BIST schemes and reusing pre-existing modules of the data path for functional-based BIST hardware synthesis under heterogenous test schemes was proposed in [21]. Another functional-based BIST hardware synthesis [63] uses the controller netlist to extract the test control/data flow to derive a set of symbolic justification and propagation paths. In [164] regular expression based high level symbolic testability analysis further reduces test area overhead under delay constraints by carefully selecting a small subset of registers to serve as test pattern generators and output response analysers. Recently, redundancy identification and testability improvement of digital filter data paths was proposed in [79] which restricts to circuits which are described as a network of shift, add, delay, sign-extension and truncation elements. All the previous functional-based BIST hardware synthesis techniques [21, 63, 76, 79, 164] depend strongly on the functional information of data path modules and/or high level control/data flow. Therefore they can not deal with complex multi-functional library modules used in custom applications [57] and for complex data paths the volume of test data is excessively large. On the other hand, structural-based BIST hardware synthesis inserts test registers by analysing interconnections between registers and modules in a given RTL netlist, without using the functional information of data path modules or high level control/data flow. This makes structural-based BIST hardware synthesis more suitable at RTL than functional-based BIST hardware synthesis.

An early structural-based BIST hardware synthesis algorithm at RTL was presented in [28] without taking into account the test application time. Another structural-based BIST hardware synthesis algorithm that minimises test application time and BIST area overhead was proposed in [14]. The algorithm, however, has an inefficient testable design space exploration due to fixed test resource allocation, which means that the test hardware is allocated before the test scheduling process. Furthermore, the optimisation algorithm limits the number of test plans to only four per module, leading to limited number of explored testable designs. To overcome the fixed test resource allocation, simultaneous test hardware insertion and test scheduling was proposed in [115]. While previous test scheduling algorithms [33, 47, 92] assumed fixed test resource allocation, the work in [115] presented an incremental test scheduling procedure which overcomes the limited testable design space exploration encountered with fixed test resources. Despite its good performance, the algorithm in [115] is not capable of dealing in low computational time with complex designs such as 32-point discrete cosine transform (DCT), since a branch and bound-based algorithm is employed to explore the testable design space.

5.1.2 Motivation and Objectives

Up to this point, the described structural-based BIST hardware synthesis algorithms have assumed the BIST embedding methodology where every module port is embedded between a test pattern generator and a signature analysis register (Section 1.3.2 of Chapter 1). This methodology is inefficient due to the following four problems:

a. to achieve low test application time, a high number of test registers is required which leads to a large BIST area overhead and performance degradation.

b. since every module belongs to a different BIST embedding, aliasing can occur for every module tested separately leading to an increase in fault-escape probability for the entire data path.

c. the increased number of signature analysis registers yields a large volume of test data and increases the test application time due to the time required to shift out test responses.

d. the huge size of the testable design space where test synthesis and test scheduling are strictly interrelated (Figure 1.9 from Chapter 1) leads to long computational time for efficient testable design space exploration.

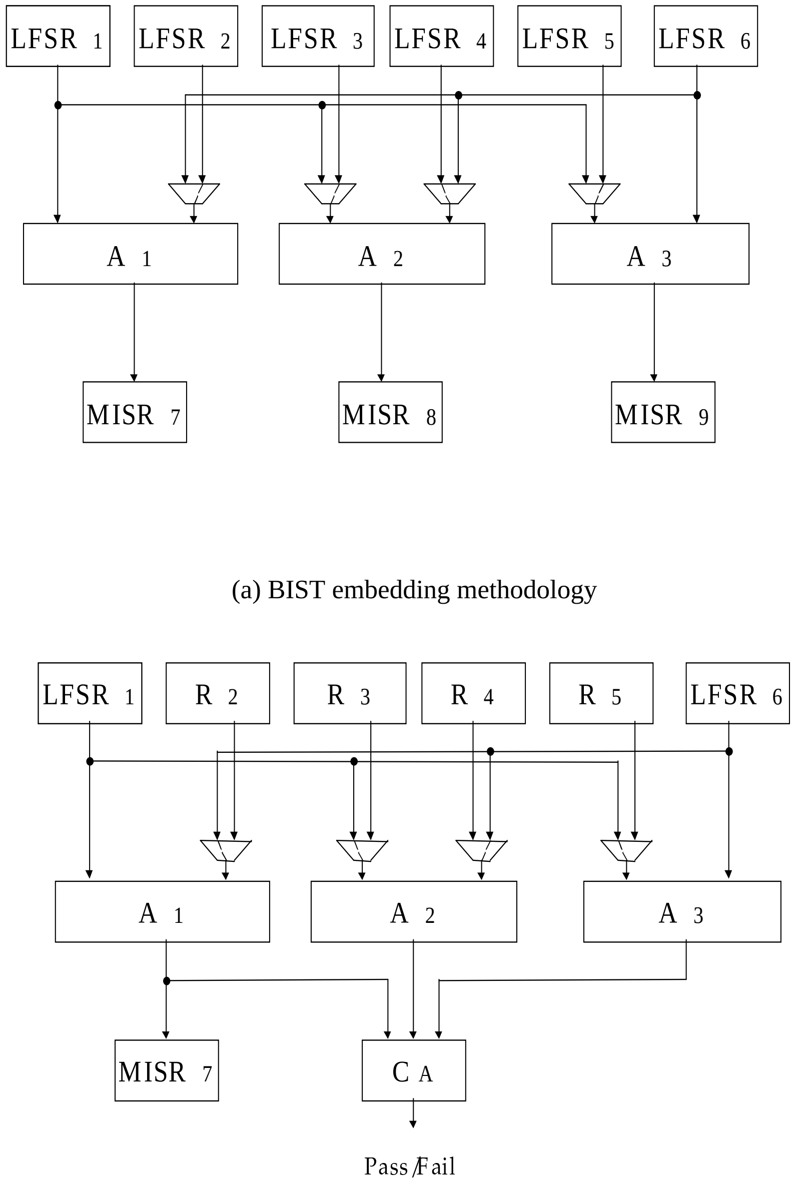
To overcome the large number of test registers in the BIST embedding methodology (problem (a)), a methodology based on chaining modules into test paths was described in [85, 150]. Randomness and transparency of data path modules [40] are used to guide the simultaneous test path generation and test scheduling. Despite reducing the performance degradation, the greater number of test patterns for each test path, which are no longer truly pseudorandom, increased the test application time. The test path generation algorithm lacked the global view of the design space and the suboptimum solution depends on the order in which the modules are processed. Furthermore, the pipelined test scheduling for multiple clock cycles test paths increases the complexity of the BIST controller as the design complexity is enlarged. Concurrent checkers [82, 127] were used for reducing fault-escape probability (problem (b)) during off-line self-test. While large BIST area overhead solutions based on duplicate circuitry realized in complementary form are described in [127], the results presented in [82] show that extra test hardware required to achieve low fault-escape probability, if designed as a combination of a concurrent checker and signature analysis registers, is more cost-effective than the design using only signature analysis registers. Recently a different approach which combines mutual and signature testing schemes [1] was proposed for reducing fault-escape probability. This approach uses test registers that combine equality comparators and signature analysis registers leading also to reduction in the volume of test data (problem (c)). However, due to large number of test registers when maximum test concurrency is targeted the problem of BIST area overhead and performance degradation are not solved. The previous approaches [1, 40, 82, 85, 127, 150] proposed separate solutions for solving only one of the problems (a) - (c) at the expense of the other problems of the BIST embedding methodology. Furthermore, the interrelation between test synthesis and test scheduling which leads to huge size of the testable design space (problem (d)) was not solved efficiently by the previously described approaches [14, 28, 33, 47, 92, 115] which trade off the quality of the final solution and computational time. Therefore, this chapter introduces a new BIST methodology for RTL data paths using a new concept called test compatibility classes which overcomes problems (a)-(d).

5.2 New BIST Methodology for RTL Data Paths

This section motivates the key ideas presented in this chapter through examples and gives formal concepts and definitions for the proposed BIST methodology. First the shortcomings of the traditional BIST embedding methodology are identified and benefits of the proposed BIST methodology are outlined using a detailed example. Then the formal definition of test compatibility classes is given.

5.2.1 An Illustrative Example

Traditional BIST embedding methodology embeds every module port between a test pattern generator and a signature analysis register (Section 1.3.2). This may lead to conflicts between different test resources when maximum test concurrency is targeted. Furthermore the number of test resources for low test application time is extremely high leading to both high BIST area overhead and performance degradation. The proposed BIST methodology takes advantage of the structural information of RTL data path and reduces the test application time by grouping same-type modules into test compatibility classes (TCCs). Two modules are of the same type if they are two different instances of the same module library prototype and hence they have the identical physical and structural information. Due to the identical physical and structural information the fault sets of two same-type modules have the same detection probability profile [13]. Thus, the same test pattern generators can be used simultaneously (no need to schedule the tests at different test times) for two or more same-type modules without decreasing the fault coverage. On the other hand fault sets of different-type modules have different detection probability profiles and hence different test pattern generators and different test application times are needed to satisfy the required fault coverage. The use of hard macro implementations of library modules which have identical physical and structural information can significantly improve the final design [39]. Furthermore, design methodologies which use regular elements and identify similarity need to be incorporated in state of the art CAD tools [39, 50]. Therefore the proposed BIST methodology is targeting design flows that use few pre-designed module types with identical physical and structural information and exploits the regularity of the data path to reduce test application time and BIST area overhead as explained in the following example.



(a) BIST embedding methodology

(b) The proposed BIST methodology

Figure 5.2: Comparison of data path testing using the traditional BIST embedding methodology and the proposed BIST methodology

Example 5.1 To give an insight into the proposed BIST methodology consider the simple data path shown in Figure 5.2(a). The data path has 3 modules of module-type A_{type} and 9 registers. Each module is annotated with its name (A_1-A_3). In order to test all the modules in a single test session, all the registers are modified into linear feedback shift registers (LFSRs) and multiple-input signature registers (MISRs). Each test pattern generator ($LFSR_1$-$LFSR_6$) applies test patterns to each input port of every module and each signature analysis register ($MISR_7$-$MISR_9$) evaluates the output responses of every module. The paths from test pattern generators to module input ports through multiplexers are denoted by dotted lines. In order to test all the modules simultaneously 6 LFSRs and 3 MISRs are required. Note the number of 2-to-1 multiplexers in the data path is 4 and hence there are 2^4 paths to apply test patterns to module input ports. Any other configuration of test registers which implies sharing of a test resource in order to reduce BIST area overhead will lead to an increased number of test sessions and hence test application time. However, if A_{type} modules are instances of the same module library prototype, then they have identical physical and structural information and hence identical detection probability profile. Thus if test patterns are applied to modules $A_1 - A_3$ simultaneously same fault coverage is achieved when compared to applying test patterns in different test sessions. Note that $LFSR_1$ is the only test register which applies test patterns to left input port of module A_1 and is connected to left input ports of A_2 and A_3. If multiplexers at left input ports of A_2 and A_3 select $LFSR_1$ as shown in Figure 5.2(b), the test registers $LFSR_3$ and $LFSR_5$ (Figure 5.2(a)) are unnecessary. This clearly leads to savings both in BIST area overhead and performance degradation, since registers R_3 and R_5 (Figure 5.2(b)) are not modified to perform test functions. Similarly if multiplexers at right input ports of A_1 and A_2 select $LFSR_6$ (Figure 5.2(b)), the test registers $LFSR_3$ and $LFSR_5$ are unnecessary. When same test patterns are generated by $LFSR_1$ and $LFSR_6$ at input ports of A_1, A_2, and A_3 same output responses are expected at the *same time*. Hence a comparator C_A is used to check the output responses. Only a single signature analysis register $MISR_7$ is allocated to compress the output responses of all the modules A_1,A_2, and A_3 which are tested simultaneously. The signature analysis register $MISR_7$ is necessary to detect faults in the case when output responses of all the three modules A_1,A_2, and A_3 are equal during the entire test application period but different from the fault-free output response. The use of comparators to check the output responses of all the same-type modules tested concurrently solves three problems. Firstly it reduces both BIST area overhead (1 MISR and 1 comparator vs. 3 MISRs) and performance degradation (1 MISR vs. 3 MISRs embedded in data path). Secondly it reduces fault-escape probability since faulty output responses which map into fault-free signatures in the BIST embedding methodology will be detected by the comparators. And thirdly, the number of signatures is reduced which has the following two implications: volume of test data is reduced which leads to less storage requirements and test application time is minimised due to less clock cycles needed to shift out the test responses. For example, given the data path width has 8 bits width, the time required to shift out the output response stored in $MISR_7$, $MISR_8$, and $MISR_9$ (Figure 5.2(a)) is 24 clock cycles when compared to only 8 clock cycles required to shift out the output response stored in $MISR_7$ (Figure 5.2(b)).

Solutions using comparators described in [127] to enhance fault escape probability are based on duplicate circuitry realized in complementary form leading to huge BIST area overhead. The proposed BIST methodology is fundamentally different where no duplicate circuitry is required and comparators are checking the responses of same-type modules which are instances of same module prototype. This makes the proposed BIST methodology suitable for complex data paths, such as data flow intensive application domains (digital signal processing, communications, and graphics) that have a high number of same-type modules generated automatically by modern CAD tools. The goal of the proposed BIST methodology is to test all the modules of the data path which are random pattern resistant and present testability problems. A part of the steering logic and interconnections are tested for free while testing the modules. It is known that a set of four vectors is sufficient to test a 2-to-1 multiplexer of any bit width. Similarly the functional registers are C-testable and non-random pattern resistant structures [2]. The comparators are the only extra DFT hardware added for BIST purposes. Approximately $2 \times n \times k$ test patterns are required to test a n-input k-bit comparator. Any portion of the data path not tested by the proposed BIST methodology is tested using a small global set of functional patterns. Since comparators check the responses of same-type modules which are inherently different cones-of-logic the small global test of functional patterns can be generated easily using the justification/propagation techniques [65, 118]. The small global set of functional patterns is applied in a preliminary phase and has no major impact on test application time. However, an alternative solution to test the comparators and untested registers and multiplexers is to scan all the functional registers that were not modified to test registers and apply an extra test session at the end of the functional modules testing process. This will avoid the usage of justification/propagation techniques [65, 118] at the expense of higher test area overhead. A comparative discussion between using justification/propagation techniques and full scan approach to test the untested steering logic and interconnections is provided in the experimental results Section 5.4.

5.2.2 Definition of Test Compatibility Classes

The formal introduction of the new BIST methodology is given in this section. The applicability of the proposed BIST methodology to variable bit width data paths and its relation to traditional BIST methodology are discussed.

An RTL data path consists of n_{reg} registers, n_{mod} two-input modules of n_{res} module-types, and multiplexers. Before the test compatibility class concept is introduced, it is necessary to present the following preliminary definitions.

Definition 5.1 A test register R_x performs the test pattern generation function (TPGF) for input port k (IP_k) of module M_a if test patterns for IP_k of M_a are provided by R_x. Let $IRS(M_a, IP_k)$ denote the set of registers that are connected to IP_k of M_a through only multiplexers. One and only one test register from $IRS(M_a, IP_k)$ is performing TPGF for IP_k of M_a.

The test registers used to perform TPGF are LFSRs, built-in logic block observers (BILBOs) and concurrent BILBOs (CBILBOs). If for each input port l (IP_l) of every data path module, $l = 1 \ldots 2 \times n_{mod}$, there is an m_l-to-1 multiplexer then the total number of paths to drive test patterns to data path modules is $\prod_{l=1}^{2 \times n_{mod}} m_l$. The testable data paths where both input ports of a module receive the same test patterns are not valid in the proposed BIST methodology due to the correlation between identical test patterns at both input ports which leads to a substantial decrease in fault coverage.

Definition 5.2 Two same-type modules, M_a and M_b, are incompatible, i.e. they cannot be tested simultaneously, if there is a test register R_x that performs TPGF for input port 1 (IP_1) of M_a and input port 2 (IP_2) of M_b, or if there is a test register R_y that performs TPGF for IP_2 of M_a and IP_1 of M_b. Two different-type modules are incompatible if there is a test register R_z that performs TPGF for any input port of M_a and any input port of M_b. Two modules are compatible if they are not incompatible.

The previous two Definitions 5.1 and 5.2 are important to formally introduce test compatibility classes. Having described test pattern generation function and test module compatibility and incompatibility, now the definition of test compatibility classes is given.

Definition 5.3 A test compatibility class $TCC_{i,j}$ (i is called the class index, while j is called the module-type index) is a set of modules that satisfies the following three properties:

i. all the modules from $TCC_{i,j}$ are compatible and belong to the same module-type j

ii. two test compatibility classes, $TCC_{p,j}$ and $TCC_{q,l}$, are incompatible if for at least one module M_a from $TCC_{p,j}$ there exists at least one module M_b from $TCC_{q,l}$ such that M_a and M_b are incompatible.

iii. modules from $TCC_{i,j}$ are tested simultaneously by the same test patterns and the output responses are checked by an n-input k-bit comparator, where n is the cardinality of $TCC_{i,j}$ and k is the bit-width of the data path. A single signature analysis register compresses the output response of a single module from $TCC_{i,j}$ to verify that the output response sequence is correct.

The first property of TCCs (Definition 5.3-(i)) guarantees that all the modules from a TCC can share test pattern generators leading to less BIST area overhead and performance degradation, without any penalty in test efficiency or test application time. The second property of TCCs (Definition 5.3-(ii)) indicates that high number of incompatible modules should be merged in a small number of incompatible TCCs leading to maximum test concurrency and hence reduction in test application time. The use of comparators described in the third property of TCCs (Definition 5.3-(iii)) decreases fault-escape probability. Furthermore the reduction in the number of signature analysis registers leads to smaller volume of test data and hence test application time (the sum of the time required to complete the test schedule and shifting time required to shift in generators seeds and shift out analysers' signatures). Let $ORS(M_k)$ denote the set of registers that are connected to the output of module M_k through multiplexers only. The output register set of $TCC_{i,j}$, ORS($TCC_{i,j}$), is the union of output register sets of all the modules from $TCC_{i,j}$. The signature analysis register for $TCC_{i,j}$ is chosen from ORS($TCC_{i,j}$). The test registers used for signature analysis are MISRs, BILBOs and CBILBOs. The number of highly expensive CBILBOs required for testing the self-loops in the data path is reduced when using the proposed methodology due to the greater number of potential signature analysis registers for each TCC. While the traditional BIST embedding methodology has n_{mod} signatures,

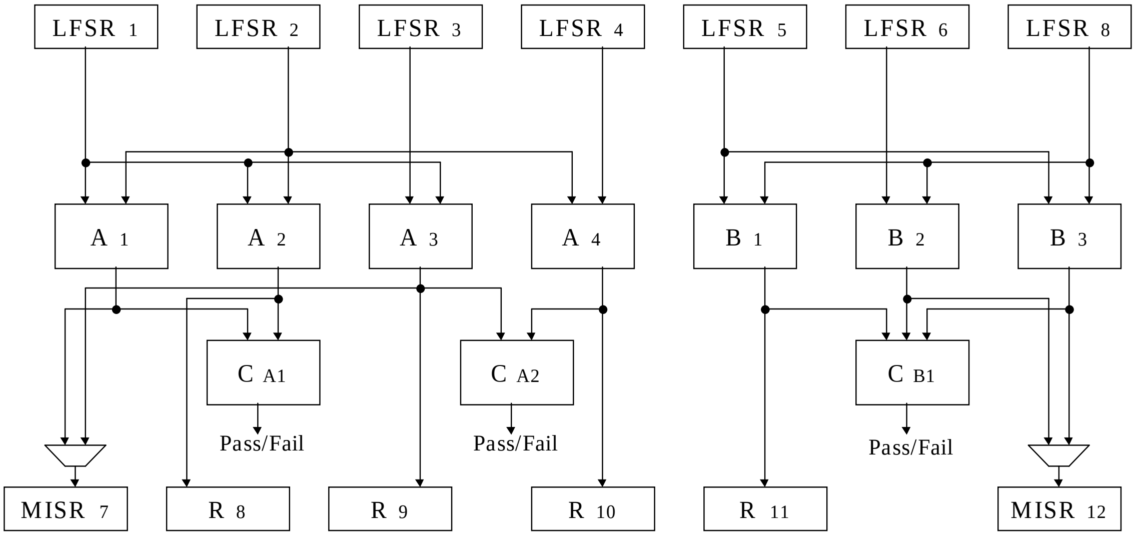


Figure 5.3: Example of data path testing using the proposed TCC grouping methodology

the proposed BIST methodology has only $\sum_{j=1}^{n_{res}} n_{classes}(j)$ signatures, where $n_{classes}(j)$ is the number of classes for module-type j. When $n_{mod} = n_{res}$ every module is an instance of a different module library prototype and according to Definitions 5.1-5.3 the proposed **TCC grouping** methodology is identical with the traditional BIST embedding methodology for RTL data paths. Hence, the BIST embedding methodology is a particular case of the TCC grouping methodology when each TCC consists of a single module. The proposed TCC grouping methodology is not limited to only uniform bit width data paths. Same-type modules which belong to a TCC have the same bit width. However different module types can have different bit width which does not imply any change in the methodology. Definitions 5.1-5.3 hold for variable bit width data paths. In the variable bit width case TCCs of *different* bit width are tested using *different* bit-width for test registers and n-input k-bit comparators. Moreover, the proposed methodology can handle both several modules chained together without any registers between them and particular cases when logic/RTL synthesis tools transform different instances of the same module type into different implementations by considering them as new module-types with new detection probability profile.

Testing an RTL data path using the proposed **TCC grouping** methodology and clarification of the previously introduced formal concepts are described in the following example.

Example 5.2 To illustrate Definitions 5.1-5.3 consider the data path example of Figure 5.3, where $LFSR_1$, $LFSR_2$, $LFSR_3$ and $LFSR_4$ test four modules of A_{type}, and $LFSR_5$, $LFSR_6$ and $LFSR_7$ test three modules of B_{type}. $LFSR_1$ generates test patterns for input port 1 (IP_1) of A_1 and A_2 and for input port 2 (IP_2) of A_3. $LFSR_2$ generates test patterns for the IP_2 of A_1 and A_2 and for IP_1 of A_4. $LFSR_3$ generates test patterns for IP_1 of A_3 whilst $LFSR_4$ provides test patterns for IP_2 of A_4. Modules A_1 and A_2 belong to $TCC_{0,0}$ (class index is 0 and A_{type} index is 0). Due to incompatibilities between A_1 and A_3 and between A_2 and A_4, modules A_3 and A_4 belong to $TCC_{1,0}$ (class index is 1). Similarly, modules B_1, B_2 and B_3 are all compatible and belong to $TCC_{0,1}$ (class index is 0 and B_{type} index is 1). Given the bit-width of the data path as 8 bits the output responses of modules from $TCC_{0,0}$ are compared by a 2-input 8-bit comparator (CA_1). Similarly, the output responses of modules from $TCC_{1,0}$ are compared by another 2-input 8-bit comparator (CA_2). On the other hand, output responses of modules from $TCC_{0,1}$ are compared by a 3-input 8-bit comparator (CB_1). If any error occurs during testing, the $Pass/Fail$ signal will be activated and the testing process will stop. The output register set ORS(A_1) is $\{R_7\}$, the ORS(A_2) is $\{R_8\}$, the ORS(A_3) is $\{R_7,\ R_9\}$ and the ORS(A_4) is $\{R_{10}\}$. Hence the ORS($TCC_{0,0}$) is $\{R_7, R_8\}$ and the ORS($TCC_{1,0}$) is $\{R_7, R_9, R_{10}\}$. Similarly, the ORS($TCC_{0,1}$) is $\{R_{11}, R_{12}\}$. Any of these two registers R_{11} and R_{12} can be configured as signature analysis register for $TCC_{0,1}$. The procedure that chooses the best signature analysis register is presented in Section 5.3.3. For data path example in Figure 5.3 the chosen signature analysis register for $TCC_{0,1}$ is R_{12} whilst both $TCC_{0,0}$ and $TCC_{1,0}$ use R_7 as signature analysis register at different test times. It should be noted that for testing modules A_3 and A_4 concurrently, test pattern generators $LFSR_1$ and $LFSR4$, and $LFSR_2$ and $LFSR_3$ need to generate the same patterns. This is achieved by shifting in identical seeds for test pattern generators for modules belonging to the same test compatibility class, at the beginning of each test session.

It should be noted that examples in this chapter do not illustrate BIST controller as described in Figure 1.10 from Section 1.3 of Chapter 1. However, the control overhead is considered in the experimental results (Section 5.4) and it is shown that the TCC grouping methodology yields lower BIST area overhead when compared to BIST embedding methodology.

5.3 New BIST Hardware Synthesis Algorithm for TCC Grouping

Having described the TCC grouping methodology, now a BIST hardware synthesis algorithm is considered. As outlined in Section 5.2, the BIST embedding methodology is a particular case of the TCC grouping methodology where each TCC consists of a single module. Therefore, the testable design space for the TCC grouping methodology is much larger and more complex than the testable design space for the BIST embedding methodology. The previous approaches [14, 28, 33, 47, 92, 115] which trade off the quality of the final solution and the computational time are unsuitable for the size and complexity of the TCC grouping methodology. This section presents a new and efficient testable design space exploration which combines the accuracy of incremental test scheduling algorithms [115] with the exploration speed of test scheduling algorithms based on fixed test resource allocation [33, 47, 92]. Section 5.3.1 outlines the general framework of tabu search-based testable design space exploration. Section 5.3.2 presents the generation of new solutions and speed up techniques for local neighbourhood search. Finally in Section 5.3.3 an incremental TCC scheduling algorithm for each solution is proposed.

5.3.1 Tabu Search-Based Testable Design Space Exploration

Tabu search [78] was proposed as a general combinatorial optimisation technique. Tabu search falls under the larger category of move-based heuristics which iteratively construct new candidate solutions based on the neighbourhood that is defined over the set of feasible solutions and the history of optimisation. The neighbourhood is implicitly defined by a move that specifies how one solution is transformed into another solution in a single step. The philosophy of tabu search is to derive and exploit a collection of principles of intelligent problem solving. Tabu search controls uphill moves and stimulates convergence toward global optima by maintaining a tabu list of its r most recent moves, where r is called tabu tenure and it is a prescribed constant. Occasionally, it is useful to override the tabu status of a move when the move is aspirated (i.e., improves the search and does not produce cycling near a local minimum). Tabu search based heuristics are simple to describe and implement. Furthermore, a well defined cost function and the use of topological information of the design space will lead to an intelligent search of high quality solutions in very low computational time. Before the proposed tabu search-based testable design space exploration is described, it is necessary to present the following definition.

Definition 5.4 A solution in the testable design space is a partially testable data path **PT-DP** where test pattern generators are allocated for each data path module. A fully testable data path **FT-DP** is generated by allocating signature analysis registers for each test compatibility class of the partially testable data path.

The proposed tabu search-based testable design space exploration is summarised in Figure 5.4. The algorithm starts with an initial solution which is a partially testable data path PT-DP$_{init}$ obtained by randomly assigning a single test pattern generator to each input port of every module from the data path as shown from lines 1 to 4. During the optimisation process (lines 5 to 21) for each current solution PT-DP$_{current}$, a number of n_{reg} neighbour solutions are generated as described in Section 5.3.2. Test application time T_x and BIST area overhead A_x are computed after a fully testable data path FT-DP$_x$ and a test schedule S_x are generated using the algorithms from Section 5.3.3, as shown from lines 8 to 12. The optimisation process is guided towards the objective of minimum test application time design by a cost function which is defined as follows.

Definition 5.5 The cost function is a 2-tuple $C_x = (T_x, A_x)$, where T_x is the test application time, A_x is the BIST area overhead and the following relations are defined:

i. $C_{x1} = C_{x2}$ if $(T_{x1} = T_{x2})$ and $(A_{x1} = A_{x2})$

ii. $C_{x1} < C_{x2}$ if $(T_{x1} < T_{x2})$ or $(T_{x1} = T_{x2}$ and $A_{x1} < A_{x2})$

iii. $C_{x1} > C_{x2}$ if $(T_{x1} > T_{x2})$ or $(T_{x1} = T_{x2}$ and $A_{x1} > A_{x2})$

The main objective of the cost function is test application time with BIST area overhead used as tie-breaking mechanism among many possible solutions with same test application time, as outlined in Figure 1.11 from Section 1.4.1 of Chapter 1. It should be noted that the minimisation of other parameters outlined in Section 1.3.2, performance degradation, volume of test data, and fault-escape probability, is a by-product of the proposed optimisation using the previously defined cost function. Based on the value of the cost function and on the tabu status of a move, a new solution is accepted or rejected as described from lines 14 to 19 in Figure 5.4. The tabu list contains registers involved in a

```
ALGORITHM: **Testable Design Space Exploration**
INPUT: *Data Path* **DP**
OUTPUT: *Fully Testable Data Path* **FT-DP**$_{best}$

1     for every module $M_a$ from DP with $a = 1, \ldots, n_{mod}$ do
2         for every input port $IP_k$ of $M_a$ with $k = 1, 2$ do
3             choose randomly $R_x$ from $IRS(M_a, IP_k)$
              and assign it to perform TPGF (this results into PT-DP$_{init}$)
4     PT-DP$_{current}$ ←PT-DP$_{init}$
5     repeat
6         for each register $R_x$ from PT-DP$_{current}$ with $x = 1, \ldots, n_{reg}$ do {
7             generate the new solution PT-DP$_x$ (Section 5.3.2)
8             generate a global test incompatibility graph T using PT-DP$_x$
              (Section 5.3.3)
9             generate test schedule $S_x$ and fully testable data path FT-DP$_x$
              using T and PT-DP$_x$ by simultaneous test scheduling and
              signature analysis registers allocation (Section 5.3.3)
10            compute test application time $T_x$ using test schedule $S_x$
11            compute BIST area overhead $A_x$ using FT-DP$_x$
12        }
13        for each FT-DP$_x$ ordered using $T_x$ and $A_x$ do {
14            if not tabu(FT-DP$_x$) or aspirated(FT-DP$_x$) then {
15                PT-DP$_{current}$ ←PT-DP$_x$
16                if best solution so far then
17                    FT-DP$_{best}$ ←FT-DP$_x$
18                break
19            }
20        }
21    until iterations since previous best solution $> N_{iter}$
22    return FT-DP$_{best}$
```

Figure 5.4: Tabu search-based testable design space exploration

move as described in Section 5.3.2. A move is classified as tabu if a register involved in the move is present in the tabu list. The tabu tenure (length of the tabu list) varies from 5 (small designs) to 10 (complex designs). A move is aspirated as shown in line 14 if it has produced a solution which is better than the best solution reached so far. The testable design space exploration continues until the number of iterations since the previous best solution exceeds a predefined N_{iter}.

5.3.2 Generation of New Solutions and Speed Up Techniques for Local Neighbourhood Search

The neighbourhood of the current solution in the testable design space PT-DP$_{current}$ is defined with n_{reg} feasible neighbour solutions. For each data path register there is a single neighbour solution. Each of the n_{reg} solutions is provided by an independent subroutine designed to identify better configuration of test registers based on two new metrics. Due to the huge size and complexity of the testable design space, speed up techniques for efficient exploration are required. Before defining the neighbour solution for each register two new metrics and a theorem used for reducing the testable design space are presented.

Definition 5.6 The current spatial sharing degree $C_{SSD}(R_x, j, IP_k)$ of register R_x for input port k (IP_k) of module-type j is the number of modules of j for which R_x performs test pattern generation function (TPGF) for IP_k in the current partially testable data path.

Definition 5.7 The maximum spatial sharing degree $M_{SSD}(R_x, j, IP_k)$ of register R_x for input port k (IP_k) of module-type j is the number of modules of j for which R_x can perform TPGF for IP_k. The value of $M_{SSD}(R_x, j, IP_k)$ is the cardinality of the set of modules of module-type j whose IP_k is connected to R_x through only multiplexers.

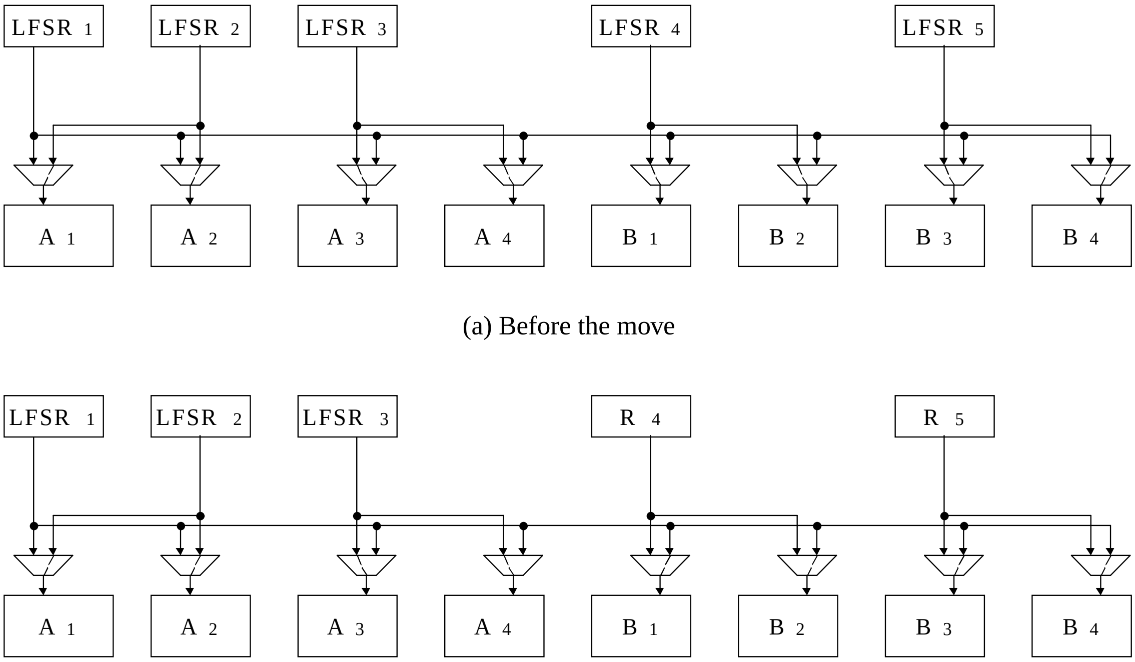
Theorem 5.1 Consider two current solutions, PT-DP$^1_{current}$ and PT-DP$^2_{current}$, with different $C_{SSD}(R_x, j, IP_k)$ for given R_x, j and IP_k. In PT-DP$^1_{current}$ the current spatial sharing degree is $0 < C_{SSD}(R_x, j, IP_k) < M_{SSD}(R_x, j, IP_k)$, whilst in PT-DP$^2_{current}$ the current spatial sharing degree is $C_{SSD}(R_x, j, IP_k) = M_{SSD}(R_x, j, IP_k)$. Then PT-DP$^2_{current}$ has at most the number of TCCs as PT-DP$^1_{current}$.

Proof: Let $\{M_1, \ldots, M_n\}$ be the set of modules of module-type j whose IP_k is connected to R_x through only multiplexers. In PT-DP$^1_{current}$, R_x performs TPGF for $\{M_1, \ldots, M_t\}$, whilst $\{R_{y_1}, R_{y_2}, \ldots, R_{y_m}\}$ perform TPGF for $\{M_{t+1}, \ldots, M_n\}$. Because $C_{SSD}(R_x, j, IP_k) > 0$ all the incompatibilities with both same-type modules and different-type modules are already created. In PT-DP$^2_{current}$, by increasing $C_{SSD}(R_x, j, IP_k)$ to $M_{SSD}(R_x, j, IP_k)$ no more incompatibilities will be created, which implies that the number of TCCs is not increased. Furthermore, according to Definition 5.1 of Section 5.2.2, each of $\{M_{t+1}, \ldots, M_n\}$ has one and only one test register that performs TPGF for IP_k in PT-DP$^2_{current}$. Hence test registers $\{R_{y_1}, R_{y_2}, \ldots, R_{y_m}\}$ will not perform TPGF for IP_k of $\{M_{t+1}, \ldots, M_n\}$ any more. This can lead to a decrease in $C_{SSD}(R_{y_i}, j, IP_k)$, where $i = 1 \ldots m$. If any of the $C_{SSD}(R_{y_i}, j, IP_k)$

becomes 0, the incompatibilities are reduced and the number of TCCs is decreased in PT-DP$^2_{current}$.

The above theorem presents a very important theoretical result which has the following two implications. The first implication reduces the *total* testable design space to the *representative* testable design space. The total testable design space consists of partially testable data paths with all the possible values $0 \leq C_{SSD}(R_x, j, IP_k) \leq M_{SSD}(R_x, j, IP_k)$ such that all the modules are assigned one and only one test pattern generator. The representative testable design space consists of partially testable data paths for which $C_{SSD}(R_x, j, IP_k)$ is considered only $M_{SSD}(R_x, j, IP_k)$ such that all the modules are assigned one and only one test pattern generator. Consider the simple data path of Figure 5.2. In the first case when the current spatial sharing degree for R_1 is $C_{SSD}(R_1, A_{type}, IP_1) = 1$ two more test registers $LFSR_3$ and $LFSR_5$ are necessary to generate test patterns for IP_1 of modules A_2 and A_3 as shown in Figure 5.2(a). On the other hand when $C_{SSD}(R_1, A_{type}, IP_1) = M_{SSD}(R_1, A_{type}, IP_1) = 3$ only one test pattern generator is necessary to generate test patterns for IP_1 of all the three modules as shown in Figure 5.2(b). The case when $C_{SSD}(R_1, A_{type}, IP_1) = 1$ has greater BIST area overhead and performance degradation due to $LFSR_3$ and $LFSR_5$. Furthermore if the simple data path of Figure 5.2 is a small part of a more complex data path, where $LFSR_3$ and $LFSR_5$ are already allocated to perform TPGF for different module-types, assigning $LFSR_3$ and $LFSR_5$ to perform TPGF for IP_1 of A_2 and A_3 respectively, will introduce conflicts between test resources leading to incompatible modules and hence increase in test application time. Theorem 5.1 justifies the reduction of the total testable design space where all the $C_{SSD}(R_1, A_{type}, IP_1) = \{0, 1, 2, 3\}$ are examined in the search of feasible partially testable data paths to the representative testable design space where only $C_{SSD}(R_1, A_{type}, IP_1) = 3$ is considered. The second implication of the theoretical result of Theorem 5.1 is concerned with efficient generation of moves in the representative testable design space. Generation of a move in the testable design space for register R_x consists of two phases:

i. The first phase computes: $\triangle_x(j, IP_k) = M_{SSD}(R_x, j, IP_k) - C_{SSD}(R_x, j, IP_k)$; $\triangle_x$ is a metric that measures the difference between the potential and actual use of R_x as a test pattern generator for IP_k of j modules. Note there are $2 \times n_{res}$ values of $\triangle_x$ for each register R_x.



(a) Before the move

(b) After the move

Figure 5.5: Example of a partially testable data path to illustrate generation of new solutions and speed up techniques for local neighbourhood search

ii. In the second phase the move for R_x that has the maximum value of $\triangle_{max}$ is chosen. If there are two or more j_m and/or IP_{k_n} for which $\triangle_x(j_m, IP_{k_n}) = \triangle_{max}$ the move for j_m and IP_k with the maximum value of $M_{SSD}(R_x, j_m, IP_k)$ is chosen.

Let j_{max} be the index of module-type and k_{max} be the index of input port for which $\triangle_{max}$ is maximum. Let $\{M_1, \ldots, M_n\}$ be the set of modules of module-type j_{max} whose $IP_{k_{max}}$ is connected to R_x through only multiplexers. Before the move, R_x performs TPGF for $\{M_1, \ldots, M_t\}$, whilst $\{R_{y_1}, R_{y_2}, \ldots, R_{y_m}\}$ perform TPGF for $\{M_{t+1}, \ldots, M_n\}$. After the move, R_x performs TPGF for $\{M_1, \ldots, M_n\}$, whilst $C_{SSD}(R_{y_i}, j_{max}, IP_{k_{max}})$ are decreased, with $i = 1 \ldots m$. The previously described two phases are repeated for each data path register and hence a neighbourhood of n_{reg} feasible solutions is generated. Increasing the current spatial sharing degree of the selected test registers leads to a smaller number of test pattern generators and hence reductions in BIST area overhead and performance degradation. Furthermore, the number of incompatibilities between TCCs is decreased which leads to lower test application time. The reduction in the number of incompatibilities between TCCs can also be explained as minimising the number of conflicts in the resource allocation graph (Figure 1.8(a) from section 1.3.2) due to high current spatial sharing degrees. Moreover the most important feature of the local neighbourhood search is the speed up technique for efficient exploration caused by reduction in the size of the testable design space to be explored. To illustrate generation of new solutions consider the following example.

