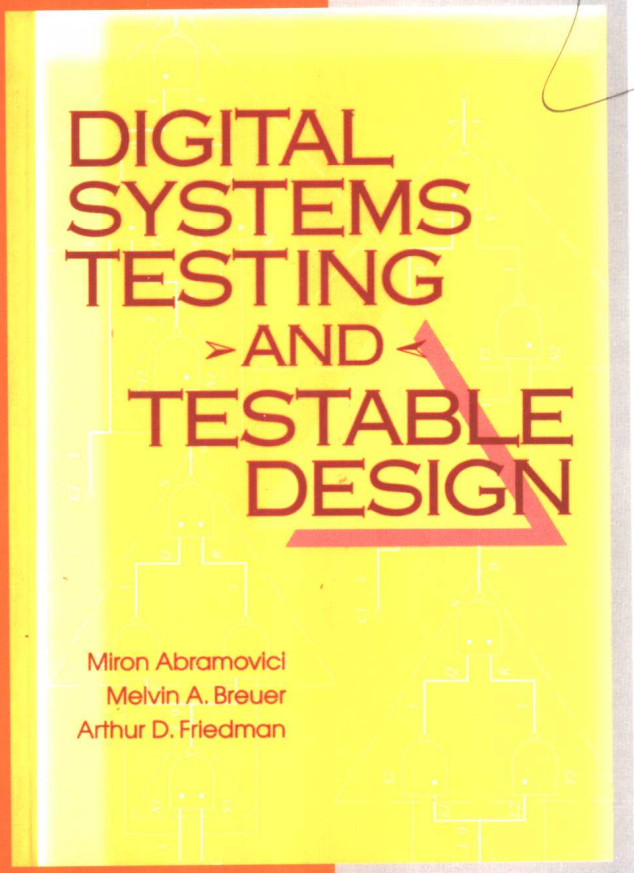
国外大学优秀教材 —— 微电子类系列（影印版）

Miron Abramovici
Melvin A. Breuer
Arthur D. Friedman

数字系统测试和可测性设计



清华大学出版社

国外大学优秀教材 —— 微电子类系列（影印版）

数字系统测试和可测性设计

Digital Systems Testing and Testable Design

Miron Abramovici
Melvin A. Breuer
Arthur D. Friedman

清华大学出版社
北京

Miron Abramovici, Melvin A. Breuer, Arthur D. Friedman
Digital Systems Testing and Testable Design (Revised Printing)

北京市版权局著作权合同登记号　图字：01-2003-8066

本书封面贴有清华大学出版社激光防伪标签，无标签者不得销售。

图书在版编目(CIP)数据

数字系统测试和可测性设计 = Digital Systems Testing and Testable Design / 阿布拉莫维奇(Abramovici, M.)，布鲁尔（Breuer, M. A.），弗里德曼（Friedman, A. D.）著. —影印本. —北京：清华大学出版社，2003
（国外大学优秀教材——微电子类系列（影印版））
ISBN 7-302-07747-9

Ⅰ. 数…　Ⅱ. ①阿…　②布…　③弗…　Ⅲ. ①数字系统－测试－高等学校－教材－英文　②数字系统－系统设计－高等学校－教材－英文　Ⅳ. TP271

中国版本图书馆 CIP 数据核字（2003）第 112305 号

出 版 者：清华大学出版社
http://www.tup.com.cn
社总机：(010) 6277 0175
责任编辑：邹开颜
印 刷 者：北京鑫霸印务有限公司
装 订 者：三河市新茂装订有限公司
发 行 者：新华书店总店北京发行所
开　　本：185×260　**印张**：42.25
版　　次：2004 年 1 月第 1 版　2004 年 1 月第 1 次印刷
书　　号：ISBN 7-302-07747-9/TN • 161
印　　数：1~3000
定　　价：65.00 元

地　　址：北京清华大学学研大厦
邮　　编：100084
客户服务：(010) 6277 6969

出 版 前 言

微电子技术是信息科学技术的核心技术之一，微电子产业是当代高新技术产业群的核心和维护国家主权、保障国家安全的战略性产业。我国在《信息产业“十五”计划纲要》中明确提出：坚持自主发展，增强创新能力和核心竞争力，掌握以集成电路和软件技术为重点的信息产业的核心技术，提高具有自主知识产权产品的比重。发展集成电路技术的关键之一是培养具有国际竞争力的专业人才。

