

PREFACE

This book provides a comprehensive and detailed treatment of digital systems testing and testable design. These subjects are increasingly important, as the cost of testing is becoming the major component of the manufacturing cost of a new product. Today, design and test are no longer separate issues. The emphasis on the quality of the shipped products, coupled with the growing complexity of VLSI designs, require testing issues to be considered early in the design process so that the design can be modified to simplify the testing process.

This book was designed for use as a text for graduate students, as a comprehensive reference for researchers, and as a source of information for engineers interested in test technology (chip and system designers, test engineers, CAD developers, etc.). To satisfy the different needs of its intended readership the book (1) covers thoroughly both the fundamental concepts and the latest advances in this rapidly changing field, (2) presents only theoretical material that supports practical applications, (3) provides extensive discussion of testable design techniques, and (4) examines many circuit structures used to realize built-in self-test and self-checking features.

Chapter 1 introduces the main concepts and the basic terminology used in testing. Modeling techniques are the subject of Chapter 2, which discusses functional and structural models for digital circuits and systems. Chapter 3 presents the use of logic simulation as a tool for design verification testing, and describes compiled and event-driven simulation algorithms, delay models, and hardware accelerators for simulation. Chapter 4 deals with representing physical faults by logical faults and explains the concepts of fault detection, redundancy, and the fault relations of equivalence and dominance. The most important fault model — the single stuck-fault model — is analyzed in detail. Chapter 5 examines fault simulation methods, starting with general techniques — serial, parallel, deductive, and concurrent — and continuing with techniques specialized for combinational circuits — parallel-pattern single-fault propagation and critical path tracing. Finally, it considers approximate methods such as fault sampling and statistical fault analysis.

Chapter 6 addresses the problem of test generation for single stuck faults. It first introduces general concepts common to most test generation algorithms, such as implication, sensitization, justification, decision tree, implicit enumeration, and backtracking. Then it discusses in detail several algorithms — the *D*-algorithm, the 9V-algorithm, PODEM, FAN, and critical path test generation — and some of the techniques used in TOPS, SOCRATES, RAPS, SMART, FAST, and the subscripted *D*-algorithm. Other topics include random test generation, test generation for sequential circuits, test generation using high-level models, and test generation systems.

Chapter 7 looks at bridging faults caused by shorts between normally unconnected signal lines. Although bridging faults are a "nonclassical" fault model, they are dealt with by simple extensions of the techniques used for single stuck faults. Chapter 8 is concerned with functional testing and describes heuristic methods, techniques using binary decision diagrams, exhaustive and pseudoexhaustive testing, and testing methods for microprocessors.

Chapter 9 presents design for testability techniques aimed at simplifying testing by modifying a design to improve the controllability and observability of its internal signals. The techniques analyzed are general ad hoc techniques, scan design, board and system-level approaches, partial scan and boundary scan (including the proposed JTAG/IEEE 1149.1 standard).

Chapter 10 is dedicated to compression techniques, which consider a compressed representation of the response of the circuit under test. The techniques examined are ones counting, transition counting, parity checking, syndrome checking, and signature analysis. Because of its widespread use, signature analysis is discussed in detail. The main application of compression techniques is in circuits featuring built-in self-test, where both the generation of input test patterns and the compression of the output response are done by circuitry embedded in the circuit under test. Chapter 11 analyzes many built-in self-test design techniques (CSBL, BEST, RTS, LOCST, STUMPS, CBIST, CEBS, RTD, SST, CATS, CSTP, and BILBO) and discusses several advanced concepts such as test schedules and partial intrusion built-in self-test.

Chapter 12 discusses logic-level diagnosis. The covered topics include the basic concepts in fault location, fault dictionaries, guided-probe testing, expert systems for diagnosis, effect-cause analysis, and a reasoning method using artificial intelligence concepts.

Chapter 13 presents self-checking circuits where faults are detected by a subcircuit called a checker. Self-checking circuits rely on the use of coded inputs. Some basic concepts of coding theory are first reviewed, followed by a discussion of specific codes — parity-check codes, Berger codes, and residue codes — and of designs of checkers for these codes.

Chapter 14 surveys the testing of programmable logic arrays (PLAs). First it reviews the fault models specific to PLAs and test generation methods for external testing of these faults. Then it describes and compares many built-in self-test design methods for PLAs.