Example 5.3 The move for register R_1 of the data path shown in Figure 5.5(a), is explained in detail. R_1 is configured as an LFSR and generates test patterns for input port 2 (IP_2) of B_4. The following metrics $\triangle_1(A_{type}, IP_1)$, $\triangle_1(A_{type}, IP_2)$, $\triangle_1(B_{type}, IP_1)$ and $\triangle_1(B_{type}, IP_2)$ are computed. Because $M_{SSD}(R_1, A_{type}, IP_1) = 2$ and $C_{SSD}(R_1, A_{type}, IP_1) = 0$ the following $\triangle_1(A_{type}, IP_1) = 2-0 = 2$ is obtained. Similarly $\triangle_1(A_{type}, IP_2) = 2-0 = 2$ and $\triangle_1(B_{type}, IP_1) = 0-0 = 0$. Since R_1 already performs TPGF for IP_2 of B_{type} then $\triangle_1(B_{type}, IP_2) = 4-1 = 3$. Because $\triangle_1(B_{type}, IP_2)$ is maximum the TPGF for IP_2 of B_{type} is moved to R_1 and area overhead is reduced because R_4 and R_5 do not perform TPGF any more (Figure 5.5(b)).

5.3.3 Incremental TCC Scheduling Algorithm

So far the testable design space to be explored was reduced with respect to the number of test registers required for test pattern generation using the speed up techniques for local neighbourhood search. The algorithms outlined in this section further shrink the size of the testable design space by considering simultaneous TCC scheduling and signature analysis registers allocation for each partially testable data path generated by local neighbourhood search. The steps which lead to a fully testable data path with low test application time and reduced BIST area overhead are introduced. Firstly the assignment of every data path module to test compatibility classes to maximise test concurrency is presented. Secondly the algorithm for simultaneous TCC scheduling and signature analysis registers allocation is described.

Generation of the Global Test Incompatibility Graph

To achieve maximum test concurrency it is required that a large number of different-type test compatibility classes are compatible. Following the second property of TCCs (Definition 5.3-(ii)) a high number of incompatible modules are sought to be merged in a small number of incompatible TCCs. This will reduce the number of edges in the global test incompatibility graph, which is based on the test incompatibility graph introduced in Figure 1.8(c) from section 1.3.2, and is defined as follows. A global test incompatibility graph (G-TIG) is a graph where a node appears for every TCC and an edge exists between nodes $TCC_{i,j}$ and $TCC_{k,l}$ if test compatibility classes $TCC_{i,j}$ and $TCC_{k,l}$ are incompatible. All the edges from G-TIG belong to the edge set E. The generation of G-TIG is outlined in Figure 5.6. For each partial testable data path PT-DP generated by the previously described local neighbourhood search a G-TIG is generated which is used for simultaneous test scheduling and signature analysis registers allocation. The generation of G-TIG is carried out in three steps.

Step 1: The first step assigns incompatible same-type modules into test compatibility classes and generates the initial G-TIG. Local test incompatibility graphs L-TIG(j) are created for each module-type j, with $j = 1 \ldots n_{res}$, of the current partially testable data path according to module incompatibility described in Definition 5.2. Each L-TIG is partitioned in TCCs using an efficient graph partitioning algorithm [47].

Step2: Data path modules that are incompatible with different-type modules are considered in the second step. A module is called an *assigned module* if it belongs to a TCC. A module is called an *unassigned module* if it is not assigned to any TCC during steps 1 and 2. If for two assigned different-type incompatible modules M_a and M_b, with $M_a \in TCC_{i,j}$ and $M_b \in TCC_{k,l}$, test compatibility classes $TCC_{i,j}$ and $TCC_{k,l}$ are compatible, then an edge between $TCC_{i,j}$ and $TCC_{k,l}$ is added to G-TIG as shown from lines 5 to 7 in Figure 5.6. Unassigned modules should be assigned to the already existing TCCs such that the number of incompatibilities between different-type TCCs is decreased leading to maximum test concurrency. If there is an unassigned module M_a which is incompatible with an assigned module M_b, with $M_b \in TCC_{k,l}$, then an edge is sought in the edge set E between $TCC_{k,l}$ and a TCC of the same-type with M_a. If at least one edge is found M_a is added to $TCC_{i,j}$ such that the following metric:

$\phi(TCC_{i,j}, M_a) = |ORS(TCC_{i,j} \cup M_a)| - |ORS(TCC_{i,j})|$

is maximum. If no edge is found then M_a is assigned to $TCC_{q,j}$ such that $\phi(TCC_{q,j}, M_a)$ is maximum and a new edge between $TCC_{q,j}$ and $TCC_{k,l}$ is added to G-TIG as shown from lines 12 to 15. If ϕ has the same value for all the classes of a module-type then the class

```
ALGORITHM: Generate Global Test Incompatibility Graph (G-TIG)
INPUT: Partially Testable Data Path PT-DP
OUTPUT: Global Test Incompatibility Graph T

Step1:  for every module-type $j$ from PT-DP with $j = 1, \ldots, n_{res}$ do
            Partition same-type incompatible modules in TCCs
            and generate initial G-TIG T with edge set E
Step2:  for every pair of different-type incompatible modules $(M_a, M_b)$ do {
            $j \leftarrow$ module-type$(M_a)$; $l \leftarrow$ module-type$(M_b)$;
            if $\exists i$ such that $M_a \in TCC_{i,j}$ and $\exists k$ such that $M_b \in TCC_{k,l}$ then
                if $(TCC_{i,j}, TCC_{k,l}) \notin E$ then
                    Add Edge $(TCC_{i,j}, TCC_{k,l})$ to E
            if $\nexists\ i \in \{1 \ldots n_{classes}(j)\}$ such that $M_a \in TCC_{i,j}$
            and $\exists k$ such that $M_b \in TCC_{k,l}$ then {
                if $\exists i$ such that $(TCC_{i,j}, TCC_{k,l}) \in E$ then
                    Add $M_a$ to $TCC_{i,j}$ such that $\phi(TCC_{i,j}, M_a)$ is maximum
                else {
                    Add $M_a$ to $TCC_{q,j}$ such that $\phi(TCC_{q,j}, M_a)$ is maximum
                    Add Edge $(TCC_{q,j}, TCC_{k,l})$ to E
                }
            }
            if $\nexists\ i \in \{1 \ldots n_{classes}(j)\}$ such that $M_a \in TCC_{i,j}$
            and $\nexists k \in \{1 \ldots n_{classes}(l)\}$ such that $M_b \in TCC_{k,l}$ then {
                if $\exists i$ and $\exists k$ such that $(TCC_{i,j}, TCC_{k,l}) \in E$ then {
                    Add $M_a$ to $TCC_{i,j}$ such that $\phi(TCC_{i,j}, M_a)$ is maximum
                    Add $M_b$ to $TCC_{k,l}$ such that $\phi(TCC_{k,l}, M_b)$ is maximum
                }
                else {
                    Add $M_a$ to $TCC_{q,j}$ such that $\phi(TCC_{q,j}, M_a)$ is maximum
                    Add $M_b$ to $TCC_{r,l}$ such that $\phi(TCC_{r,l}, M_b)$ is maximum
                    Add Edge $(TCC_{q,j}, TCC_{r,l})$ to E
                }
            }
        }
Step3:  for every module $M_k$ that is compatible with all modules from PT-DP do {
            $j \leftarrow$ module-type$(M_c)$
            Add $M_c$ to $TCC_{i,j}$ such that $\phi(TCC_{i,j}, M_c)$ is maximum
        }
        return T
```

Figure 5.6: Generation of global test incompatibility graph for maximum test concurrency

with the lowest index is considered. The newly introduced metric ϕ increases the number of potential signature analysis registers which has the following implications during the test scheduling process described in Section 5.3.3:

- reduction of conflicts between signature analysis registers that are allocated during the test scheduling process leading to lower test application time;
- reuse of same signature analysis registers at different test times for different TCCs, which has straight impact on BIST area overhead, performance degradation and shifting time required to shift out signatures when the test process is accomplished;
- minimisation of the number of highly expensive CBILBOs required for testing the self-loops in the data path since the difference between the output register set of a TCC and the input register sets of every module from a TCC is maximised;

Finally, if two unassigned different-type modules M_a and M_b are incompatible then the same assignment reasoning, outlined previously for an unassigned module incompatible with an assigned one, is applied as shown from lines 17 to 27.

Step3: In the third step unassigned modules M_c of module-type j which are compatible with all TCCs are added to a class i such that $(TCC_{i,j}, M_c)$ is maximum.

The proposed algorithm for generation of G-TIG guarantees by construction that every module is assigned to a TCC and the number of nodes and edges in G-TIG is minimum. This implies maximum test concurrency of the partially testable data path which is a good starting point for the incremental test scheduling algorithm.

Example 5.4 To illustrate the generation of G-TIG consider the data path shown in Figure 5.7. The module-type indexes of A_{type}, B_{type} and C_{type} are 0,1 and 2 respectively. In the first step, local test incompatibility graphs L-TIG(A_{type}), L-TIG(B_{type}), and L-TIG(C_{type}) are created. Since modules that belong to C_{type} do not share any test registers with same-type modules then L-TIG(C_{type}) is void. The initial test compatibility classes are: $TCC_{0,0} = \{A_1\}$, $TCC_{1,0} = \{A_2\}$, $TCC_{2,0} = \{A_3\}$, $TCC_{0,1} = \{B_1, B_2\}$, and $TCC_{1,1} = \{B_3, B_4\}$. In the second step edges between TCCs representing different module-types are added to G-TIG. So far the only edges in the edge set are the ones between the TCCs of identical module-types: $(TCC_{0,0}, TCC_{1,0})$, $(TCC_{1,0}, TCC_{2,0})$,

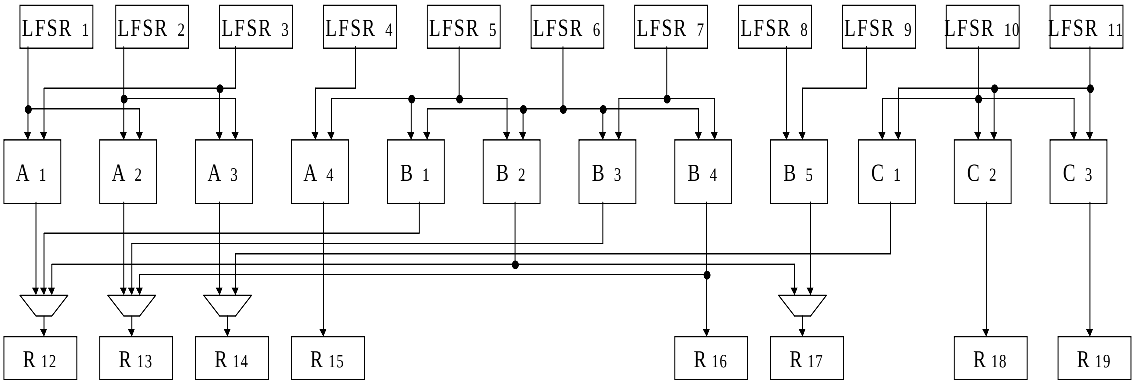


Figure 5.7: Data path example to illustrate the incremental TCC scheduling algorithm

$(TCC_{2,0}, TCC_{0,0})$ and $(TCC_{0,1}, TCC_{1,1})$. Due to sharing of $LFSR_5$ between A_4 and B_1 an edge between $TCC_{0,1}$ and a TCC of A_{type} must be added to G-TIG. First of all, A_4 is assigned to one of the TCCs of A_{type}. The computed output register set increase measures are $\phi(TCC_{0,0}, A_4) = 1$, $\phi(TCC_{1,0}, A_4) = 1$, and $\phi(TCC_{2,0}, A_4) = 1$. All the three $TCC_{0,0}$, $TCC_{1,0}$ and $TCC_{2,0}$ are candidates. Due to the smallest class index of $TCC_{0,0}$, module A_4 is assigned to $TCC_{0,0}$ and the incompatibility edge $(TCC_{0,0}, TCC_{0,1})$ is added to G-TIG. In the third step B_5 is assigned to one of the TCCs of B_{type}. The computed output register set increase measures are $\phi(TCC_{0,1}, B_5) = 0$ and $\phi(TCC_{1,1}, B_5) = 1$. B_5 is assigned to $TCC_{1,1}$ due to greater increase in the number of potential signature analysis registers. The test compatibility classes are $TCC_{0,0} = \{A_1, A_4\}$, $TCC_{1,0} = \{A_2\}$, $TCC_{2,0} = \{A_3\}$, $TCC_{0,1} = \{B_1, B_2\}$, $TCC_{1,1} = \{B_3, B_4, B_5\}$ and $TCC_{0,2} = \{C_1, C_2, C_3\}$ and the G-TIG is shown in Figure 5.9.

Simultaneous TCC Scheduling and Signature Analysis Registers Allocation

Test scheduling is performed using the global test incompatibility graph outlined in Section 5.3.3. Fault sets of different-type modules have different detection probability profiles as outlined in Section 5.2.2. Hence, TCCs of different module-types need different test application times to satisfy the required fault coverage. Thus, the TCC scheduling algorithm deals with unequal test lengths. The test scheduling algorithm for partitioned testing with run to completion from [47] needs to be modified such that test scheduling and signature analysis registers allocation is done simultaneously, as shown in Figure 5.8. For the formal presentation of the algorithm, the sets and notations from [47] are preserved. The

```
ALGORITHM: **Simultaneous TCC Scheduling and
            Signature Analysis Registers Allocation**
INPUT: *Global Test Incompatibility Graph* **T**
       *Partially Testable Data Path* **PT-DP**
OUTPUT: *Test Schedule* **S**
        *Fully Testable Data Path* **FT-DP**

$AN \leftarrow T; S \leftarrow \emptyset; L_i \leftarrow a_i; I_i \leftarrow L_i - i + \sum_{j|t_j \in N(\{t_i\})} L_j;$
while $AN \neq \emptyset$ do {
    $M \leftarrow N(S); C \leftarrow AN \cap \overline{M}; C' \leftarrow C \cap N(M);$
    while $C \neq \emptyset$ do {
        if $C' \neq \emptyset$ then choose $t_i \in C'$ where $I_i$ is maximum
        else choose $t_i \in C$ where $I_i$ is maximum
        if $\exists\ R_k \in ORS(t_i)$ and $R_k \notin B$ then {
            if $\exists\ R_k \in U$ then
                choose $R_k \in U$ where $fanin(R_k)$ is maximum
            else
                choose $R_k \in ORS(t_i)$ where $fanin(R_k)$ is maximum
            $AN \leftarrow AN - \{t_i\}; B \leftarrow B \cup \{R_k\}; S \leftarrow S \cup \{t_i\};$
            $I_i \leftarrow I_i - 1; I_j \leftarrow I_j - 1$ where $t_j \in N(\{t_i\});$
        }
        $M \leftarrow M \cup (N\{t_i\} \cap AN);$
        $C \leftarrow C \cup (\{t_i\} \cup N\{t_i\}); C' \leftarrow C \cap N\{M\};$
    }
}
$time \leftarrow min_{i|t_i \in S}\{L_i\};$
for all $t_i \in S$ do {
    $L_i \leftarrow L_i - time; I_i \leftarrow I_i - (time - 1);$
    for all $t_j \in N(\{t_i\})$ do
        $I_j \leftarrow I_j - (time - 1);$
    if $L_i = 0$ then {
            $S \leftarrow S - \{t_i\};$
            $B \leftarrow B - \{R_k\};$
            $U \leftarrow U \cup \{R_k\};$
    }
}
for all $R_k \in U$ do
    modify $R_k$ from **PT-DP** into a test register in **FT-DP**
return (S,FT-DP)
```

Figure 5.8: Algorithm for simultaneous TCC scheduling and signature analysis registers allocation

test scheduling algorithm is based on resource allocation graph, test compatibility graph and test incompatibility graphs described in Figures 1.8(a)-1.8(c) introduced in Section 1.3.2. For the sake of clarity the notations and a brief overview of the algorithm 2M from [47] are described in the following: S is the concurrent test set (test compatibility classes scheduled at the same time represented by the independent set in the test incompatibility graph), AN is the set of active nodes (test compatibility classes that need to be assigned a signature analysis register and scheduled in a time slot), T is the set of all the nodes (all the test compatibility classes from the global test incompatibility graph outlined in Section 5.3.3), C is the set of candidate nodes (test compatibility classes that can be scheduled at the current time slot), C' is the set of distance 2 candidate nodes, M is the set of marked nodes, and $N(\{t_i\})$ is the adjacency relation (set of nodes incompatible with a set of nodes $\{t_i\}$). The algorithm 2M from [47] schedules tests for a fixed test resource allocation, and the aim is to generate a compatible test set by adding candidate tests to an expanding set. Any node which is incompatible with a node from concurrent test set S cannot be a candidate to be added in the expanding set. Therefore, all incompatible (adjacent) nodes are added to the marked set M and excluded from further consideration during the expansion of the current S as shown in line 3 of Figure 5.8. During the execution of the algorithm when no additional nodes can be added to the expanding set, a concurrent test set was found and M is the remaining graph of the test incompatibility graph. To maximise the concurrent test set S the algorithm first considers as candidates tests which are of distance 2 from the expanding set as shown in line 6 of Figure 5.8. The node with maximum I_i is selected where I_i is the number of equivalent incompatibilities (degree) associated with each token t_i. If no distance 2 candidates exist, then a node from all candidate tests C is considered. The tests incompatible with t_i are added to M, candidate sets are updated, and process continues until there are no new candidates. When the shortest currently active tests are completed, the concurrent test set S is modified to reflect the removal of all the completed tests, and this modified set is used as a seed for further processing in order to preserve the run to completion requirement (lines 19 to 29 of Figure 5.8).

Since the previously described algorithm 2M from [47] schedules tests for a fixed test resource allocation, the following three necessary modifications are carried to perform simultaneous TCC scheduling and signature analysis register allocation. Two more notations are introduced: U is the set of used test registers that have compressed output responses at a previous test time and B is the set of busy test registers that are compressing output responses at the current test time.

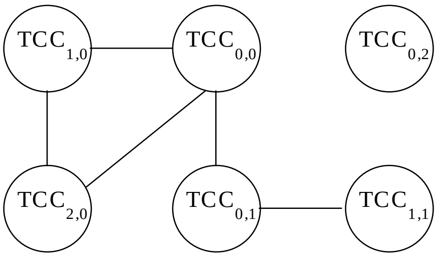
i. if all the registers in $ORS(t_i)$ are busy at the current test time then test t_i is removed from the candidate node set being postponed for a later test time, as shown from lines 7 to 16; otherwise for every available register R_k in $ORS(t_i)$, it is checked whether R_k belongs to the used test register set and the R_k with the maximum fanin is chosen; this choice will allow R_k to be reused at a later test time.

ii. when the shortest currently active test t_i is completed, the test register R_k that has served as signature analysis register is removed from the busy register set B and added to the used register set U, as shown from lines 24 to 28.

iii. after the completion of test scheduling all the registers from the used register set U are modified to signature analysis registers as shown in lines 30 and 31; the algorithm returns a test schedule S and a fully testable data path FT-DP which are used to compute test application time and BIST area overhead in the tabu search testable design space exploration (Figure 5.4).

The first modification solves the conflicts between signature analysis registers during the test scheduling process reducing both the size of the testable design space to be explored and test application time. Thus the efficiency of testable design space exploration is improved by combining the accuracy of incremental test scheduling algorithms with the exploration speed of test scheduling algorithms based on fixed test resource allocation. The second and third modifications reduce the number of signature analysis registers by reusing them at different test times leading to further reductions in BIST area overhead, performance degradation, and volume of test data.

Example 5.5 To illustrate the above three modifications consider Figure 5.9 which shows the G-TIG of the data path shown in Figure 5.7. It is assumed that the number of test patterns to test A_{type} modules is $T_0 = T$, the number of patterns to test of B_{type} modules is $T_0 = 2\times T$, and the number of test patterns to test C_{type} modules is $T_2 = 3\times T$, with T a reasonable large integer. The first scheduled test is $TCC_{0,0}$ at test time 0. The signature analysis (SA) register for $TCC_{0,0}$ is R_{12} which is chosen from $ORS(TCC_{0,0})$ because its fanin is maximum. Two more tests, $TCC_{1,1}$ and $TCC_{0,2}$ are scheduled at test time



$ORS(TCC_{0,0}) = \{R_{12}, R_{15}\}$	$T_0 = T$
$ORS(TCC_{1,0}) = \{R_{13}\}$	$T_0 = T$
$ORS(TCC_{2,0}) = \{R_{14}\}$	$T_0 = T$
$ORS(TCC_{0,1}) = \{R_{12}, R_{17}\}$	$T_1 = 2xT$
$ORS(TCC_{1,1}) = \{R_{13}, R_{16}, R_{17}\}$	$T_1 = 2xT$
$ORS(TCC_{0,2}) = \{R_{14}, R_{18}, R_{19}\}$	$T_2 = 3xT$

Figure 5.9: Global test incompatibility graph for data path of Figure 5.7

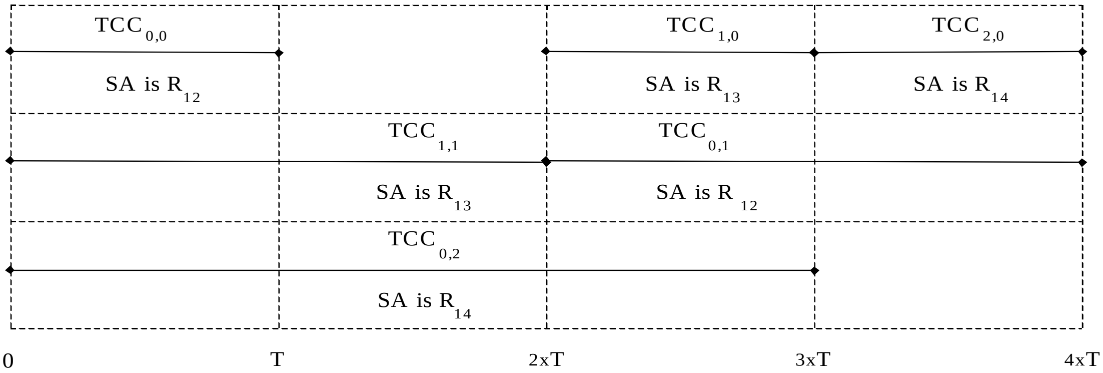


Figure 5.10: TCC schedule for data path of Figure 5.7 using the proposed algorithm for simultaneous TCC scheduling and signature analysis registers allocation

0. The signature analysis registers for $TCC_{1,1}$ and $TCC_{0,2}$ are R_{13} and R_{14} respectively. Registers R_{12}, R_{13} and R_{14} are added to the busy register set B. At test time T test $TCC_{0,0}$ is completed and R_{12} is removed from B and added to the used register set U. The attempt to schedule $TCC_{1,0}$ at test time T fails because R_{13} belongs to B. Similarly the attempt to schedule $TCC_{2,0}$ at test time T fails because R_{14} also belongs to B. At test time $2 \times T$ test $TCC_{1,1}$ is completed and R_{13} is removed from the busy register set B and added to the used register set U. A new attempt to schedule $TCC_{1,0}$ succeeds and the allocated signature analysis register is R_{13}. In addition at test time $2 \times T$ test $TCC_{0,1}$ is scheduled and the chosen register from U is R_{12}. Finally at test time $3 \times T$ test $TCC_{0,2}$ is completed and test $TCC_{2,0}$ is scheduled reusing the signature analysis register R_{14}. Note that only three test registers are used to analyse test responses. A graphical representation of the test schedule is presented in Figure 5.10.

5.4 Experimental Results

Experimental results are divided in four separate sections. Section 5.4.1 gives a comparison of the proposed TCC grouping methodology (Section 5.2.2) and the traditional BIST embedding methodology for benchmark circuits. Section 5.4.2 provides the experimental results when benchmark circuits are technology mapped into AMS 0.35 micron technology [9]. Section 5.4.3 outlines the benefits of the proposed methodology for complex hypothetical data paths, and Section 5.4.4 gives the reduction in fault-escape probability.

5.4.1 Comparison of the TCC Grouping and BIST Embedding Methodologies for Benchmark Circuits

The BIST hardware synthesis for the TCC grouping methodology was implemented on a SUN SPARC 20 workstation using 6000 lines of C++ code. To give insight into the efficiency of testability achieved using the presented approach Tables 5.1 and 5.2 show a comparison of BIST resources and number of test sessions using the BIST embedding methodology and the TCC grouping methodology. The results for the BIST embedding methodology are obtained using the same BIST hardware synthesis algorithm (Section 5.3) assuming that every pair of modules in the data path are different $(n_{res} = n_{mod})$ as described in Section 5.2.2. The comparison is carried out for a number of benchmark examples including elliptic wave digital filter (EWF) and 8 and 32 point discrete cosine transform (DCT). The benchmarks were synthesised using the ARGEN high level synthesis system [99, 100] for different execution time constraints ranging from 10 to 40. Data paths consisting of different number of modules and registers were generated by ARGEN. For example, in the case of *EWF-17* we have 6 modules (MOD), 3 multipliers (*) and 3 adders (+), and 12 registers (REG). Test lengths for adders and multipliers are assumed to be $T_+ = T_u$, and respectively $T_* = 4 \times T_u$, where T_u is a reasonably large integer and can be estimated for the required fault coverage using the techniques from [117]. The proposed BIST hardware synthesis algorithm for RTL data paths (Section 5.3) provides the flexibility of specifying the test length for the expected fault coverage of 100% for each data path module. In the experimental results reported in this section it was considered that $T_u = 128$, and hence $T_+ = 128$ and $T_* = 512$, for achieving 100% fault coverage for each 8 bit data path module which is comparable with the test length values reported in [115]. To validate the assumption regarding test length, adder and multiplier modules were syn-

Design	**M** **O** **D**	**R** E **G**	**BIST embedding**	
			BIST resources	Test
			Test registers	Sessions
EWF-17	3*,3+	12	8 LFSR, 3 MISR	5×T_u
EWF-21	2*,3+	13	6 LFSR, 3 MISR, 1 BILBO	4×T_u
EWF-23	1*,2+	11	5 LFSR, 2 MISR	4×T_u
EWF-25	1*,2+	12	5 LFSR, 3 MISR	4×T_u
EWF-28	1*,2+	11	3 LFSR, 2 MISR	4×T_u
8DCT-10	4*,4+	17	9 LFSR, 5 MISR, 3 BILBO	4×T_u
8DCT-12	4*,3+	16	9 LFSR, 5 MISR	5×T_u
8DCT-13	4*,4+	16	8 LFSR, 4 MISR, 1 BILBO	5×T_u
8DCT-14	3*,3+	16	6 LFSR, 5 MISR	5×T_u
8DCT-16	3*,2+	16	7 LFSR, 5 MISR	4×T_u
8DCT-21	2*,2+	15	6 LFSR, 4 MISR	4×T_u
32DCT-30	9*,12+	60	33 LFSR, 21 MISR	4×T_u
32DCT-31	9*,12+	62	33 LFSR, 21 MISR	4×T_u
32DCT-32	8*,12+	62	32 LFSR, 20 MISR	4×T_u
32DCT-33	8*,11+	62	30 LFSR, 19 MISR	4×T_u
32DCT-37	8*,9+	63	26 LFSR, 17 MISR	4×T_u
32DCT-38	9*,9+	59	27 LFSR, 18 MISR	4×T_u
32DCT-40	7*,10+	61	27 LFSR, 17 MISR	4×T_u

Table 5.1: BIST resources and number of test sessions when using the BIST embedding methodology for benchmark examples

thesised and technology mapped into AMS 0.35 micron technology [9]. Subsequently a parallel pattern single fault propagation fault simulator [112] has shown that $T_u = 128$ is valid for 8 bit data path modules. In general the TCC grouping methodology produces less test registers than the BIST embedding methodology as shown in Tables 5.1 and 5.2. For example, in the case of *8DCT-10* the number of MISRs is reduced from 5 to 2, and the number of BILBOs is reduced from 3 to 0. There is further reduction as the design complexity increases. For example, in the case of *32DCT-30* the number of LFSRs is reduced from 33 to 18, and the number of MISRs is reduced from 21 to 2. The reduction in test registers in case of TCC grouping is achieved at the expense of comparators. In the case of *32DCT-30* there are one 5 input comparator (C5), one 7 input comparator (C7) and one 9 input comparator (C9). However, the TCC grouping methodology requires reduced BIST area overhead when compared with the BIST embedding methodology as shown in Table 5.5.

Design	**TCC grouping**			**CPU**
	BIST resources		Test	**time**
	Test Registers	Comparators	Sessions	**(s)**
EWF-17	6 LFSR, 1 MISR, 1 BILBO	2 C3	$4\times T_u$	1
EWF-21	7 LFSR, 2 MISR	1 C3, 1 C2	$4\times T_u$	1
EWF-23	4 LFSR, 2 MISR	1 C2	$4\times T_u$	1
EWF-25	4 LFSR, 2 MISR	1 C2	$4\times T_u$	1
EWF-28	5 LFSR, 2 MISR	1 C2	$4\times T_u$	1
8DCT-10	9 LFSR, 2 MISR	2 C4	$4\times T_u$	1
8DCT-12	9 LFSR, 2 MISR	1 C3, 1 C4	$4\times T_u$	1
8DCT-13	9 LFSR, 2 MISR	2 C4	$4\times T_u$	1
8DCT-14	7 LFSR, 2 MISR	2 C3	$4\times T_u$	1
8DCT-16	5 LFSR, 2 MISR	1 C2, 1 C3	$4\times T_u$	1
8DCT-21	4 LFSR, 2 MISR	2 C2	$4\times T_u$	1
32DCT-30	18 LFSR, 2 MISR	1 C5, 1 C7, 1 C9	$4\times T_u$	129
32DCT-31	19 LFSR, 2 MISR	1 C5, 1 C7, 1 C9	$4\times T_u$	124
32DCT-32	16 LFSR, 2 MISR	1 C4, 2 C8	$4\times T_u$	103
32DCT-33	14 LFSR, 2 MISR	1 C5, 1 C6, 1 C8	$4\times T_u$	55
32DCT-37	16 LFSR, 2 MISR	1 C3, 1 C6, 1 C8	$4\times T_u$	86
32DCT-38	16 LFSR, 2 MISR	2 C9	$4\times T_u$	38
32DCT-40	16 LFSR, 2 MISR	1 C7, 1 C10	$4\times T_u$	45

Table 5.2: BIST resources and number of test sessions when using the proposed TCC grouping methodology for benchmark examples

The BIST hardware synthesis algorithm has excellent computational time. The CPU time required to achieve lowest number of test sessions for benchmark circuits is shown in the last column of Table 5.2. For example, in the case of EWF and 8 point DCT designs, the computational time is very low under 1s. In the case of designs with huge testable design space like 32 point DCT, high quality solutions are achieved in computational times ranging from 38s to 129s. A high quality solution is a fully testable data path with the time to complete the test sessions equal (or almost equal) to the longest test length required to test the most random pattern resistant module ($4\times T_u$ in the case of benchmark circuits from Tables 5.1 and 5.2). It should be noted that despite the fact that test registers and test schedule of the final solution are dependent on the initial random assignment of test registers, the quality of the final solution (in terms of test application time and BIST area overhead) is independent on the initial random assignment due to the intelligent neighbourhood search outlined in Section 5.3.

5.4.2 Experimental Validation of the TCC Grouping Methodology

To give an indication how the BIST resources and the number of test sessions shown in Tables 5.1 and 5.2 translate into BIST parameters outlined in Section 1.3.2 from Chapter 1 (test application time, BIST area overhead, performance degradation, and volume test data), the following experimental validation framework is used. The elliptic wave digital filter and 8 and 32 point discrete cosine transforms were synthesised and technology mapped into AMS 0.35 micron technology [9] using VHDL as entry point. The experimental validation flow and software organisation used to integrate the BIST embedding and the TCC grouping methodologies into third party electronic design automation tools [55], and sample VHDL files describing BIST RTL data paths, are given in appendix A and appendix C respectively. The results for test application time (TAT) in terms of clock cycles, BIST area overhead (BAO) in terms of square mils, circuit performance (Perf) in terms of clock frequency in MHz, and volume of test data (VTD) in terms of data bits are reported for BIST embedding and TCC grouping methodologies in Table 5.3 and Table 5.4 respectively. Experimental results shown in columns 2 to 5 of Tables 5.3 and 5.4 consider that untested multiplexers and registers use a small global test of functional patterns that can be generated using the justification/propagation techniques [65, 118] and which does not have an influence on test application time as outlined in Section 5.2.1. However, an alternative solution to test the untested registers and multiplexers is to scan all the functional registers that were not modified to test registers, and apply an extra test session at the end of the functional modules testing process (columns 6 to 9 from Tables 5.3 and 5.4). It can be seen that both in the case of the BIST embedding methodology (Table 5.3) and the TCC grouping methodology (Table 5.4) there is an increase in test application time, BIST area overhead, and volume of test data, and similar or a lower clock frequency, when scanning the untested functional registers. For example, in the case of *8DCT-14* when applying BIST embedding methodology, there is an increase from 760 to 864 clock cycles in TAT, from 59 to 71 square mils in BAO, from 120 to 232 data bits in VTD, and a decrease from 91 to 90 MHz in clock frequency (line 9 from Table 5.3). The lower clock frequency caused by longer critical paths when employing scan leads to higher performance degradation. However, by scanning all the registers the methodology is self-contained without any reliance on justification/propagation techniques [65, 118] for a small global set of functional patterns.

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	760	45	86	120	784	54	86	152
EWF-21	736	43	98	104	760	50	98	136
EWF-23	704	46	91	64	808	49	90	176
EWF-25	576	41	90	64	672	50	90	152
EWF-28	704	49	98	64	832	54	98	200
8DCT-10	656	61	90	160	656	72	90	160
8DCT-12	776	51	91	136	856	51	91	224
8DCT-13	792	49	89	160	880	57	87	256
8DCT-14	760	59	91	120	864	71	90	232
8DCT-16	608	53	90	96	752	56	90	232
8DCT-21	592	51	93	80	744	65	89	232
32DCT-30	1064	124	92	432	1544	136	85	832
32DCT-31	1064	124	92	432	1504	158	88	800
32DCT-32	1056	115	92	416	1512	139	92	856
32DCT-33	904	113	87	392	1400	115	86	784
32DCT-37	856	101	91	344	1400	113	91	776
32DCT-38	864	109	94	360	1328	137	93	696
32DCT-40	864	105	92	352	1344	106	90	704

Table 5.3: Experimental data for BIST embedding methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping in 0.35 micron AMS technology

It is interesting to note that BIST area overhead in terms of square mils reflects *not only* the additional test hardware required by test registers, but *also* the additional gates required to integrate the functional and test controller as outlined in Figure 1.10 of Section 1.3.2 from Chapter 1. For benchmark circuit *32DCT-30* the reductions in BIST area overhead are of 18.55% in the case of a 8-bit data path when no scan is employed (column 3 and line 12 of Table 5.5), and 15.77% when scan is employed for untested registers and multiplexers (column 7 and line 12 of Table 5.5). It should be noted that savings in TAT reflect not only the time required to complete the test sessions shown in Tables 5.1 and 5.2, but also the time required to shift in the generators' seeds and shift out analysers' signatures at the end of each test session. Due to less test registers required by the proposed methodology and lower number of test sessions, there is consistent reduction in test application time. For example, in the case of *32DCT-30* there is 15.79% reduction

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	584	40	89	40	664	48	89	128
EWF-21	584	43	98	40	664	50	98	128
EWF-23	560	41	91	40	664	44	91	152
EWF-25	568	36	90	40	632	45	90	128
EWF-28	568	35	98	40	624	39	98	104
8DCT-10	608	45	94	40	648	56	92	88
8DCT-12	600	46	97	40	696	46	97	144
8DCT-13	600	46	90	40	728	55	90	176
8DCT-14	592	45	91	40	704	58	90	160
8DCT-16	568	48	93	40	688	51	93	184
8DCT-21	560	45	94	40	688	59	94	184
32DCT-30	896	101	92	64	1456	114	92	720
32DCT-31	880	100	93	64	1424	132	92	696
32DCT-32	848	96	93	64	1488	120	93	736
32DCT-33	664	95	92	40	1128	100	92	624
32DCT-37	672	85	92	40	1104	100	91	600
32DCT-38	680	86	94	40	1136	114	93	640
32DCT-40	688	84	93	40	1152	90	93	648

Table 5.4: Experimental data for TCC grouping methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping in 0.35 micron AMS technology

when using only BIST hardware and 5.70% when employing scan to test the untested registers and multiplexers. The lower reduction obtained when using scan is explained by the lower number of test registers and hence a higher number of untested registers which need to be scanned in the case of the proposed TCC grouping methodology. This implies that the extra test session required to test the untested multiplexers will require longer test application time in the case of TCC grouping. However, this is compensated over the entire test application period by the reduction in the number of test sessions and the number of seeds to be shifted in and signatures to be shifted out in the case of TCC grouping (Table 5.2). So far the reductions in TAT and BIST area overhead achieved by the TCC grouping methodology when compared to the BIST embedding methodology were outlined. Table 5.5 also shows the reductions in performance degradation (PD) and volume of test data. The reduction in PD represents the increase in clock frequency for TCC

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)
EWF-17	23.16	11.11	3.37	66.67	15.31	10.78	3.37	15.79
EWF-21	20.65	0.00	0.00	61.54	12.63	0.00	0.00	5.88
EWF-23	20.45	10.87	0.00	37.50	17.82	10.22	1.10	13.64
EWF-25	1.39	12.20	0.00	37.50	5.95	11.22	0.00	15.79
EWF-28	19.32	28.57	0.00	37.50	25.00	27.71	0.00	48.00
8DCT-10	7.32	26.23	4.26	75.00	1.22	22.56	2.17	45.00
8DCT-12	22.68	9.80	6.19	70.59	18.69	8.62	6.19	35.71
8DCT-13	24.24	6.12	1.11	75.00	17.27	4.53	3.33	31.25
8DCT-14	22.11	23.73	0.00	66.67	18.52	19.22	0.00	31.03
8DCT-16	6.58	9.43	3.23	58.33	8.51	7.92	3.23	20.69
8DCT-21	5.41	11.76	1.06	50.00	7.53	8.94	5.32	20.69
32DCT-30	15.79	18.55	0.00	85.19	5.70	15.77	7.61	13.46
32DCT-31	17.29	19.35	1.08	85.19	5.32	16.64	4.35	13.00
32DCT-32	19.70	16.52	1.08	84.62	1.59	13.71	1.08	14.02
32DCT-33	26.55	15.93	5.43	89.80	19.43	13.22	6.52	20.41
32DCT-37	21.50	15.84	1.09	88.37	21.14	11.56	0.00	22.68
32DCT-38	21.30	21.10	0.00	88.89	14.46	16.46	0.00	8.05
32DCT-40	20.37	20.00	1.08	88.64	14.29	14.80	3.23	7.95

Table 5.5: The reduction of the proposed TCC grouping methodology over the BIST embedding methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data

grouping after the full description of the circuit (including functional and test controller) were technology mapped into 0.35 AMS technology. For example the reduction in PD for *32DCT-30* is 0.00% when using only BIST hardware and 7.61% when also employing scan, as shown in Table 5.5. The reduction when employing scan can be explained that for both BIST embedding and TCC grouping methodologies applied to *32DCT-30*, the longest delay on the critical path is dependent not only on the number of BIST resources, but also on the functional registers that are scanned to test the untested multiplexers. Also, due to higher number of vectors to be scanned in untested registers the reduction in VTD when employing BIST hardware and scan is lower when compared to using only BIST hardware. In the case of *32DCT-30* the reduction values for VTD are 85.19% when using only BIST hardware and 13.46% when employing both BIST hardware and scan. All the reduction values of the proposed TCC grouping methodology over the traditional BIST embedding methodology shown in Table 5.5 are computed using the experimental values given in Table 5.3 for BIST embedding methodology and Table 5.4 respectively for TCC grouping methodology. It is interesting to note that the constant savings of the proposed TCC grouping over the BIST embedding methodology are due to the regularity of the data paths where a large number of functional units, such as adders and multipliers, have identical implementations exploited by TCCs (Section 5.2.2).

