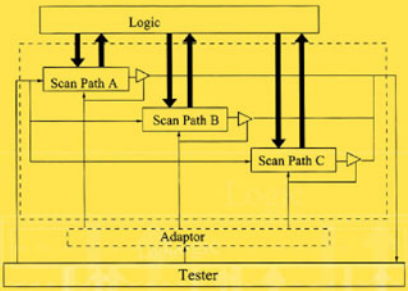
Power-Constrained Testing of VLSI Circuits

by

Nicola Nicolici and Bashir M. Al-Hashimi



Kluwer Academic Publishers

FRONTIERS IN ELECTRONIC TESTING

Consulting Editor

Vishwani D. Agrawal

Books in the series:

High Performance Memory Memory Testing
R. Dean Adams
ISBN: 1-4020-7255-4

SOC (System-on-a-Chip) Testing for Plug and Play Test Automation
K. Chakrabarty
ISBN: 1-4020-7205-8

Test Resource Partitioning for System-on-a-Chip
K. Chakrabarty, Iyengar & Chandra
ISBN: 1-4020-7119-1

A Designers' Guide to Built-in Self-Test
C. Stroud
ISBN: 1-4020-7050-0

Boundary-Scan Interconnect Diagnosis
J. de Sousa, P.Cheung
ISBN: 0-7923-7314-6

Essentials of Electronic Testing for Digital, Memory, and Mixed Signal VLSI Circuits
M.L. Bushnell, V.D. Agrawal
ISBN: 0-7923-7991-8

Analog and Mixed-Signal Boundary-Scan: A Guide to the IEEE 1149.4 Test Standard
A. Osseiran
ISBN: 0-7923-8686-8

Design for At-Speed Test, Diagnosis and Measurement
B. Nadeau-Dosti
ISBN: 0-79-8669-8

Delay Fault Testing for VLSI Circuits
A. Krstic, K-T. Cheng
ISBN: 0-7923-8295-1

Research Perspectives and Case Studies in System Test and Diagnosis
J.W. Sheppard, W.R. Simpson
ISBN: 0-7923-8263-3

Formal Equivalence Checking and Design Debugging
S.-Y. Huang, K.-T. Cheng
ISBN: 0-7923-8184-X

Defect Oriented Testing for CMOS Analog and Digital Circuits
M. Sachdev
ISBN: 0-7923-8083-5

Reasoning in Boolean Networks: Logic Synthesis and Verification Using Testing Techniques
W. Kunz, D. Stoffel
ISBN: 0-7923-9921-8

Introduction to I_{DDQ} Testing
S. Chakravarty, P.J. Thadikaran
ISBN: 0-7923-9945-5

Multi-Chip Module Test Strategies
Y. Zorian
ISBN: 0-7923-9920-X

Testing and Testable Design of High-Density Random-Access Memories
P. Mazumder, K. Chakraborty
ISBN: 0-7923-9782-7

From Contamination to Defects, Faults and Yield Loss
J.B. Khare, W. Maly
ISBN: 0-7923-9714-2

POWER-CONSTRAINED TESTING OF VLSI CIRCUITS

by

NICOLA NICOLICI
McMaster University, Hamilton, Canada

and

BASHIR M. AL-HASHIMI
University of Southampton, U.K.

KLUWER ACADEMIC PUBLISHERS
NEW YORK, BOSTON, DORDRECHT, LONDON, MOSCOW

eBook ISBN: 0-306-48731-4
Print ISBN: 1-4020-7235-X

Created in the United States of America

Visit Springer's eBookstore at: http://www.ebooks.kluweronline.com
and the Springer Global Website Online at: http://www.springeronline.com

Foreword

Increased levels of chip integration combined with physical limitations of heat removal devices, cooling mechanisms and battery capacity, have established energy-efficiency as an important design objective in the implementation flow of modern electronic products. To meet these low energy objectives, new low power techniques, including circuits, architectures, methodologies, algorithms and computer-aided design tool flows, have emerged.

If the integration trend continues in the coming decade, i.e. transistors on lead microprocessors double every two years, die size grows by 14% every two years, supply voltage scales meagerly, and frequency doubles every two years, then what would happen to power and energy? Expected power consumption of such microprocessors, which goes beyond 100watts today, will grow by an order of magnitude every two years reaching 10Kwatts in 2008. It is clear that excessive power usage may become prohibitive and total power consumption will be a limiting factor in the near future. These two factors will become even more critical for lower performance applications, such as in portable products, where low power techniques becomes a necessity. Planning for power need to be incorporated into the design flow of such systems.