微电子技术发展迅速，内容更新快，而我国微电子专业图书数量少，且内容和体系不能反映科技发展的水平，不能满足培养人才的需求，为此，我们系统挑选了一批国外经典教材和前沿著作，组织分批出版。图书选择的几个基本原则是：在本领域内广泛采用，有很大影响力；内容反映科技的最新发展，所述内容是本领域的研究热点；编写和体系与国内现有图书差别较大，能对我国微电子教育改革有所启示。本套丛书还侧重于微电子技术的实用性，选取了一批集成电路设计方面的工程技术用书，使读者能方便地应用于实践。本套丛书不仅能作为相关课程的教科书和教学参考书，也可作为工程技术人员的自学读物。

我们真诚地希望，这套丛书能给国内高校师生、工程技术人员以及科研人员的学习和工作有所帮助，对推动我国集成电路的发展有所促进。也衷心期望着广大读者对我们一如既往的关怀和支持，鼓励我们出版更多、更好的图书。

清华大学出版社
理工出版事业部
2003.9

Digital Systems Testing and Testable Design

影 印 版 序

集成电路发展到今天的 SoC 时代，遇到最为棘手的问题是 SoC 片的可测性问题和测试方法问题。SoC 测试（testing for system on chip）又称微系统芯片测试。它是制造 SoC 的一部分，以确信物理器件从行为定义、行为和逻辑综合、物理综合到工艺制造都毫无缺陷；微系统芯片测试或 IP 模块测试是对集成起来的电路或模块进行检测，它是通过测量以及对于具体的微系统芯片或集成的 IP 模块施加测试激励，采集其输出响应和预期输出进行比较，确定或评估产品功能和性能的过程，是验证设计、监控生产 、保证质量、分析失效以及指导应用的重要手段。

SoC 测试的完成要实现两个目标。其一是完成微系统芯片质量的评估，得到合格芯片以保证应用成功；其二是在芯片出现失效或错误时，分析失效，诊断和定位错误，改进设计，推动新的合格微系统芯片尽快产生。

根据现有的数字系统可测性设计理论和度量方法，数字系统的可控制性和可观测性是与系统的电路结构和数据传输路径的长度有关，而它的测试复杂度（测试矢量长度和宽度、以及所能达到的测试出故障覆盖率）与系统内部存在的环路长度和数量成正比。系统内部环路长度越长测试复杂度越高，系统内部环路数量越多测试复杂度和难度也越大，并且它们间的关系是指数性增长关系。

如何研究和改善微系统芯片的测试效率和可测性设计的效能，必须研究和精通数字系统可测性设计理论和度量方法。一般而言，人们对于一个数字电路测试对象总是希望进行全面的功能测试。随着集成电路的规模和单位功能密度的指数增加，要想实现全面的功能测试的希望有可能落空。以一个 64 位加法器为例，它要对于两个 64 位数进行相加，就需要 128 个输入和 1 个低位进位输入。为了保证不犯例如奔腾处理器除法设计的错误，有较好的功能测试的完备性，人们势必需要穷举输入 2^{129} 个矢量以确定芯片功能的正确性。但人们忽略了一个基本事实，这么多的测试矢量所需的测试时间是多少？我们用一台测试数据速率为 1GHz 的集成电路自动系统（ATE）来测试该芯片，粗略计算一下大约需要 2.15×10^{22} 年。这是一个人穷一生努力也不能实现的一个天文数字。而应用结构测试方法用同样机器去测试仅要 0.000001728 秒。多么鲜明的对比呀！人们自然将希望、将目光转向科学家所提出的理论——结构测试的一系列理论和方法。