Although the emphasis of the work presented in this chapter is on built-in self-testable data paths the following discussion clarifies the issue of testing the controller itself. Since the proposed BIST methodology targets data flow intensive applications there are only a few flip flops required to implement the functional controller. For example, in the case of *32DCT-30* only 5 state elements implement the 30 control states. Furthermore, the size of the BIST controller is small and the BIST controller is merged with the functional controller for both controlling the signals during data path testing and testing the controller itself as outlined in Figure 1.10(b) from Section 1.3.2. The merged controller leads to small area overhead when compared to the size of the data path of 60 registers, 9 multipliers and 12 adders (Table 5.1). One way of testing the controller is by configuring all the control sequential elements into LFSR. This does not add any performance degradation for data flow intensive applications since the clock period is constrained by the critical path in the complex data path. Furthermore, both the size of the merged controller and test application time for the controller are smaller in the case of the proposed TCC grouping when compared to BIST embedding due to the smaller number of cycles required to shift out test responses and less logic required to implement control signals during testing. If the random patterns generated by the LFSR are not satisfactory to detect all the faults in the controller, an alternative way of testing the controller is to add some additional DFT such as scan, and shift in each pattern required to test the controller.

5.4.3 Comparison for Complex Hypothetical Data Paths

The BIST hardware synthesis for TCC grouping methodology allows the huge testable design space to be explored efficiently by combining the accuracy of incremental test scheduling algorithms and the exploration speed of test scheduling algorithms based on fixed test resource allocation, as outlined in Section 5.3. This means it can be used with extremely complex hypothetical designs of dimensions not often reported in literature.

Design	**M**	**R**	**BIST embedding**	
	O	E	BIST resources	Test
	D	**G**	Test registers	Sessions
EX-01	35	90	54 LFSR, 30 MISR	8×T_u
EX-02	35	95	55 LFSR, 28 MISR, 1 BILBO	9×T_u
EX-03	35	100	56 LFSR, 29 MISR, 2 BILBO	8×T_u
EX-04	35	105	67 LFSR, 32 MISR, 1 BILBO	5×T_u
EX-05	35	110	65 LFSR, 32 MISR, 1 BILBO	9×T_u
EX-06	35	115	67 LFSR, 34 MISR, 1 BILBO	5×T_u
EX-07	40	90	58 LFSR, 25 MISR	8×T_u
EX-08	40	95	58 LFSR, 26 MISR, 4 BILBO	8×T_u
EX-09	40	100	59 LFSR, 29 MISR, 1 BILBO	8×T_u
EX-10	40	105	61 LFSR, 30 MISR, 3 BILBO	8×T_u
EX-11	40	110	65 LFSR, 32 MISR, 1 BILBO	8×T_u
EX-12	40	115	63 LFSR, 35 MISR, 2 BILBO	8×T_u
EX-13	45	90	56 LFSR, 24 MISR, 4 BILBO	8×T_u
EX-14	45	95	57 LFSR, 26 MISR, 3 BILBO	9×T_u
EX-15	45	100	57 LFSR, 32 MISR, 4 BILBO	8×T_u
EX-16	45	105	60 LFSR, 31 MISR, 3 BILBO	12×T_u
EX-17	45	110	63 LFSR, 30 MISR, 9 BILBO	8×T_u
EX-18	45	115	67 LFSR, 32 MISR, 2 BILBO	8×T_u

Table 5.6: BIST resources and number of test sessions when using the BIST embedding methodology for complex hypothetical data paths with 35 to 45 modules and 90 to 115 registers

Complex hypothetical data paths were generated as described in the following. The number of modules n_{mod} varies from 35 to 45, and the number of registers n_{reg} varies from 90 to 115. The number of module-types is $n_{res} = 5$. The maximum fanin for every register or input port of a module is $M_{fanin} = 8$. The input register set of each input port of every module contains a random number n_r, with $1 \leq n_r \leq M_{fanin}$, of randomly chosen registers. Similarly, the number of modules multiplexed at the input of each register is a random number n_m, with $1 \leq n_m \leq M_{fanin}$, of randomly chosen modules. The test length of three module-types is assumed T_u, and in the case of the other two module-types the test length is considered 4×T_u. Tables 5.6 and 5.7 show a comparison of BIST resources and number of test sessions using the TCC grouping and the BIST embedding methodologies for complex hypothetical data paths. For most of the complex hypothetical data paths the BIST embedding methodology needed BILBO registers to achieve low number of test sessions. For example, in the case of *EX-16* with 45 modules and 105 registers, the

Design	**TCC grouping**			**CPU time (s)**
	BIST resources		Test Sessions	
	Test Registers	Comparators		
EX-01	42 LFSR, 4 MISR	C8, 2C7, 2C6	$5\times T_u$	242
EX-02	35 LFSR, 3 MISR	C9, 2 C7, C5, 2 C3	$8\times T_u$	383
EX-03	38 LFSR, 3 MISR	C9, C7, C6, C5, 2 C4	$5\times T_u$	491
EX-04	43 LFSR, 4 MISR	2 C8, 3 C6	$4\times T_u$	317
EX-05	44 LFSR, 3 MISR	C8, C7, 2 C6, C4, 2 C2	$9\times T_u$	559
EX-06	44 LFSR, 4 MISR	C9, 2 C7, C6, C5	$5\times T_u$	515
EX-07	39 LFSR, 4 MISR	C12, C6, C5, 2 C4, 2 C3	$6\times T_u$	480
EX-08	41 LFSR, 3 MISR	C9, C8, 2 C6, 2 C4, C3	$8\times T_u$	416
EX-09	47 LFSR, 3 MISR	C9, C8, 2 C7, C5, C3	$8\times T_u$	352
EX-10	40 LFSR, 3 MISR	2 C10, 2 C6, 2 C4	$6\times T_u$	532
EX-11	45 LFSR, 4 MISR	C9, 2 C8, C7, C6	$8\times T_u$	588
EX-12	48 LFSR, 4 MISR	C9, C8, 2 C7, C6	$5\times T_u$	554
EX-13	48 LFSR, 3 MISR	C9, C8, 2 C7, C5, C4	$6\times T_u$	543
EX-14	46 LFSR, 3 MISR	C12, C9, C7, C6, 2 C4, C3	$8\times T_u$	561
EX-15	52 LFSR, 4 MISR	C13, C7, 3 C6, C5, C2	$5\times T_u$	417
EX-16	47 LFSR, 2 MISR	C12, C10, 3 C7, C2	$6\times T_u$	594
EX-17	52 LFSR, 5 MISR	C13, C7, C6, 2 C5, C4, C2	$8\times T_u$	502
EX-18	53 LFSR, 4 MISR	C12, C11, C9, C7, C5	$5\times T_u$	593

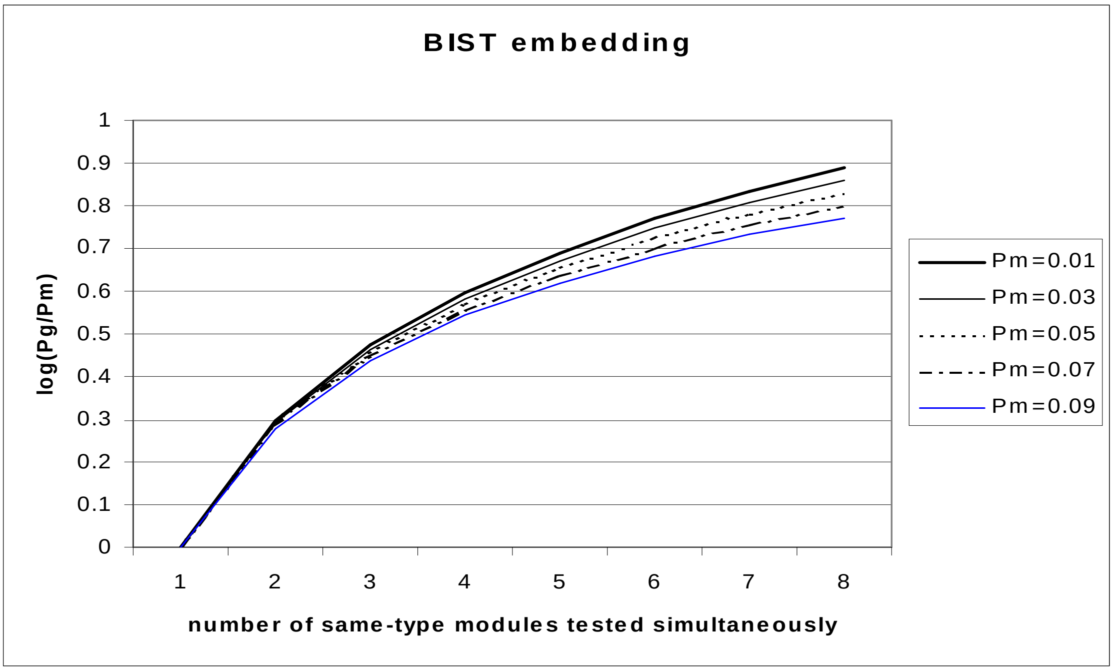
Table 5.7: BIST resources and number of test sessions when using the TCC grouping methodology for complex hypothetical data paths with 35 to 45 modules and 90 to 115 registers

BIST embedding methodology required 60 LFSRs, 31 MISRs and 3 BILBOs to achieve a time to complete the test sessions of $12\times T_u$. The TCC grouping achieves a time to complete the test sessions of $6\times T_u$ and reduces the number of test registers from 94 to only 49, without use of BILBO registers. Furthermore, it requires only 2 signature analysis registers reusable in different test sessions. This proves the efficiency of simultaneous TCC scheduling and signature analysis register allocation described in Section 5.3.3. Furthermore, the computational time is still very low related to the size of the testable design space. For example it took less than 600s to find high quality solutions for data paths with 45 modules and up to 115 registers. The proposed BIST methodology is applicable to variable data path widths as outlined in Section 5.2. Appendix B gives exhaustive experimental data and reductions achieved by the proposed TCC grouping methodology for data path widths of 4, 8, and 16 bits for all the previously described benchmark circuits (Tables 5.1 and 5.2) and hypothetical data paths (Tables 5.6 and 5.7).

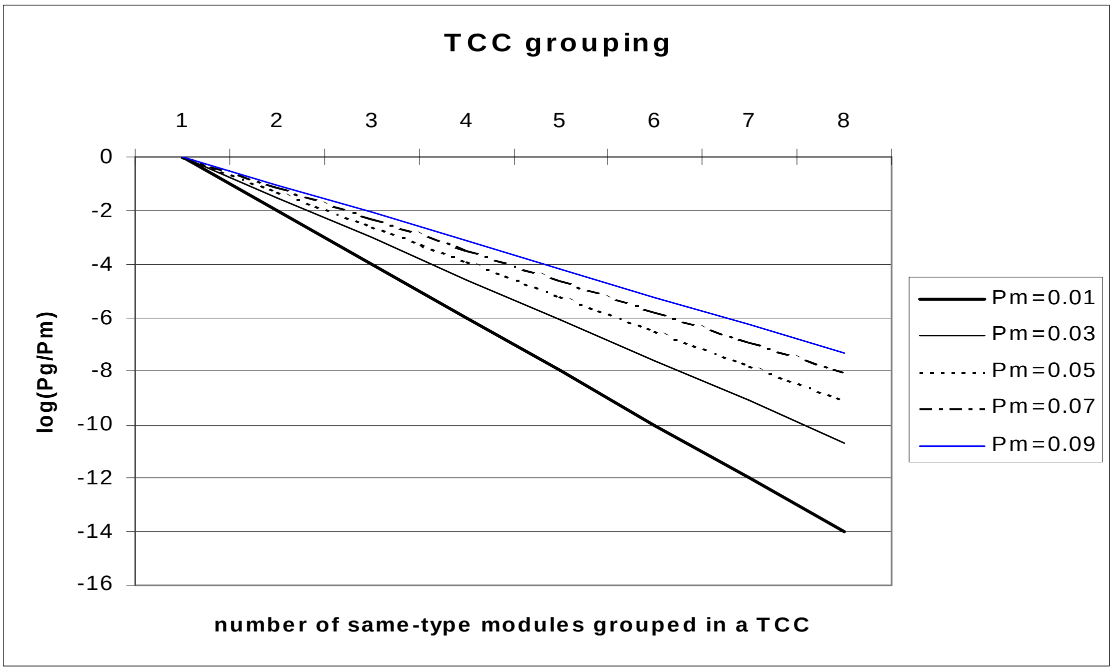
5.4.4 Reduction in Fault-Escape Probability of the Proposed TCC Grouping Methodology Over the BIST Embedding Methodology

So far the experimental results have shown that the proposed TCC grouping methodology yields savings over the traditional BIST embedding methodology in the following BIST parameters: test application time, BIST area overhead, performance degradation, and volume of test data. Further, it was shown that the proposed BIST hardware synthesis algorithm efficiently explores the testable design space in low computational time. This section gives the results for fault-escape probability which is an important BIST parameter as outlined in Section 1.3.2 from Chapter 1.

Figure 5.11 shows how the proposed TCC grouping methodology decreases the fault-escape probability [139, 173] when compared to the BIST embedding methodology. The experiments were done for a data path module with 10^6 possible error sequences, where the aliasing error sequences, for a given characteristic polynomial of signature analysis register, vary from 10 to 90. Fault-escape probability of a module varies from $P_m = 0.01\%$ to $P_m = 0.09\%$. As it can be seen from Figure 5.11(a), in the case of BIST embedding methodology the fault-escape probability for group of modules (P_g) increases as the number of modules tested simultaneously increases. On the other hand, in the case of the TCC grouping, the fault-escape probability decreases exponentially with the number of modules tested simultaneously as shown in Figure 5.11(b). This is due to the fact that a fault is not detected in the TCC grouping methodology only when initially the n-input k-bit comparator fails to detect the fault and subsequently the signature of a TCC also fails to detect the fault. A previous work on reducing fault-escape probability at the expense of increased area overhead, performance degradation, and volume of test data was presented in [114]. Note that the proposed methodology does not introduce either extra area overhead, nor performance degradation, whilst the reduction in fault-escape probability is exponential. This is due to the fact that the reduction in fault-escape probability is a by-product of the proposed test compatibility classes that use comparators to test the *concurrent* same-type modules (Definition 5.3 from Section 5.2.2), and it is also a by-product of the testable design space exploration which aims to increase test concurrency as outlined in Section 5.3.1.



(a) Increase in fault-escape probability for BIST embedding



(b) Decrease in fault-escape probability for TCC grouping

Figure 5.11: Comparison in fault-escape probability when 1 to 8 same-type modules are tested simultaneously in BIST embedding and TCC grouping methodologies

5.5 Concluding Remarks

This chapter has addressed the testability of RTL data paths using BIST. The new BIST methodology is based on grouping modules with identical physical information into TCCs and testing the compatible modules by sharing a small number of test pattern generators at the same test time. An *n*-input *k*-bit comparator checks module output responses from each TCC reducing the fault-escape probability and the number of signatures that have to be shifted out. The proposed TCC grouping methodology is suitable for RTL data paths with both uniform and variable bit width. A new BIST hardware synthesis uses efficient tabu search-based testable design space exploration which combines the accuracy of incremental test scheduling algorithms with the exploration speed of test scheduling algorithms based on fixed test resource allocation. The huge size of the testable design space is reduced by considering only the representative partially testable data paths during the local neighbourhood search. An incremental TCC scheduling algorithm further shrinks the size of the testable design space by generating a fully testable data path using simultaneous test scheduling and signature analysis registers allocation. BIST hardware synthesis algorithm for the proposed TCC grouping methodology was validated exhaustively for benchmark and complex hypothetical data paths. When compared to the traditional BIST embedding methodology, the TCC grouping methodology is capable of reducing the test application time with lower BIST area overhead and yielding savings in performance degradation, volume of test data, and fault-escape probability. Furthermore the proposed BIST hardware synthesis algorithm achieves high quality of the final solution in low computational time.

Chapter 6

Low Power BIST for RTL Data Paths

Considering power dissipation during test application at logic level of abstraction of the VLSI design flow was considered in Chapters 3 and 4. It was shown that the novel Best Primary Input Change (BPIC) test application strategy introduced in Chapter 3 reduces power dissipation during test application in small to medium sized scan sequential circuits with no penalty in test area, performance, test efficiency, test application time or volume of test data. To extend power minimisation during test application at the logic level to large scan sequential circuits, the new multiple scan chain technique presented in Chapter 4 is based on a test set independent approach and achieves significant power savings in low computational time with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance. However, considering testability at the logic level of abstraction using scan requires expensive external automatic test equipment and does not support in-field and at-speed testing as outlined in Section 1.2 from Chapter 1. Furthermore, to avoid unnecessary iterations in the design flow, recent research interests have shifted towards the investigation of addressing testability at high levels of abstraction during the early stages of the VLSI design flow [54]. Therefore, Chapter 5 introduced a novel BIST methodology for RTL data paths using a new concept called test compatibility classes (TCC) which overcomes the problems of test application time, BIST area overhead, performance degradation, volume of test data, and fault-escape probability associated with traditional BIST embedding methodology [114, 115]. To fully exploit the testability benefits of BIST RTL data paths, power dissipation during test application in BIST RTL data paths needs to be accounted, and novel power conscious test synthesis and scheduling algorithms equally applicable to BIST embedding and TCC grouping methodologies need to be developed. This is of particular importance when power dissipation during the functional operation is not exceeding a given power constraint as it is the case for RTL data paths synthesised using low power high level synthesis algorithms [30, 96, 103, 104, 105, 106, 107] outlined in Figures 2.2 and 2.3 in Section 2.2.1 from Chapter 2.

The aim of this chapter is to introduce new power conscious test synthesis and test scheduling algorithms that account for power dissipation during the testable design space exploration. It was established in Section 1.4 from Chapter 1, that power dissipation and BIST area overhead decrease as test application time increases. Since power dissipation is dependent on switching activity of *all* the active elements during every test session, there is a significant variation in power dissipation due to *useless power dissipation* introduced in Section 6.2. This chapter shows that considering the interrelation between test synthesis and test scheduling described in Section 1.3.2 from Chapter 1, and investigating their impact on power dissipation during test application, *useless power dissipation* is eliminated which leads to low power BIST RTL data paths. This chapter explains how the proposed power conscious test synthesis and test scheduling algorithms are integrated in the testable design space exploration described in Figure 5.4 of Section 5.3.1 from Chapter 5, and how this leads to savings in power dissipation both during test application and while shifting out of test responses, at the expense of low overhead in computational time. Moreover, it is shown that when combining the new power conscious test synthesis and test scheduling algorithms with the proposed test compatibility classes (Chapter 5), simultaneous reduction in test application time and power dissipation is achieved with constant savings in BIST area overhead when compared to the traditional BIST embedding methodology.

The rest of the chapter is organised as follows. The motivation and objectives of the undertaken research are given in Section 6.1. Section 6.2 accounts for sources of power dissipation in BIST RTL data paths and outlines a taxonomy for power dissipation in BIST RTL data paths. The effect of test synthesis and test scheduling on power dissipation during test application is investigated in Section 6.3. Power constrained testable design space exploration, using novel power conscious test synthesis and test scheduling algorithms, is described in Section 6.4. Experimental results of both traditional BIST embedding and TCC grouping methodologies are given Section 6.5. Finally, some concluding remarks are given in Section 6.6.

6.1 Motivation and Objectives

Section 2.2 from Chapter 2 has reviewed the previous work on low power BIST both at logic level and register-transfer level of abstraction of the VLSI design flow. While low power BIST techniques at logic level of abstraction [86, 60, 68, 71, 189, 196, 198] yield modest savings in power dissipation they can be combined with techniques proposed at higher levels of abstraction to produce further savings in power dissipation during test application. Considering power minimisation at high level of abstraction is of further importance when various alternatives in the low power design space are explored at high levels of abstraction [30, 96, 103, 104, 105, 106, 107]. Despite their efficiency the previous approaches [25, 41, 110, 132, 165, 199] for minimising power dissipation during test application at higher level of abstraction than logic level *are not suitable* for BIST RTL data paths due to the following two problems:

a. test scheduling assumes fixed amount of power dissipation associated with each test which is not the case for BIST RTL data paths as outlined in Section 6.2;

b. test scheduling is performed on a fixed test resource allocation without considering the strong interrelation between test synthesis and test scheduling as outlined in Figure 1.9 of Section 1.3.2 from Chapter 1;

The fixed amount of power dissipation assumption (problem (a)) is not valid in the case of BIST RTL data paths where transitions associated with necessary power dissipation required for testing each module can propagate to other registers and further to untested modules leading to useless power dissipation. The useless power dissipation introduced in Section 6.2 does not have any influence on test efficiency and is not caused only by test scheduling but also by test synthesis as outlined in Section 6.3. Therefore, since test synthesis and test scheduling are strictly interrelated as described in Section 1.3.2 from Chapter 1, the previously proposed power constrained test scheduling algorithms [25, 41, 110, 132, 165, 199] based on fixed test resource allocation (problem (b)) will lead to prohibitively large computational time hindering efficient exploration of the testable design space. Thus, new power conscious test synthesis and test scheduling algorithms which take into account the interrelation between test synthesis and test scheduling for minimisation of necessary power dissipation and elimination of useless power dissipation are required.

6.2 Power Dissipation Classification During Test Application in BIST RTL Data Paths

This section gives a taxonomy of power dissipation during test application in BIST RTL data paths. According to the necessity for achieving the required test efficiency, power dissipation is classified into necessary and useless power dissipation, as defined in the following.

Definition 6.1 *Necessary power dissipation* is the power dissipated in test registers and tested modules during each test session and the power dissipated in test registers while shifting in seeds for test pattern generators and shifting out responses from signature analysers.

Necessary power dissipation is compulsory for achieving the required test efficiency, however, the useless power dissipation must be eliminated. In order to introduce useless power dissipation, firstly spurious transitions in BIST RTL data paths are defined. While previous Chapters 3 and 4 have introduced *spurious transitions during test application* in scan sequential circuits at the logic level of abstraction (Definitions 3.1 and 4.1), the following definition introduces spurious transitions when using BIST for RTL data paths.

Definition 6.2 A spurious transition when employing BIST for RTL data paths is a transition which occurs in modules and/or registers which are not used in the current test session. These transitions do not have any influence on test efficiency since the values at the input and output of modules and/or values loaded in registers are not useful test data.

Definition 6.3 *Useless power dissipation* is the power dissipated in registers and untested modules due to spurious transitions which cannot be eliminated by any configuration of control signals of data path multiplexers.

Example 6.1 To show sources of useless power dissipation, consider the BIST RTL data path shown in Figure 6.1, where $MISR_0$ and $MISR_1$ are active simultaneously. For the sake of simplicity, the modules whose output responses are analysed by $MISR_0$ and $MISR_1$, do not appear in Figure 6.1. The output of $MISR_0$ is connected to M_1 while $MISR_1$ is connected to M_1 and M_2. In the case of M_2, the inactive register R_2 can be selected by using the appropriate value of control signal c_2 which stops the propagation of

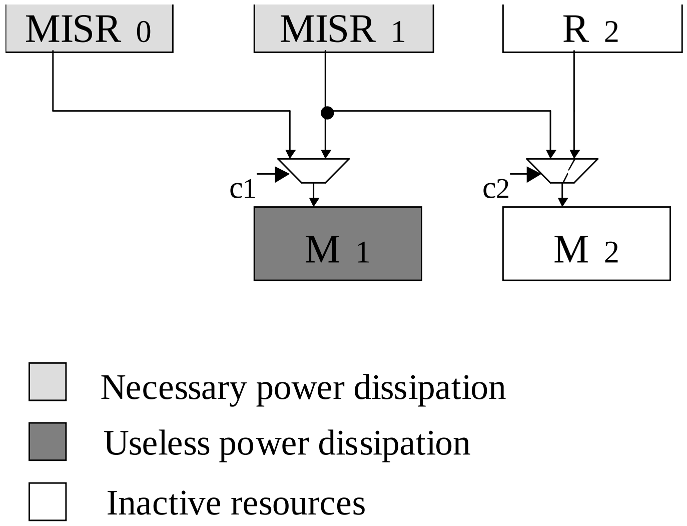


Figure 6.1: Useless power dissipation during test application in BIST RTL data paths.

transitions that occur in $MISR_1$. However, in the case of M_1 there is no value for control signal c_1 such that transitions which occur in $MISR_0$ or $MISR_1$ are eliminated at the input of M_1. Thus spurious transitions occur in M_1 without any influence on test efficiency leading to useless power dissipation.

According to the occurrence during the testing process power dissipation is classified into test application and shifting power dissipation, as defined in the following.

Definition 6.4 *Test Application Power dissipation* (TAP) is the power which occurs during execution of each test session when test patterns, necessary to achieve the required test efficiency, are applied to modules.

Definition 6.5 *Shifting Power dissipation* (SP) is the power which occurs while shifting in the seeds for test pattern generators, required for next test session, and shifting out the responses stored in signature analysers at the end of the previous test session.

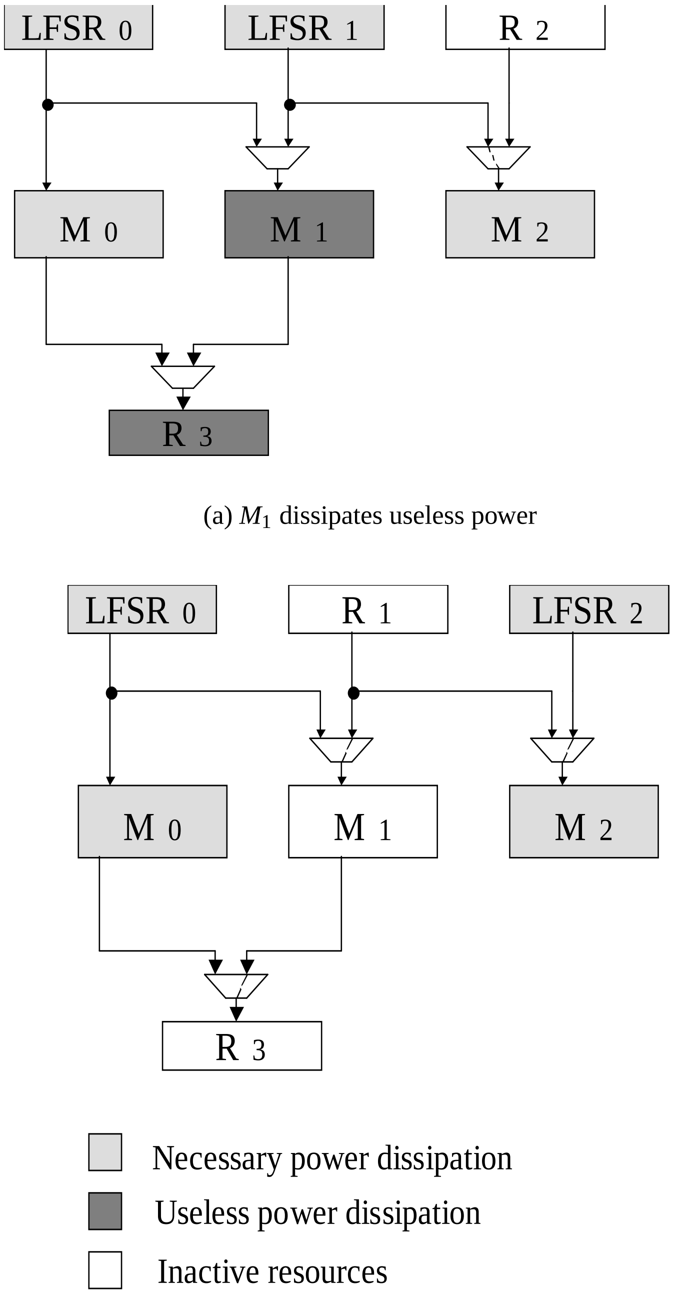
It should be noted that multiplexer power is very low during test application, which is unlike the case of functional operation where power dissipated by multiplexers is large [162]. This is due to the fact that multiplexer control signals are modified only at the start and at the end of every test session, thus avoiding any glitching activity which can propagate from control logic.

6.3 Effect of Test Synthesis and Scheduling on Useless Power Dissipation

In order to eliminate useless power dissipation, the effect of test synthesis and scheduling on useless power dissipation is analysed through three detailed examples. The first example analyses the effect of test synthesis on both test application and shifting power in modules and registers. The second example examines the effect of module selection during test scheduling on elimination of useless power dissipation in both registers and modules. The third example illustrates the effect of power conscious test synthesis on BIST area overhead, performance degradation and volume of test data for the example data path illustrated shown in Figure 2.3 from Section 2.2.1 of Chapter 2. From now onwards the illustrative examples from Section 6.3 and algorithms from Section 6.4 are equally applicable to both BIST embedding and TCC grouping. The only modification exclusively applicable to test compatibility classes is outlined in Section 6.4.3.

In order to show the need for eliminating useless power dissipation, outlined in Section 6.2, during the testable design space exploration described in Section 5.3.1 of the previous chapter, the following Example 6.2 investigates the effect of test synthesis on test application and shifting power dissipation.

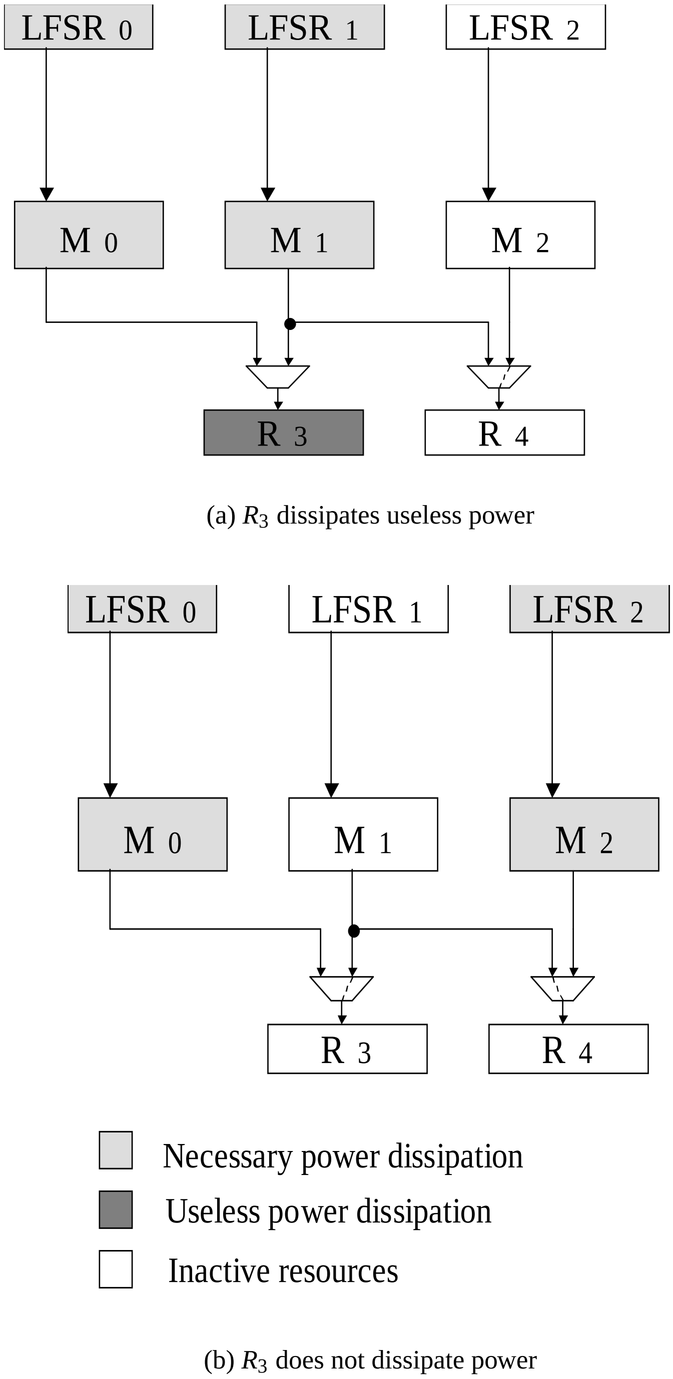
Example 6.2 Consider the BIST RTL data path shown in Figure 6.2 and assume that modules M_0 and M_2 are tested simultaneously without exceeding the given power constraints. When linear feedback shift register $LFSR_1$ generates test patterns for M_2 any configuration of control signals for multiplexer at the input of M_1 will lead to useless power dissipation in M_1 (Figure 6.2(a)). Moreover, the useless power dissipation in M_1 will further be propagated to R_3 leading to useless power dissipation in both modules and registers. However when $LFSR_2$ generates test patterns for M_2, by selecting the inactive register R_1 at the input of M_1 will lead to the elimination of useless power in M_1 without any penalty in test area or test efficiency (Figure 6.2(b)). $LFSR_2$ selection also implies that useless power dissipation in R_3 is eliminated. It should be noted that test synthesis has a profound impact also on shifting power (SP in section 6.2) since an inappropriate selection of test registers may lead to useless power dissipation in M_1 and R_3 in modules while shifting in the seeds for test pattern generators $LFSR_0$ and $LFSR_1$ (Figure 6.2(a)).



(a) M_1 dissipates useless power

(b) M_1 does not dissipate power

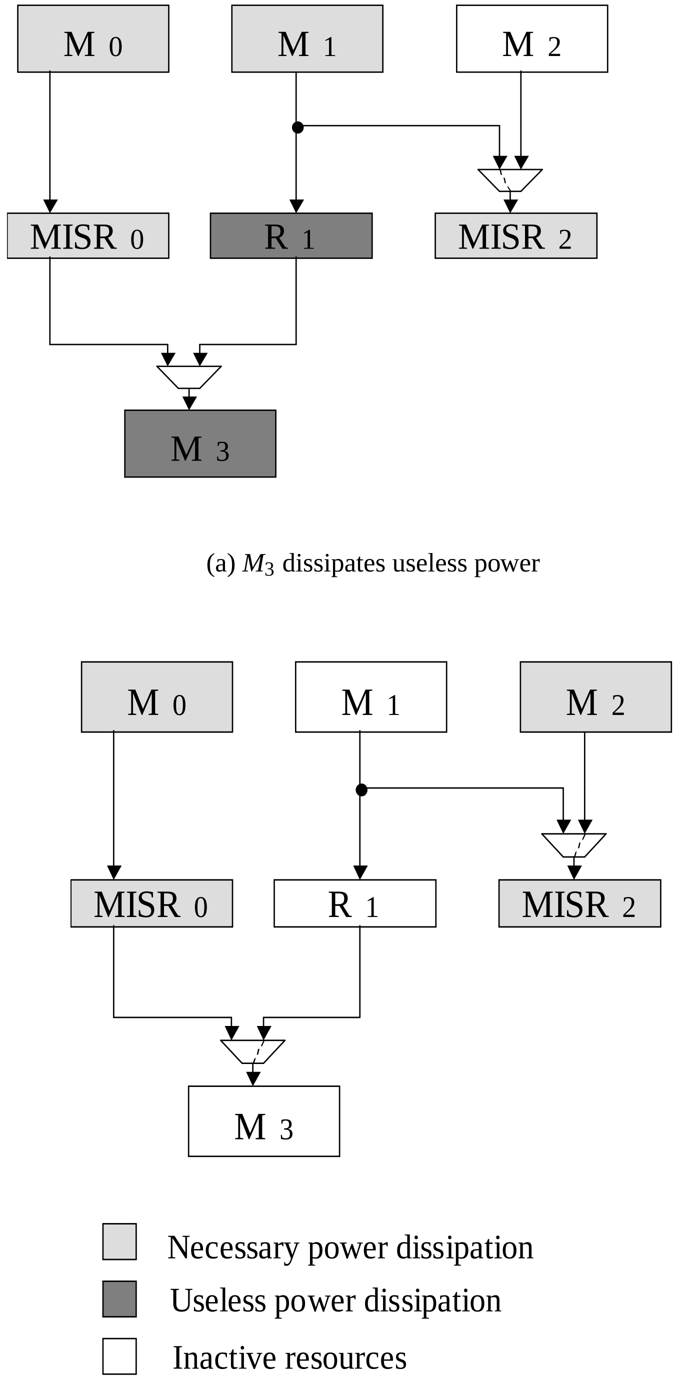
Figure 6.2: Test synthesis for useless power elimination (Example 6.2).



(a) R_3 dissipates useless power

(b) R_3 does not dissipate power

Figure 6.3: Module selection during test scheduling for useless power elimination in registers (Example 6.3).



(a) M_3 dissipates useless power

(b) M_3 does not dissipate power

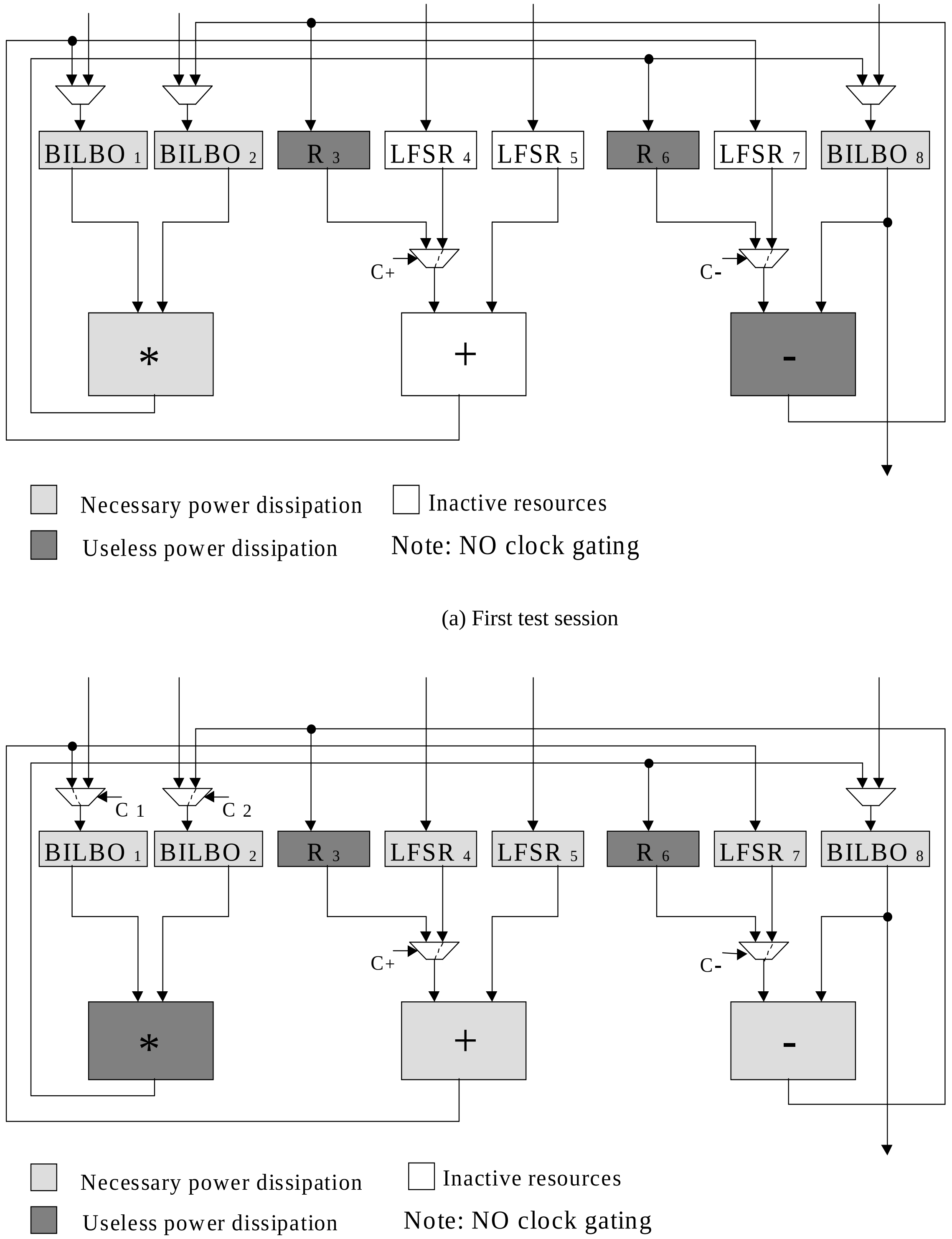
Figure 6.4: Module selection during test scheduling for useless power elimination in modules (Example 6.3).

Having described the effect of test synthesis on useless power dissipation, the following Example 6.3 examines the effect of module selection during test scheduling on useless power elimination in both registers and modules.