Since most of the existing low power techniques aim to reduce the switching activity during the functional operation, they may conflict with the state-of-the-art manufacturing test flow.
In fact, power management is not limited to the design space only, testing chips with high power consumption is a major problem too. For example, a complex chip may consume three or four times higher power during testing when compared to its functional operation. This leads to a reliability problem since overheating can cause destructive test. An additional concern for testing low power circuits is caused by the interaction between the existing design-for-test methods and voltage drop on power/ground networks. Due to high circuit activity when employing scan or built-in self-test, the voltage drop which occurs

only during test will cause some good circuits to fail the testing process, thus leading to unnecessary manufacturing yield loss. Therefore, accounting for power dissipation during test is emerging as a necessary step in the implementation flow, which will ultimately influence both the quality and the cost of test. To keep the pace with the low power design practices, it is essential that the emerging system-on-a-chip test methodologies regard power-constrained testing as an important parameter when establishing the manufacturing test requirements. Today, we have started to see certain embedded test solutions that are designed with infrastructure IP to perform power management on-chip. With the increasing use of embedded cores from third party IP providers, it is expected that power-constrained test solutions be implemented at the cores level by the third party IP providers.

This book is the first comprehensive book that covers all aspects of power-constrained test solutions. It is a reflection of authors' own research and also a survey of the major contributions in this domain. I strongly recommend this book to all engineers involved in design and test of system-on-chip, who want to understand the impact of power on test and design-for-test.

Fremont, November 2002

Dr Yervant Zorian,
Vice President & Chief Scientist,
Virage Logic Corp,
Fremont, California,
U.S.A.

Contents

Preface

Multi-billion transistor chips will be manufactured by the end of this decade, and new design and test methodologies will be emerging to cope with the product complexity. Power dissipation has already become a major design concern and it is turning into a key challenge for the deep submicron digital integrated circuits. Power dissipation concerns cover a large spectrum of products ranging from high performance computing to wireless communication. The continuous growth in power requirements is driven by the ever increasing chip complexity. While smaller devices operate at lower voltages, the currents grow to drive the large number of transistors on a complex circuit and consequently the power dissipation increases. Since devices are sources of heat, the chip temperatures are also increasing. If the temperature increase is excessive (i.e., above heat removal limits) the devices being heated may either permanently degrade or totally fail. Also, due to variations in chip temperatures and since the slew rate (slope) is temperature dependent, differences in delay between different parts of the circuit can occur and they may lead to skew problems that can affect the functionality. Hence, placing more and more functions on a silicon die has resulted in higher power/heat densities, which impose stringent constraints on packaging and thermal management in order to preserve performance and reliability. If the packaging and thermal management parameters (e.g., heat sinks) are determined only based on the normal (functional) operating conditions, then what are the implications of the high circuit activity during test on yield and/or reliability? To answer this question, the factors responsible for power dissipation in digital integrated circuits and the solutions that address the excessive power dissipation during test are discussed in this book.

This page intentionally left blank

Acknowledgments

This book could have not been completed without the help and assistance of many people. The members of the Computer-Aided Design and Test Group at McMaster University and Electronic Systems Design Group in the Department of Electronics and Computer Science of University of Southampton have inspired us with their support. We are thankful for their constant friendship and providing appreciated feedback on several occasions. The authors would like to express sincere gratitude to two very special friends who have contributed generously to this book: Paul Rosinger and Theo Gonciari.

This venture has been initiated with the enthusiastic encouragement from Mark de Jongh of Kluwer Academic Publishers. The editorial assistance and patience of the staff at Kluwer has been invaluable. We are particularly indebted to our editor, Vishwani Agrawal of Agere Systems, who has provided valuable comments and helpful corrections, which have been incorporated with deep appreciation. Finally, we wish to thank our families for understanding the time we needed to complete this project. Without their support this book would have not been possible.

Nicola Nicolici (http://www.ece.mcmaster.ca/~ nicola/)
Bashir M. Al-Hashimi (http://www.ecs.soton.ac.uk/~ bmah/)

This page intentionally left blank

Chapter 1

DESIGN AND TEST OF DIGITAL INTEGRATED CIRCUITS

1.1 Introduction

The topic of this book is power-constrained testing of 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 the manufacturing process for integrated circuits (IC) and the end user's expectations of defect-free chips. Manufacturers test their products to discard the faulty components to ensure that only the defect-free chips make their way to the consumer [1, 15]. With the advent of deep sub-micron technology [69], 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. Since much of the power consumed by the circuit is dissipated as heat, the relationship between the test activity and the cooling capacity need to be taken into consideration in order to avoid destructive test [148]. Also in the long term power limitations will be driven more by system level cooling and test constraints than packaging [70].