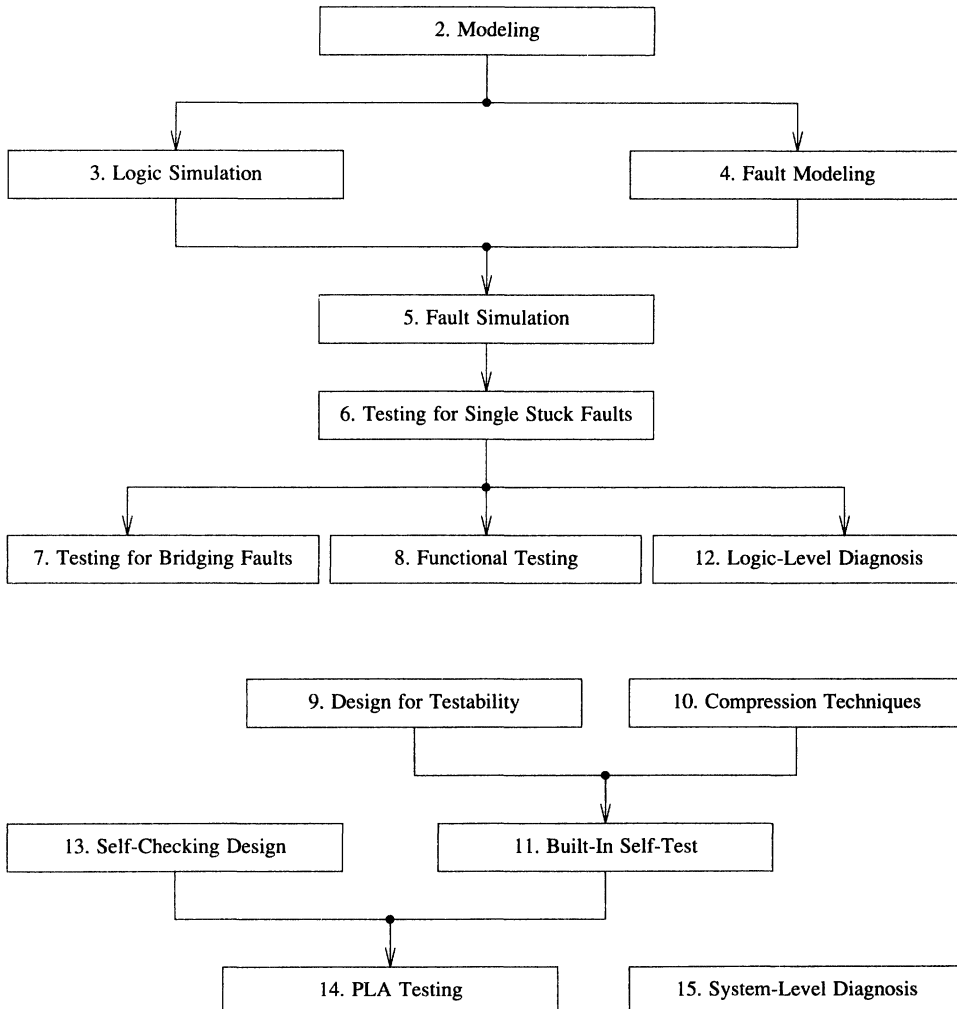
Chapter 15 deals with the problem of testing and diagnosis of a system composed of several independent processing elements (units), where one unit can test and diagnose other units. The focus is on the relation between the structure of the system and the levels of diagnosability that can be achieved.

In the Classroom

This book is designed as a text for graduate students in computer engineering, electrical engineering, or computer science. The book is self-contained, most topics being covered extensively, from fundamental concepts to advanced techniques. We assume that the students have had basic courses in logic design, computer science, and probability theory. Most algorithms are presented in the form of pseudocode in an easily understood format.

The progression of topics follows a logical sequence where most chapters rely on material presented in preceding chapters. The most important precedence relations among chapters are illustrated in the following diagram. For example, fault simulation (5) requires understanding of logic simulation (3) and fault modeling (4). Design for

testability (9) and compression techniques (10) are prerequisites for built-in self-test (11).



Precedence relations among chapters

The book requires a two-semester sequence, and even then some material may have to be glossed over. For a one-semester course, we suggest a "skinny path" through Chapters 1 through 6 and 9 through 11. The instructor can hope to cover only about half of this material. This "Introduction to Testing" course should emphasize the fundamental concepts, algorithms, and design techniques, and make only occasional forays into the more advanced topics. Among the subjects that could be skipped or only briefly discussed in the introductory course are Simulation Engines (Section 3.11),

The Multiple Stuck-Fault Model (4.6), Fault Sampling (5.4), Statistical Fault Analysis (5.5), Random Test Generation (6.2.3), Advanced Scan Techniques (9.9), and Advanced BIST Concepts (11.5). Most of the material in Chapter 2 and the ad hoc design for testability techniques (9.2) can be given as reading assignments.

Most of the topics not included in the introductory course can be covered in a second semester "Advanced Testing" course.

Acknowledgments

We have received considerable help in developing this book. We want to acknowledge Xi-An Zhu, who contributed to portions of the chapter on PLA testing. We are grateful for the support provided by the Bell Labs managers who made this work possible — John Bierbauer, Bill Evans, Al Fulton, Hao Nham, and Bob Taylor. Special thanks go to the Bell Labs word processing staff — Yvonne Anderson, Deborah Angell, Genevieve Przeor, and Lillian Pilz — for their superb job in producing this book, to David Hong for helping with troff and related topics, to Jim Coplien for providing the indexing software, and to John Pautler and Tim Norris for helping with the phototypesetter. We want to thank our many colleagues who have been using preprints of this book for several years and who have given us invaluable feedback, and especially S. Reddy, S. Seth, and G. Silberman. And finally, many thanks to our students at the University of Southern California and the Illinois Institute of Technology, who helped in "debugging" the preliminary versions.

Miron Abramovici
Melvin A. Breuer
Arthur D. Friedman