Example 6.3 Consider the BIST RTL data path shown in Figure 6.3. Assume that module M_0 is already scheduled in the current test session and the selection of M_1 and M_2 is examined. It should be noted that for the sake of simplicity signature analysis registers for M_0, M_1, and M_2 are not shown in Figure 6.3 and registers R_3 and R_4 are not used as analysers in the current test session. By selecting M_1 to be tested simultaneously with M_0, and by choosing the inactive module M_2 at the input of register R_4 useless power is eliminated in R_4. However, any configuration of control signals for the multiplexer at the input of R_3 will lead to useless power dissipation in R_3 (Figure 6.3(a)). The useless power in both R_3 and R_4 is eliminated by selecting M_2 to be tested simultaneously with M_0 and setting the appropriate values on control signals of multiplexers at the inputs of R_3 and R_4 (Figure 6.3(b)). So far the effect of test scheduling on useless power in registers was outlined. To examine the effect of test scheduling on useless power in modules consider the circuit shown in Figure 6.4. Assume that $MISR_0$ analyses test responses from M_0 and $MISR_2$ can analyse test responses from either M_1 or M_2. Scheduling the test for M_1 at the same time with the test for M_0 will lead not only to useless power dissipation in R_1 but also in M_3. This is because both $MISR_0$ and R_1 are active at the same time which leads to propagation of spurious transitions from R_1 to M_3 (Figure 6.4(a)). The useless power is eliminated in both R_1 and M_3 by selecting M_2 to be tested simultaneously with M_0 and setting the appropriate values on control signals of multiplexers at the input of M_3 (Figure 6.4(b)). This clearly shows that test scheduling has effect on useless power dissipation in both registers and modules.

The previous two examples have investigated the effect of test synthesis and test scheduling on useless power dissipation in BIST RTL data paths. It was found that both test synthesis and test scheduling have a profound impact on useless power dissipation. Detailed description of the modifications required during the testable design space exploration described in Figure 5.4 from section 5.3.1 in chapter 5, such that useless power dissipation is accounted, are described in the following section 6.4. The following Example 6.4 investigates the impact of power conscious test synthesis and test scheduling on BIST area overhead, performance degradation and volume of test data.

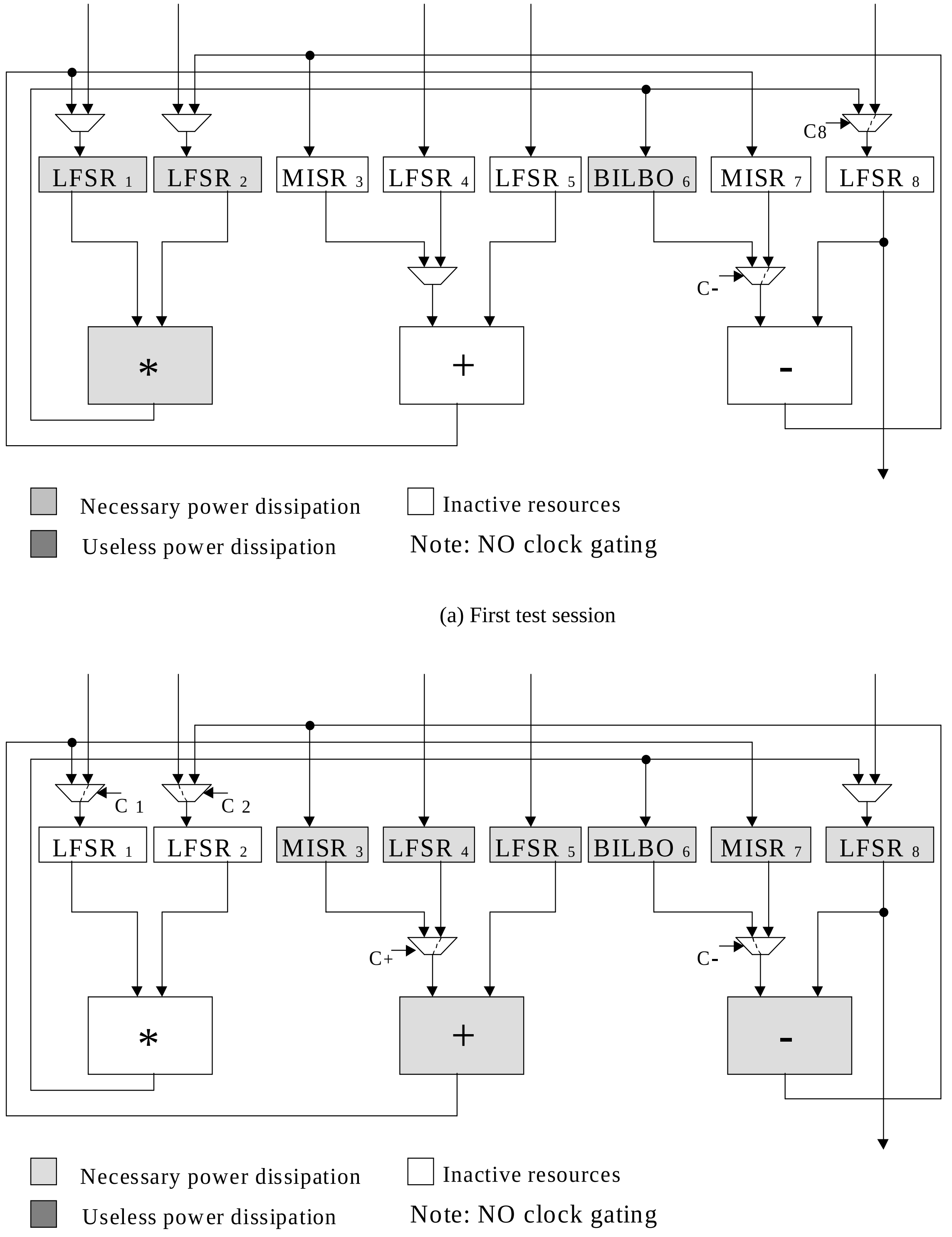
Example 6.4 It was shown in section 2.2.1 from chapter 2, how data flow graph shown in Figure 2.2 is synthesised in the data path shown in Figure 2.3 such that low power dissipation during functional operation is achieved. As described in Example 2.2 from chapter 2, during clock cycles 1 and 4 the active elements are registers R_1, R_2, and multiplier $(*)$. Since excessive power dissipation during BIST can damage the circuit under test, as described in section 2.2.1 from chapter 2, it is important that data path circuit is tested in two separate sessions, one for multiplier $(*)$, and one for adder $(+)$ and subtracter $(-)$. Figure 6.5 illustrates the self-testable data path in two test sessions: first session for the multiplier $(*)$ (Figure 6.5(a)) and second session for the adder $(+)$ and subtracter $(-)$ (Figure 6.5(b)). The BIST RTL data path shown in Figure 6.5 is obtained using the testable design space exploration algorithm from the section 5.3 in chapter 5, such that a given power constraint derived from functional operation is not exceeded during test application. From now onwards the test synthesis and test scheduling algorithm described in section 5.3 such that a given power constraint is not exceeded is referred to as Time and Area Test Synthesis and Scheduling (TA-TSS). Figure 6.6 illustrates the BIST RTL data path in the two test sessions when applying the Power Conscious Test Synthesis and Scheduling (PC-TSS) detailed in the next section 6.4 using the observations from Examples 6.2 and 6.3. The main objective of TA-TSS is to minimise test application time under the given power constraint with BIST area overhead used as tie-breaking mechanism among many possible solutions with same test application time as outlined in Definition 5.5 of Section 5.3.1. Unlike TA-TSS, the main objective of PC-TSS is to eliminate useless power dissipation, and then it uses test application time and BIST area overhead as tie-breaking mechanism as outlined in Section 6.4. Therefore, PC-TSS leads to more test registers than TA-TSS with the benefit of eliminating useless power dissipation. It should be noted that the power constraint is exceeded in both test sessions when TA-TSS is employed due to useless power dissipation shown in registers R_3, R_6 and subtracter $(-)$ of Figure 6.5(a) and registers R_3, R_6 and multiplier $(*)$ of Figure 6.5(b). In order to show that TA-TSS ignores useless power dissipation and hence exceeds power constraints the following experiment was conducted. Registers, test registers, and functional units (modules) were synthesised and technology mapped into AMS 0.35 micron technology [9]. Using a real delay model simulator [130] and cell library timing and power information operating at supply voltage 3.3V and clock frequency 100MHz, the following power values were obtained for 8 bit data path width using pseudorandom sequences



(a) First test session

(b) Second test session

Figure 6.5: Time and Area Test Synthesis and Scheduling (TA-TSS) for data path circuit from Figure 2.3 in section 2.2.1 from chapter 2.



(a) First test session

(b) Second test session

Figure 6.6: Power Conscious Test Synthesis and Scheduling (PC-TSS) for data path circuit from Figure 2.3 in section 2.2.1 from chapter 2.

applied during testing: $P_{REG} = 0.8mW$, $P_{LFSR} = 1mW$, $P_{MISR} = 2mW$, $P_{BILBO} = 2.5mW$, $P_{+} = 3.5mW$, $P_{-} = 3.5mW$, and $P_{*} = 11.5mW$. Using the register and module activity from Example 2.2 from Chapter 2, the power dissipated during functional operation of the data path from Figure 6.5 is $16.5mW$ in clock cycles 1 and 4, due to the activity of the following elements: $BILBO_1$, $BILBO_2$, and $(*)$. Considering manufacturing process tolerance the power constraint during testing is set to $20mW$. When using TA-TSS, due to ignorance of useless power dissipation during test synthesis and scheduling of the algorithm from Section 5.3.1, power value for the first test session (Figure 6.5(a)) is $24.1mW$, and $30.1mW$ for the second test session (Figure 6.5(b)). This clearly shows that ignoring useless power dissipation there is significant violation of the power constraint and substantially higher power dissipation during testing. However, when employing PC-TSS the data path circuit from Figure 6.6 dissipates $13.5mW$ in clock cycles 1 and 4, due to the activity of the following elements: $LFSR_1$, $LFSR_2$, and $(*)$. During testing, useless power dissipation is eliminated and $16mW$ are dissipated in the first test session (Figure 6.6(a)) and $16.5mW$ during the second test session (Figure 6.6(b)). It should be noted that for both self-testable data paths of Figures 6.5 and 6.6 the volume of test data consists of 6 seeds for test pattern generators and 3 signatures to be shifted out and compared with the fault-free responses. So far the effect of power conscious test synthesis on BIST area overhead and volume of test data was analysed. On the other hand, power conscious test scheduling has direct impact on test application time and indirect impact on BIST area overhead due to the merged implementation of the functional and BIST controller (Figure 1.10 of Section 1.3.2) which controls the execution of test sessions, and shifting in seeds to LFSRs and shifting out signatures from MISRs. Therefore, when both circuits from Figures 6.5 and 6.6 are synthesised and technology mapped to AMS 0.35 micron technology [9] the following results are obtained for 8 bit data path width. Total area of circuit from Figure 6.5 is 96 sqmil, whereas total area of the circuit from Figure 6.6 is 97 sqmil. This leads to minor increase in BIST area overhead at the benefit of an improvement in performance (highest clock frequency) from 145 MHz for the circuit from Figure 6.5 to 147 MHz for the circuit from Figure 6.6, due to fewer performance degrading test registers such as BILBOs. Therefore, it was shown, for the data path example shown in Figures 6.5 and 6.6, that PC-TSS has a minor impact on BIST area overhead and performance degradation, and no effect on volume of test data.

6.4 Power Conscious Test Synthesis and Scheduling (PC-TSS)

Having described in Section 6.3 the effect of test synthesis and test scheduling on useless power dissipation, now power conscious test synthesis and scheduling (PC-TSS) is considered. The proposed PC-TSS was integrated into the efficient tabu search-based testable design space exploration described in Section 5.3.1 (Figure 5.4), which combines the accuracy of incremental test scheduling algorithms with the exploration speed of test scheduling algorithms based on fixed test resource allocation. Elimination of useless power dissipation introduced in Section 6.2 is carried out in two steps. Section 6.4.1 introduces the first step based on power conscious test synthesis moves during testable design space exploration, while Section 6.4.2 gives the second step by describing module selection during power conscious test scheduling. It should be noted that test synthesis and test scheduling algorithms proposed in Sections 6.4.1 and 6.4.2 equally apply to both BIST embedding and TCC grouping methodologies. A new incremental power conscious generation of TCCs for satisfying a given power constraint is given in Section 6.4.3.

6.4.1 Move Acceptance During Power Conscious Test Synthesis

Chapter 5 has introduced tabu search-based testable design space exploration in Section 5.3.1. In order to minimise power dissipation, move generation criteria (Section 5.3.2 and line 7 from Figure 5.4) must be modified to examine if the newly generated testable design leads to useless power dissipation (see Example 6.2 from Section 6.3). If a move generates a testable design with useless power dissipation then it is rejected. Figure 6.7 shows the new *ACCEPT-MOVE* algorithm. Given the partially testable data path **PT-DP** (Definition 5.4 from Section 5.3.1) and the test registers of the current solution (i.e. left and right test pattern generators, $\mathbf{TPG_L}$ and $\mathbf{TPG_R}$), the proposed algorithm accepts or rejects the new testable designs by analysing the interconnect between test registers and modules. For every module from the output module set of test registers of the current solution, the left and right input register sets (*LIRS* and *RIRS*) are examined. If all the registers from either input register set are test registers then the move is rejected. This is because there will be no value of control signals for multiplexers at the input of data path modules that will eliminate the propagation of spurious transitions introduced in Definition 6.2

```
ALGORITHM: **ACCEPT-MOVE**
INPUT:Partially Testable Data Path **PT-DP**
        Potential test registers $\{\mathbf{TPG_L}, \mathbf{TPG_R}\}$
OUTPUT:boolean variable **ACCEPT**

**ACCEPT** = TRUE;
**for** every $TR_i$ from $\{\mathbf{TPG_L}, \mathbf{TPG_R}\}$ {
    $OMS_i$ = output module set of $TR_i$
    **for** every $M_j$ from $OMS_i$ {
        $LIRS_j$ = left input register set of $M_j$
        $RIRS_j$ = right input register set of $M_j$
        **if** (every $R_k$ from $LIRS_j$ is test register)
        **or** (every $R_l$ from $RIRS_j$ is test register)
            **ACCEPT** = FALSE;
    }
}
**return ACCEPT**;
```

Figure 6.7: Proposed algorithm for power conscious test synthesis moves during testable design space exploration (step 1).

of Section 6.2. By rejecting the testable data paths using the proposed *ACCEPT-MOVE* algorithm useless power will be avoided both during test application and while shifting out test responses. On one hand, the useless power dissipation during test application (TAP introduced in Section 6.2) will be avoided by guaranteeing that regardless of the generated test schedule there will be a value of control signals for multiplexers at the input of data path modules such that spurious transitions are eliminated when activating the synthesised test registers for each test session. On the other hand, useless power dissipation during shifting (SP introduced in Section 6.2) will be avoided by providing at least one value of control signals for multiplexers at the input of data path modules while shifting in seeds for test pattern generators and shifting out signatures. It should be noted that if all the moves lead to useless power dissipation then the move which leads to lowest test application time is accepted as described in Section 5.3.2, and the useless power dissipation is minimised using power conscious test scheduling described in the following section.

6.4.2 Module Selection During Power Conscious Test Scheduling

In order to guarantee that built-in self-testable data path eliminates useless power dissipation during testing, test application time is computed by carrying out the following two modifications to the test scheduling algorithm based on partitioned testing with run to completion [47] described in Section 5.3.3. It should be noted that the following two modifications consider a test associated with each module and are applicable to the BIST embedding methodology which is a particular case of the TCC grouping methodology as outlined in Section 5.2.2. Further extensions required for TCC grouping methodology are introduced in the following Section 6.4.3.

a. A new module selection algorithm *SELECT-MODULE* is proposed such that useless power is eliminated;

b. The power dissipated by scheduling the selected module M_i is computed and if the power constraint is not satisfied during the current test time then test t_i for M_i is removed from the candidate node set [47] being postponed for a later test time;

Figure 6.8 shows the proposed algorithm for module selection during power conscious test scheduling. The module selection aims to eliminate useless power dissipation not only in *useless registers* (UR) at the output of currently tested modules, but it considers also the useless power dissipation in *useless modules* (UM) to which spurious transitions are propagated through *useless registers* (see Example 6.3 of Section 6.3). Given the testable data path, the modules scheduled at the current test time (tested modules) and the candidate modules to be scheduled according to the resource conflict graph [47, 140], the algorithm *SELECT-MODULE* selects the candidate module which when scheduled at the current test time will lead to the minimum increase in power dissipation. Initially the active module set (*AMS*) contains the tested modules, while the active register set (*ARS*) contains the test registers which generate test patterns and analyse test responses for currently tested modules. For each candidate module the power dissipation is computed by recursively propagating spurious transitions through UR and UM (lines 4 to 11 in Figure 6.8). Initially both sets of useless registers and useless modules are null. UM is computed using *ARS* and UR. A module is assigned to UM if all the registers in its left *or* right input register set are active at the current test time. The useless modules are considered for detecting the propagation of spurious transitions to useless registers. Once

```
ALGORITHM: SELECT-MODULE
INPUT:Testable Data Path DP
        Tested Modules $\{\mathbf{TM_0},\ldots,\mathbf{TM_n}\}$
        Candidate Modules $\{\mathbf{CM_0},\ldots,\mathbf{CM_m}\}$
OUTPUT:module to be scheduled $\mathbf{CM_s}$

1   Active Module Set $AMS$ = $\{\mathbf{TM_0},\ldots,\mathbf{TM_n}\}$;
2   Active Register Set $ARS$=TestRegisters($AMS$);
3   for every $CM_i$ from $\{\mathbf{CM_0},\ldots,\mathbf{CM_m}\}$ {
4       $UR \leftarrow \varnothing$; $UM \leftarrow \varnothing$;
5       repeat
6         $ARS \leftarrow ARS \cup UR$;
7         $UM \leftarrow$ GetUselessMod($ARS$);
8         $AMS \leftarrow AMS \cup UM$;
9         $UR \leftarrow$ GetUselessReg($AMS \cup UM$);
10      until $UR \neq \varnothing$;
11      compute $P_i$ using $AMS \cup ARS$;
12  }
13  select the $\mathbf{CM_s}$ with least power dissipation $P_s$;
14  return $\mathbf{CM_s}$;
```

Figure 6.8: Proposed algorithm for module selection during power conscious test scheduling (step 2).

AMS is updated with *UM*, useless registers *UR* are computed using the updated *AMS*. A register is assigned to *UR* if all the modules in its input module set are active at the current test time. All the useless registers detected in the current iteration are used to update the set of active registers *ARS* in the next iteration. Once *ARS* is updated, new useless modules are detected and this recursive propagation of spurious transitions continues until no new useless registers are detected. At the end of the recursive propagation of spurious transitions (lines 5 to 10 in Figure 6.8) *AMS* and *ARS* contain *not only* the tested modules and their test registers, but also *all* the active data path elements during the current test time. *AMS* and *ARS* are used to compute both necessary and useless power dissipation associated with selecting candidate module CM_i to be scheduled at the current test time. Finally, the candidate module CM_s which leads to minimum power dissipation is selected to be scheduled at the current test time.

6.4.3 Power Conscious Generation of Test Compatibility Classes

Power conscious test synthesis (Section 6.4.1) and power conscious test scheduling (Section 6.4.2) are equally applicable to both BIST embedding and TCC grouping methodologies. In order to further increase test concurrency of test compatibility classes while satisfying a given power constraint two modifications for an incremental TCC scheduling algorithm (Section 5.3.3 from Chapter 5) are necessary:

- While generating the global test incompatibility graph (G-TIG) for maximum test concurrency (Figure 5.6 in Section 5.3.3 from Chapter 5) the unassigned modules which are compatible with all the existing TCCs need not to be assigned such that a maximum increase in output register set is achieved (third step - lines 29-32 from Figure 5.6). Each unassigned module creates a new TCC consisting of a single module. This will lead to higher number of nodes in the G-TIG. However the total necessary power dissipated by each TCC is lower. Thus instead of merging all the compatible modules prior to TCC scheduling, there are more nodes created in the G-TIG and they will be merged later if the power constraint is satisfied. Therefore, if two TCCs of the same type present in G-TIG are compatible then during the incremental test scheduling they are merged as explained in the following modification.

- During the simultaneous TCC scheduling and signature analysis allocation algorithm (Figure 5.8 in Section 5.3.3 from Chapter 5) if two test compatibility classes are compatible then they are merged if they meet the given power constraint. This will lead to an incremental generation of maximal test concurrency within the G-TIG ***only if*** power constraints are satisfied. The assignment of conflict free modules to TCCs is driven by power constraint and not by the increase in the size of the output register. This causes both high test concurrency and satisfaction of power constraints at the expense of the size of the output register set and hence the potential reuse of signature analysis registers.

The previous two modifications are incorporated in algorithms shown in Figures 5.6 and 5.8. As shown in the Section 6.5 test compatibility classes preserve both lower test application time and BIST area overhead when compared to BIST embedding for a given power constraint.

It is interesting to note that both algorithms *ACCEPT-MOVE* and *SELECT-MODULE* introduced in this section guarantee that as far as there is a potential solution leading to lower test application time and BIST area overhead, and which eliminates useless power dissipation then it will be selected. This is unlike traditional testable design space exploration algorithms introduced in [115, 140] and previous Chapter 5 where the single aim is to reduce test application time and BIST area overhead without any consideration of power dissipation. Furthermore, the previous power constrained test scheduling approaches [25, 41, 110, 132, 165, 199] have assumed fixed test resource allocation and fixed amount of power dissipation associated with each test. As outlined in the motivational Section 6.1 this is untrue for RTL data paths. The fixed test resource allocation problem is overcome by incorporating both algorithms *ACCEPT-MOVE* (Figure 6.7) and *SELECT-MODULE* (Figure 6.8) in the testable design space exploration from Figure 5.3.1 which combines the accuracy of incremental test scheduling algorithms [115] with the exploration speed of test scheduling algorithms based on fixed test resource allocation [33, 47, 92]. The problem of fixed amount of power dissipation associated with each test is overcome by accounting for useless power dissipation during the following three stages: (a) move acceptance - by incorporation of *ACCEPT-MOVE* in the new moves generation described in Section 5.3.2; (b) test scheduling - by incorporation of *SELECT-MODULE* in the test scheduling algorithm based on partitioned testing with run to completion [47] described in Section 5.3.3; (c) TCC generation - by incorporation of the modifications described in this section in the algorithms shown in Figures 5.6 and 5.8.

Finally, it should be noted that for designs where gated clocks are employed at register-transfer level, useless power dissipation in registers can also be eliminated by gating the clock of the inactive registers. However, elimination of useless power dissipation in modules by controlling multiplexer inputs is necessary even in the case when modules are highly sequential and use power enable/disable signals to turn off the modules which are not targeted. This is due to the fact that useless power dissipation is eliminated in the combinational logic up to the first sequential boundary in the module only where power enable/disable signals take effect. Therefore, although the techniques proposed in this chapter are geared towards design styles which do not employ gated clocks and/or power enable/disable signals they can successfully be combined with clock gating power reduction methodologies leading to further savings in power dissipation.

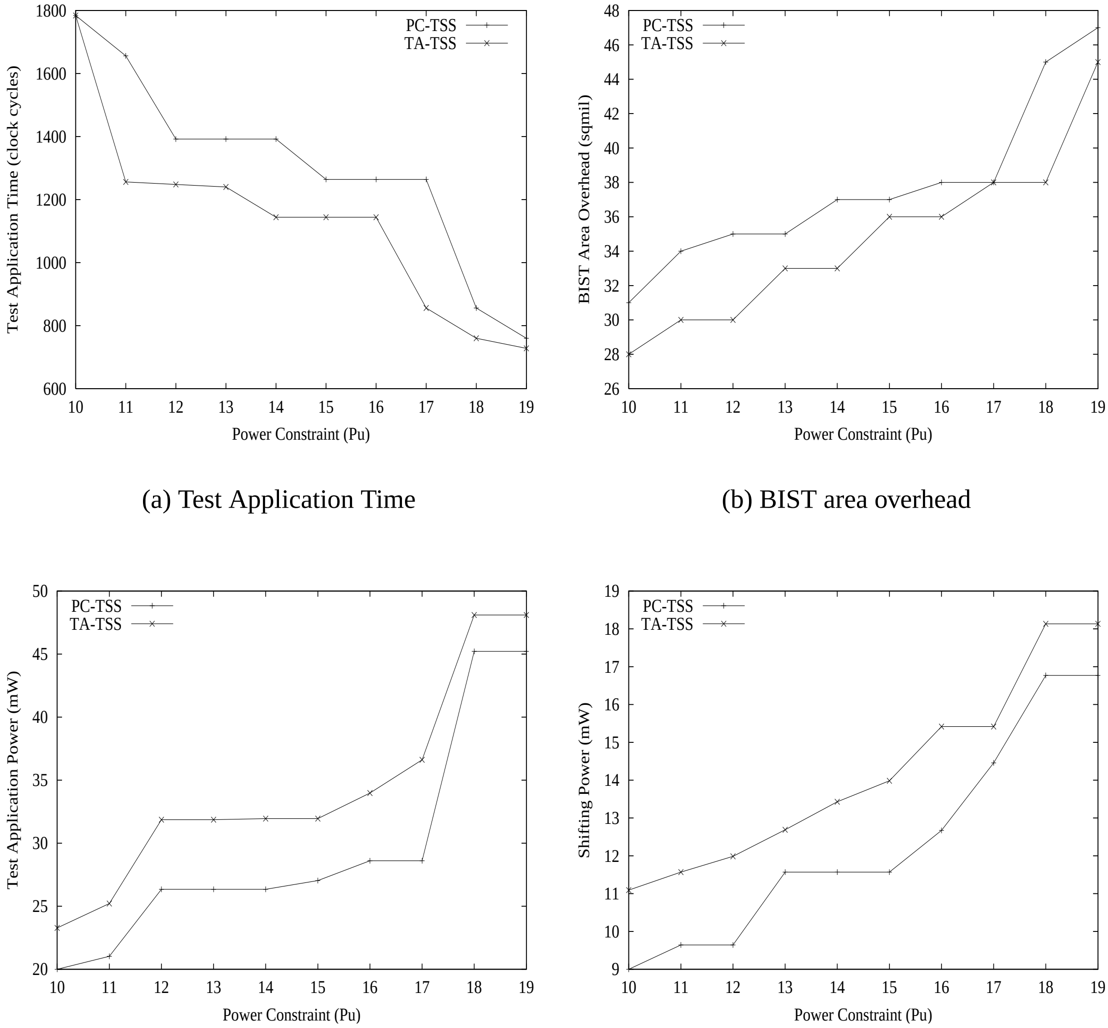
6.5 Experimental Results

Power conscious test synthesis and test scheduling were successfully implemented on a SUN SPARC 20 workstation and integrated within the testable design space exploration environment described in Section 5.3.1 from Chapter 5. Section 6.5.1 gives the experimental results for the BIST embedding methodology when applying power conscious test synthesis and test scheduling algorithms from Sections 6.4.1 and 6.4.2. Section 6.5.2 provides the experimental results for the TCC grouping methodology when power conscious generation of test compatibility classes, described in Section 6.4.3, is employed.

6.5.1 Experimental Results for BIST Embedding Methodology

To give insight into the importance of considering the effect of test synthesis and test scheduling on useless power dissipation, Figures 6.9 to 6.11 show a comparison of test application time (TAT), BIST area overhead (BAO), test application power (TAP) and shifting power (SP) when using the proposed Power Conscious Test Synthesis and Scheduling (PC-TSS) and Time and Area Test Synthesis and Scheduling (TA-TSS) for BIST embedding methodology. The main objective of TA-TSS is to minimise test application time under the given power constraint with BIST area overhead used as tie-breaking mechanism among many possible solutions with same test application without any consideration of useless power dissipation during test application. Therefore TA-TSS assumes fixed amount of power dissipation associated with each test which is untrue for RTL data paths as outlined in Section 6.1 (problem (a)). The results when applying TA-TSS to BIST embedding methodology are obtained using the same BIST hardware synthesis algorithm presented in Section 5.3 assuming that every pair of modules in the data path are different $(n_{res} = n_{mod})$ as described in Section 5.2.2, since BIST embedding is a particular case of TCC grouping. Unlike TA-TSS, the main objective of PC-TSS is to eliminate useless power dissipation, and then it uses test application time and BIST area overhead as tie-breaking mechanism as outlined in Example 6.4 from Section 6.3. Therefore, PC-TSS leads to more test registers than TA-TSS at the benefit of eliminating useless power dissipation. The comparison is carried out for a number of benchmark examples including elliptic wave digital filter with execution time constraint of 17 control steps (Figure 6.9), 8 point discrete cosine transform with execution time constraint of 10 control steps

(Figure 6.10), and 32 point discrete cosine transform with execution time constraint of 30 control steps (Figure 6.11). The benchmarks were synthesised using the ARGEN high level synthesis system [99, 100]. When integrating power conscious test synthesis and test scheduling algorithms (Section 6.4) into testable design space exploration (Section 5.3.1 from Chapter 5), generic values for test application time, and power dissipation need to be considered. Therefore, test application time (TAT) for adders and multipliers are assumed to be $T_{+} = T_u$, and respectively $T_{*} = 4 \times T_u$, where $T_u = 128$ for achieving 100% fault coverage for 8 bit data path modules. As outlined in Section 5.4 to validate the assumption regarding test length, 8 bit width adder and multiplier modules were synthesised and technology mapped into AMS 0.35 micron technology [9], and subsequently a parallel pattern single fault propagation fault simulator [112] has shown that $T_u = 128$ is a valid assumption. Similarly, during the power conscious testable design space exploration power dissipation for registers, adders and multipliers is assumed to be $P_{REG} = P_u$, $P_{+} = P_u$ and $P_{*} = 4 \times P_u$, where P_u can be derived using the techniques from [116]. The generic high level model for power dissipation provides the flexibility of applying the proposed algorithms to various library modules with different power characterisation. To validate the generic power model, registers, test registers, and functional modules were synthesised and technology mapped to AMS 0.35 micron technology [9]. Using a real delay model simulator [130] and AMS 0.35 micron timing and power information operating at supply voltage 3.3V and clock frequency 100MHz, and hence accounting for glitching activity, the following power values were obtained for 8 bit data path width using pseudorandom sequences applied during testing: $P_{REG} = 0.8mW$, $P_{LFSR} = 1mW$, $P_{MISR} = 2mW$, $P_{BILBO} = 2.5mW$, $P_{+} = 3.5mW$, and $P_{*} = 11.5mW$. To compute the power dissipation during test application in the entire data path (Figures 6.9, 6.10, and 6.11), the power dissipation of all the active elements is summed. This hierarchical power dissipation computation provides a trade-off between the accuracy of low level power simulators such as SPICE and the computational complexity for large circuits such as elliptic wave digital filters and discrete cosine transform. Finally in order to compute BIST area overhead, BIST RTL data paths are specified in VHDL code at RTL and synthesised and technology mapped using [55] into AMS 0.35 micron technology [9]. It should be noted that BIST area overhead includes not only the overhead caused by data path test registers, but also the overhead caused by the merged functional and BIST controller as described in Figure 1.10 from Chapter 1.



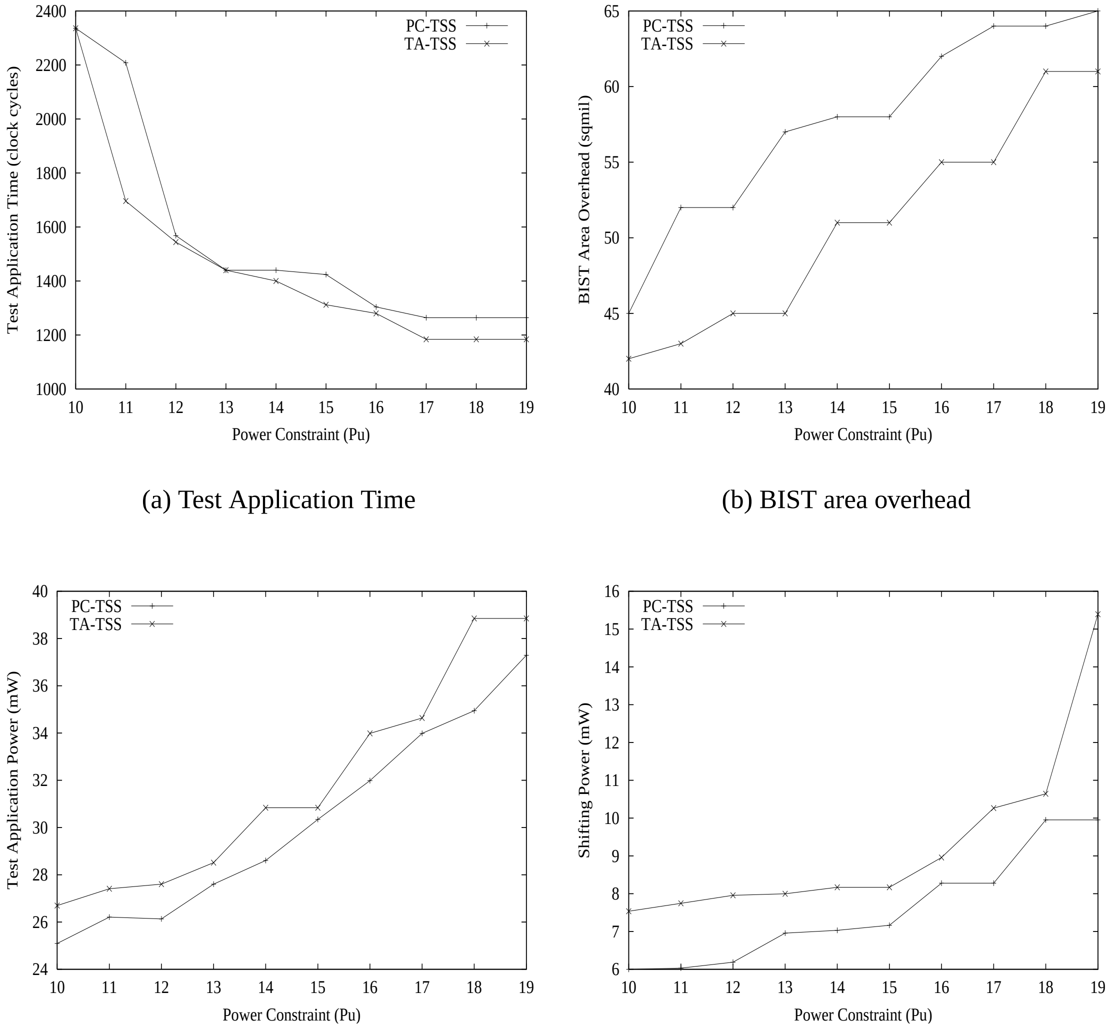
(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

(d) Shifting Power Dissipation

Figure 6.9: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TA-TSS and PC-TSS for BIST embedding methodology applied to elliptic waveform digital filter for 10 power constraints



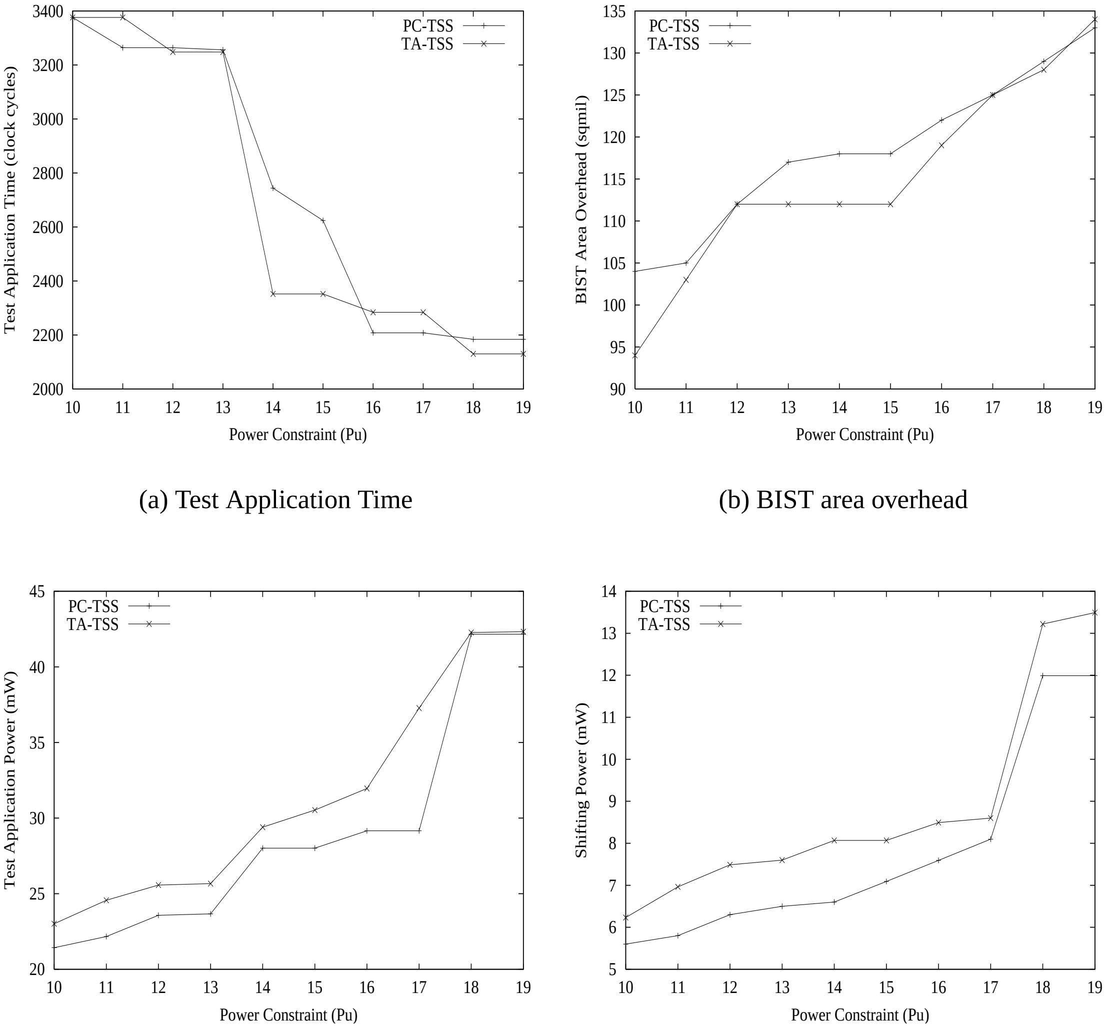
(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

(d) Shifting Power Dissipation

Figure 6.10: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TA-TSS and PC-TSS for BIST embedding methodology applied to 8 point discrete cosine transform for 10 power constraints



(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

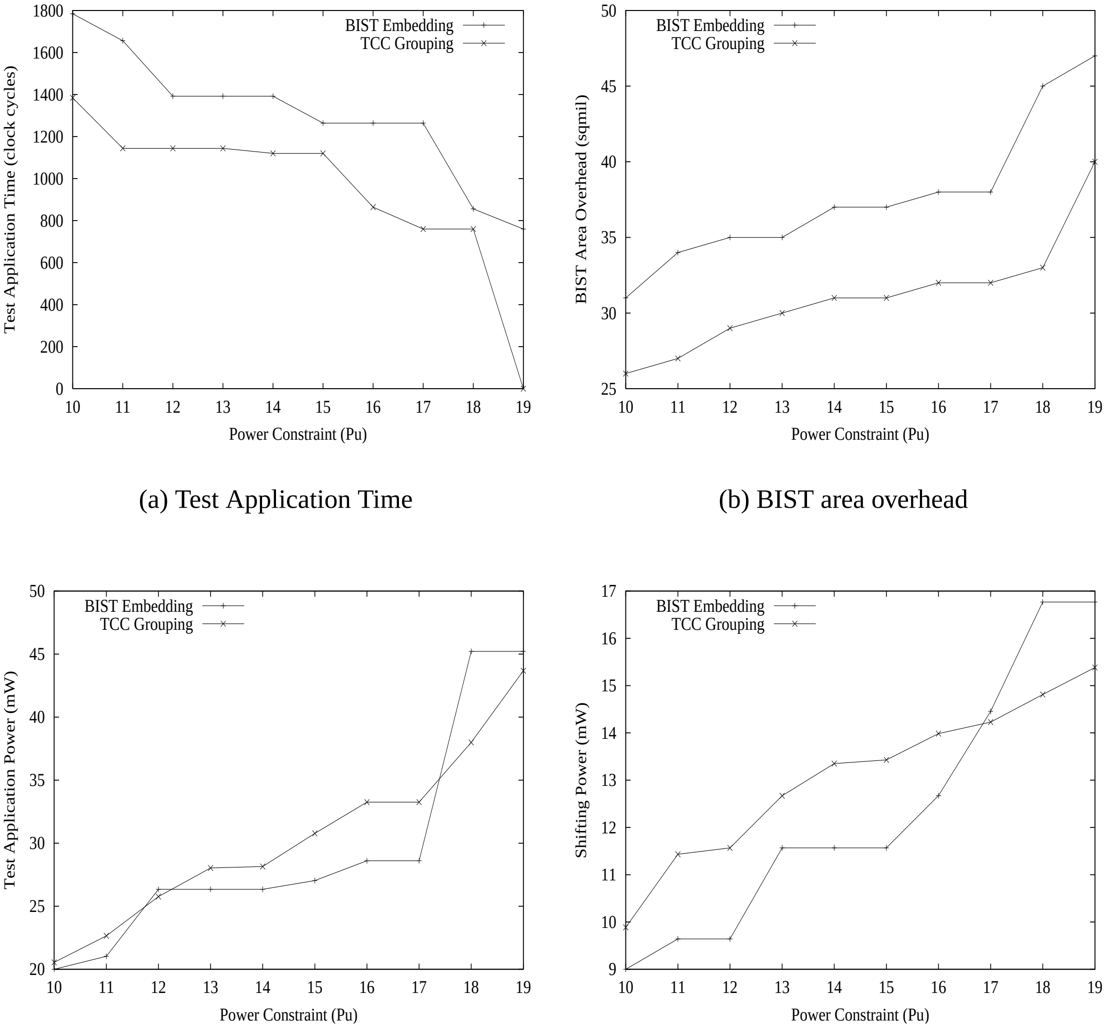
(d) Shifting Power Dissipation

Figure 6.11: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TA-TSS and PC-TSS for BIST embedding methodology applied to 32 point discrete cosine transform for 10 power constraints

To asses the effectiveness of the proposed techniques, PC-TSS and TA-TSS were compared for 10 power constraints ranging from 10×P_u to 19×P_u. For all the evaluated circuits the proposed PC-TSS in Section 6.4 produces less TAP and less SP when compared to TA-TSS which does not account for useless power dissipation. For example, as shown in Figure 6.9(c) in the case of elliptic wave digital filter, TAP varies from 20mW to 45mW in the case of PC-TSS, whereas in the case of TA-TSS it varies from 23mW to 48mW. While TAP, SP and BAO increase simultaneously with power constraint, TAT in terms of clock cycles decreases as the power constraint increases, as shown in Figures 6.9(a), 6.10(a), and 6.11(a). Test application time values consider that untested multiplexers and registers use a small global test set of functional patterns that can be generated using the justification/propagation techniques [65, 118] as outlined in Section 5.2.1 of the previous Chapter 5. It should be noted that TA-TSS which assumes fixed amount of power dissipation with each test (problem (a) of Section 6.1) yields lower test application time and BIST area overhead when compared to PC-TSS for most of the power constraints. However, lower test application time and BIST area overhead in the case of TA-TSS leads *constantly* to higher power dissipation during test application due to ignorance of useless power dissipation during test synthesis and test scheduling, and hence causes a violation of the power constraint which can decrease the reliability of the circuit and lead to manufacturing yield loss [191, 199]. Further, as the complexity of the circuit increases as it is the case of 32 point discrete cosine transform, PC-TSS has comparable test application time (Figure 6.11(a)) and BIST area overhead (Figure 6.11(b)) to TA-TSS at the benefit of lower power dissipation. The integration of algorithms *ACCEPT-MOVE* (Section 6.4.1) and *MODULE-SELECT* (Section 6.4.2) into testable design space exploration algorithm (Figure 5.4 from Chapter 5) comes at the expense of low overhead in computational time. For example, the computational time for elliptic wave digital filter and 8 point discrete cosine transform is below 10s, where for complex 32 point discrete transform is under 500s which is within reasonable computational limits.