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 organized as follows. Section 1.2 overviews the VLSI design flow and outlines the importance of testing integrated circuits. External testing using automatic test equipment (ATE) and the need for design for test (DFT) methods are described in Section 1.3. Section 1.4 introduces built-in self-test (BIST) and provides the terminology used throughout the book with the help of detailed examples. Section 1.5 describes the importance of power minimization during test and Section 1.6 provides an overview of the book.

1.2 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 [69]. While the increase in integration comes with numerous beneficial effects, the perception of the faulty behavior 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 [39].

As shown in Figure 1.1 the design flow of VLSI circuits is divided into three main steps: specification, implementation and manufacturing [113]. *Specification* is the step of describing the functionality of the VLSI circuit. The specification is done in hardware description languages (HDLs), such as VHDL or Verilog [39] in two different design domains, the behavioral domain or the structural domain [39], at various levels of abstraction. For example the *logic level of abstraction* is represented by means of expressions in Boolean algebra in the behavioral 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 present 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 semi-custom [34]. In the full custom design methodology the design is hand-crafted requiring an extensive effort of a design team to optimize each detailed feature of the circuit. In semi-custom 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 behavior into structural implementation, to optimize the resulted netlist, to map the circuit into physical logic gates, and to route the connections between gates.

Manufacturing is the final step of the VLSI design flow and it results in a physical circuit realized 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 characterized 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., 0.13 micron). CMOS technology is the dominant technology for manufacturing VLSI circuits and it is considered throughout this book. Due to significant improve-

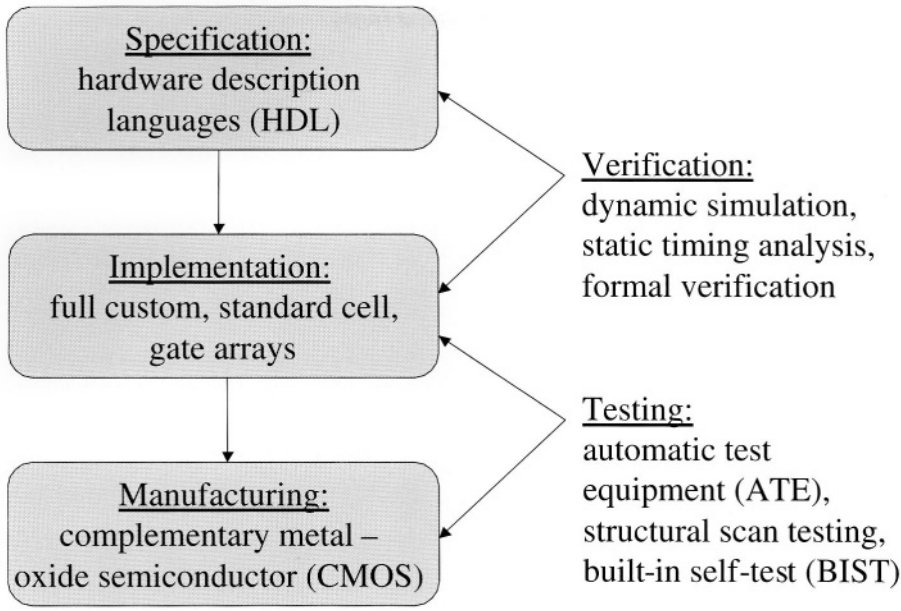


Figure 1.1. VLSI design flow.

ment 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 [72]. 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 [72].

Testing assures that the function of each manufactured circuit corresponds to the function of the implementation [1]. Producing reliable VLSI circuits depends strongly on testing to eliminate various defects caused by the manufacturing process. Basic types of defects in VLSI circuits [99] 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 behavior of the circuit which can be determined either by parametric testing or logic testing [15]. Testing circuits parametrically includes measuring the current flowing through the power supply in the quiescent or static state [15]. As CMOS technology scales down parametric testing is no longer practical due to an increase in sub-threshold leakage current. Logic testing involves modeling manufacturing defects at the logic level of abstraction of the VLSI design flow, where faulty behavior is measured by the logic value of the primary outputs of the circuit [1]. 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 [15]. 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 book 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 ATE or internally using BIST.

1.3 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 (CUT) or device under test (DUT)* is the component which is tested for manufacturing defects; *ATE* including the control processor, timing module, power module, and format module; and *ATE memory* that supplies test patterns and measures test responses. In the following an overview [99] of each of these components is presented.

The *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. Due to the complexity of state-of-the-art circuits many modern CUTs are heterogenous systems-on-a-chip (SOCs) which combine different types of entities such as digital cores, embedded processors, memory macro-cells and analog interfaces.