Digital Systems Testing and Testable Design 一书，是全美本科和研究生优秀教材，比较系统地介绍了结构测试的理论和方法、可测性设计理论和度量方法、测试数据的处理及简化的理论和方法以及智能芯片（处理器、数字信号处理器和自动机等）测试理论和方法等。该书共有 15 章，分为 3 部分。前 8 章为第一部分，主要介绍数字系统、数字微系统芯片缺陷的来源、逻辑描述的方法——故障的建模、故障模拟、测试单固定故障、测试桥接故障、智能数字系统的功能测试及其范围等；第 9 章～第 14 章是第二部分，主要介绍数字系统的可测性设计理论和方法、建内自测试（built-in self test，BIST）测试数据压缩方法等现代测试理论和方法；第 15 章是第三部分，主要讨论系统测试的方法，应该说这部分内容讲得少

了些，对于发展微系统芯片测试而言，需要发展和补充诸如内容：SoC 的测试结构、嵌入内核的测试访问机制和标准、测试资源的优化及利用和测试内核的标准语言、以及混合结构 SoC 的电路测试方法等。但是这些丝毫没有掩盖本书全美大学用了近 20 年的优秀教材本色。全书主要贡献在于系统全面总结了故障模型的描述方法，数字组合逻辑电路和时序逻辑电路结构测试中测试生成算法，可测性设计（design for testability，DFT）基本理论和具体方法，专用的 BIST 结构和 PLA 的测试方法等，这些都为今天每个学习和应用者提供很好实践的思路。该书概念清晰层次分明、定义和证明准确、算法推导和阐述简练。每章附有大量练习题可帮助读者对于概念的消化吸收。

引进国际先进的教材可以帮助我们尽快实现大学和研究生教育与国际接轨，更好借鉴国外办学的经验，缩短和国外大学的差距，发展我们自己的高水平高等教育和研究生教育，培养更优秀的电子信息科学的人才。故作此序，和读者共勉。

孙义和于清华大学

2003 年 11 月

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.

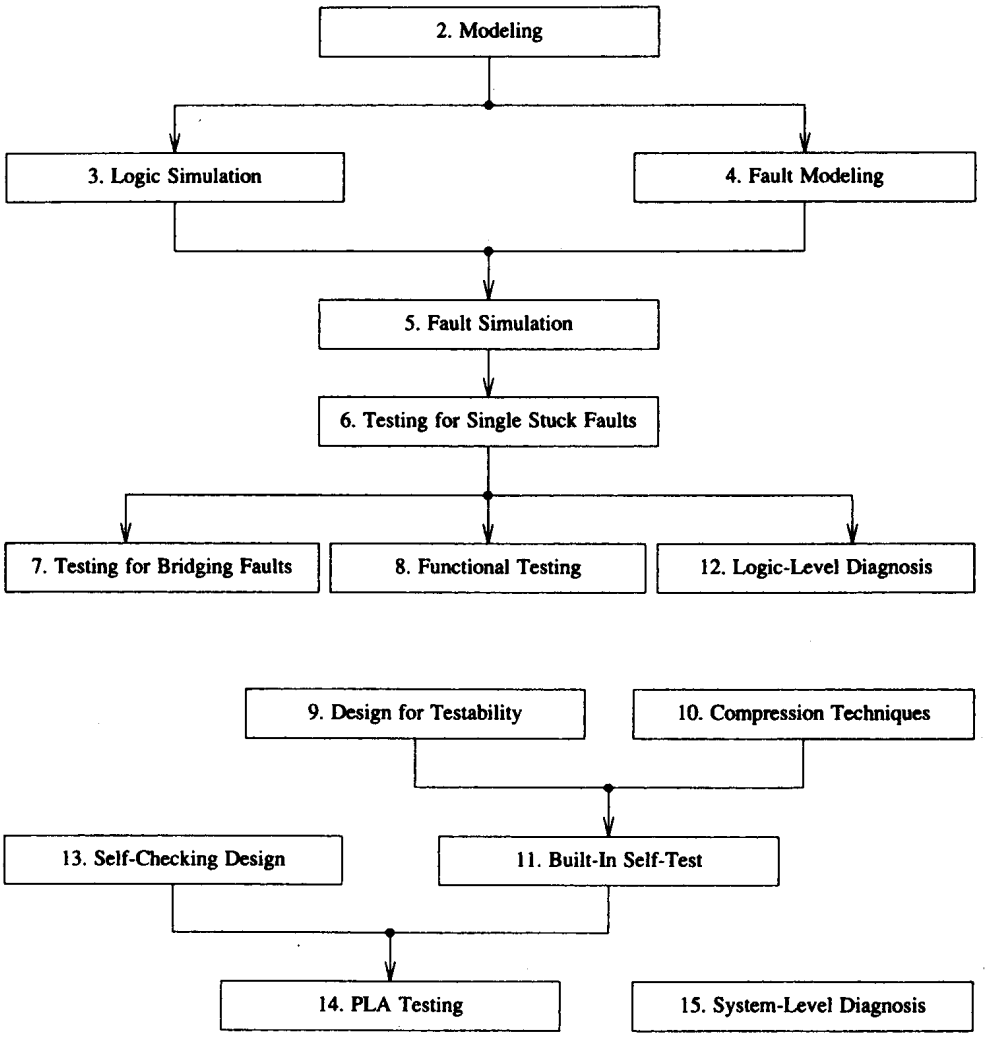
Chapter 15 deals with the problem of testing and diagnosis of a system composed of several independent processing elements (units), where one unit can test and diagnose other units. The focus is on the relation between the structure of the system and the levels of diagnosability that can be achieved.