6.5.2 Experimental Results for TCC Grouping Methodology

To outline the advantages of combining the proposed TCC grouping methodology (Section 5.2 from Chapter 5) with the newly introduced power conscious test synthesis and scheduling (Section 6.4.3), Figures 6.12, 6.13, and 6.14 give a comparison in terms of



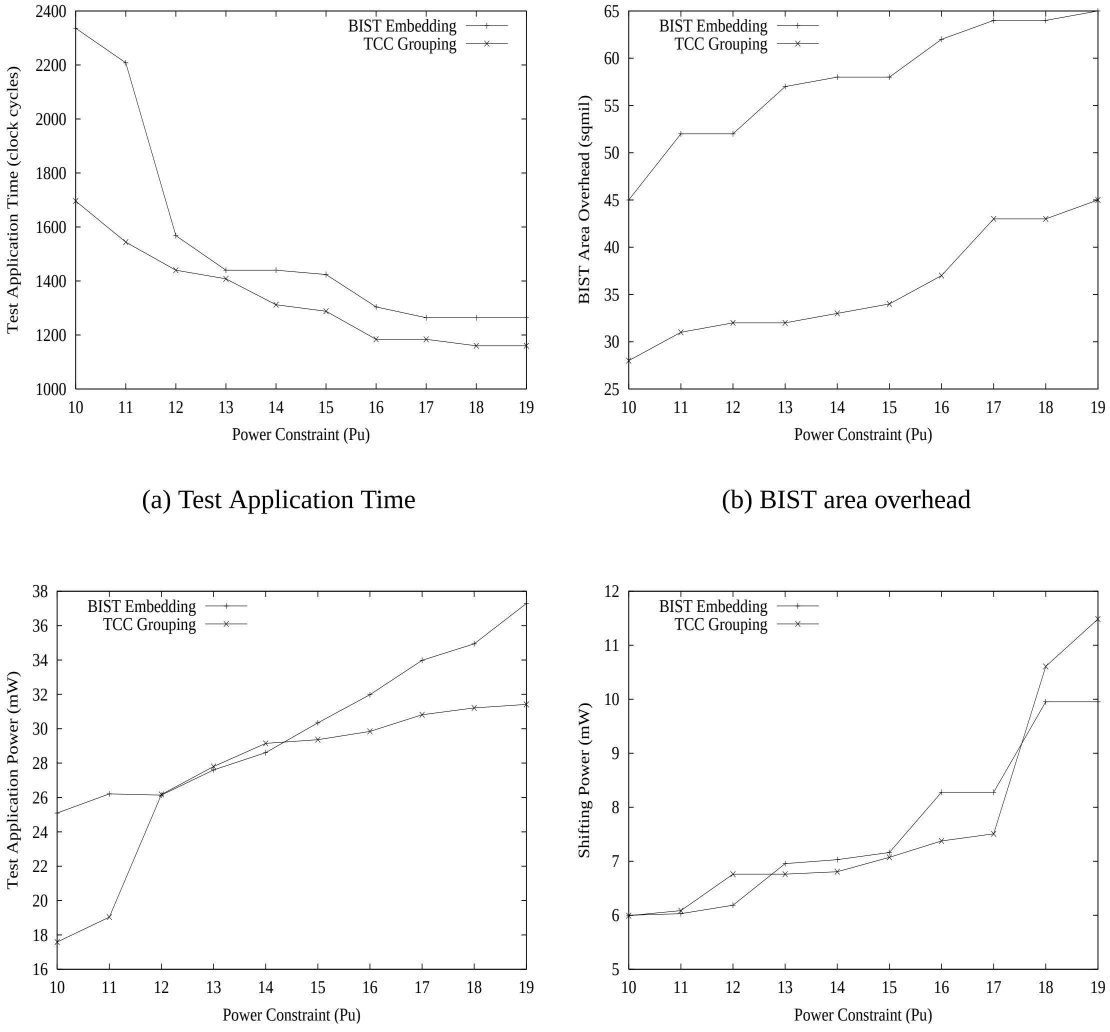
(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

(d) Shifting Power Dissipation

Figure 6.12: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TCC grouping and BIST embedding methodologies applied to elliptic waveform digital filter for 10 power constraints



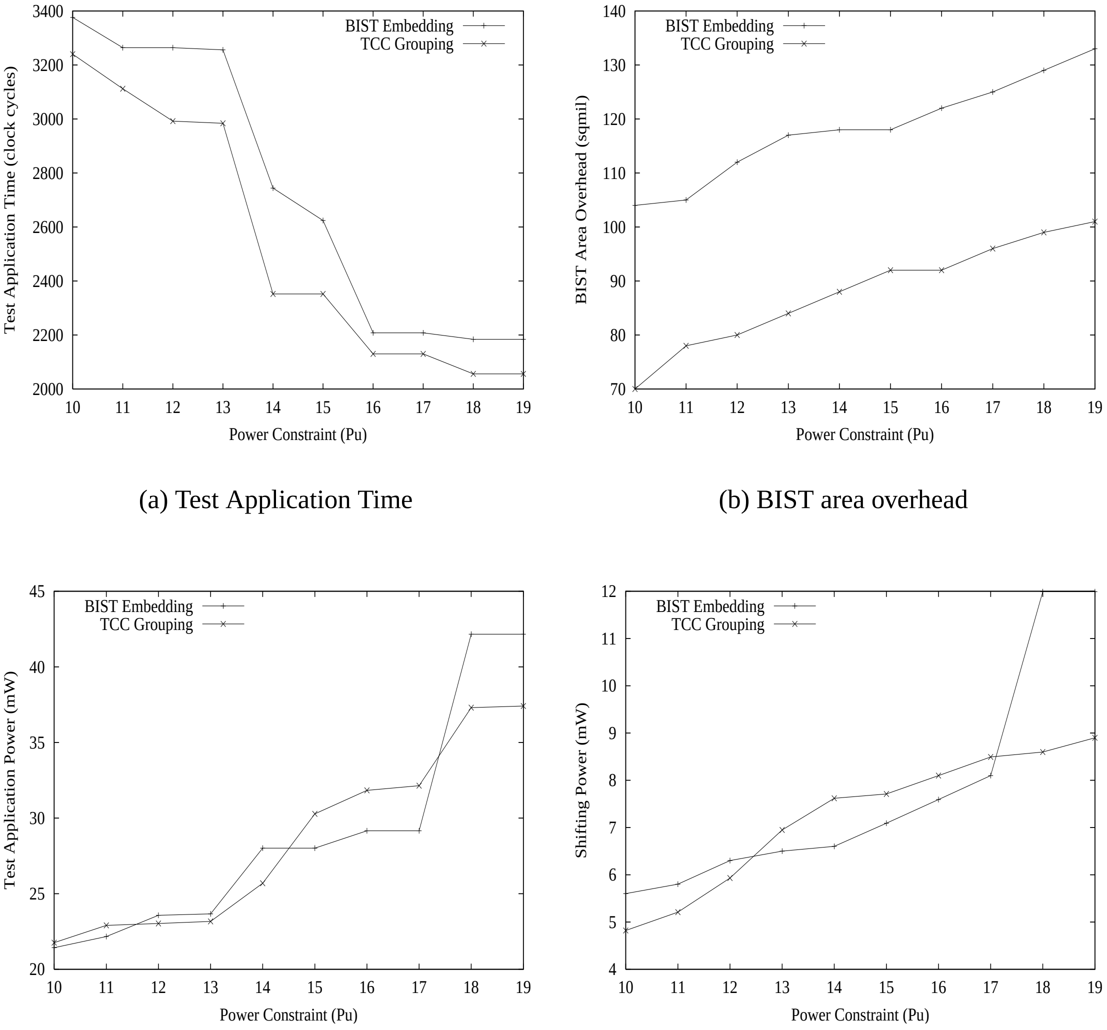
(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

(d) Shifting Power Dissipation

Figure 6.13: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TCC grouping and BIST embedding methodologies applied to 8 point discrete cosine transform for 10 power constraints



(a) Test Application Time

(b) BIST area overhead

(c) Test Application Power Dissipation

(d) Shifting Power Dissipation

Figure 6.14: Comparison of test application time, BIST area overhead, test application power, and shifting power when using TCC grouping and BIST embedding methodologies applied to 32 point discrete cosine transform for 10 power constraints

TAT, BAO, TAP, and SP for traditional BIST embedding and the proposed TCC grouping methodologies. So far it was shown in Section 5.4 that test compatibility classes overcome the problems of high test application time, BIST area overhead, performance degradation, volume of test data, fault-escape probability and large computational time required for testable design space exploration associated with traditional BIST methodologies. It is shown in the following that test compatibility classes lead to an increase in test concurrency and hence decrease in test application time with no penalty in power dissipation. To show the suitability of TCC grouping for various power constraints, experiments were done on elliptic wave digital filter (Figure 6.12), and 8 and 32 point discrete cosine transform (Figures 6.13 and 6.14) using the same validation methodology as described previously for TA-TSS and PC-TSS in Section 6.5.1. The values for BIST embedding methodology are obtained applying algorithms *ACCEPT-MOVE* and *SELECT-MODULE* described in Sections 6.4.1 and 6.4.2 respectively, whereas for TCC grouping also the extensions described in Section 6.4.3 are employed. As it is clearly seen in Figures 6.12(a), 6.13(a), and 6.14(a) the proposed TCC grouping methodology *constantly* yields lower test application time when compared to the traditional BIST embedding methodology. Similarly, Figures 6.12(b), 6.13(b), and 6.14(b) show that TCC grouping methodology produces lower BIST area overhead with *similar* values for test application power (Figures 6.12(c), 6.13(c), and 6.14(c)) and shifting power (Figures 6.12(d), 6.13(d), and 6.14(d)). For example, in the case of elliptic wave digital filter (Figure 6.12(c)) test application power for TCC grouping varies from $20mW$ to $44mW$ while for BIST embedding it varies from $21mW$ to $45mW$. This implies that when applying PC-TSS with extensions to test compatibility classes there is no penalty in power dissipation at the benefit of lower test application time and BIST area overhead when compared to traditional BIST embedding methodology [114, 115]. Therefore it may be concluded that when power conscious algorithms proposed in Section 6.4 are combined with TCC grouping methodology introduced in Chapter 5 not only fixed test resource allocation and fixed amount of power dissipation are overcome, problems (a) and (b) of previous approaches [25, 41, 110, 132, 165, 199] outlined in Section 6.1), but also a new methodology for simultaneously reducing test application time and power dissipation is provided. Reducing test application time and power dissipation simultaneously using TCCs is particularly important for high performance low power RTL data paths generated using recently reported low power high level synthesis algorithms [30, 96, 103, 104, 105, 106, 107].

6.6 Concluding Remarks

This chapter has shown how power dissipation during test application is minimised at higher levels of abstraction than the logic level of abstraction (Chapters 3 and 4) of the VLSI design flow. When using BIST for low power RTL data paths [30, 96, 103, 104, 105, 106, 107], useless power dissipation during test application is eliminated by novel power conscious test synthesis and test scheduling. In order to achieve this goal power dissipation was classified into necessary and useless power during test application in Section 6.2. Then the effect of test synthesis and test scheduling on power dissipation was analysed in Section 6.3. Using the analysis from Section 6.3 power minimisation was achieved by power conscious test synthesis moves during the testable design space exploration and by power conscious module selection during test scheduling, which equally apply to BIST embedding [114, 115] and TCC grouping (Chapter 5) methodologies. Unlike previous power constrained test scheduling approaches [25, 41, 110, 132, 165, 199] which have assumed fixed test resource allocation and fixed amount of power dissipation associated with each test (Section 6.1) the proposed power conscious test synthesis and scheduling (PC-TSS) (Section 6.4) employs novel testable design space exploration and accounts for useless power dissipation. To further increase test concurrency while satisfying power constraints in the case of TCC grouping two modifications for incremental generation of TCCs were presented and their integration with algorithms from previous Chapter 5 was outlined (algorithms from Figures 5.6 and 5.8). The algorithms proposed in this chapter are of prime importance for achieving higher yield and reliability by satisfying power constraints during test application in BIST RTL data paths at the expense of low overhead in computational time. This is due to the fact that previously reported algorithms that do not consider power dissipation during test application (Chapter 5, [114, 115]) or consider a fixed amount of power dissipation associated with each test [25, 41, 110, 132, 165, 199] consistently lead to higher power dissipation when compared to the proposed PC-TSS as shown in the experimental results section. Finally, it was shown in Section 6.5 that when power conscious test synthesis and test scheduling proposed in Section 6.4 are combined with TCC grouping methodology introduced in Chapter 5, simultaneous reduction in test application time and power dissipation is achieved.

Chapter 7

Conclusions and Future Work

The demand for low power VLSI circuits in the growth area of portable communications and computing systems will continue to increase in the future. Cost and lifetime cycle of near future portable communications and computing systems will depend not only on VLSI circuits designed using low power synthesis techniques, but also on new DFT methods targeting power minimisation during test application. This is because traditional DFT methods are not suitable for testing low power VLSI circuits leading to lower reliability and manufacturing yield. This dissertation has focused on minimising power dissipation during test application at two different levels of abstraction of the VLSI design flow. Two novel techniques for power minimisation during test application in scan sequential circuits, were proposed at logic level of abstraction. A new BIST methodology for RTL data paths, based on a novel concept called test compatibility classes, which overcomes the problems associated with traditional BIST methodologies, was introduced. Finally, power dissipation during test application in BIST RTL data paths was investigated and new techniques for useless power elimination for both traditional BIST methodologies and newly introduced test compatibility classes, were presented. This research findings further motivate the need for new DFT methods for testing low power VLSI circuits, and provide important and necessary new theoretical and experimental results on power minimisation during test application. Consequently, this research findings contribute towards reducing the cost and increasing lifetime of portable systems due to improved reliability and manufacturing yield of low power VLSI circuits. In the following, Section 7.1 summarises the main contributions of the dissertation, and Section 7.2 outlines directions for future exploration.

7.1 Summary of Thesis Contributions

Motivation for low power testing and a comprehensive review of previously reported approaches for minimising power dissipation during test application was provided in Chapter 2. It was shown that switching activity during test application is significantly higher than during functional operation of the circuit which causes higher power dissipation and hence may lead to lower circuit yield and reliability. The previous approaches which reduce power dissipation during test application were classified in two broad classes based on the level of abstraction in the VLSI design flow: low level techniques such as *logic level of abstraction*, and high level techniques such as *register-transfer level of abstraction*.

Chapter 3 has proposed a new technique for minimising power dissipation in scan sequential circuits during test application at *logic level of abstraction*. The new technique is based on increasing the correlation between successive states during shifting in test vectors and shifting out test responses by changing the primary inputs such that the smallest number of transitions is achieved. A novel algorithm which is test set dependent, and computes best primary input change (BPIC) time for each test vector, was presented. It was shown that combining the described technique with the recently reported scan cell and test vector ordering using a simulated annealing-based design space exploration yields substantial reductions in power dissipation during test application in small to medium sized full scan sequential circuits. Exhaustive experimental results, using both compact and non-compact test sets, have shown that compact test sets have similar power dissipation during test application with substantial reduction in test application time and computational time when compared to non-compact test sets. Moreover, new BPIC test application strategy is equally applicable for minimising power dissipation in partial scan sequential circuits. Since the proposed test application strategy depends only on controlling primary input change time, power is minimised with no penalty in test area, performance, test efficiency, test application time or volume of test data. Furthermore, it was shown that partial scan does not provide only the commonly known benefits such as less test area overhead and test application time, but also less power dissipation during test application and computational time required for design space exploration, when compared to full scan.

Chapter 4 has presented a new test set independent technique based on multiple scan chains and it has shown how with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, considerable savings in power dissipation during test application in large scan sequential circuits can be achieved in very low computational time. The technique is based on a new DFT architecture and a novel test application strategy which is applied at *logic level of abstraction*. To facilitate the reduction of spurious transitions, the proposed DFT architecture is based on classifying scan cells into compatible, incompatible and independent scan cells, without any knowledge of the test set which will be applied to achieve the required fault coverage which makes the approach test set independent. Based on their classification scan cells are partitioned into multiple scan chains and a single extra test vector associated with each scan chain is computed. A new test application strategy which applies the extra test vector to primary inputs while shifting out test responses for each scan chain, was presented. Unlike previous approaches which are test set dependent and hence are not able to handle large scan sequential circuits due to the complexity of the design space, it has shown that with low overhead in test area and volume of test data, and with no penalty in test application time, test efficiency, or performance, in low computational time for both small and large tests. Finally, it was outlined that the synthesisable extra hardware required by the new DFT architecture and the novel test application strategy make the proposed multiple scan chain based technique easily embeddable in the existing VLSI design flow using state of the art third party electronic design automation tools.

Chapter 5 has addressed the testability of RTL data paths using BIST. It was shown that an improvement in terms of test application time, BIST area overhead, performance degradation, volume of test data, and fault-escape probability is achieved when using the newly introduced test compatibility classes-based methodology. The new BIST methodology is based on grouping modules with identical physical information into test compatibility classes (TCC) and testing the compatible modules by sharing a small number of test pattern generators at the same test time. A novel BIST hardware synthesis which uses efficient tabu search-based testable design space exploration and combines the accuracy of incremental test scheduling algorithms with the exploration speed of test scheduling algorithms based on fixed test resource allocation was presented. When compared to the traditional BIST embedding methodology, it was shown that the proposed TCC grouping methodology is capable of reducing the test application time with lower BIST area overhead and yielding savings in performance degradation, volume of test data, and fault-escape probability. Furthermore, the proposed BIST hardware synthesis algorithm achieved high quality of the final solution in low computational time.

Chapter 6 has shown how power dissipation during test application is minimised at *register-transfer level* of abstraction of the VLSI design flow. When using BIST for low power RTL data paths, power dissipation during test application is minimised by new power conscious test synthesis and test scheduling. To achieve this goal power dissipation was classified into necessary and useless power and then the effect of test synthesis and test scheduling on power dissipation was analysed. It was shown how power dissipation is minimised when using power conscious test synthesis moves during the testable design space exploration and by power conscious module selection during test scheduling which equally apply to traditional BIST methodologies and the newly introduced TCC grouping methodology (Chapter 5). Novel power conscious test synthesis and test scheduling algorithms are of prime importance to achieving higher yield and reliability by satisfying power constraints during test application in BIST RTL data paths at the expense of low overhead in computational time. Finally, it was shown that when the proposed power conscious test synthesis and test scheduling is combined with new TCC grouping methodology simultaneous reduction in test application time and power dissipation is achieved.

7.2 Future Research Directions

During the course of this research, a number of challenging research topics related to low power testing were identified. In the following a short review of three research areas that require further investigation is given.

Low Power BIST Methodology for Integrated Testing of Controller/Data Paths

It was shown that an improvement in terms of test application time, BIST area overhead, performance degradation, volume of test data, and fault-escape probability is achieved using the newly introduced test compatibility classes in Chapter 5. An interesting extension of this BIST methodology is the investigation of the effect of the faults in the test controller which will provide a structured methodology for integrated controller/data path testing. Since the implementation of the test controller and the flow of test data will define the test application strategy for the BIST RTL data path it is anticipated that this investigation will allow the combination of techniques proposed in Chapters 3 and 4 to be extended to RTL data paths. The key to this work is to exploit the redundant information in the test control definition including primary input values and coding of test registers and multiplexers control signals. An advantage of exploiting the redundant information, as in the case of logic level techniques described in Chapters 3 and 4, is that power optimised test application strategies for BIST RTL datapaths are generated together with functional and test controllers optimised for area and performance. It was shown in Chapter 6 that useless power dissipation is eliminated when employing power conscious test synthesis and test scheduling. It is anticipated that measured power dissipation data, obtained from manufactured built-in self-testable circuits, should facilitate the further acceptance and maturity of this new area of research. Furthermore, the feedback and incorporation of measured power dissipation data into flexible power conscious test synthesis and test scheduling algorithms will lead to further optimised self testable data paths in terms of area, performance, power and testability.

High Level Synthesis for Low Power BIST

High level synthesis is the process of automatically mapping a behavioural description at the algorithmic level of abstraction of the VLSI design flow to a structural implementation at the register-transfer level in terms of functional units, registers and interconnect [59, 101, 146, 147]. Clearly, while some progress was made in high level synthesis of digital circuits, new issues like integrating testability [24, 193] in the tasks of high level synthesis were not fully explored. The mandatory tasks during high level synthesis are allocation, scheduling and binding, all of which were shown to have significant impact on the testability of the synthesised design [54, 62]. Recent research work involving introduction of BIST constraints to guide high level synthesis [21, 149, 151, 156, 157] proved to be efficient to reduce test application time and/or BIST area overhead of the resulted RTL data path. However, all the previous high level synthesis algorithms have considered low power [30, 96, 103, 104, 105, 106, 107] and testability [21, 24, 149, 151, 156, 157, 193] as two mutually exclusive optimisation objectives. It is anticipated that by integrating low power testability constraints into the tasks of high level synthesis will lead to RTL data paths that will be suitable to exploit full capabilities of the new TCC grouping methodology described in Chapter 5 and novel power conscious test synthesis and test scheduling algorithms introduced in Chapter 6. To facilitate the evaluation of design trade-offs at a higher level of abstraction the migration of the appropriated testability and low power related decisions in the behavioural domain needs further investigation. Furthermore, in order to achieve low computational time, fast and accurate high level testability metrics will be required to guide the synthesis process towards high quality solutions in terms of performance, cost, and testability.

System on a Chip Test

Recent advances in manufacturing technology have provided the opportunity to integrate millions of transistors on a single chip leading to the development of new design methodologies such as embedded-core based systems on a chip [81]. Even though the design process of embedded core-based systems on a chip is conceptually analogous to the traditional board design, their manufacturing processes are fundamentally different [125, 194, 200, 201]. While design reuse of embedded cores contributes to the efficiency of systems on a chip design, true inter-operability can be achieved only if the tests for these cores can also be reused. Therefore, novel problems need to be addressed and new challenges arise to research community in order to provide unified solutions that will simplify the design flow, increase the inter-operability of different vendor cores, cut down the manufacturing cost, speed-up time to market and provide a plug-and-play methodology for core-based design paradigm. It is anticipated that test synthesis and test scheduling algorithms proposed in Chapters 5 and 6 can successfully be applied, subject to appropriate design style modifications for the new embedded core-based systems on a chip design paradigm. Further, to improve test access mechanism for embedded cores the relation between test hardware requirements, performance degradation and power dissipation during test application needs further investigation.

Appendix A

Experimental Validation Flow

The proposed techniques in Chapters 3, 4, 5, and 6 were integrated into the VLSI design flow using third party electronic design automation (EDA) tools. This appendix gives an overview of the experimental validation flow. Firstly, Section A.1 outlines the integration of the techniques proposed in Chapters 5, and 6 at register-transfer level of abstraction. Secondly, Section A.2 describes the integration of the techniques proposed in Chapters 3 and 4 at logic level of abstraction.

A.1 Integration of the Proposed Register-Transfer Level Techniques

The experimental validation flow for technology mapping RTL data paths into a target technology when employing the proposed test compatibility classes (Chapter 5) and power conscious test synthesis and test scheduling (Chapter 6) is shown in Figure A.1. The parts of the experimental validation flow that were developed for the purpose of integration to third party tools are shown in the gray boxes. The behavioural descriptions of the elliptic wave digital filter, and 8 and 32 point discrete cosine transforms are synthesised using the ARGEN high level synthesis system [99, 100]. The output of the high level synthesis system are functional control and functional structural RTL data path. The functional structural RTL data path and test registers described at RTL (see appendix C) serve as input to BIST hardware synthesis which is equally applicable to TCC grouping methodology (TCC - Chapter 5), and power conscious test synthesis and scheduling (PC-TSS- Chapter 6). The output of BIST hardware synthesis are: test application time and volume of test data, BIST control, and BIST structural data path. The functional control,

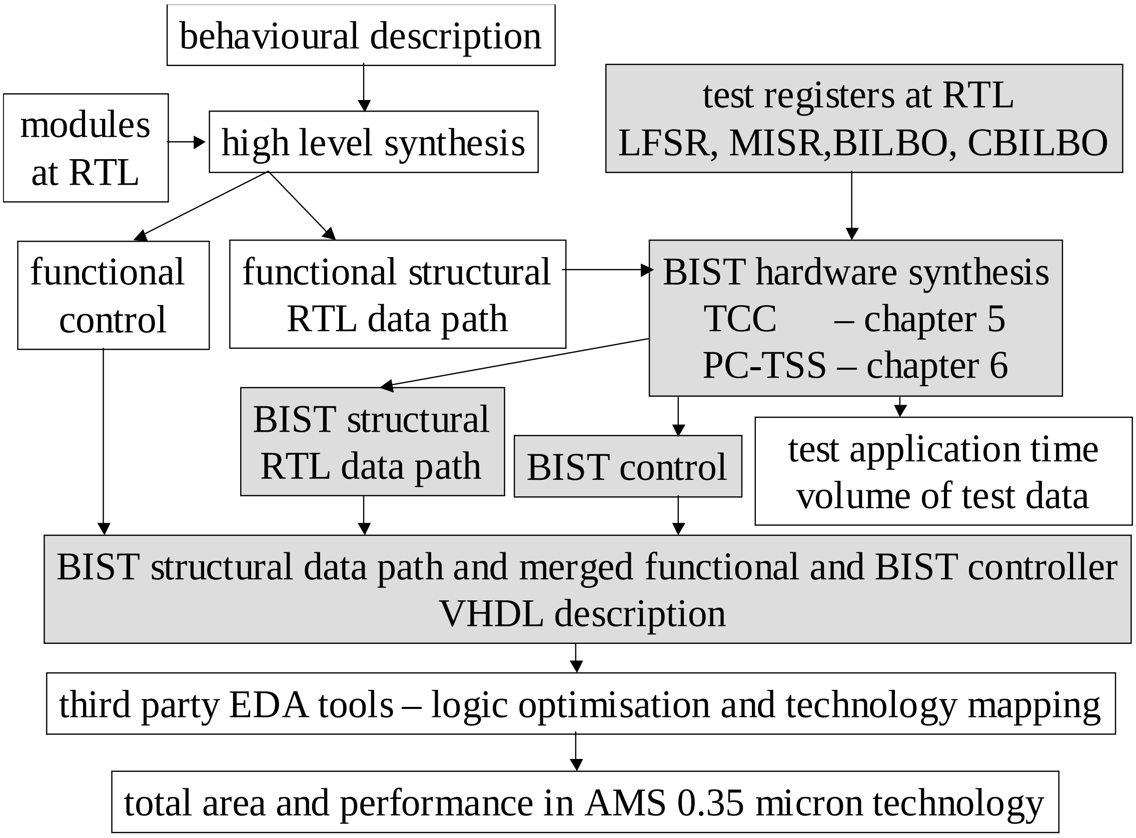

behavioural description
modules at RTL
high level synthesis
test registers at RTL
LFSR, MISR,BILBO, CBILBO
functional control
functional structural RTL data path
BIST hardware synthesis
TCC – chapter 5
PC-TSS – chapter 6
BIST structural RTL data path
BIST control
test application time
volume of test data
BIST structural data path and merged functional and BIST controller VHDL description
third party EDA tools – logic optimisation and technology mapping
total area and performance in AMS 0.35 micron technology

Figure A.1: Area and performance estimation for BIST RTL data paths

BIST control, and BIST structural data path are specified in VHDL [137] and technology mapped using third party EDA tools [55] into AMS 0.35 micron technology [9]. The results obtained after technology mapping provide total area and performance. BIST area overhead is computed by subtracting the total area of the functional structural RTL data path where test registers and BIST control are not inserted. The BIST area overhead, performance, test application time, and volume of test data obtained using the experimental validation flow shown in Figure A.1 were reported in Tables 5.3 - 5.5 from Section 5.4 in Chapter 5, and Tables B.1 - B.9 from appendix B.

In order to compute the power dissipation reported in Section 6.5 from Chapter 6, the experimental validation flow shown in Figure A.2 is employed. The modules at RTL used by the high level synthesis system, as shown in Figure A.1, and test registers are synthesised and technology mapped into AMS 0.35 micron technology [9]. Having obtained the BIST structural data path and BIST control as described in Figure A.1, the number of active data path elements, and the test patterns applied during each test state, serve as

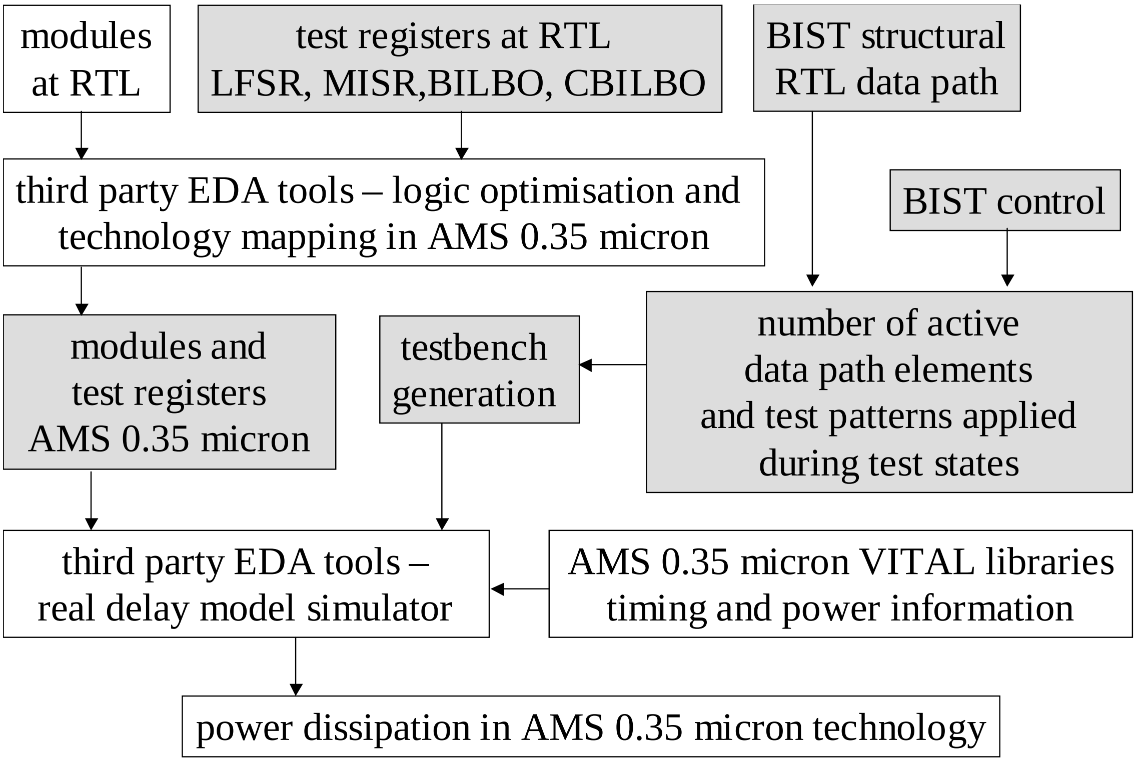

modules at RTL
test registers at RTL LFSR, MISR,BILBO, CBILBO
BIST structural RTL data path
third party EDA tools – logic optimisation and technology mapping in AMS 0.35 micron
BIST control
modules and test registers AMS 0.35 micron
testbench generation
number of active data path elements and test patterns applied during test states
third party EDA tools – real delay model simulator
AMS 0.35 micron VITAL libraries timing and power information
power dissipation in AMS 0.35 micron technology

Figure A.2: Power estimation for BIST RTL data paths

input for the generation of a testbench. The testbench consists of an *activity profile* of all the data path elements in every test state which can either be a test application state (during a test session) or shifting state (during shifting in seeds and shifting out signatures). Also, for every data path element a *power profile* is created, using the following: pseudorandom patterns applied during testing; AMS 0.35 micron VITAL libraries with timing and power information; and a real delay model simulator [130] which accounts for the glitching activity. A data path element is either a module, test register, multiplexer or comparator in the case of TCC grouping. Finally, average value of power dissipation is computed hierarchically by summing the *power profile* of active data path elements in every test state using the *activity profile* over the entire test application period.

A.2 Integration of the Proposed Logic Level Techniques

The experimental validation flow for technology mapping logic level circuits into a target technology when using the proposed BPIC technique (Chapter 3) and multiple scan chain-

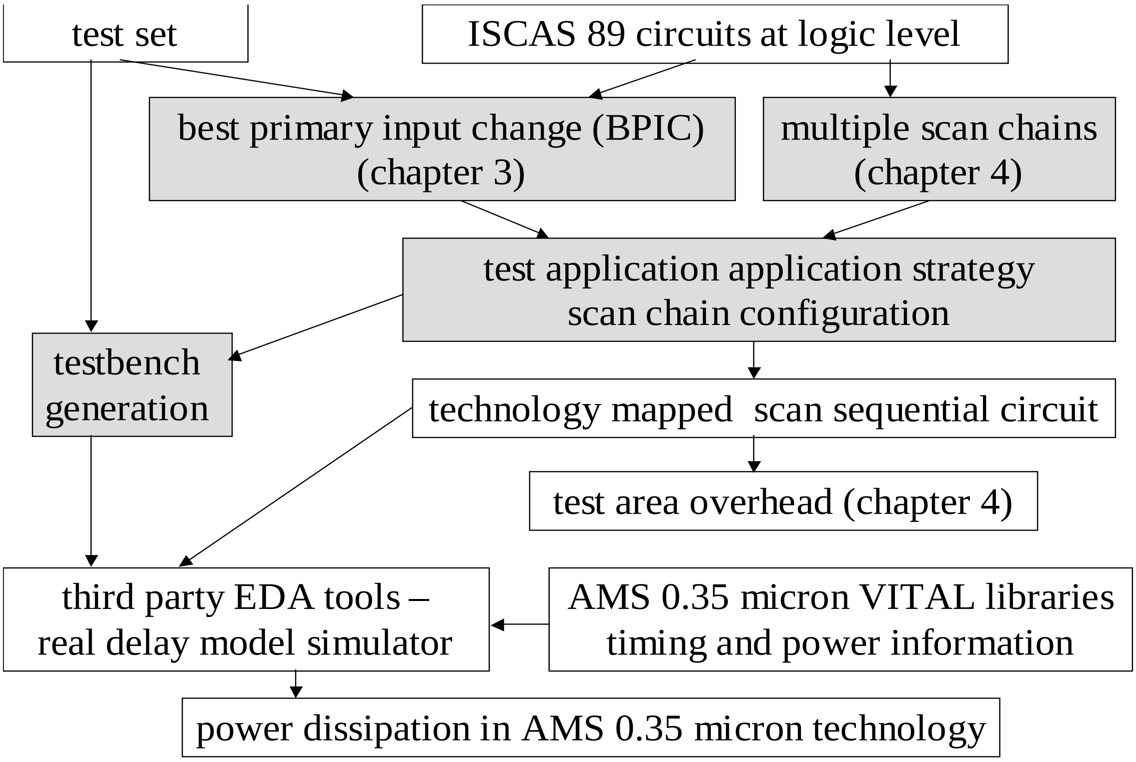

test set
ISCAS 89 circuits at logic level
best primary input change (BPIC)
(chapter 3)
multiple scan chains
(chapter 4)
test application application strategy
scan chain configuration
testbench
generation
technology mapped scan sequential circuit
test area overhead (chapter 4)
third party EDA tools –
real delay model simulator
AMS 0.35 micron VITAL libraries
timing and power information
power dissipation in AMS 0.35 micron technology

Figure A.3: Power estimation for logic level scan sequential circuits

based technique (Chapter 4) is shown in Figure A.3. The circuit description at logic level of abstraction is provided to the algorithms described in Sections 3.4 and 4.4. In the case of the test set dependent technique proposed in Chapter 3, the test set generated using ATOM [83] or MINTEST [84] is also provided as input. The output of the techniques proposed in Chapters 3 and 4 is a description of the scan sequential circuit including the scan chain configuration (scan cell order for Chapter 3 or multiple scan chains for Chapter 4), and the test application strategy defined for both techniques proposed in Chapters 3 and 4. The definition of the test application strategy will serve for the generation of a testbench in VHDL that together with the technology mapped scan sequential circuit, and AMS 0.35 micron VITAL libraries containing timing and power information [9] will be the input of a real delay model simulator [130]. The output of the real delay model simulator which accounts for the glitching activity is the average value of power dissipation over the entire test application period. Similarly, the difference in area between the technology mapped multiple scan chain circuit and the standard scan circuit, provides the overhead in test area shown in Table 4.3(b) from Chapter 4.

Appendix B

Supplementary Experimental Results

Section B.1 shows that node transition count, reported in Chapters 3 and 4, provides reliable relative power information. Section B.2 gives exhaustive supplementary experimental results that show the advantages of the proposed TCCs (Chapter 5) for different data path widths.

B.1 Node Transition Count as Reliable Relative Power Information

This section shows that NTC (Equation 3.1 from Chapter 3) provides reliable relative power information using the example circuit *s344* from ISCAS89 benchmark set [23]. Reliable relative power information provided by NTC, means that savings in NTC under zero delay model reported in experimental results sections from Chapters 3 and 4, and savings in power dissipation obtained after technology mapping the circuit and accounting for glitching activity during test application, are within the same range.