The *ATE* includes the control processor, timing module, power module, and format module. The control processor is a host computer that controls the

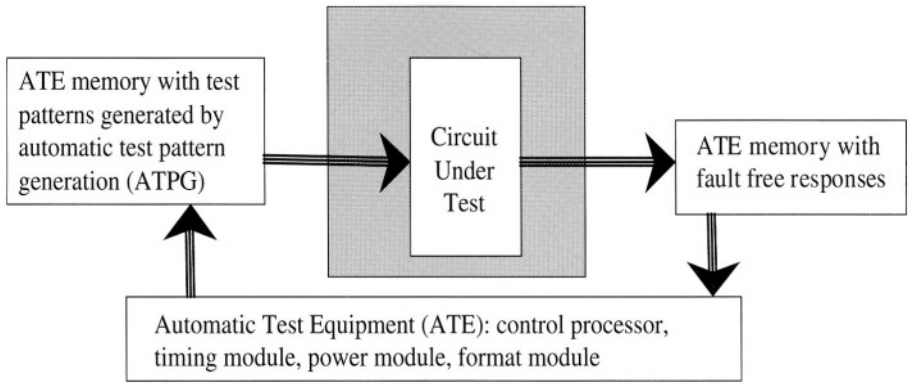


Figure 1.2. Basic principle of external testing using ATE.

flow of the test process and communicates to the other ATE modules whether the CUT is faulty or fault free. The timing module defines the clock edges needed for each pin of the CUT. The format module extends the test pattern information with timing and format information that specifies when the signal to a pin will go high or low, and the 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 [15]. *Test patterns* or *test vectors* stored in the ATE memory are obtained using automatic test pattern generation (ATPG) algorithms [1]. From now onwards throughout this book the terms *test patterns* and *test vectors* are used interchangeably. The number and size of the test patterns/responses which need to be provided to/analyzed from the CUT determine the *volume of test data.* 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 [1]. 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). Compared to random ATPG algorithms, deterministic ATPG algorithms produce shorter and higher quality tests in terms of test efficiency, at the expense of longer computation time. High computation 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 [15] computation 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.

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 modifies 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.* In a scan cycle, the number of clock cycles required to shift in the present state (pseudo input) part of a test vector equals the total number of scan latches, and then the test response is loaded in a single clock 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 (clock cycle) $t + m$ 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 $S_0\ldots S_{m-1}$. 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}. *In the case when ATE channels control directly the primary inputs of the CUT,* then while shifting in the pseudo input value of the test vector the redundant information at primary inputs can be exploited for defining new test application strategies which do not affect test efficiency.

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.*

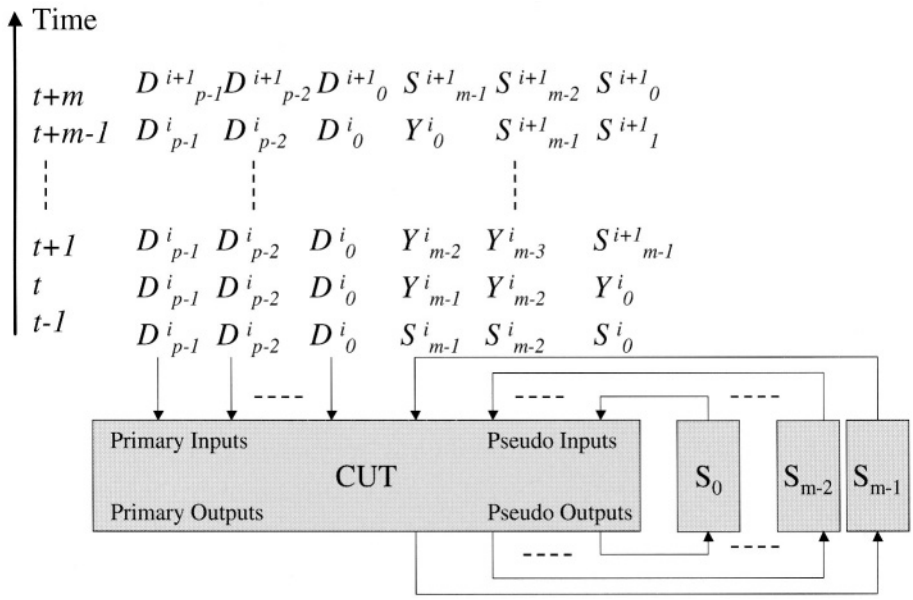


Figure 1.3. Scan based design.

1.4 Internal Testing Using Built-In Self-Test

Despite its benefits of detecting manufacturing defects, external testing using ATE has two problems. Firstly, ATE is extremely expensive and its cost is expected to grow in the future as the number of chip pins increases [15]. 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 difficult.

These problems have led to development of *BIST* [1, 7, 15]. 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), but they are generated *internally* using BIST circuitry. To a large 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 to store the test patterns (Figure 1.2), is eliminated by BIST which employs on chip test pattern generator (TPG) and signature analyzer (SA). When the circuit is in the test mode, TPG generates patterns that