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

How This Book Was Written

Don't worry. We will not begin by saying that "because of the rapid increases in the complexity of VLSI circuitry, the issues of testing, design-for-test and built-in-self-test, are becoming increasingly more important." You have seen this type of opening a million times before. Instead, we will tell you a little of the background of this book.

The story started at the end of 1981. Miron, a young energetic researcher (at that time), noted that Breuer and Friedman's *Diagnosis & Reliable Design of Digital Systems* — known as the "yellow book" — was quickly becoming obsolete because of the rapid development of new techniques in testing. He suggested co-authoring a new book, using as much material from the yellow book as possible and updating it where necessary. He would do most of the writing, which Mel and Art would edit. It all sounded simple enough, and work began in early 1982.

Two years later, Miron had written less than one-third of the book. Most of the work turned out to be new writing rather than updating the yellow book. The subjects of modeling, simulation, fault modeling, fault simulation, and test generation were reorganized and greatly expanded, each being treated in a separate chapter. The end, however, was nowhere in sight. Late one night Miron, in a state of panic and frustration, and Mel, in a state of weakness, devised a new course of action. Miron would finalize the above topics and add new chapters on bridging faults testing, functional testing, logic-level diagnosis, delay-faults testing, and RAM testing. Mel would write the chapters dealing with new material, namely, PLA testing, MOS circuit testing, design for testability, compression techniques, and built-in self-test. And Art would update the material on self-checking circuits and system-level diagnosis.

The years went by, and so did the deadlines. Only Art completed his chapters on time. The book started to look like an encyclopedia, with the chapter on MOS testing growing into a book in itself. Trying to keep the material up to date was a continuous and endless struggle. As each year passed we came to dread the publication of another proceedings of the DAC, ITC, FTCS, or ICCAD, since we knew it would force us to go back and update many of the chapters that we had considered done.

Finally we acknowledged that our plan wasn't working and adopted a new course of action. Mel would set aside the MOS chapter and would concentrate on other, more essential chapters, leaving MOS for a future edition. Miron's chapters on delay fault testing and RAM testing would have the same fate. A new final completion date was set for January 1989.

This plan worked, though we missed our deadline by some 10 months. Out of love for our work and our profession, we have finally accomplished what we had set out to do. As this preface was being written, Miron called Mel to tell him about a paper he just read with some nice results on test generation. Yes, the book is obsolete already. If you are a young, energetic researcher — don't call us.

国外大学优秀教材——微电子类系列（影印）

- Semiconductor Physics and Devices (3rd ed.) *Donald A. Neamen*
- RF Microelectronics *Behzad Razavi*
- The Art of Analog Layout *Alan Hastings*
- Digital Systems Testing and Testable Design *Miron Abramovici*
- Complete Digital Design A Comprehensive Guide to Digital Electronics and Computer System Architecture *Mark Balch*
- Phase-Locked Loops Design, Simulation, and Applications(5th ed.) *Roland E. Best*
- IC Mask Design *Christopher Saint*
- Digital Integrated Circuits: A Design Perspective(2nd ed.) *Jan M. Rabaey*

CONTENTS