Logic level description of *s344* was technology mapped in AMS 0.35 micron technology [9] using the experimental validation flow shown in Figure A.3 from appendix A. It should be noted that technology mapping accounts for the fanout of each gate by choosing the appropriate library cell with correct timing and power information. It was shown using MINTEST [84] that 13 test vectors are sufficient for obtaining 100% fault coverage. The number of primary inputs is 9 and the number of scan cells (pseudo inputs) is 15. After combining test vector and scan cell ordering, and the proposed BPIC test application strategy (Figure 3.12 in Section 3.4.2 from Chapter 3) two testbench processes were implemented in VHDL. The first testbench process (shown in Example B.1) implements the ASAP test application strategy (Definition 3.2) and the second testbench process implements the BPIC test application strategy (Definition 3.5). Using a state of the art real delay model simulator [130] with power and timing information of every library cell [9], it was obtained, by applying ASAP test application strategy, an average power dissipation of 4.553*mW* for supply voltage 3.3V and and clock frequency 100MHz. By applying BPIC test application strategy, an average power dissipation of 3.326*mW* was obtained. Therefore, power savings when employing the real delay model using VITAL libraries for 0.35 micron AMS technology are of 26.94% which is greater than 24.39% as shown in column 7, row 3, of Table 3.6 in Section 3.5.1 from Chapter 3. This concludes that *NTC* provides reliable relative power information since the savings in both cases are within the same range. The higher savings when employing the real delay model can also be explained by the fact that by eliminating spurious transitions (Definition 3.1) the propagation of hazards and glitches is also eliminated leading to even greater reductions in power dissipation in the case of VITAL libraries for 0.35 micron AMS technology [9]. The following Example B.1 shows the VHDL implementations of ASAP and BPIC test application strategies, including the necessary definitions.

Example B.1

```
-- constant and type definitions for s344

CONSTANT  NO_PI: INTEGER := 9;       -- number of primary inputs
CONSTANT  NO_SC: INTEGER := 15;      -- number of scan cells
CONSTANT  NO_TV: INTEGER := 13;      -- number of test vectors generated by MINTEST
CONSTANT  VSIZE: INTEGER := 24;      -- entire vector size (primary inputs + pseudo inputs)

-- primary input part of the test vector
TYPE primary_vector IS ARRAY (1 TO NO_PI) OF std_logic;
-- pseudo input part of the test vector
TYPE pseudo_vector  IS ARRAY (1 TO NO_SC) OF std_logic;
TYPE test_vector    IS ARRAY (1 TO VSIZE) OF std_logic;   -- entire test vector
TYPE test_set       IS ARRAY (1 TO NO_TV) OF test_vector; -- test set
TYPE bpic_vector    IS ARRAY (1 TO NO_TV) OF integer;     -- best primary input change time
                                                           -- for every test vector
-- signals defined in the architecture of the testbench

signal piv    : primary_vector;      -- primary input vector
signal psv    : pseudo_vector;       -- pseudo  input vector
signal SI   : std_logic;             -- scan input
signal NT   : std_logic;             -- normal/test signal
                                     -- NT = 1 - shift mode
                                     -- NT = 0 - apply mode
-- ASAP process implements the As Soon As Possible (ASAP) test application strategy

ASAP: process
    -- MINTEST test set
    variable TS: test_set :=         -- first NO_SC bits are the pseudo input part
                                     -- of the test vector and the last NO_PI bits
                                     -- are the primary input part of the test vector
      (
        "101000000001111101011010",
        "001111001111001001100100",
        "001101000000101001010100",
        "111000010111111000000001",
        "000111100001111000000000",
        "100000000101111001000111",
        "110111100001110000100000",
        "110000011100010001001100",
        "101000011111010011111010",
        "010110000001101001010010",
        "011011001001011001100101",
        "000101101001111011111111",
        "010011111100011110110001"
      );
begin

  for i in 1 to NO_TV loop           -- loop to apply all the test vectors

    for k in 1 to NO_SC loop
      psv(k) <= TS(i)(k);            -- assign the pseudo input part of the test vector
    end loop;

    for k in 1 to NO_PI loop         -- assign the primary input part of the test vector
      piv(k) <= TS(i)(NO_SC + k);    -- as soon as possible and keep that value
    end loop;                        -- during the entire scan cycle

    NT  <= '1';                      -- set the shift mode to shift in the
                                     -- pseudo input part of the test vector
    for j in NO_SC downto 1 loop
      SI <= psv(j);                  -- assign the scan in value
      wait for 10 ns;
    end loop;

    NT  <= '0';                      -- set the apply mode and load the responses
    wait for 10 ns;

  end loop;

  NT  <= '1';                        -- set the shift mode to shift out the
                                     -- pseudo out part of the last response
  for j in NO_SC downto 1 loop
    wait for 10 ns;
  end loop;

end process;
```

```
-- BPIC process implements the Best Primary Input Change (BPIC) test application strategy
-- after combining test vector and scan cell ordering, and best primary input change time

BPIC: process
    -- MINTEST test set after the pseudo input part of every test vector
    -- was reordered as the newly obtained scan cell order
    variable TS: test_set :=         -- first NO_SC bits are the pseudo input part
                                     -- of the test vector and the last NO_PI bits
                                     -- are the primary input part of the test vector
      (
        "101111111000000011111111",  -- new scan cell order is
        "100101101101110001100100",  -- (9,1,14,12,13,15,4,7,6,5,2,3,11,10,8)
        "011111000001000101011010",
        "000011101001000001010100",  -- new test vector order is
        "001111111100000000000000",  -- (12,2,1,3,5,13,9,4,8,11,6,7,10)
        "101001011110011110110001",
        "111100000001111011111010",
        "011111000011111000000001",
        "111000000010011001001100",
        "101101001111000001100101",
        "011111000000010001000111",
        "011110111110000000100000",
        "000111100110000001010010"
      );
    -- the best primary input change (BPIC) time for every test vector
    variable bpic: bpic_vector := (8,0,0,15,7,0,15,0,1,11,0,0,10);

begin
  for i in 1 to NO_TV loop           -- loop to apply all the test vectors
    for k in 1 to NO_SC loop         -- assign the pseudo input part of the test vector
      psv(k) <= TS(i)(k);
    end loop;
    NT  <= ’1’;                      -- set the shift mode to shift in the
                                     -- pseudo input part of the test vector
    for j in NO_SC downto 1 loop
      if (j < bpic(i)) then          -- check if the primary inputs need to change
        if (i>1) then
          for k in 1 to NO_PI loop
            piv(k) <= TS(i-1)(NO_SC + k);  -- keep the primary inputs to the value of the
          end loop;                        -- primary part of the previous test vector
        else
          for k in 1 to NO_PI loop
            piv(k) <= ’0’;           -- in the case of the first vector
          end loop;                  -- keep the primary inputs to all zero value
        end if;
      else
        for k in 1 to NO_PI loop           -- change the primary inputs to the value of the
          piv(k) <= TS(i)(NO_SC + k);      -- primary input part of the actual test vector
        end loop;
      end if;
      SI <= psv(j);                  -- assign the scan in value
      wait for 10 ns;
    end loop;
    if (bpic(i) = NO_SC) then
      for k in 1 to NO_PI loop       -- assign the primary input part of the test vector
        piv(k) <= TS(i)(NO_SC + k);  -- if best primary input change time is
      end loop;                      -- as late as possible
    end if

    NT  <= '0';                      -- set the apply mode and load the responses
    wait for 10 ns;
  end loop;

  NT  <= '1';                        -- set the shift mode to shift out the
                                     -- pseudo out part of the last response
  for j in NO_SC downto 1 loop
    wait for 10 ns;
  end loop;
end process;
```

B.2 Supplementary Experimental Results on Test Compatibility Classes

This section gives exhaustive results after technology mapping BIST structural data paths in 0.35 micron AMS technology [9] when employing both BIST embedding and TCC grouping methodologies (Chapter 5) for different data path widths.

Tables B.1 - B.9 show the experimental results in terms of test application time, BIST area overhead, performance, and volume of test data for the BIST embedding methodology, (Tables B.1,B.4, and B.7), TCC grouping methodology, (Tables B.2,B.5, and B.8), and the reductions obtained using the TCC grouping methodology over the BIST embedding methodology (Tables B.3,B.6, and B.9) for 4, 8, and 16 bit width data paths. The results were obtained using the benchmark and hypothetical data paths described in Tables 5.1 and 5.6 from Chapter 5, and using the experimental validation flow shown in Figure A.1 from appendix A. As outlined in Section 5.4 from Chapter 5, test lengths for adders and multipliers are assumed to be $T_+ = T_u$, and respectively $T_* = 4 \times T_u$, where T_u is a reasonably large integer and can be estimated for the required fault coverage using the techniques from [117]. In the experimental results reported in this section it was considered that $T_u = 64$, for achieving 100% fault coverage for each 4 bit data path module, $T_u = 128$, for achieving 100% fault coverage for each 8 bit data path module, and $T_u = 256$, for achieving 100% fault coverage for each 16 bit data path module.

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	380	36	135	60	392	41	135	76
EWF-21	368	35	156	52	380	43	156	68
EWF-23	352	43	150	32	404	53	148	88
EWF-25	288	37	148	32	336	41	148	76
EWF-28	352	45	159	32	416	45	158	100
8DCT-10	328	45	150	80	328	48	147	80
8DCT-12	388	43	158	68	428	51	152	112
8DCT-13	396	41	145	80	440	47	143	128
8DCT-14	380	51	148	60	432	54	145	116
8DCT-16	304	46	147	48	376	53	137	116
8DCT-21	296	45	155	40	372	48	142	116
32DCT-30	532	97	136	216	772	105	129	416
32DCT-31	532	95	146	216	752	114	115	400
32DCT-32	528	91	141	208	756	102	119	428
32DCT-33	452	79	125	196	700	98	115	392
32DCT-37	428	73	144	172	700	81	108	388
32DCT-38	432	77	137	180	664	90	117	348
32DCT-40	432	75	130	176	672	87	120	352
EX-01	932	150	111	420	984	183	110	476
EX-02	992	152	103	420	1260	156	103	568
EX-03	932	160	114	420	1068	206	112	564
EX-04	740	136	113	420	988	145	108	556
EX-05	864	145	108	420	1472	165	106	696
EX-06	740	149	113	420	1136	149	112	628
EX-07	992	148	117	480	1092	171	112	544
EX-08	992	162	113	480	1236	194	112	544
EX-09	988	157	104	480	1236	161	103	576
EX-10	988	160	105	480	1160	192	83	596
EX-11	992	164	108	480	1308	165	104	584
EX-12	992	172	105	480	1128	197	103	608
EX-13	1044	177	106	540	1096	224	105	596
EX-14	1108	179	106	540	1208	187	102	620
EX-15	1048	178	88	540	1108	194	77	604
EX-16	1304	199	99	540	1396	232	91	636
EX-17	1032	192	107	540	1312	220	86	612
EX-18	1048	184	109	540	1164	211	103	660

Table B.1: Experimental data for BIST embedding methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 4 bit width data paths in 0.35 micron AMS technology

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	292	34	142	20	332	39	142	64
EWF-21	292	35	156	20	332	43	155	64
EWF-23	280	36	150	20	332	47	150	76
EWF-25	284	31	148	20	316	35	148	64
EWF-28	284	31	159	20	312	32	157	52
8DCT-10	304	37	150	20	324	39	140	44
8DCT-12	300	39	158	20	348	47	149	72
8DCT-13	300	38	145	20	364	44	137	88
8DCT-14	296	38	148	20	352	40	140	80
8DCT-16	284	39	158	20	344	47	158	92
8DCT-21	280	37	158	20	344	40	158	92
32DCT-30	448	81	136	32	728	93	97	360
32DCT-31	440	80	146	32	712	96	102	348
32DCT-32	424	78	141	32	744	88	107	368
32DCT-33	332	76	125	20	564	95	107	312
32DCT-37	336	68	144	20	552	77	107	300
32DCT-38	340	70	137	20	568	83	102	320
32DCT-40	344	66	130	20	576	77	111	324
EX-01	576	115	111	72	940	140	110	440
EX-02	780	142	103	84	1084	147	97	516
EX-03	580	115	114	72	1040	160	104	532
EX-04	508	126	113	72	784	136	99	468
EX-05	740	143	108	84	840	163	106	524
EX-06	584	121	113	72	848	122	112	532
EX-07	704	122	117	96	1052	148	112	488
EX-08	812	147	113	84	1052	176	112	512
EX-09	820	145	104	84	1080	152	103	504
EX-10	640	144	107	72	1080	173	107	576
EX-11	800	150	108	60	1092	153	104	572
EX-12	604	145	105	72	1116	171	103	600
EX-13	712	141	106	84	1036	185	105	412
EX-14	828	153	106	84	1184	167	102	468
EX-15	660	139	114	84	1024	158	114	452
EX-16	688	153	112	72	1156	185	112	544
EX-17	872	162	113	84	1100	194	113	528
EX-18	628	132	109	60	1104	158	103	540

Table B.2: Experimental data for TCC grouping methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 4 bit width data paths in 0.35 micron AMS technology

Design	only BIST hardware				BIST hardware and scan			
	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)
EWF-17	23.16	5.56	4.93	66.67	15.31	4.84	4.93	15.79
EWF-21	20.65	0.00	0.00	61.54	12.63	0.00	0.65	5.88
EWF-23	20.45	16.28	0.00	37.50	17.82	12.54	1.33	13.64
EWF-25	1.39	16.22	0.00	37.50	5.95	13.62	0.00	15.79
EWF-28	19.32	31.11	0.00	37.50	25.00	28.31	0.64	48.00
8DCT-10	7.32	17.78	0.00	75.00	1.22	17.42	5.00	45.00
8DCT-12	22.68	9.30	0.00	70.59	18.69	7.44	2.01	35.71
8DCT-13	24.24	7.32	0.00	75.00	17.27	5.64	4.38	31.25
8DCT-14	22.11	25.49	0.00	66.67	18.52	25.49	3.57	31.03
8DCT-16	6.58	15.22	6.96	58.33	8.51	11.87	13.29	20.69
8DCT-21	5.41	17.78	1.90	50.00	7.53	15.82	10.13	20.69
32DCT-30	15.79	16.49	0.00	85.19	5.70	12.04	32.99	13.46
32DCT-31	17.29	15.79	0.00	85.19	5.32	15.47	12.75	13.00
32DCT-32	19.70	14.29	0.00	84.62	1.59	14.00	11.21	14.02
32DCT-33	26.55	3.80	0.00	89.80	19.43	3.53	7.48	20.41
32DCT-37	21.50	6.85	0.00	88.37	21.14	4.86	0.93	22.68
32DCT-38	21.30	9.09	0.00	88.89	14.46	8.00	14.71	8.05
32DCT-40	20.37	12.00	0.00	88.64	14.29	10.44	8.11	7.95
EX-01	38.20	23.33	0.00	82.86	4.47	23.10	0.00	7.56
EX-02	21.37	6.58	0.00	80.00	13.97	5.99	6.19	9.15
EX-03	37.77	28.12	0.00	82.86	2.62	22.21	7.69	5.67
EX-04	31.35	7.35	0.00	82.86	20.65	6.32	9.09	15.83
EX-05	14.35	1.38	0.00	80.00	42.93	1.27	0.00	24.71
EX-06	21.08	18.79	0.00	82.86	25.35	17.85	0.00	15.29
EX-07	29.03	17.57	0.00	80.00	3.66	13.53	0.00	10.29
EX-08	18.15	9.26	0.00	82.50	14.89	9.17	0.00	5.88
EX-09	17.00	7.64	0.00	82.50	12.62	5.65	0.00	12.50
EX-10	35.22	10.00	1.87	85.00	6.90	9.80	22.43	3.36
EX-11	19.35	8.54	0.00	87.50	16.51	7.17	0.00	2.05
EX-12	39.11	15.70	0.00	85.00	1.06	13.35	0.00	1.32
EX-13	31.80	20.34	0.00	84.44	5.47	17.49	0.00	30.87
EX-14	25.27	14.53	0.00	84.44	1.99	10.75	0.00	24.52
EX-15	37.02	21.91	22.81	84.44	7.58	18.40	32.46	25.17
EX-16	47.24	23.12	11.61	86.67	17.19	20.11	18.75	14.47
EX-17	15.50	15.62	5.31	84.44	16.16	11.87	23.89	13.73
EX-18	40.08	28.26	0.00	88.89	5.15	24.87	0.00	18.18

Table B.3: The reduction of the proposed TCC grouping methodology over the BIST embedding methodology for 4 bit width data paths in terms of test application time, BIST area overhead, performance degradation, and volume of test data

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	760	45	86	120	784	54	86	152
EWF-21	736	43	98	104	760	50	98	136
EWF-23	704	46	91	64	808	49	90	176
EWF-25	576	41	90	64	672	50	90	152
EWF-28	704	49	98	64	832	54	98	200
8DCT-10	656	61	90	160	656	72	90	160
8DCT-12	776	51	91	136	856	51	91	224
8DCT-13	792	49	89	160	880	57	87	256
8DCT-14	760	59	91	120	864	71	90	232
8DCT-16	608	53	90	96	752	56	90	232
8DCT-21	592	51	93	80	744	65	89	232
32DCT-30	1064	124	92	432	1544	136	85	832
32DCT-31	1064	124	92	432	1504	158	88	800
32DCT-32	1056	115	92	416	1512	139	92	856
32DCT-33	904	113	87	392	1400	115	86	784
32DCT-37	856	101	91	344	1400	113	91	776
32DCT-38	864	109	94	360	1328	137	93	696
32DCT-40	864	105	92	352	1344	106	90	704
EX-01	1864	204	76	840	1968	263	75	952
EX-02	1984	198	75	840	2520	255	74	1136
EX-03	1864	234	80	840	2136	255	80	1128
EX-04	1480	195	79	840	1976	208	79	1112
EX-05	1728	203	82	840	2944	225	80	1392
EX-06	1480	249	76	840	2272	261	76	1256
EX-07	1984	189	76	960	2184	202	76	1088
EX-08	1984	211	77	960	2472	244	77	1088
EX-09	1976	215	78	960	2472	221	78	1152
EX-10	1976	217	76	960	2320	227	75	1192
EX-11	1984	224	78	960	2616	275	77	1168
EX-12	1984	249	79	960	2256	283	78	1216
EX-13	2088	227	79	1080	2192	274	79	1192
EX-14	2216	230	78	1080	2416	296	78	1240
EX-15	2096	249	78	1080	2216	321	75	1208
EX-16	2608	259	78	1080	2792	305	75	1272
EX-17	2064	266	77	1080	2624	279	74	1224
EX-18	2096	258	78	1080	2328	281	78	1320

Table B.4: Experimental data for BIST embedding methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 8 bit width data paths in 0.35 micron AMS technology

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	584	40	89	40	664	48	89	128
EWF-21	584	43	98	40	664	50	98	128
EWF-23	560	41	91	40	664	44	91	152
EWF-25	568	36	90	40	632	45	90	128
EWF-28	568	35	98	40	624	39	98	104
8DCT-10	608	45	94	40	648	56	92	88
8DCT-12	600	46	97	40	696	46	97	144
8DCT-13	600	46	90	40	728	55	90	176
8DCT-14	592	45	91	40	704	58	90	160
8DCT-16	568	48	93	40	688	51	93	184
8DCT-21	560	45	94	40	688	59	94	184
32DCT-30	896	101	92	64	1456	114	92	720
32DCT-31	880	100	93	64	1424	132	92	696
32DCT-32	848	96	93	64	1488	120	93	736
32DCT-33	664	95	92	40	1128	100	92	624
32DCT-37	672	85	92	40	1104	100	91	600
32DCT-38	680	86	94	40	1136	114	93	640
32DCT-40	688	84	93	40	1152	90	93	648
EX-01	1152	149	78	144	1880	202	78	880
EX-02	1560	172	76	168	2168	223	76	1032
EX-03	1160	148	80	144	2080	170	80	1064
EX-04	1016	160	79	144	1568	179	79	936
EX-05	1480	175	82	168	1680	202	80	1048
EX-06	1168	157	78	144	1696	185	78	1064
EX-07	1408	158	78	192	2104	172	78	976
EX-08	1624	182	77	168	2104	220	77	1024
EX-09	1640	179	78	168	2160	190	78	1008
EX-10	1280	182	76	144	2160	193	76	1152
EX-11	1600	191	78	120	2184	241	77	1144
EX-12	1208	182	79	144	2232	229	78	1200
EX-13	1424	182	80	168	2072	235	79	824
EX-14	1656	194	79	168	2368	251	79	936
EX-15	1320	182	80	168	2048	237	80	904
EX-16	1376	193	80	144	2312	227	80	1088
EX-17	1744	206	78	168	2200	233	78	1056
EX-18	1256	181	81	120	2208	205	81	1080

Table B.5: Experimental data for TCC grouping methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 8 bit width data paths in 0.35 micron AMS technology

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)
EWF-17	23.16	11.11	3.37	66.67	15.31	10.78	3.37	15.79
EWF-21	20.65	0.00	0.00	61.54	12.63	0.00	0.00	5.88
EWF-23	20.45	10.87	0.00	37.50	17.82	10.22	1.10	13.64
EWF-25	1.39	12.20	0.00	37.50	5.95	11.22	0.00	15.79
EWF-28	19.32	28.57	0.00	37.50	25.00	27.71	0.00	48.00
8DCT-10	7.32	26.23	4.26	75.00	1.22	22.56	2.17	45.00
8DCT-12	22.68	9.80	6.19	70.59	18.69	8.62	6.19	35.71
8DCT-13	24.24	6.12	1.11	75.00	17.27	4.53	3.33	31.25
8DCT-14	22.11	23.73	0.00	66.67	18.52	19.22	0.00	31.03
8DCT-16	6.58	9.43	3.23	58.33	8.51	7.92	3.23	20.69
8DCT-21	5.41	11.76	1.06	50.00	7.53	8.94	5.32	20.69
32DCT-30	15.79	18.55	0.00	85.19	5.70	15.77	7.61	13.46
32DCT-31	17.29	19.35	1.08	85.19	5.32	16.64	4.35	13.00
32DCT-32	19.70	16.52	1.08	84.62	1.59	13.71	1.08	14.02
32DCT-33	26.55	15.93	5.43	89.80	19.43	13.22	6.52	20.41
32DCT-37	21.50	15.84	1.09	88.37	21.14	11.56	0.00	22.68
32DCT-38	21.30	21.10	0.00	88.89	14.46	16.46	0.00	8.05
32DCT-40	20.37	20.00	1.08	88.64	14.29	14.80	3.23	7.95
EX-01	38.20	26.96	2.56	82.86	4.47	22.92	3.85	7.56
EX-02	21.37	13.13	1.32	80.00	13.97	12.60	2.63	9.15
EX-03	37.77	36.75	0.00	82.86	2.62	33.08	0.00	5.67
EX-04	31.35	17.95	0.00	82.86	20.65	14.18	0.00	15.83
EX-05	14.35	13.79	0.00	80.00	42.93	10.07	0.00	24.71
EX-06	21.08	36.95	2.56	82.86	25.35	29.19	2.56	15.29
EX-07	29.03	16.40	2.56	80.00	3.66	14.92	2.56	10.29
EX-08	18.15	13.74	0.00	82.50	14.89	10.03	0.00	5.88
EX-09	17.00	16.74	0.00	82.50	12.62	13.89	0.00	12.50
EX-10	35.22	16.13	0.00	85.00	6.90	15.16	1.32	3.36
EX-11	19.35	14.73	0.00	87.50	16.51	12.37	0.00	2.05
EX-12	39.11	26.91	0.00	85.00	1.06	19.11	0.00	1.32
EX-13	31.80	19.82	1.25	84.44	5.47	14.27	0.00	30.87
EX-14	25.27	15.65	1.27	84.44	1.99	15.34	1.27	24.52
EX-15	37.02	26.91	2.50	84.44	7.58	26.10	6.25	25.17
EX-16	47.24	25.48	2.50	86.67	17.19	25.48	6.25	14.47
EX-17	15.50	22.56	1.28	84.44	16.16	16.24	5.13	13.73
EX-18	40.08	29.84	3.70	88.89	5.15	26.86	3.70	18.18

Table B.6: The reduction of the proposed TCC grouping methodology over the BIST embedding methodology for 8 bit width data paths in terms of test application time, BIST area overhead, performance degradation, and volume of test data

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	1520	59	42	240	1568	71	42	304
EWF-21	1472	55	47	208	1520	57	47	272
EWF-23	1408	53	44	128	1616	67	44	352
EWF-25	1152	49	44	128	1344	52	44	304
EWF-28	1408	55	47	128	1664	66	47	400
8DCT-10	1312	88	43	320	1312	96	43	320
8DCT-12	1552	67	44	272	1712	69	44	448
8DCT-13	1584	67	43	320	1760	69	43	512
8DCT-14	1520	75	43	240	1728	94	43	464
8DCT-16	1216	64	44	192	1504	77	44	464
8DCT-21	1184	60	47	160	1488	75	43	464
32DCT-30	2128	180	45	864	3088	208	45	1664
32DCT-31	2128	180	45	864	3008	199	45	1600
32DCT-32	2112	167	46	832	3024	213	45	1712
32DCT-33	1808	179	45	784	2800	209	45	1568
32DCT-37	1712	161	46	688	2800	162	46	1552
32DCT-38	1728	172	46	720	2656	215	46	1392
32DCT-40	1728	163	46	704	2688	184	46	1408
EX-01	3728	298	41	1680	3936	321	40	1904
EX-02	3968	296	43	1680	5040	358	42	2272
EX-03	3728	306	41	1680	4272	333	41	2256
EX-04	2960	298	41	1680	3952	309	41	2224
EX-05	3456	305	43	1680	5888	344	42	2784
EX-06	2960	308	40	1680	4544	332	40	2512
EX-07	3968	272	42	1920	4368	299	42	2176
EX-08	3968	317	43	1920	4944	405	42	2176
EX-09	3952	303	41	1920	4944	357	41	2304
EX-10	3952	325	43	1920	4640	412	42	2384
EX-11	3968	325	41	1920	5232	344	41	2336
EX-12	3968	351	42	1920	4512	361	41	2432
EX-13	4176	320	41	2160	4384	393	40	2384
EX-14	4432	327	42	2160	4832	421	42	2480
EX-15	4192	357	43	2160	4432	374	43	2416
EX-16	5216	368	41	2160	5584	368	41	2544
EX-17	4128	391	42	2160	5248	504	42	2448
EX-18	4192	351	43	2160	4656	407	42	2640

Table B.7: Experimental data for BIST embedding methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 16 bit width data paths in 0.35 micron AMS technology

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits	TAT clock cycles	BAO sqmil	Perf MHz	VTD data bits
EWF-17	1168	49	43	80	1328	62	43	256
EWF-21	1168	54	47	80	1328	56	47	256
EWF-23	1120	49	44	80	1328	64	44	304
EWF-25	1136	45	44	80	1264	49	44	256
EWF-28	1136	43	47	80	1248	51	47	208
8DCT-10	1216	61	44	80	1296	70	44	176
8DCT-12	1200	62	45	80	1392	65	45	288
8DCT-13	1200	64	43	80	1456	67	43	352
8DCT-14	1184	59	44	80	1408	77	44	320
8DCT-16	1136	64	45	80	1376	77	45	368
8DCT-21	1120	57	47	80	1376	71	47	368
32DCT-30	1792	140	46	128	2912	166	46	1440
32DCT-31	1760	142	45	128	2848	166	45	1392
32DCT-32	1696	133	46	128	2976	177	45	1472
32DCT-33	1328	132	45	80	2256	157	45	1248
32DCT-37	1344	119	46	80	2208	130	46	1200
32DCT-38	1360	122	46	80	2272	157	46	1280
32DCT-40	1376	118	46	80	2304	140	46	1296
EX-01	2304	214	41	288	3760	238	41	1760
EX-02	3120	234	43	336	4336	285	42	2064
EX-03	2320	214	41	288	4160	251	41	2128
EX-04	2032	229	42	288	3136	256	42	1872
EX-05	2960	237	43	336	3360	277	43	2096
EX-06	2336	222	41	288	3392	261	41	2128
EX-07	2816	236	43	384	4208	269	43	1952
EX-08	3248	254	43	336	4208	328	41	2048
EX-09	3280	251	41	336	4320	307	41	2016
EX-10	2560	256	43	288	4320	333	42	2304
EX-11	3200	268	42	240	4368	294	42	2288
EX-12	2416	260	42	288	4464	280	41	2400
EX-13	2848	267	42	336	4144	328	42	1648
EX-14	3312	277	42	336	4736	370	42	1872
EX-15	2640	269	43	336	4096	302	43	1808
EX-16	2752	276	42	288	4624	279	42	2176
EX-17	3488	294	43	336	4400	385	43	2112
EX-18	2512	273	43	240	4416	329	43	2160

Table B.8: Experimental data for TCC grouping methodology in terms of test application time, BIST area overhead, performance degradation, and volume of test data after technology mapping of 16 bit width data paths in 0.35 micron AMS technology

Design	**only BIST hardware**				**BIST hardware and scan**			
	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)	TAT red (%)	BAO red (%)	PD red (%)	VTD red (%)
EWF-17	23.16	16.95	2.33	66.67	15.31	12.54	2.33	15.79
EWF-21	20.65	1.82	0.00	61.54	12.63	1.80	0.00	5.88
EWF-23	20.45	7.55	0.00	37.50	17.82	5.36	0.00	13.64
EWF-25	1.39	8.16	0.00	37.50	5.95	7.02	0.00	15.79
EWF-28	19.32	21.82	0.00	37.50	25.00	21.60	0.00	48.00
8DCT-10	7.32	30.68	2.27	75.00	1.22	26.69	2.27	45.00
8DCT-12	22.68	7.46	2.22	70.59	18.69	6.42	2.22	35.71
8DCT-13	24.24	4.48	0.00	75.00	17.27	3.40	0.00	31.25
8DCT-14	22.11	21.33	2.27	66.67	18.52	17.70	2.27	31.03
8DCT-16	6.58	0.00	2.22	58.33	8.51	0.00	2.22	20.69
8DCT-21	5.41	5.00	0.00	50.00	7.53	4.15	8.51	20.69
32DCT-30	15.79	22.22	2.17	85.19	5.70	20.22	2.17	13.46
32DCT-31	17.29	21.11	0.00	85.19	5.32	16.47	0.00	13.00
32DCT-32	19.70	20.36	0.00	84.62	1.59	16.90	0.00	14.02
32DCT-33	26.55	26.26	0.00	89.80	19.43	24.95	0.00	20.41
32DCT-37	21.50	26.09	0.00	88.37	21.14	19.83	0.00	22.68
32DCT-38	21.30	29.07	0.00	88.89	14.46	26.74	0.00	8.05
32DCT-40	20.37	27.61	0.00	88.64	14.29	23.47	0.00	7.95
EX-01	38.20	28.19	0.00	82.86	4.47	25.93	2.44	7.56
EX-02	21.37	20.95	0.00	80.00	13.97	20.32	0.00	9.15
EX-03	37.77	30.07	0.00	82.86	2.62	24.66	0.00	5.67
EX-04	31.35	23.15	2.38	82.86	20.65	17.13	2.38	15.83
EX-05	14.35	22.30	0.00	80.00	42.93	19.40	2.33	24.71
EX-06	21.08	27.92	2.44	82.86	25.35	21.50	2.44	15.29
EX-07	29.03	13.24	2.33	80.00	3.66	9.80	2.33	10.29
EX-08	18.15	19.87	0.00	82.50	14.89	19.08	2.44	5.88
EX-09	17.00	17.16	0.00	82.50	12.62	14.07	0.00	12.50
EX-10	35.22	21.23	0.00	85.00	6.90	19.32	0.00	3.36
EX-11	19.35	17.54	2.38	87.50	16.51	14.56	2.38	2.05
EX-12	39.11	25.93	0.00	85.00	1.06	22.30	0.00	1.32
EX-13	31.80	16.56	2.38	84.44	5.47	16.56	4.76	30.87
EX-14	25.27	15.29	0.00	84.44	1.99	12.23	0.00	24.52
EX-15	37.02	24.65	0.00	84.44	7.58	19.23	0.00	25.17
EX-16	47.24	25.00	2.38	86.67	17.19	24.00	2.38	14.47
EX-17	15.50	24.81	2.33	84.44	16.16	23.57	2.33	13.73
EX-18	40.08	22.22	0.00	88.89	5.15	19.11	2.33	18.18

Table B.9: The reduction of the proposed TCC grouping methodology over the BIST embedding methodology for 16 bit width data paths in terms of test application time, BIST area overhead, performance degradation, and volume of test data

Appendix C

VHDL Descriptions for Elliptic Wave Digital Filter

This appendix gives the VHDL descriptions for the elliptic wave digital filter example with execution time constraint of 21 control steps. Firstly, the structural BIST data path using the traditional BIST embedding methodology is described. Secondly, structural BIST data path using the proposed TCC grouping methodology is given, and finally RTL descriptions for multiplexers, comparators, and test registers are provided.

The merged functional and BIST controller (Figure 1.10(b)) and BIST data path for the elliptic wave digital filter when applying the traditional BIST embedding methodology, are described in the following. Experimental results in terms of test application time, BIST area overhead, performance and volume of test data are shown in the second row of Table 5.3 in Section 5.4 from Chapter 5. It was outlined in Section B.2 from appendix B that test length for achieving 100% fault coverage in an adder is $T_u = 64$ for 4 bit data paths, $T_u = 128$ for 8 bit data paths, and $T_u = 256$, for 16 bit data paths. This can also be translated as $T_u = 16 \times data\_width$, where $data\_width$ is the data path width (4, 8, and 16 respectively). This observation has served for easier implementation of the BIST controller as explained in the following VHDL description.

```
-- Functional and BIST control, data path and interconnect
-- BIST methodology - BIST EMBEDDING

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity FSM is
        generic (data_width : natural := WIDTH);
        port(
                clk, reset      : in        std_logic;
                -- signals to control module and register input multiplexers
                selM_1_a        : out       mux3sel;
                selM_1_b        : out       mux2sel;
                selM_4_a        : out       mux5sel;
                selM_4_b        : out       mux4sel;
                selM_5_a        : out       mux2sel;
                selM_7_a        : out       mux2sel;
                selM_8_a        : out       mux2sel;
                selR_2_a        : out       mux2sel;
                selR_7_a        : out       mux2sel;
                selR_16_a       : out       mux2sel;
                selR_18_a       : out       mux2sel;
                -- signals to control scan multiplexers
                selS_13_a       : out       mux2sel;
                selS_16_a       : out       mux2sel;
                selS_OUT_a      : out       mux3sel;
                -- signals to control test registers
                shfR_1_a,  genR_1_a,  shfR_2_a,  genR_2_a,  shfR_3_a,
                anlR_3_a,  shfR_7_a,  genR_7_a,  anlR_7_a,  shfR_10_a,
                genR_10_a, shfR_11_a, genR_11_a, shfR_13_a, genR_13_a,
                shfR_16_a, anlR_16_a, shfR_17_a, anlR_17_a, shfR_20_a,
                genR_20_a       : out       std_logic;
                -- switching between functional and test operation
                NORMAL_TEST     : in        std_logic
        );
end FSM;

architecture BEHAVIOUR of FSM is
        type states is (
                NST0,  NST1,  NST2,  NST3,  NST4,  NST5,  NST6,  NST7,
                NST8,  NST9,  NST10, NST11, NST12, NST13, NST14, NST15,
                NST16, NST17, NST18, NST19, NST20, NST21, TST0,  TST1,
                TST2,  TST3,  TST8,  TST9,  TST10);
        signal PRESENT_STATE, NEXT_STATE: states;

        signal count: std_logic_vector(11 downto 0);  -- pattern counter

        -- monitoring the status during testing using the pattern counter
        -- if switch_apply_shift = 1 then test state is changed
        -- from apply mode to shift mode or viceversa

        signal switch_apply_shift: std_logic;

begin

        STATE_MACHINE: process(clk,reset)
        begin
                if reset = '1'
                then PRESENT_STATE <= NST0;
                     count <= conv_std_logic_vector(0,12);
                elsif rising_edge(clk)
                then PRESENT_STATE <= NEXT_STATE;
                     count <= count + 1;
                     if switch_apply_shift = '1'
                     then count <= conv_std_logic_vector(0,12);
                     end if;
                end if;
        end process STATE_MACHINE;

        STATE_TRANS: process(PRESENT_STATE,switch_apply_shift)
        begin
                case PRESENT_STATE is

                        -- normal operation of the circuit

                        when NST0 =>  if NORMAL_TEST = '0'
                                      then NEXT_STATE <= NST1;
                                      else NEXT_STATE <= TST0;
                                      end if;
                        when NST1 =>  NEXT_STATE <= NST2;
                        when NST2 =>  NEXT_STATE <= NST3;
                        when NST3 =>  NEXT_STATE <= NST4;
                        when NST4 =>  NEXT_STATE <= NST5;
                        when NST5 =>  NEXT_STATE <= NST6;
                        when NST6 =>  NEXT_STATE <= NST7;
                        when NST7 =>  NEXT_STATE <= NST8;
                        when NST8 =>  NEXT_STATE <= NST9;
                        when NST9 =>  NEXT_STATE <= NST10;
                        when NST10 => NEXT_STATE <= NST11;
                        when NST11 => NEXT_STATE <= NST12;
                        when NST12 => NEXT_STATE <= NST13;
                        when NST13 => NEXT_STATE <= NST14;
                        when NST14 => NEXT_STATE <= NST15;
                        when NST15 => NEXT_STATE <= NST16;
                        when NST16 => NEXT_STATE <= NST17;
                        when NST17 => NEXT_STATE <= NST18;
                        when NST18 => NEXT_STATE <= NST19;
                        when NST19 => NEXT_STATE <= NST20;
                        when NST20 => NEXT_STATE <= NST21;
                        when NST21 => NEXT_STATE <= NST0;

                        -- test operation of the circuit
                        -- sequence: scan in seed - apply - scan out signature

                        when TST0 =>    if NORMAL_TEST = '1' then
                                          if switch_apply_shift = '1'
                                          then NEXT_STATE <= TST1;
                                          else NEXT_STATE <= TST0;
                                          end if;
                                        else NEXT_STATE <= NST0;
                                        end if;
                        when TST1 =>    if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST2;
                                        else NEXT_STATE <= TST1;
                                        end if;
                        when TST2 =>    if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST3;
                                        else NEXT_STATE <= TST2;
                                        end if;
                        when TST3 =>    if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST8;
                                        else NEXT_STATE <= TST3;
                                        end if;
                        when TST8 =>    if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST9;
                                        else NEXT_STATE <= TST8;
                                        end if;
                        when TST9 =>    if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST10;
                                        else NEXT_STATE <= TST9;
                                        end if;
                        when TST10 =>   if switch_apply_shift = ’1’
                                        then NEXT_STATE <= TST0;
                                        else NEXT_STATE <= TST10;
                                        end if;
                end case;
        end process STATE_TRANS;

        SWITCH_APPLY_SHIFT_UPDATE: process(PRESENT_STATE,count)
        begin
                case PRESENT_STATE is
                        when TST0 =>    if count = 4 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST1 =>    if count = 1 * 16 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST2 =>    if count = 1 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST3 =>    if count = 3 * 16 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST8 =>    if count = 5 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST9 =>    if count = 1 * 16 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when TST10 =>   if count = 2 * data_width
                                        then switch_apply_shift <= ’1’;
                                        else switch_apply_shift <= ’0’;
                                        end if;
                        when others =>  switch_apply_shift <= ’0’;
                end case;
        end process SWITCH_APPLY_SHIFT_UPDATE;

        LOGIC_OUTPUT: process(clk,PRESENT_STATE)
        begin

        -- selection signals for modules and registers

        if (PRESENT_STATE = NST0 or PRESENT_STATE = NST19 or
            PRESENT_STATE = TST0 or PRESENT_STATE = TST1 or
            PRESENT_STATE = TST9 or PRESENT_STATE = TST10) then
                            selM_1_a <= sel1;
        end if;
        if (PRESENT_STATE = NST1 or PRESENT_STATE = NST2 or
            PRESENT_STATE = NST4 or PRESENT_STATE = NST8 or
            PRESENT_STATE = TST8) then
                            selM_1_a <= sel2;
        end if;
        if (PRESENT_STATE = NST3  or PRESENT_STATE = NST6 or
            PRESENT_STATE = NST10 or PRESENT_STATE = NST12 or
            PRESENT_STATE = NST17) then
                            selM_1_a <= sel3;
        end if;
        if (PRESENT_STATE = NST0  or PRESENT_STATE = NST9 or
            PRESENT_STATE = NST10 or PRESENT_STATE = NST11 or
            PRESENT_STATE = TST9) then
                            selM_1_b <= sel1;
        end if;
        if (PRESENT_STATE = NST1  or PRESENT_STATE = NST5 or
            PRESENT_STATE = NST6  or PRESENT_STATE = NST7 or
            PRESENT_STATE = NST14 or PRESENT_STATE = NST16) then
                            selM_1_b <= sel2;
        end if;
        if (PRESENT_STATE = NST6 or PRESENT_STATE = NST14 or
            PRESENT_STATE = TST0 or PRESENT_STATE = TST1 or
            PRESENT_STATE = TST3) then
                            selM_4_a <= sel1;
        end if;
        if (PRESENT_STATE = NST0 or PRESENT_STATE = TST9) then
                            selM_4_a <= sel2;
        end if;
        if (PRESENT_STATE = NST2  or PRESENT_STATE = NST4 or
            PRESENT_STATE = NST12 or PRESENT_STATE = NST13 or
            PRESENT_STATE = TST8) then
                            selM_4_a <= sel3;
        end if;
        if (PRESENT_STATE = NST5 or PRESENT_STATE = NST19) then
                            selM_4_a <= sel4;
        end if;
```

```
if (PRESENT_STATE = NST1 or PRESENT_STATE = NST3 or
    PRESENT_STATE = NST15) then
                    selM_4_a <= sel5;
end if;
if (PRESENT_STATE = NST3 or PRESENT_STATE = NST6 or
    PRESENT_STATE = TST8) then
                    selM_4_b <= sel1;
end if;
if (PRESENT_STATE = NST2  or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST20 or
    PRESENT_STATE = TST10) then
                    selM_4_b <= sel2;
end if;
if (PRESENT_STATE = NST1) then
                    selM_4_b <= sel3;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST13 or
    PRESENT_STATE = TST9) then
                    selM_4_b <= sel4;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST3 or PRESENT_STATE = NST5 or
    PRESENT_STATE = NST9 or PRESENT_STATE = NST10 or
    PRESENT_STATE = TST1 or PRESENT_STATE = TST9) then
                    selM_5_a <= sel1;
end if;
if (PRESENT_STATE = NST1 or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST6 or PRESENT_STATE = TST0) then
                    selM_5_a <= sel2;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST14 or PRESENT_STATE = NST15 or
    PRESENT_STATE = NST20 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST2  or PRESENT_STATE = TST3 or
    PRESENT_STATE = TST10) then
                    selM_7_a <= sel1;
end if;
if (PRESENT_STATE = NST1 or PRESENT_STATE = TST0 or
    PRESENT_STATE = TST8) then
    selM_7_a <= sel2;
end if;
if (PRESENT_STATE = NST1  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST10 or
    PRESENT_STATE = NST12 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST3  or PRESENT_STATE = TST8) then
                    selM_8_a <= sel1;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST16) then
                    selM_8_a <= sel2;
end if;
if (PRESENT_STATE = NST2 or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST5 or PRESENT_STATE = TST3 or
    PRESENT_STATE = TST9) then
                    selR_2_a <= sel1;
```

```
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST1 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST13 or
    PRESENT_STATE = NST17 or PRESENT_STATE = NST20) then
                    selR_2_a <= sel2;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST1 or
    PRESENT_STATE = NST4 or PRESENT_STATE = NST6 or
    PRESENT_STATE = NST7) then
                    selR_7_a <= sel1;
end if;
if (PRESENT_STATE = NST2 or PRESENT_STATE = NST18 or
    PRESENT_STATE = TST1 or PRESENT_STATE = TST3) then
                    selR_7_a <= sel2;
end if;
if (PRESENT_STATE = NST1 or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST4 or PRESENT_STATE = NST18 or
    PRESENT_STATE = TST9) then
                    selR_16_a <= sel1;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST11 or
    PRESENT_STATE = NST17 or PRESENT_STATE = TST1) then
                    selR_16_a <= sel2;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST3 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST5 or
    PRESENT_STATE = NST8  or PRESENT_STATE = NST9 or
    PRESENT_STATE = NST10 or PRESENT_STATE = NST11 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST19 or
    PRESENT_STATE = NST20 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST3) then
                    selR_18_a <= sel1;
end if;
if (PRESENT_STATE = NST1  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST6  or PRESENT_STATE = NST13 or
    PRESENT_STATE = NST16 or PRESENT_STATE = NST17 or
    PRESENT_STATE = NST21) then
                    selR_18_a <= sel2;
end if;

-- multiplexer signals for scan chains

if (PRESENT_STATE = TST8) then selS_13_a <= sel1; end if;
if (PRESENT_STATE = TST0) then selS_13_a <= sel2; end if;
if (PRESENT_STATE = TST2) then selS_16_a <= sel1; end if;
if (PRESENT_STATE = TST10) then selS_16_a <= sel2; end if;
if (PRESENT_STATE = TST0) then  selS_OUT_a <= sel1; end if;
if (PRESENT_STATE = TST2 or PRESENT_STATE = TST10)
then selS_OUT_a <= sel2;
end if;
if (PRESENT_STATE = TST8) then selS_OUT_a <= sel3; end if;

-- shift, generate and analyse signals for test registers
```

```
        if (PRESENT_STATE = TST9) then genR_1_a <= ’1’; else genR_1_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1) then genR_2_a <= ’1’; else genR_2_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST9) then anlR_3_a <= ’1’; else anlR_3_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST9) then genR_7_a <= ’1’; else genR_7_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3)
        then anlR_7_a <= ’1’; else anlR_7_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1) then genR_10_a <= ’1’; else genR_10_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3)
        then genR_11_a <= ’1’; else genR_11_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3 or
            PRESENT_STATE = TST9) then genR_13_a <= ’1’;
        else genR_13_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST9)
        then anlR_16_a <= ’1’; else anlR_16_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3)
        then anlR_17_a <= ’1’; else anlR_17_a <= ’0’;
        end if;
        if (PRESENT_STATE = TST9) then genR_20_a <= ’1’; else genR_20_a <= ’0’;
        end if;

        end process LOGIC_OUTPUT;

end BEHAVIOUR;

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

package comp_fsm is
-- define FSM as a component used in the following entity
end comp_fsm;

package body comp_fsm is
end comp_fsm;

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity circuit is
        generic (data_width : natural := WIDTH);
        port(
                clk, reset : in  std_logic;
                inM_7      : in  std_logic_vector(data_width - 1 downto 0);
                inM_8      : in  std_logic_vector(data_width - 1 downto 0);
                inR_10     : in  std_logic_vector(data_width - 1 downto 0);
                outR_5     : out std_logic_vector(data_width - 1 downto 0);
                SCAN_IN    : in  std_logic;
                SCAN_OUT   : out std_logic;
                NORMAL_TEST: in  std_logic
        );
end circuit;

architecture BEHAVIOUR of circuit is
        signal sig_selM_1_a:        mux3sel;
        signal sig_selM_1_b:        mux2sel;
        signal sig_selM_4_a:        mux5sel;
        signal sig_selM_4_b:        mux4sel;
        signal sig_selM_5_a:        mux2sel;
        signal sig_selM_7_a:        mux2sel;
        signal sig_selM_8_a:        mux2sel;
        signal sig_selR_2_a:        mux2sel;
        signal sig_selR_7_a:        mux2sel;
        signal sig_selR_16_a:       mux2sel;
        signal sig_selR_18_a:       mux2sel;
        signal sig_selS_13_a:       mux2sel;
        signal sig_selS_16_a:       mux2sel;
        signal sig_selS_OUT_a:      mux3sel;
        signal sig_shfR_1_a,  sig_genR_1_a,  sig_shfR_2_a,  sig_genR_2_a,
               sig_shfR_3_a,  sig_anlR_3_a,  sig_shfR_7_a,  sig_genR_7_a,
               sig_anlR_7_a,  sig_shfR_10_a, sig_genR_10_a, sig_shfR_11_a,
               sig_genR_11_a, sig_shfR_13_a, sig_genR_13_a, sig_shfR_16_a,
               sig_anlR_16_a, sig_shfR_17_a, sig_anlR_17_a, sig_shfR_20_a,
               sig_genR_20_a: std_logic;
        signal sig_inM_1_a,   sig_inM_1_b,  sig_outM_1_a, sig_inM_4_a,
               sig_inM_4_b,   sig_outM_4_a, sig_inM_5_a,   sig_inM_5_b,
               sig_outM_5_a,  sig_inM_7_a,  sig_inM_7_b,   sig_outM_7_a,
               sig_inM_8_a,   sig_inM_8_b,  sig_outM_8_a,  sig_inR_1_a,
               sig_outR_1_a,  sig_inR_2_a,  sig_outR_2_a,  sig_inR_3_a,
               sig_outR_3_a,  sig_inR_5_a,  sig_outR_5_a,  sig_inR_7_a,
               sig_outR_7_a,  sig_inR_10_a, sig_outR_10_a, sig_inR_11_a,
               sig_outR_11_a, sig_inR_13_a, sig_outR_13_a, sig_inR_16_a,
               sig_outR_16_a, sig_inR_17_a, sig_outR_17_a, sig_inR_18_a,
               sig_outR_18_a, sig_inR_19_a, sig_outR_19_a, sig_inR_20_a,
               sig_outR_20_a: std_logic_vector(data_width - 1 downto 0);
        signal sig_inS_1_a,  sig_outS_1_a,  sig_inS_2_a,  sig_outS_2_a,
               sig_inS_3_a,  sig_outS_3_a,  sig_inS_5_a,  sig_outS_5_a,
               sig_inS_7_a,  sig_outS_7_a,  sig_inS_10_a, sig_outS_10_a,
               sig_inS_11_a, sig_outS_11_a, sig_inS_13_a, sig_outS_13_a,
               sig_inS_16_a, sig_outS_16_a, sig_inS_17_a, sig_outS_17_a,
               sig_inS_18_a, sig_outS_18_a, sig_inS_19_a, sig_outS_19_a,
               sig_inS_20_a, sig_outS_20_a: std_logic;

begin

        -- map the multiplexers at the inputs of modules
```

```
pm_inM_1_a: mux3 port map(sig_selM_1_a, sig_outR_1_a,
                          sig_outR_2_a, sig_outR_3_a, sig_inM_1_a);
pm_inM_1_b: mux2 port map(sig_selM_1_b, sig_outR_13_a,
                          sig_outR_17_a, sig_inM_1_b);
pm_inM_4_a: mux5 port map(sig_selM_4_a, sig_outR_1_a, sig_outR_7_a,
                          sig_outR_11_a, sig_outR_17_a, sig_outR_19_a,
                          sig_inM_4_a);
pm_inM_4_b: mux4 port map(sig_selM_4_b, sig_outR_3_a, sig_outR_5_a,
                          sig_outR_18_a,  sig_outR_20_a, sig_inM_4_b);
pm_inM_5_a: mux2 port map(sig_selM_5_a, sig_outR_10_a, sig_outR_18_a,
                          sig_inM_5_a);
sig_inM_5_b <= sig_outR_2_a;
pm_inM_7_a: mux2 port map(sig_selM_7_a, sig_outR_13_a,
                          sig_outR_16_a, sig_inM_7_a);
sig_inM_7_b <= inM_7;
pm_inM_8_a: mux2 port map(sig_selM_8_a, sig_outR_11_a,
                          sig_outR_13_a, sig_inM_8_a);
sig_inM_8_b <= inM_8;

-- map the multiplexers at the inputs of registers

sig_inR_1_a <= sig_outM_4_a;
pm_inR_2_a: mux2 port map(sig_selR_2_a, sig_outM_5_a,
                          sig_outM_7_a, sig_inR_2_a);
sig_inR_3_a <= sig_outM_1_a;
sig_inR_5_a <= sig_outM_4_a;
pm_inR_7_a: mux2 port map(sig_selR_7_a, sig_outM_5_a,
                          sig_outM_7_a, sig_inR_7_a);
sig_inR_10_a <= inR_10;
sig_inR_11_a <= sig_outM_4_a;
sig_inR_13_a <= sig_outM_4_a;
pm_inR_16_a: mux2 port map(sig_selR_16_a, sig_outM_4_a,
                           sig_outM_5_a, sig_inR_16_a);
sig_inR_17_a <= sig_outM_8_a;
pm_inR_18_a: mux2 port map(sig_selR_18_a, sig_outM_4_a,
                           sig_outM_7_a, sig_inR_18_a);
sig_inR_19_a <= sig_outM_4_a;
sig_inR_20_a <= sig_outM_1_a;

-- map the multiplexers at the scan inputs of shift registers

sig_inS_1_a  <= SCAN_IN;
sig_inS_2_a  <= SCAN_IN;
sig_inS_3_a  <= SCAN_IN;
sig_inS_7_a  <= sig_outS_1_a;
sig_inS_10_a <= sig_outS_2_a;
sig_inS_11_a <= sig_outS_10_a;
pm_inS_13_a: smux2 port map(sig_selS_13_a, sig_outS_7_a,
                            sig_outS_11_a, sig_inS_13_a);
pm_inS_16_a: smux2 port map(sig_selS_16_a, SCAN_IN,
                            sig_outS_3_a, sig_inS_16_a);
sig_inS_17_a <=  sig_outS_13_a;
sig_inS_20_a <=  sig_outS_17_a;
pm_inS_OUT_a: smux3 port map(sig_selS_OUT_a, sig_outS_13_a,
                                   sig_outS_16_a, sig_outS_20_a, SCAN_OUT);

        -- map the modules and registers

        pm_M_1: add   port map(sig_inM_1_a, sig_inM_1_b, sig_outM_1_a);
        pm_M_4: add   port map(sig_inM_4_a, sig_inM_4_b, sig_outM_4_a);
        pm_M_5: add   port map(sig_inM_5_a, sig_inM_5_b, sig_outM_5_a);
        pm_M_7: mul   port map(sig_inM_7_a, sig_inM_7_b, sig_outM_7_a);
        pm_M_8: mul   port map(sig_inM_8_a, sig_inM_8_b, sig_outM_8_a);
        pm_R_1: lfsr  port map(clk,sig_shfR_1_a,sig_genR_1_a,sig_inS_1_a,
                               sig_inR_1_a,sig_outR_1_a,sig_outS_1_a);
        pm_R_2: lfsr  port map(clk,sig_shfR_2_a,sig_genR_2_a,sig_inS_2_a,
                               sig_inR_2_a,sig_outR_2_a,sig_outS_2_a);
        pm_R_3: misr  port map(clk,sig_shfR_3_a,sig_anlR_3_a,sig_inS_3_a,
                               sig_inR_3_a,sig_outR_3_a,sig_outS_3_a);
        pm_R_5: reg   port map(clk,sig_inR_5_a,sig_outR_5_a);
        pm_R_7: bilbo port map(clk,sig_shfR_7_a,sig_genR_7_a,sig_anlR_7_a,
                               sig_inS_7_a,sig_inR_7_a,sig_outR_7_a,
                               sig_outS_7_a);
        pm_R_10: lfsr port map(clk,sig_shfR_10_a,sig_genR_10_a,sig_inS_10_a,
                               sig_inR_10_a,sig_outR_10_a,sig_outS_10_a);
        pm_R_11: lfsr port map(clk,sig_shfR_11_a,sig_genR_11_a,sig_inS_11_a,
                               sig_inR_11_a,sig_outR_11_a,sig_outS_11_a);
        pm_R_13: lfsr port map(clk,sig_shfR_13_a,sig_genR_13_a,sig_inS_13_a,
                               sig_inR_13_a,sig_outR_13_a,sig_outS_13_a);
        pm_R_16: misr port map(clk,sig_shfR_16_a,sig_anlR_16_a,sig_inS_16_a,
                               sig_inR_16_a,sig_outR_16_a,sig_outS_16_a);
        pm_R_17: misr port map(clk,sig_shfR_17_a,sig_anlR_17_a,sig_inS_17_a,
                               sig_inR_17_a,sig_outR_17_a,sig_outS_17_a);
        pm_R_18: reg  port map(clk,sig_inR_18_a,sig_outR_18_a);
        pm_R_19: reg  port map(clk,sig_inR_19_a,sig_outR_19_a);
        pm_R_20: lfsr port map(clk,sig_shfR_20_a,sig_genR_20_a,sig_inS_20_a,
                               sig_inR_20_a,sig_outR_20_a,sig_outS_20_a);

        -- map the finite state machine

        pm_FSM: work.comp_fsm.FSM port map(clk, reset, sig_selM_1_a,
                                  sig_selM_1_b, sig_selM_4_a, sig_selM_4_b,
                                  sig_selM_5_a, sig_selM_7_a, sig_selM_8_a,
                                  sig_selR_2_a, sig_selR_7_a, sig_selR_16_a,
                                  sig_selR_18_a, sig_selS_13_a, sig_selS_16_a,
                                  sig_selS_OUT_a, sig_shfR_1_a, sig_genR_1_a,
                                  sig_shfR_2_a, sig_genR_2_a, sig_shfR_3_a,
                                  sig_anlR_3_a, sig_shfR_7_a, sig_genR_7_a,
                                  sig_anlR_7_a, sig_shfR_10_a, sig_genR_10_a,
                                  sig_shfR_11_a, sig_genR_11_a, sig_shfR_13_a,
                                  sig_genR_13_a, sig_shfR_16_a, sig_anlR_16_a,
                                  sig_shfR_17_a, sig_anlR_17_a, sig_shfR_20_a,
                                  sig_genR_20_a, NORMAL_TEST);

        y outR_5 <= sig_outR_5_a;

end BEHAVIOUR;
```

The merged functional and BIST controller (Figure 1.10(b)) and BIST data path for the elliptic wave digital filter when applying the proposed TCC grouping methodology, are described in the following VHDL description. Experimental results and reduction over the traditional BIST embedding methodology are given in the second row of Table 5.2, Table 5.4, and Table 5.5 respectively.

```
-- Functional and BIST control, data path and interconnect
-- BIST methodology - TEST COMPATIBILITY CLASSES

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity FSM is
        generic (data_width : natural := WIDTH);
        port(
                clk, reset      : in          std_logic;
                -- signals to control module and register input multiplexers
                selM_1_a        : out         mux3sel;
                selM_1_b        : out         mux2sel;
                selM_4_a        : out         mux5sel;
                selM_4_b        : out         mux4sel;
                selM_5_a        : out         mux2sel;
                selM_7_a        : out         mux2sel;
                selM_8_a        : out         mux2sel;
                selR_2_a        : out         mux2sel;
                selR_7_a        : out         mux2sel;
                selR_16_a       : out         mux2sel;
                selR_18_a       : out         mux2sel;
                -- signals to control scan multiplexers
                selS_OUT_a      : out         mux3sel;
                -- signals to control test registers
                shfR_1_a,  genR_1_a,  shfR_2_a,  genR_2_a,  shfR_3_a,
                anlR_3_a,  shfR_10_a, genR_10_a, shfR_11_a, genR_11_a,
                shfR_13_a, genR_13_a, shfR_17_a, genR_17_a, shfR_18_a,
                anlR_18_a, shfR_20_a, genR_20_a: out std_logic;
                -- signals checking the output of comparators
                TCC_5           : in          std_logic;
                TCC_8           : in          std_logic;
                -- FAIL is activated if comparators indicate an error
                FAIL            : out         std_logic;
                -- switching between functional and test operation
                NORMAL_TEST     : in          std_logic
        );
end FSM;

architecture BEHAVIOUR of FSM is
```

```
        type states is (
                NST0,  NST1,  NST2,  NST3,  NST4,  NST5,  NST6,  NST7,  NST8,
                NST9,  NST10, NST11, NST12, NST13, NST14, NST15, NST16, NST17,
                NST18, NST19, NST20, NST21, TST0,  TST1,  TST2,  TST3,  TST8);
        signal PRESENT_STATE, NEXT_STATE: states;

        signal count: std_logic_vector(11 downto 0);  -- pattern counter

        -- monitoring the status during testing using the pattern counter
        -- if switch_apply_shift = 1 then test state is changed
        -- from apply mode to shift mode or viceversa

        signal switch_apply_shift: std_logic;

begin

        STATE_MACHINE: process(clk,reset)
        begin
                if reset = '1'
                then PRESENT_STATE <= NST0;
                     count <= conv_std_logic_vector(0,12);
                elsif rising_edge(clk) then
                      PRESENT_STATE <= NEXT_STATE;
                      count <= count + 1;
                      if switch_apply_shift = '1'
                      then count <= conv_std_logic_vector(0,12);
                      end if;
                end if;
        end process STATE_MACHINE;

        STATE_TRANS: process(PRESENT_STATE,switch_apply_shift)
        begin
                case PRESENT_STATE is

                        -- normal operation of the circuit

                        when NST0 =>         if NORMAL_TEST = '0'
                                             then NEXT_STATE <= NST1;
                                             else NEXT_STATE <= TST0;
                                             end if;
                        when NST1 =>         NEXT_STATE <= NST2;
                        when NST2 =>         NEXT_STATE <= NST3;
                        when NST3 =>         NEXT_STATE <= NST4;
                        when NST4 =>         NEXT_STATE <= NST5;
                        when NST5 =>         NEXT_STATE <= NST6;
                        when NST6 =>         NEXT_STATE <= NST7;
                        when NST7 =>         NEXT_STATE <= NST8;
                        when NST8 =>         NEXT_STATE <= NST9;
                        when NST9 =>         NEXT_STATE <= NST10;
                        when NST10 =>        NEXT_STATE <= NST11;
                        when NST11 =>        NEXT_STATE <= NST12;
                        when NST12 =>        NEXT_STATE <= NST13;
                        when NST13 =>        NEXT_STATE <= NST14;
                        when NST14 =>        NEXT_STATE <= NST15;
                        when NST15 =>        NEXT_STATE <= NST16;
                        when NST16 =>        NEXT_STATE <= NST17;
                        when NST17 =>        NEXT_STATE <= NST18;
                        when NST18 =>        NEXT_STATE <= NST19;
                        when NST19 =>        NEXT_STATE <= NST20;
                        when NST20 =>        NEXT_STATE <= NST21;
                        when NST21 =>        NEXT_STATE <= NST0;

                        -- test operation of the circuit
                        -- sequence: scan in seed - apply - scan out signature

                        when TST0 =>    if NORMAL_TEST = '1' then
                                          if switch_apply_shift = '1'
                                          then NEXT_STATE <= TST1;
                                          else NEXT_STATE <= TST0;
                                          end if;
                                        else NEXT_STATE <= NST0;
                                        end if;
                        when TST1 =>    if switch_apply_shift = '1'
                                        then NEXT_STATE <= TST2;
                                        else NEXT_STATE <= TST1;
                                        end if;
                        when TST2 =>    if switch_apply_shift = '1'
                                        then NEXT_STATE <= TST3;
                                        else NEXT_STATE <= TST2;
                                        end if;
                        when TST3 =>    if switch_apply_shift = '1'
                                        then NEXT_STATE <= TST8;
                                        else NEXT_STATE <= TST3;
                                        end if;
                        when TST8 =>    if switch_apply_shift = '1'
                                        then NEXT_STATE <= TST0;
                                        else NEXT_STATE <= TST8;
                                        end if;
                end case;
        end process STATE_TRANS;

        SWITCH_APPLY_SHIFT_UPDATE: process(PRESENT_STATE,count)
        begin
                case PRESENT_STATE is
                        when TST0 =>    if count = 7 * data_width
                                        then switch_apply_shift <= '1';
                                        else switch_apply_shift <= '0';
                                        end if;
                        when TST1 =>    if count = 1 * 16 * data_width
                                        then switch_apply_shift <= '1';
                                        else switch_apply_shift <= '0';
                                        end if;
                        when TST2 =>    if count = 1 * data_width
                                        then switch_apply_shift <= '1';
                                        else switch_apply_shift <= '0';
                                        end if;
                        when TST3 =>    if count = 3 * 16 * data_width
                                        then switch_apply_shift <= '1';
                                        else switch_apply_shift <= '0';
                                        end if;
                        when TST8 =>    if count = 1 * data_width
                                        then switch_apply_shift <= '1';
                                        else switch_apply_shift <= '0';
                                        end if;
                        when others =>  switch_apply_shift <= '0';
                end case;
        end process SWITCH_APPLY_SHIFT_UPDATE;

        LOGIC_OUTPUT: process(clk,PRESENT_STATE)
        begin

        -- selection signals for modules and registers
        if (PRESENT_STATE = NST0 or PRESENT_STATE = NST19 or
            PRESENT_STATE = TST1 or PRESENT_STATE = TST2) then
                            selM_1_a <= sel1;
        end if;
        if (PRESENT_STATE = NST1 or PRESENT_STATE = NST2 or
            PRESENT_STATE = NST4 or PRESENT_STATE = NST8) then
                            selM_1_a <= sel2;
        end if;
        if (PRESENT_STATE = NST3  or PRESENT_STATE = NST6 or
            PRESENT_STATE = NST10 or PRESENT_STATE = NST12 or
            PRESENT_STATE = NST17 or PRESENT_STATE = TST0) then
                            selM_1_a <= sel3;
        end if;
        if (PRESENT_STATE = NST0  or PRESENT_STATE = NST9 or
            PRESENT_STATE = NST10 or PRESENT_STATE = NST11) then
                            selM_1_b <= sel1;
        end if;
        if (PRESENT_STATE = NST1  or PRESENT_STATE = NST5 or
            PRESENT_STATE = NST6  or PRESENT_STATE = NST7 or
            PRESENT_STATE = NST14 or PRESENT_STATE = NST16 or
            PRESENT_STATE = TST1) then
                            selM_1_b <= sel2;
        end if;
        if (PRESENT_STATE = NST6 or PRESENT_STATE = NST14 or
            PRESENT_STATE = TST1 or PRESENT_STATE = TST3) then
                            selM_4_a <= sel1;
        end if;
        if (PRESENT_STATE = NST0 or PRESENT_STATE = TST0) then
                            selM_4_a <= sel2;
        end if;
        if (PRESENT_STATE = NST2  or PRESENT_STATE = NST4 or
            PRESENT_STATE = NST12 or PRESENT_STATE = NST13) then
                            selM_4_a <= sel3;
        end if;
        if (PRESENT_STATE = NST5  or PRESENT_STATE = NST19) then
                            selM_4_a <= sel4;
        end if;
        if (PRESENT_STATE = NST1 or PRESENT_STATE = NST3 or
            PRESENT_STATE = NST15) then
                            selM_4_a <= sel5;
```

```
end if;
if (PRESENT_STATE = NST3  or PRESENT_STATE = NST6 or
    PRESENT_STATE = TST0  or PRESENT_STATE = TST3 or
    PRESENT_STATE = TST8) then
                   selM_4_b <= sel1;
end if;
if (PRESENT_STATE = NST2  or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST20 or
    PRESENT_STATE = TST2) then
                    selM_4_b <= sel2;
end if;
if (PRESENT_STATE = NST1) then
                    selM_4_b <= sel3;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST13 or
    PRESENT_STATE = TST1) then
                    selM_4_b <= sel4;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST3 or PRESENT_STATE = NST5 or
    PRESENT_STATE = NST9 or PRESENT_STATE = NST10 or
    PRESENT_STATE = TST1 or PRESENT_STATE = TST3 or
    PRESENT_STATE = TST8) then
                    selM_5_a <= sel1;
end if;
if (PRESENT_STATE = NST1 or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST6 or PRESENT_STATE = TST0) then
                    selM_5_a <= sel2;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST14 or PRESENT_STATE = NST15 or
    PRESENT_STATE = NST20 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST3) then
                    selM_7_a <= sel1;
end if;
if (PRESENT_STATE = NST1 or PRESENT_STATE = TST0) then
                    selM_7_a <= sel2;
end if;
if (PRESENT_STATE = NST1  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST10 or
    PRESENT_STATE = NST12 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST3) then
                    selM_8_a <= sel1;
end if;
if (PRESENT_STATE = NST0 or PRESENT_STATE = NST16) then
                    selM_8_a <= sel2;
end if;
if (PRESENT_STATE = NST2 or PRESENT_STATE = NST4 or
    PRESENT_STATE = NST5 or PRESENT_STATE = TST3) then
                    selR_2_a <= sel1;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST1 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST13 or
    PRESENT_STATE = NST17 or PRESENT_STATE = NST20) then
```

```
                    selR_2_a <= sel2;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST1 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST6 or
    PRESENT_STATE = NST7  or PRESENT_STATE = TST3) then
                    selR_7_a <= sel1;
end if;
if (PRESENT_STATE = NST2  or PRESENT_STATE = NST18) then
                    selR_7_a <= sel2;
end if;
if (PRESENT_STATE = NST1  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST18) then
                    selR_16_a <= sel1;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST11 or
    PRESENT_STATE = NST17) then
                    selR_16_a <= sel2;
end if;
if (PRESENT_STATE = NST0  or PRESENT_STATE = NST3 or
    PRESENT_STATE = NST4  or PRESENT_STATE = NST5 or
    PRESENT_STATE = NST8  or PRESENT_STATE = NST9 or
    PRESENT_STATE = NST10 or PRESENT_STATE = NST11 or
    PRESENT_STATE = NST12 or PRESENT_STATE = NST19 or
    PRESENT_STATE = NST20) then
                    selR_18_a <= sel1;
end if;
if (PRESENT_STATE = NST1  or PRESENT_STATE = NST2 or
    PRESENT_STATE = NST6  or PRESENT_STATE = NST13 or
    PRESENT_STATE = NST16 or PRESENT_STATE = NST17 or
    PRESENT_STATE = NST21 or PRESENT_STATE = TST1 or
    PRESENT_STATE = TST3) then
                    selR_18_a <= sel2;
end if;

-- multiplexer signals for scan chains

if (PRESENT_STATE = TST2) then selS_OUT_a <= sel1; end if;
if (PRESENT_STATE = TST8) then selS_OUT_a <= sel2; end if;
if (PRESENT_STATE = TST0) then selS_OUT_a <= sel3; end if;

-- shift, generate and analyse signals for test registers

if (PRESENT_STATE = TST1) then genR_1_a <= '1';
else genR_1_a <= '0';
end if;
if (PRESENT_STATE = TST1) then genR_2_a <= '1';
else genR_2_a <= '0';
end if;
if (PRESENT_STATE = TST1) then anlR_3_a <= '1';
else anlR_3_a <= '0';
end if;
if (PRESENT_STATE = TST1) then genR_10_a <= '1';
else genR_10_a <= '0';
end if;
```

```
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3) then
             genR_11_a <= '1';
        else genR_11_a <= '0';
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3) then
             genR_13_a <= '1';
        else genR_13_a <= '0';
        end if;
        if (PRESENT_STATE = TST1) then genR_17_a <= '1';
        else genR_17_a <= '0';
        end if;
        if (PRESENT_STATE = TST1 or PRESENT_STATE = TST3) then
             anlR_18_a <= '1';
        else anlR_18_a <= '0';
        end if;
        if (PRESENT_STATE = TST1) then genR_20_a <= '1';
        else genR_20_a <= '0';
        end if;

        -- fail signal for test compatibility classes

        if ((TCC_5 = '0' and ( PRESENT_STATE = TST1)) or
            (TCC_8 = '0' and ( PRESENT_STATE = TST1 or PRESENT_STATE = TST3)))
        then FAIL <= '1'; else FAIL <= '0';
        end if;

        end process LOGIC_OUTPUT;

end BEHAVIOUR;

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

package comp_fsm is
-- define FSM as a component used in the following entity
end comp_fsm;

package body comp_fsm is
end comp_fsm;

Library IEEE;
use IEEE.std_logic_1164.all;
use IEEE.std_logic_arith.all;
use IEEE.std_logic_unsigned.all;

entity circuit is
        generic (data_width : natural := WIDTH);
        port(
                clk, reset : in  std_logic;
                inM_7      : in  std_logic_vector(data_width - 1 downto 0);
                inM_8      : in  std_logic_vector(data_width - 1 downto 0);
                inR_10     : in  std_logic_vector(data_width - 1 downto 0);
                outR_5     : out std_logic_vector(data_width - 1 downto 0);
                SCAN_IN    : in  std_logic;
                SCAN_OUT   : out std_logic;
                FAIL       : out std_logic;
                NORMAL_TEST: in  std_logic
        );
end circuit;

architecture BEHAVIOUR of circuit is
        signal sig_selM_1_a:        mux3sel;
        signal sig_selM_1_b:        mux2sel;
        signal sig_selM_4_a:        mux5sel;
        signal sig_selM_4_b:        mux4sel;
        signal sig_selM_5_a:        mux2sel;
        signal sig_selM_7_a:        mux2sel;
        signal sig_selM_8_a:        mux2sel;
        signal sig_selR_2_a:        mux2sel;
        signal sig_selR_7_a:        mux2sel;
        signal sig_selR_16_a:       mux2sel;
        signal sig_selR_18_a:       mux2sel;
        signal sig_selS_OUT_a:      mux3sel;
        signal sig_shfR_1_a,  sig_genR_1_a,  sig_shfR_2_a,  sig_genR_2_a,
               sig_shfR_3_a,  sig_anlR_3_a,  sig_shfR_10_a, sig_genR_10_a,
               sig_shfR_11_a, sig_genR_11_a, sig_shfR_13_a, sig_genR_13_a,
               sig_shfR_17_a, sig_genR_17_a, sig_shfR_18_a, sig_anlR_18_a,
               sig_shfR_20_a, sig_genR_20_a, sig_TCC_5,     sig_TCC_8,
               sig_FAIL:      std_logic;
        signal sig_inM_1_a,   sig_inM_1_b,  sig_outM_1_a,  sig_inM_4_a,
               sig_inM_4_b,   sig_outM_4_a, sig_inM_5_a,   sig_inM_5_b,
               sig_outM_5_a,  sig_inM_7_a,  sig_inM_7_b,   sig_outM_7_a,
               sig_inM_8_a,   sig_inM_8_b,  sig_outM_8_a,  sig_inR_1_a,
               sig_outR_1_a,  sig_inR_2_a,  sig_outR_2_a,  sig_inR_3_a,
               sig_outR_3_a,  sig_inR_5_a,  sig_outR_5_a,  sig_inR_7_a,
               sig_outR_7_a,  sig_inR_10_a, sig_outR_10_a, sig_inR_11_a,
               sig_outR_11_a, sig_inR_13_a, sig_outR_13_a, sig_inR_16_a,
               sig_outR_16_a, sig_inR_17_a, sig_outR_17_a, sig_inR_18_a,
               sig_outR_18_a, sig_inR_19_a, sig_outR_19_a, sig_inR_20_a,
               sig_outR_20_a: std_logic_vector(data_width - 1 downto 0);
        signal sig_inS_1_a,  sig_outS_1_a,  sig_inS_2_a,  sig_outS_2_a,
               sig_inS_3_a,  sig_outS_3_a,  sig_inS_5_a,  sig_outS_5_a,
               sig_inS_7_a,  sig_outS_7_a,  sig_inS_10_a, sig_outS_10_a,
               sig_inS_11_a, sig_outS_11_a, sig_inS_13_a, sig_outS_13_a,
       sig_inS_16_a, sig_outS_16_a, sig_inS_17_a, sig_outS_17_a,
       sig_inS_18_a, sig_outS_18_a, sig_inS_19_a, sig_outS_19_a,
       sig_inS_20_a, sig_outS_20_a: std_logic;

begin

        -- map the multiplexers at the inputs of modules

        pm_inM_1_a: mux3 port map(sig_selM_1_a, sig_outR_1_a, sig_outR_2_a,
                                  sig_outR_3_a, sig_inM_1_a);
        pm_inM_1_b: mux2 port map(sig_selM_1_b,  sig_outR_13_a,
                                  sig_outR_17_a, sig_inM_1_b);
```

```
pm_inM_4_a: mux5 port map(sig_selM_4_a,  sig_outR_1_a,  sig_outR_7_a,
                          sig_outR_11_a, sig_outR_17_a, sig_outR_19_a,
                          sig_inM_4_a);
pm_inM_4_b: mux4 port map(sig_selM_4_b,  sig_outR_3_a,  sig_outR_5_a,
                          sig_outR_18_a, sig_outR_20_a, sig_inM_4_b);
pm_inM_5_a: mux2 port map(sig_selM_5_a,  sig_outR_10_a,
                          sig_outR_18_a, sig_inM_5_a);
sig_inM_5_b <= sig_outR_2_a;
pm_inM_7_a: mux2 port map(sig_selM_7_a,  sig_outR_13_a,
                          sig_outR_16_a, sig_inM_7_a);
sig_inM_7_b <= inM_7;
pm_inM_8_a: mux2 port map(sig_selM_8_a,  sig_outR_11_a,
                          sig_outR_13_a, sig_inM_8_a);
sig_inM_8_b <= inM_8;

-- map the multiplexers at the inputs of registers

sig_inR_1_a <= sig_outM_4_a;
pm_inR_2_a: mux2 port map(sig_selR_2_a, sig_outM_5_a,
                          sig_outM_7_a, sig_inR_2_a);
sig_inR_3_a <= sig_outM_1_a;
sig_inR_5_a <= sig_outM_4_a;
pm_inR_7_a: mux2 port map(sig_selR_7_a, sig_outM_5_a,
                          sig_outM_7_a, sig_inR_7_a);
sig_inR_10_a <= inR_10;
sig_inR_11_a <= sig_outM_4_a;
sig_inR_13_a <= sig_outM_4_a;
pm_inR_16_a: mux2 port map(sig_selR_16_a, sig_outM_4_a,
                           sig_outM_5_a,  sig_inR_16_a);
sig_inR_17_a <= sig_outM_8_a;
pm_inR_18_a: mux2 port map(sig_selR_18_a, sig_outM_4_a,
                           sig_outM_7_a,  sig_inR_18_a);
sig_inR_19_a <= sig_outM_4_a;
sig_inR_20_a <= sig_outM_1_a;

-- map the multiplexers at the scan inputs of shift registers

sig_inS_1_a  <=  SCAN_IN;
sig_inS_2_a  <=  sig_outS_1_a;
sig_inS_3_a  <=  SCAN_IN;
sig_inS_10_a <=  sig_outS_2_a;
sig_inS_11_a <=  sig_outS_10_a;
sig_inS_13_a <=  sig_outS_11_a;
sig_inS_17_a <=  sig_outS_13_a;
sig_inS_18_a <=  SCAN_IN;
sig_inS_20_a <=  sig_outS_17_a;
pm_inS_OUT_a: smux3 port map(sig_selS_OUT_a, sig_outS_3_a,
                             sig_outS_18_a, sig_outS_20_a, SCAN_OUT);

-- map the modules and registers

pm_M_1: add   port map(sig_inM_1_a, sig_inM_1_b, sig_outM_1_a);
pm_M_4: add   port map(sig_inM_4_a, sig_inM_4_b, sig_outM_4_a);
pm_M_5: add   port map(sig_inM_5_a, sig_inM_5_b, sig_outM_5_a);
```

```
        pm_M_7: mul   port map(sig_inM_7_a, sig_inM_7_b, sig_outM_7_a);
        pm_M_8: mul   port map(sig_inM_8_a, sig_inM_8_b, sig_outM_8_a);
        pm_R_1: lfsr  port map(clk,sig_shfR_1_a,sig_genR_1_a,sig_inS_1_a,
                               sig_inR_1_a,sig_outR_1_a,sig_outS_1_a);
        pm_R_2: lfsr  port map(clk,sig_shfR_2_a,sig_genR_2_a,sig_inS_2_a,
                               sig_inR_2_a,sig_outR_2_a,sig_outS_2_a);
        pm_R_3: misr  port map(clk,sig_shfR_3_a,sig_anlR_3_a,sig_inS_3_a,
                               sig_inR_3_a,sig_outR_3_a,sig_outS_3_a);
        pm_R_5: reg   port map(clk,sig_inR_5_a,sig_outR_5_a);
        pm_R_7: reg   port map(clk,sig_inR_7_a,sig_outR_7_a);
        pm_R_10: lfsr port map(clk,sig_shfR_10_a,sig_genR_10_a,sig_inS_10_a,
                               sig_inR_10_a,sig_outR_10_a,sig_outS_10_a);
        pm_R_11: lfsr port map(clk,sig_shfR_11_a,sig_genR_11_a,sig_inS_11_a,
                               sig_inR_11_a,sig_outR_11_a,sig_outS_11_a);
        pm_R_13: lfsr port map(clk,sig_shfR_13_a,sig_genR_13_a,sig_inS_13_a,
                               sig_inR_13_a,sig_outR_13_a,sig_outS_13_a);
        pm_R_16: reg  port map(clk,sig_inR_16_a,sig_outR_16_a);
        pm_R_17: lfsr port map(clk,sig_shfR_17_a,sig_genR_17_a,sig_inS_17_a,
                               sig_inR_17_a,sig_outR_17_a,sig_outS_17_a);
        pm_R_18: misr port map(clk,sig_shfR_18_a,sig_anlR_18_a,sig_inS_18_a,
                               sig_inR_18_a,sig_outR_18_a,sig_outS_18_a);
        pm_R_19: reg  port map(clk,sig_inR_19_a,sig_outR_19_a);
        pm_R_20: lfsr port map(clk,sig_shfR_20_a,sig_genR_20_a,sig_inS_20_a,
                               sig_inR_20_a,sig_outR_20_a,sig_outS_20_a);

        -- map the test compatibility classes onto comparators

        pm_TCC_5_a: comp3 port map(sig_outM_1_a, sig_outM_4_a,
                                   sig_outM_5_a, sig_TCC_5);
        pm_TCC_8_a: comp2 port map(sig_outM_7_a, sig_outM_8_a,
                                  sig_TCC_8);

        -- map the finite state machine

        pm_FSM: work.comp_fsm.FSM port map(clk, reset,sig_selM_1_a,
                                  sig_selM_1_b, sig_selM_4_a, sig_selM_4_b,
                                  sig_selM_5_a, sig_selM_7_a, sig_selM_8_a,
                                  sig_selR_2_a, sig_selR_7_a, sig_selR_16_a,
                                  sig_selR_18_a, sig_selS_OUT_a, sig_shfR_1_a,
                                  sig_genR_1_a, sig_shfR_2_a, sig_genR_2_a,
                                  sig_shfR_3_a, sig_anlR_3_a, sig_shfR_10_a,
                                  sig_genR_10_a, sig_shfR_11_a, sig_genR_11_a,
                                  sig_shfR_13_a, sig_genR_13_a, sig_shfR_17_a,
                                  sig_genR_17_a, sig_shfR_18_a, sig_anlR_18_a,
                                  sig_shfR_20_a, sig_genR_20_a, sig_TCC_5,
                                  sig_TCC_8, sig_FAIL, NORMAL_TEST);

         FAIL   <= sig_FAIL;
         outR_5 <= sig_outR_5_a;

end BEHAVIOUR;
```

Finally, the VHDL descriptions at RTL of multiplexers, comparators, and test registers used in the previous two data path VHDL descriptions are given. Test registers LFSR, MISR, and BILBO are implemented using external XOR implementation [2] based on the following primitive polynomials:

- 4 bit width data path - primitive polynomial x^4+x+1;
- 8 bit width data path - primitive polynomial $x^8+x^4+x^3+x+1$;
- 16 bit width data path - primitive polynomial $x^{16}+x^5+x^3+x+1$;

```
-- type definitions of multiplexer selection signals

type mux3sel is (sel1, sel2, sel3)

-- 3 input multiplexer for data path modules and registers
-- NOTE: n input multiplexers are described similarly by extending the number of inputs

entity mux3 is
   generic (data_width    : natural := WIDTH);
   port (
             sel     : in  mux3sel;
             data_out: out std_logic_vector(data_width - 1 downto 0);
             data_in1, data_in2, data_in3: in std_logic_vector(data_width - 1 downto 0)
        );
end mux3;

architecture rtl of mux3 is
  signal result: std_logic_vector(data_width - 1 downto 0);
begin
   compare: process (sel, data_in1, data_in2, data_in3)
   begin
     case sel is
       when sel1 => result <= data_in1;
       when sel2 => result <= data_in2;
       when sel3 => result <= data_in3;
     end case;
   end process;
   data_out <= result;
end rtl;

-- 3 input multiplexer for scan chains
-- NOTE: n input multiplexers are described similarly by extending the number of inputs

entity smux3 is
   port (
             sel     : in  mux3sel;
```

```
                data_out: out std_logic
                data_in1, data_in2, data_in3: in std_logic;
        );
end smux3;

architecture rtl of smux3 is
  signal result: std_logic;
begin
   compare: process (sel, data_in1, data_in2, data_in3)
   begin
     case sel is
       when sel1 => result <= data_in1;
       when sel2 => result <= data_in2;
       when sel3 => result <= data_in3;
     end case;
   end process;
   data_out <= result;
end rtl;

-- 3 input comparator
-- NOTE: n input comparators are described similarly by extending the number of inputs

entity comp3 is
   generic (data_width    : natural := WIDTH);
   port (
                pass      : out std_logic;
                data_in1, data_in2, data_in3: in std_logic_vector(data_width - 1 downto 0)
        );
end comp3;

architecture rtl of comp3 is
  signal comp: std_logic;
begin
   compare: process (data_in1, data_in2, data_in3)
   begin
     if (data_in1 = data_in2 and data_in1 = data_in3)
     then  comp <= '1'; else  comp <= '0';
     end if;
   end process;
   pass <= comp;
end rtl;

-- LINEAR FEEDBACK SHIFT REGISTER (LFSR)

entity lfsr is
   generic (data_width    : natural := WIDTH);
   port (
                clk, shift, generat, scan_in: in std_logic;
                data_in : in  std_logic_vector(data_width - 1 downto 0);
                data_out: out std_logic_vector(data_width - 1 downto 0);
                scan_out: out std_logic
        );
end lfsr;
```

```
architecture rtl of lfsr is
  signal feedback : std_logic;
  signal lfsr_reg : std_logic_vector(data_width - 1 downto 0);
  begin

--  NOTE: the following feedback equation is for 4 bits data path
--      for 8  bits add lfsr_reg(lfsr_reg'high-4) xor lfsr_reg(lfsr_reg'high-3)
--      for 16 bits add lfsr_reg(lfsr_reg'high-5) xor lfsr_reg(lfsr_reg'high-3)

  feedback <= lfsr_reg(lfsr_reg'high) xor lfsr_reg(lfsr_reg'high-1);
  latch_it: process(clk)
        begin
          if (clk = '1' and clk'event) then
            if (shift = '0') then                -- parallel load
              lfsr_reg <= data_in;
            elsif (generat = '0') then           -- shift register
              lfsr_reg <= lfsr_reg(lfsr_reg'high - 1 downto 0) & scan_in;
            else                                 -- test pattern generator
              lfsr_reg <= lfsr_reg(lfsr_reg'high - 1 downto 0) & feedback;
            end if;
          end if;
        end process ;
   data_out <= lfsr_reg;
   scan_out <= lfsr_reg(lfsr_reg'high);
end rtl;

-- MULTIPLE INPUT SIGNATURE REGISTER (MISR)

entity misr is
   generic (data_width    : natural := WIDTH);
   port (
                clk, shift, analyse, scan_in: in std_logic;
                data_in : in  std_logic_vector(data_width - 1 downto 0);
                data_out: out std_logic_vector(data_width - 1 downto 0);
                scan_out: out std_logic
        );
end misr;

architecture rtl of misr is
  signal feedback : std_logic;
  signal misr_reg : std_logic_vector(data_width - 1 downto 0);
  begin

--  NOTE: the following feedback equation is for 4 bits data path
--      for 8  bits add misr_reg(misr_reg'high-4) xor misr_reg(misr_reg'high-3)
--      for 16 bits add misr_reg(misr_reg'high-5) xor misr_reg(misr_reg'high-3)

  feedback <= misr_reg(misr_reg'high) xor misr_reg(misr_reg'high - 1);
  latch_it: process(clk)
        begin
          if (clk = '1' and clk'event) then
             if (shift = '0') then                -- parallel load
                misr_reg <= data_in;
             elsif (analyse = '0') then           -- shift register
```

```
                misr_reg <= (misr_reg(misr_reg'high - 1 downto 0) & scan_in);
             else                                  -- signature analyser
                misr_reg <= (misr_reg(misr_reg'high - 1 downto 0) & feedback)
                            xor data_in;
             end if;
          end if;
        end process ;
   data_out <= misr_reg;
   scan_out <= misr_reg(misr_reg'high);
end rtl;

-- BUILT IN LOGIC BLOCK OBSERVER (BILBO)

entity bilbo is
   generic (data_width    : natural := WIDTH);
   port (
                clk, shift, generat, analyse, scan_in: in std_logic;
                data_in  : in  std_logic_vector(data_width - 1 downto 0);
                data_out : out std_logic_vector(data_width - 1 downto 0);
                scan_out : out std_logic
        );
end bilbo;

architecture rtl of bilbo is
  signal feedback : std_logic;
  signal bilbo_reg: std_logic_vector(data_width - 1 downto 0);
  begin

--  NOTE: the following feedback equation is for 4 bits data path
--  for 8  bits add bilbo_reg(bilbo_reg'high-4) xor bilbo_reg(bilbo_reg'high-3)
--  for 16 bits add bilbo_reg(bilbo_reg'high-5) xor bilbo_reg(bilbo_reg'high-3)

  feedback <= bilbo_reg(bilbo_reg'high) xor bilbo_reg(bilbo_reg'high - 1);
  latch_it: process(clk)
        begin
          if (clk = '1' and clk'event) then
            if (shift = '0') then                -- parallel load
              bilbo_reg <= data_in;
            elsif (generat = '0') then           -- shift register
              bilbo_reg <= (bilbo_reg(bilbo_reg'high - 1 downto 0) & scan_in);
            elsif (analyse = '0') then           -- test pattern generator
              bilbo_reg <= (bilbo_reg(bilbo_reg'high - 1 downto 0) & feedback);
            else                                 -- signature analyser
              bilbo_reg <= (bilbo_reg(bilbo_reg'high - 1 downto 0) & feedback)
                           xor data_in;
            end if;
          end if;
        end process ;
   data_out <= bilbo_reg;
   scan_out <= bilbo_reg(bilbo_reg'high);
end rtl;
```

References

[1] M.F. Abdulla, C.P. Ravikumar, and A. Kumar. Optimization of mutual and signature testing schemes for highly concurrent systems. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 12(3):199–216, June 1998.

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

[3] V.D. Agrawal. Editorial - special issue on partial scan design. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 7(5):5–6, August 1995.

[4] 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.

[5] 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.

[6] V.D. Agrawal, C.-J. Lin, P.W. Rutkowski, S. Wu, and Y. Zorian. Built-in self-test for digital integrated circuits. *AT&T Technical Journal*, pages 30–39, March 1994.

[7] R.C. Aitken. Nanometer technology effects on fault models for IC testing. *Computer*, 32(11):46–51, November 1999.

[8] M. Alidina, J. Monteiro, S. Devadas, A. Ghosh, and M. Papaefthymiou. Precomputation-based sequential logic optimization for low power. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 2(4):426–436, December 1994.

[9] AMS. *0.35 Micron CMOS Process Parameters*. Austria Mikro Systeme International AG, 1998.

[10] L.J. Avra. *Synthesis Techniques for Built-In Self-Testable Designs*. PhD thesis, Stanford University, June 1994.

[11] R.I. Bahar, H. Cho, G.D. Hachtel, E. Macii, and F. Somenzi. Symbolic timing analysis and resynthesis for low power of combinational circuits containing false paths. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(10):1101–1115, October 1997.

[12] D. Bakalis, H.T. Vergos, D. Nikolos, X. Kavousianos, and G.P Alexiou. Low power dissipation in BIST schemes for modified booth multipliers. In *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 121–129, 1999.

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

[14] A. Basu, T.C. Wilson, D.K. Banerji, and J.C. Majithia. An approach to minimize testability for BILBO based built-in self-test. In *Proc. 5th International Conference on VLSI Design*, pages 354–355, 1992.

[15] 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.

[16] L. Benini, A. Bogliolo, G.A. Paleologo, and G. De Micheli. Policy optimization for dynamic power management. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):813–833, June 1999.

[17] L. Benini and G. De Micheli. State assignment for low power dissipation. *IEEE Journal of Solid-State Circuits*, 30(3):258–268, March 1995.

[18] L. Benini and G. De Micheli. Automatic synthesis of low-power gated-clock finite-state machines. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(6):630–643, June 1996.

[19] L. Benini, G. De Micheli, A. Macii, E. Macii, M. Poncino, and R. Scarsi. Glitch power minimization by selective gate freezing. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):287–298, June 2000.

[20] L. Benini, Polly Siegel, and G. De Micheli. Saving power by synthesizing gated clocks for sequential circuits. *IEEE Design and Test of Computers*, 11(4):32–41, Winter 1994.

[21] D. Berthelot, M.L. Flottes, and B. Rouzeyre. BISTing datapaths under heterogenous test schemes. *Journal of Electronic Testing: Theory and Application (JETTA)*, 14(1):115–123, January 1999.

[22] M. Brazzarola and F. Fummi. Power characterization of LFSRs. In *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 138–146, 1999.

[23] 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.

[24] A.D. Brown, K.R. Baker, and A.J. Williams. On-line testing of statically and dynamically scheduled synthesized systems. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(1):47–57, January 1997.

[25] 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.

[26] S.T. Chakradhar and S. Dey. Resynthesis and retiming for optimum partial scan. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(5):621–630, May 1999.

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

[28] P.R. Chalsani, S. Bhawmik, A. Acharya, and P. Palchaudhuri. Design of testable VLSI circuits with minimum area overhead. *IEEE Transactions on Computers*, 38(9):1460–1462, September 1989.

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

[30] A.P. Chandrakasan, M. Potkonjak, R. Mehra, J. Rabaey, and R.W. Brodersen. Optimizing power using transformations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(1):12–31, January 1995.

[31] S.-C. Chang, K.-J. Lee, Z.-Z. Wu, and W.-B. Jone. Reducing test application time by scan flip-flops sharing. *IEE Proceedings - Computers and Digital Techniques*, 147(1):42–48, January 2000.

[32] C.-S. Chen and T.T. Hwang. Layout driven selection and chaining of partial scan flip flops. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 13(1):19–27, August 1998.

[33] C.I.H. Chen. Graph partitioning for concurrent test scheduling in VLSI circuits. In *Proc. 28th Design Automation Conference (DAC)*, pages 287–290, 1991.

[34] K.-T. Cheng. Gate level test generation for sequential circuits. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(4):405–442, October 1996.

[35] K.-T. Cheng and A. Krstic. Current directions in automatic test-pattern generation. *Computer*, 32(11):58–64, November 1999.

[36] V. Chickername, J. Lee, and J.K. Patel. Addressing design for testability at the architectural level. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(7):920–934, July 1994.

[37] V. Chickername and J.H. Patel. An optmization based approach to the partial scan design problem. In *Proc. International Test Conference*, pages 377–386, 1990.

[38] V. Chickername and J.H. Patel. A fault oriented partial scan design approach. In *Proc. International Conference on Computer Aided Design*, pages 400–403, 1991.

[39] D.G. Chinnery and K. Keutzer. Closing the gap between ASIC and custom: An ASIC perspective. In *Proc. 37th IEEE/ACM Design Automation Conference (DAC)*, pages 637–642, 2000.

[40] S. Chiu and C.A. Papachristou. A desing for testability scheme with applications to data path synthesis. In *Proc. 28th IEEE/ACM Design Automation Conference (DAC)*, pages 271–277, 1991.

[41] 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.

[42] S.-H. Chow, Y.-C. Ho, T. Hwang, and C.L. Liu. Low power realization of finite state machines - a decomposition approach. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 1(3):313–340, July 1996.

[43] F. Corno, P. Prinetto, M. Rebaudengo, and M. Sonza-Reorda. A test pattern generation methodology for low power consumption. In *16th IEEE VLSI Test Symposium*, pages 453–460, 1998.

[44] F. Corno, M. Rebaudengo, M. Sonza Reorda, G. Squillero, and M. Violante. Low power BIST via non-linear hybrid cellular automata. In *18th IEEE VLSI Test Symposium*, pages 29–34, 2000.

[45] F. Corno, M. Rebaudengo, M. Sonza Reorda, and M. Violante. A new BIST architecture for low power circuits. In *IEEE European Test Workshop (ETW99)*, pages 160–164, 1999.

[46] F. Corno, M. Rebaudengo, M. Sonza 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.

[47] 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.

[48] A. L. Crouch. *Design-For-Test For Digital IC's and Embedded Core Systems*. Prentice-Hall, 1999.

[49] 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.

[50] W.J. Dally and A. Chang. The role of custom design in ASIC chips. In *Proc. 37th IEEE/ACM Design Automation Conference (DAC)*, pages 643–647, 2000.

[51] G. de Micheli. *Synthesis and Optimization of Digital Circuits*. McGraw-Hill International Editions, 1994.

[52] A. Dekkers and E. Aarts. Global optimization using simulated annealing. *Mathematical Programming*, 50(1):367–393, 1991.

[53] S. Dey and M. Potkonjak. Nonscan design-for-testability techniques using RT-level design information. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(12):1488–1506, December 1997.

[54] S. Dey, A. Raghunathan, and K.D. Wagner. Design for testability techniques at the behavioural and register-transfer levels. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 13(2):79–91, October 1998.

[55] Exemplar Logic. *Leonardo Spectrum Release Notes*. Exemplar Logic Incorporated, 1999.

[56] 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.

[57] D.D. Gajski. *Principles of Digital Design*. Prentice-Hall International, 1997.

[58] M.R. Garey and D.S. Johnson. *Computers and Intractability: A Guide to the Theory of NP-Completeness*. Morgan Freeman, New York, 1979.

[59] S.H. Gerez. *Algorithms for VLSI Design Automation*. John Wiley & Sons, 1999.

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

[61] S. Gerstendorfer and H.J. Wunderlich. Minimized power consumption for scan-based BIST. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 16(3):203–212, June 2000.

[62] I. Ghosh and N.K. Jha. High-level test synthesis: A survey. *Integration, the VLSI Journal*, 26(1-2):79–99, December 1998.

[63] I. Ghosh, N.K. Jha, and S. Bhawmik. A BIST scheme for RTL circuits based on symbolic testability analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(1):111–128, January 2000.

[64] I. Ghosh, A. Raghunathan, and N.K. Jha. Design for hierarchical testability of RTL circuits obtained by behavioral synthesis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(9):1001–1014, September 1997.

[65] I. Ghosh, A. Raghunathan, and N.K. Jha. A design for testability technique for RTL circuits using control/data flow extraction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(8):706–723, August 1998.

[66] 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.

[67] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch. Circuit partitioning for low power BIST design with minimized peak power consumption. In *8th Asian Test Symposium (ATS99)*, pages 89–94, 1999.

[68] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch. A test vector inhibiting technique for low energy BIST design. In *Proc. 17th IEEE VLSI Test Symposium*, pages 407–412, 1999.

[69] P. Girard, L. Guiller, C. Landrault, and S. Pravossoudovitch. A test vector ordering technique for switching activity reduction during test operation. In *9th Great Lakes Symposium on VLSI (GLSVLSI99)*, pages 24–27, 1999.

[70] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, J. Figueras, S. Manich, P. Teixeira, and M. Santos. Low power pseudo-random BIST: on selecting the LFSR seed. In *Design of Circuits and Integrated Systems Conference (DCIS98)*, pages 166–172, 1998.

[71] P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, J. Figueras, S. Manich, P. Teixeira, and M. Santos. Low-energy BIST design: Impact of the LFSR TPG parameters on the weighted switching activity. In *Proc. International Symposium on Circuits and Systems (ISCAS)*, pages 110–113, 1999.

[72] 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.

[73] P. Girard, C. Landrault, S. Pravossoudovitch, and D. Severac. Reducing power consumption during test application by test vector ordering. In *Proc. International Symposium on Circuits and Systems (ISCAS)*, pages 296–299, 1998.

[74] D. Gizopoulos, N. Kranitis, A. Paschalis, M. Psarakis, and Y. Zorian. Effective low power BIST for datapaths. In *Proc. of the Design, Automation and Test in Europe Conference (DATE)*, page 757, 2000.

[75] D. Gizopoulos, N. Kranitis, A. Paschalis, M. Psarakis, and Y. Zorian. Low power/energy BIST for datapaths. In *Proc. of the 18th IEEE VLSI Test Symposium*, pages 23–28, 2000.

[76] D. Gizopoulos, A. Paschalis, and Y. Zorian. An effective BIST scheme for datapaths. In *Proc. International Test Conference*, pages 76–85, 1996.

[77] D. Gizopoulos, M. Psarakis, A. Paschalis, N. Kranitis, and Y. Zorian. Low power built-in self-test for datapath architectures. In *Proc. of the 2nd International Workshop on Microprocessor Test and Verification (MTV 99)*, pages 23–28, 1999.

[78] F. Glover and M. Laguna. Tabu search. In C.R. Reeves, editor, *Modern Heuristic Techniques for Combinatorial Problems*, pages 70–150. McGraw-Hill Book Company, 1995.

[79] L. Goodby and A. Orailoglu. Redundancy and testability in digital filter datapaths. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(5):631–644, May 1999.

[80] R. Gupta and M.A. Breuer. Partial scan design of register-transfer level circuits. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 7(1):25–46, August 1995.

[81] R. Gupta and Y. Zorian. Introducing core-based system design. *IEEE Design and Test of Computers*, 14(4):15–25, December 1997.

[82] S.K. Gupta and D.K. Pradhan. Utilization of on-line (concurrent) checkers during built-in self-test and vice versa. *IEEE Transactions on Computers*, 45(1):63–73, January 1996.

[83] I. Hamzaoglu and J.H. Patel. New techniques for deterministic test pattern generation. *Journal of Electronic Testing: Theory and Application (JETTA)*, 15(1/2):63–73, August 1999.

[84] I. Hamzaoglu and J.H. Patel. Test set compaction algorithms for combinational circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(8):957–962, August 2000.

[85] I.G. Harris and A. Orailoglu. Microarchitectural synthesis of VLSI designs with high test concurrency. In *Proc. 31st Design Automation Conference*, pages 206–211, 1994.

[86] A. Hertwig and H.J. Wunderlich. Low power serial built-in self-test. In *IEEE European Test Workshop*, pages 49–53, 1998.

[87] M.S. Hsiao, E.M. Rudnick, and J.H. Patel. Effects of delay models on peak power estimation of VLSI sequential circuits. In *Proc. International Conference on Computer Aided Design*, pages 45–51, 1997.

[88] T.-C. Huang and K.-J. Lee. An input control technique for power reduction in scan circuits during test application. In *Proc. 8th Asian Test Symposium*, pages 315–320, 1999.

[89] S.-A. Hwang and C.-W. Wu. Low-power testing for C-testable iterative logic arrays. In *Proceedings International Conference on VLSI Technology, Systems, and Applications*, pages 355–358, 1997.

[90] S. Iman and M. Pedram. An approach for multi-level logic optimization targeting low power. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(8):889–901, August 1996.

[91] Y.-M. Jiang, A. Krstic, and K.-T.Cheng. Estimation for maximum instantaneous current through supply lines for CMOS circuits. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(1):61–73, February 2000.

[92] W.B. Jone, C.A. Papachristou, and M. Pereira. A scheme for overlaying concurrent testing of VLSI circuits. In *Proc. 26th Design Automation Conference*, pages 531–536, 1989.

[93] J.-Y. Jou and M.-C. Nien. Power-oriented partial scan design approach. *IEE Proceedings - Circuits, Devices and Systems*, 145(4):229–235, August 1998.

[94] R. Kapur and T.W. Williams. Tough challenges as design and test go nanometer. *Computer*, 32(11):42–45, November 1999.

[95] C. Kern and M.R. Greenstreet. Formal verification in hardware design: A survey. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 4(2):123–193, April 1999.

[96] K.S. Khouri, G. Lakshminarayana, and N.K. Jha. High-level synthesis of low-power control-flow intensive circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(December):1715–1729, 12 1999.

[97] K.S. Kim and C.R Kime. Partial scan flip-flop selection by use of empirical testability. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 7(5):47–59, August 1995.

[98] S. Kirkpatrick, C.D. Gelatt, and M.P. Vecchi. Optimization by simulated annealing. *Science*, 220(4698):671–680, 1983.

[99] P. Kollig. *Algorithms for Scheduling, Allocation and Binding in High Level Synthesis*. PhD thesis, Staffordshire University, UK, April 1998.

[100] P. Kollig and B.M. Al-Hashimi. A new approach to simultaneous scheduling, allocation and binding in high level synthesis. *IEE Electronics Letters*, 33(18):1516–1518, August 1997.

[101] P. Kollig, B.M. Al-Hashimi, and K.M. Abbott. Efficient scheduling of behavioural descriptions in high-level synthesis. *IEE Proceedings - Computers and Digital Techniques*, 144(2):75–82, March 1997.

[102] B. Kumthekar, L. Benini, E. Macii, and F. Somenzi. Power optimisation of FPGA-based designs without rewiring. *IEE Proceedings - Computers and Digital Techniques*, 147(3):167–174, May 2000.

[103] G. Lakshminarayana and N.K. Jha. FACT: A framework for applying throughput and power optimizing transformations to control-flow-intensive behavioral descriptions. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(11):1577–1594, November 1999.

[104] G. Lakshminarayana and N.K. Jha. High-level synthesis of power-optimized and area-optimized circuits from hierarchical data-flow intensive behaviors. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(3):265–281, March 1999.

[105] G. Lakshminarayana, K.S. Khouri, and N.K. Jha. Wavesched: A novel scheduling technique for control-flow intensive designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(5):505–523, May 1999.

[106] G. Lakshminarayana, A. Raghunathan, and N.K. Jha. Incorporating speculative execution into scheduling of control-flow-intensive designs. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(3):308–324, March 2000.

[107] G. Lakshminarayana, A. Raghunathan, N.K. Jha, and S. Dey. Power management in high level synthesis. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 7(1):7–15, March 1999.

[108] E. Larsson and Z. Peng. An estimation-based technique for test scheduling. In *Electronic Circuits and Systems Conference*, pages 25–28, 1999.

[109] E. Larsson and Z. Peng. A technique for test infrastructure design and test scheduling. In *Design and Diagnostic of Electronic Circuits and Systems Workshop (DDECS 2000)*, 2000.

[110] E. Larsson and Z. Peng. Test infrastructure design and test scheduling optimization. In *IEEE European Test Workshop*, 2000.

[111] H. K. Lee and D. S. Ha. On the generation of test patterns for combinational circuits. Technical Report No. 12-93, Department of Electrical Engineering, Virginia Polytechnic Institute and State University, 1991.

[112] H.K. Lee and D.S. Ha. An efficient forward fault simulation algorithm based on the parallel pattern single fault propagation. In *Proc. International Test Conference*, pages 946–955, 1991.

[113] H.-M. Lin and J.-Y. Jou. On computing the minimum feedback vertex set of a directed graph by contraction operations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(3):295–307, March 2000.

[114] S.P. Lin. *A Design System to Support Built-In Self-Test of VLSI Circuits Using BILBO-Oriented Test Methodologies*. PhD thesis, University of Southern California, May 1994.

[115] S.P. Lin, C.A. Njinda, and M.A. Breuer. Generating a family of testable designs using the BILBO methodology. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 4(2):71–89, 1993.

[116] E. Macii, M. Pedram, and F. Somenzi. High level power modeling, estimation, and optimization. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(11):1061–1079, November 1998.

[117] A. Majumdar and S.B.K. Vrudhula. Fault coverage and test length estimation for random pattern testing. *IEEE Transactions on Computers*, 44(2):234–247, February 1995.

[118] Y. Makris and A. Orailoglu. RTL test justification and propagation analysis for modular designs. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 13(2):105–120, October 1998.

[119] S. Manich, A. Gabarro, M. Lopez, J. Figueras, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, P. Teixeira, and M. Santos. Energy and average power consumption reduction in LFSR based BIST structures. In *Design of Circuits and Integrated Systems Conference (DCIS99)*, pages 651–656, 1999.

[120] 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.

[121] S. Manich, A. Gabarro, M. Lopez, J. Figueras, P. Girard, L. Guiller, C. Landrault, S. Pravossoudovitch, P. Teixeira, and M. Santos. Low power BIST by filtering non-detecting vectors. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 16(3):193–202, June 2000.

[122] W. Mao and M.D. Ciletti. Reducing correlation to improve coverage of delay faults in scan-path design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 13(5):638–646, May 1994.

[123] D. Marculescu, R. Marculescu, and M. Pedram. Theoretical bounds for switching activity analysis in finite-state machines. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(3):335–339, June 2000.

[124] R. Marculescu, D. Marculescu, and M. Pedram. Probabilistic modelling of dependecies during switching activity analysis. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(2):73–83, February 1998.

[125] E.J. Marinissen and Y. Zorian. Challenges in testing cored-based system ICs. *IEEE Communications Magazine*, pages 104–109, June 1999.

[126] B. Mathew and D.G Saab. Combining multiple DFT schemes with test generation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(6):685–696, June 1999.

[127] E.J. McCluskey. Design for testability. In *Logic Design Principles With Emphasis On Testable Semicustom Circuits*, pages 424–488. Prentice Hall, New Jersey, 1986.

[128] M.C. McFarland. Formal verification of sequential hardware: A tutorial. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 12(5):633–654, May 1993.

[129] M.C. McFarland, A.C. Parker, and R. Camposano. The high-level synthesis of digital systems. *Proceedings of the IEEE*, 78(2):301–318, February 1990.

[130] Model Technology. *ModelSim Tutorial*. Model Technology Incorporated, 2000.

[131] J. Monteiro, S. Devadas, and A. Ghosh. Sequential logic optimization for low power using input disabling precomputation architectures. *IEEE Transactions on*

Computer-Aided Design of Integrated Circuits and Systems, 17(3):279–284, March 1998.

[132] V. Muresan, V. Muresan, X. Wang, and M. Vladutiu. The left edge algorithm and the tree growing technique in block-test scheduling under power constraints. In *Proc. of the 18th IEEE VLSI Test Symposium*, pages 417–422, 2000.

[133] V. Muresan, V. Muresan, X. Wang, and M. Vladutiu. The left edge algorithm in block-test scheduling under power constraints. In *Proc. Internation Symposium on Circuits and Systems (ISCAS)*, 2000.

[134] V. Muresan, X. Wang, V. Muresan, and M. Vladutiu. Power-constrained block-test list scheduling. In *11th IEEE International Workshop on Rapid System Prototyping (RSP 2000)*, pages 182–187, 2000.

[135] S. Narayanan and M.A. Breuer. Reconfiguration techniques for a single scan chain. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 14(6):750–765, June 1995.

[136] Z. Navabi. *VHDL: Analysis and Modeling of Digital Systems*. McGraw-Hill, 1997.

[137] Z. Navabi. *Verilog Digital System Design (Professional Engineering)*. McGraw-Hill, 1999.

[138] W.M. Needham. Nanometer technology challenges for test and test equipment. *Computer*, 32(11):52–57, November 1999.

[139] N. Nicolici and B.M. Al-Hashimi. Correction to the proof of theorem 2 in Parallel Signature Analysis Design with Bounds on Aliasing. *IEEE Transactions on Computers*, 47(12):1426, December 1998.

[140] N. Nicolici and B.M. Al-Hashimi. Efficient BIST hardware insertion with low test application time for synthesized data paths. In *Proc. IEEE/ACM Design Automation and Test in Europe (DATE 99)*, pages 289–295, 1999.

[141] N. Nicolici and B.M. Al-Hashimi. Power conscious test synthesis and scheduling for BIST RTL data paths. In *Proc. IEEE International Test Conference (ITC 2000)*, IEEE TTTC Beausang Award - Best Student Paper Award 2000.

[142] N. Nicolici and B.M. Al-Hashimi. Power minimisation techniques for testing low power VLSI circuits. In *Postgraduate Research in Electronics and Photonics (PREP 2000)*, pages 7–12, 2000.

[143] N. Nicolici and B.M. Al-Hashimi. Scan latch partitioning into multiple scan chains for power minimization in full scan sequential circuits. In *Proc. IEEE/ACM Design Automation and Test in Europe (DATE 2000)*, pages 715–722, 2000.

[144] N. Nicolici, B.M. Al-Hashimi, A.D. Brown, and A.C. Williams. BIST hardware synthesis for RTL data paths based on test compatibility classes. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(11), November 2000.

[145] 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.

[146] T.P.K. Nijhar and A.D. Brown. HDL-specific source level behavioural optimization. *IEE Proceedings - Computers and Digital Techniques*, 144(2):128–144, March 1997.

[147] T.P.K. Nijhar and A.D. Brown. Source level optimisation of VHDL for behavioural synthesis. *IEE Proceedings - Computers and Digital Techniques*, 144(1):1–6, January 1997.

[148] R.B. Norwood. *Synthesis-for-Scan: Reducing Scan Overhead with High Level Synthesis*. PhD thesis, Stanford University, November 1997.

[149] K. Olcoz, F. Tirado, and H. Mecha. Unified data path allocation and BIST intrusion. *Integration, the VLSI Journal*, 28(1):55–99, September 1999.

[150] A. Orailoglu and I.G. Harris. Test path generation and test scheduling for self-testable designs. In *Proc. International Conference on Computer Design*, pages 528–531, 1993.

[151] A. Orailoglu and I.G. Harris. Microarchitectural synthesis for rapid BIST testing. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(6):573–586, June 1997.

[152] P. Pan and C.L. Liu. Partial scan with preselected scan signals. *IEEE Transactions on Computers*, 48(9):1000–1005, September 1999.

[153] P.S. Parikh and M. Abramovici. Testability-based partial scan analysis. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 7(5):61–70, August 1995.

[154] S. Park and G. Lee. Partial scan design based on levellised combinational structure. *IEE Proceedings - Computers and Digital Techniques*, 145(4):249–254, July 1998.

[155] I. Parulkar. *Optimization of BIST Resources During High-Level Synthesis*. PhD thesis, University of Southern California, May 1998.

[156] I. Parulkar, S.K. Gupta, and M.A. Breuer. Allocation techniques for reducing BIST area overhead of data paths. *Journal of Electronic Testing: Theory and Applications (JETTA)*, 13(2):149–166, October 1998.

[157] I. Parulkar, S.K. Gupta, and M.A. Breuer. Estimation of BIST resources during high-level synthesis. *Journal of Electronic Testing: Theory and Application (JETTA)*, 13(3):221–237, December 1998.

[158] 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.

[159] J.M. Rabaey. *Digital Integrated Circuits: A Design Perspective*. Prentice-Hall International, 1996.

[160] L. Rade and B. Westergren. *Mathematics Handbook for Science and Engineering*. Springer-Verlag, 4th edition, 1999.

[161] A. Raghunathan and N.K. Jha. SCALP: An iterative improvement based low-power data path synthesis algorithm. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(11):1260–1277, November 1997.

[162] A. Raghunathan, N.K. Jha, and S. Dey. Register transfer level power optimization with emphasis on glitch analysis and reduction. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 18(8):1114–1131, August 1999.

[163] R. Rajsuman. Iddq testing for CMOS VLSI. *Proceedings of the IEEE*, 88(4):544–566, April 2000.

[164] S. Ravi, N.K. Jha, and G. Lakshminarayana. TAO-BIST: A framework for testability analysis and optimization for built-in self-test of RTL circuits. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(8):894–906, August 2000.

[165] 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.

[166] C.P. Ravikumar and N.S. Prasad. Evaluating BIST architectures for low power. In *7th Asian Test Symposium*, pages 430–434, 1998.

[167] C.P. Ravikumar, A. Verma, and G. Chandra. A polynomial-time algorithm for power constrained testing of core based systems. In *8th Asian Test Symposium*, pages 107–112, 1999.

[168] K. Roy and S. Prasad. Circuit activity based CMOS logic synthesis for low power reliable operations. *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 1(4):503–513, December 1993.

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

[170] E.M. Rudnick, J.H. Patel, G.S. Greenstein, and T.M. Niermann. A genetic algorithm framework for test generation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(9):1034–1044, September 1997.

[171] D.G. Saab, Y.G. Saab, and J.A. Abrahamn. Automatic test vector cultivation for sequential VLSI circuits using genetic algorithms. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 15(10):1278–1285, October 1996.

[172] R. Sankaralingam, R.R. Oruganti, and N.A. Touba. Static compaction techniques to control scan vector power dissipation. In *Proc. of the 18th IEEE VLSI Test Symposium*, pages 35–40, 2000.

[173] N.R. Saxena and E.J. McCluskey. Parallel signature analysis design with bounds on aliasing. *IEEE Transactions on Computers*, 46(4):425–438, April 1997.

[174] Semiconductor Industry Association (SIA). *The International Technology Roadmap for Semiconductors (ITRS): 1999 Edition*. http://public.itrs.net/1999_SIA_Roadmap/Home.htm, 1999.

[175] M. Serra, T. Slater, J.C. Muzio, and M. Miller. The analysis of one-dimensional linear cellular automata and their aliasing properties. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 9(7):767–778, July 1990.

[176] A. Shen, A. Ghosh, S. Devadas, and K. Keutzer. On average power dissipation and random pattern testability of CMOS combinational logic networks. In *Proc. IEEE/ACM International Conference on Computer Aided Design*, pages 402–407, 1992.

[177] J.M. Solana and M.A. Manzano. Pase-scan design: A new full-scan structure to reduce test application time. *IEE Proceedings - Computers and Digital Techniques*, 146(6):283–294, November 1999.

[178] G. Theodoridis, S. Theoharis, D. Soudris, and C.E. Goutis. Method for minimising the switching activity of two-level logic circuits. *IEE Proceedings - Computers and Digital Techniques*, 145(5):357–363, September 1998.

[179] V. Tiwari, S. Malik, and P. Ashar. Guarded evaluation: Pushing power management to logic level synthesis/design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(10):1051–1060, October 1998.

[180] H.-C. Tsai, K.-T. Cheng, and S. Bhawmik. On improving test quality of scan-based BIST. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(8):928–938, August 2000.

[181] 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.

[182] K.-H. Tsai, J. Rajski, and M. Marek-Sadowska. Star test: The theory and its applications. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 19(9):1052–1064, September 2000.

[183] J.-M. Tseng and J.-Y. Jon. Two-level logic minimization for low power. *ACM Transactions on Design Automation of Electronic Systems (TODAES)*, 4(1):52–69, January 1999.

[184] C.-Y. Tsui, M. Pedram, and A.M. Despain. Low-power state assignment targeting two- and multilevel logic implementations. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 17(2):1281–1291, December 1998.

[185] A. Vittal and M. Marek-Sadowska. Low-power buffered clock tree design. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(9):965–975, September 1997.

[186] K.D. Wagner. Robust scan-based logic test in VDSM technologies. *Computer*, 32(11):66–74, November 1999.

[187] C.-Y. Wang and K. Roy. Maximization of power dissipation in large CMOS circuits considering spurious transitions. *IEEE Transactions on Circuits and Systems-I: Fundamental Theory and Applications*, 47(4):483–490, April 2000.

[188] S. Wang. *Minimizing Heat Dissipation During Test Application*. PhD thesis, University of Southern California, May 1998.

[189] S. Wang and S.K. Gupta. ATPG for heat dissipation minimization during scan testing. In *Proc. 34th Design Automation Conference (DAC)*, pages 614–619, 1997.

[190] S. Wang and S.K. Gupta. DS-LFSR: A new BIST TPG for low heat dissipation. In *Proc. IEEE International Test Conference*, pages 848–857, 1997.

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

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

[193] A.C. Williams, A.D. Brown, and M. Zwolinski. In-line test of synthesised systems exploiting latency analysis. *IEE Proceedings - Computers and Digital Techniques*, 147(1):33–41, January 2000.

[194] H.-J. Wunderlich. BIST for systems-on-a-chip. *Integration, the VLSI Journal*, 26(1-2):55–78, December 1998.

[195] C. Yeh, C.-C. Chang, and J.-S. Wang. Power-driven technology mapping using pattern-oriented power modelling. *IEE Proceedings - Computers and Digital Techniques*, 146(2):83–89, March 1999.

[196] M. Zellerohr, A. Hertwig, and H.J. Wunderlich. Pattern selection for low-power serial built-in self-test. In *5th IEEE International Test Synthesis Workshop*, 1998.

[197] X. Zhang and K. Roy. Design and synthesis of low power weighted random pattern generator considering peak power reduction. In *International Symposium on Defect and Fault Tolerance in VLSI Systems*, pages 148–156, 1999.

[198] X. Zhang, K. Roy, and S. Bhawmik. POWERTEST: A tool for energy conscious weighted random pattern testing. In *12th International Conference on VLSI Design*, pages 416–422, 1999.

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

[200] Y. Zorian. Testing the monster chip. *IEEE Spectrum*, 36(7):54–60, July 1999.

[201] Y. Zorian, E.J. Marinissen, and S. Dey. Testing embedded core-based system chips. *Computer*, 32(6):52–60, June 1999.

Index