版 风 暴 系 列

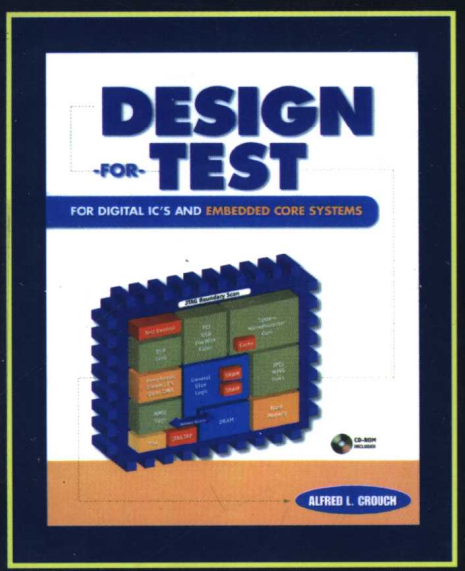
PEARSON Prentice Hall

Design-For-Test
For Digital IC's and Embedded Core Systems

数字集成电路与嵌入式内核系统可测试性设计

（影印版）

[美] Alfred L. Crouch 著



- 第一本关于可测试性设计的权威指南
- 设计和测试相关工程师及管理人员必备
- 完整教学资料帮助读者更快更简便学习

Pearson Education

中国电力出版社
www.infopower.com.cn

原 版 风 暴 系 列

Design-For-Test
For Digital IC's and Embedded Core Systems

数字集成电路与嵌入式内核系统可测试性设计

（影印版）

[美] Alfred L. Crouch 著

中国电力出版社

Design-for-test for digital IC's and embedded core systems
(ISBN 0-13-084827-1)

图书在版编目（CIP）数据
数字集成电路与嵌入式内核系统可测试性设计 /（美）阿尔佛雷德著. —影印本. —北京：中国电力出版社，2004
（原版风暴系列）
ISBN 7-5083-1904-4
Ⅰ.数...　Ⅱ.阿...　Ⅲ.①数字集成电路－测试－设计－英文②微计算机系统－测试－设计－英文　Ⅳ.①TN431.2②TP368
中国版本图书馆 CIP 数据核字（2003）第 108155 号

丛 书 名： 原版风暴系列
书　　名： 数字集成电路与嵌入式内核系统可测试性设计（影印版）
编　　著： （美）Alfred L. Crouch
责任编辑： 朱恩从
出版发行： 中国电力出版社
地址：北京市三里河路6号　邮政编码：100044
电话：（010）88515918　传真：（010）88518169
印　　刷： 北京丰源印刷厂
开　　本： 787×1092 1/16　　**印　　张：** 23.5
书　　号： ISBN 7-5083-1904-4
版　　次： 2004年1月北京第1版　　2004年1月第1次印刷
定　　价： 48.00 元

Design-for-Test for Digital IC's and Embedded Core Systems

Alfred L. Crouch

Preface

This book is made primarily for design engineers and managers, and for test and design-for-test engineers and managers. It can also be used for students of digital design and test, as well. The purpose of this book is to introduce the basic concepts of test and design-for-test (DFT), and to then address the application of these concepts with an eye toward the trade-offs of the engineering budgets (silicon area, operating frequency target, power consumption, etc.), the business drivers, and the cost factors.

Currently, some very good test and DFT texts are available. However, many of them are from an academic focus. In my years of being part of the integrated circuit design community, I have had to train many IC designers and junior DFT engineers in test and design-for-test. I have discovered that corporate education is remarkably different from academic education. A design engineer on a project, who must learn and apply DFT techniques, is learning them while being responsible for 60+ hours of other design tasks per week and while meeting regular design deadlines and milestones. In this environment, learning the DFT tasks and techniques is difficult with a book that focuses on the "mathematical" or "theoretical" point of view. History has taught me that a direct "how to do it" text is more effective.

Another interesting aspect of the competitive corporate environment is that the design process may be "factory-ized." The overall design process for a chip or a portion of a chip is no longer the responsibility of the design engineer, but of teams of chip design functions. For example, the logic gate cells may be designed and characterized by one group (standard cell and library development), and the design may be modeled and synthesized by a different group (HDL design and synthesis), verified by yet another group (formal and simulation verification), and ultimately, mapped to a physical process by yet another group (floorplanning, place&route, and physical design). In this case, the teaching of DFT techniques must be spread out to the various organizations contributing to the overall design. A teaching description of a DFT technique, such as scan design, is not effective if it is not related to the tasks, scheduling, trade-offs, and the

separations into the various organizational elements. Again, history and experience have taught me that an effective text here is one that relates the topic being addressed to the design methodology and design flow.

So direct experience in corporate technical training and teaching has led to the style and content of this "practical guide" on the test and Design-for-Test (DFT) topics of scan test, embedded memory test, and embedded core test. This text has been developed more along the lines of a "just what you need to know—and how to do it" guide that explains the topic, explains the trade-offs, and relates the topic to the design flow. My hope is that using this text will reduce the "learning curve" involved in the application of test and design-for-test techniques, and will result in designs that have a higher quality-reliability level and a lower cost-of-test.

A practical text on DFT and DFT techniques, based on the industry point of view, is needed right now for several reasons. First, the "cost of test" is beginning to dominate the recurring (per-part) cost involved in the manufacturing of the final silicon product for many of the consumer markets—parts with high volume and a low selling price. Second, shorter product lifetimes and increased time-to-market (TTM) and time-to-volume (TTV) pressures are forcing the need to have some form of structured, repeatable and automatable test features included in the device as part of the overall design methodology. Third, the move to reuse cores, and core-based design, as a reaction to shrinking process geometries and TTM pressures, is also forcing designed-in test features to become portable since design units may be distributed and reused in several different chip designs with completely different configurations. And finally, the shrinking process geometries also enable "system-on-a-chip" and ULSI (Ultra-Large Scale Integrated) designs with massive integration—more integration means more faults and more vectors—which leads to a test data management and cost-of-test problems.

Taken all together, these changes in parts of the semiconductor design industry are changing the way test and DFT are viewed, addressed, and implemented. Organizations that once ignored DFT are now being dragged kicking and screaming into modern age because of test cost, TTM, TTV, test data volume, and having to deal with the test features delivered with commercially available cores. Test is one of the three major components of recurring per-part cost involved with the manufacture and selling of digital semiconductor integrated circuits (with the cost of silicon die and the cost of packaging being the other two). As with every product, trade-offs are made to achieve a quality level and to fit within a target cost profile. I hope that this text will eliminate the view that understanding the cost-of-test and applying DFT during the design phase of a product is a "black art" for organizations and individuals that must address managing the cost factors of a chip design.

If you have questions or comments, I can be contacted at *Al_Crouch@prodigy.net.*

Acknowledgments

A book like this could never have been completed without the support of many people. I would like to thank a couple of my early mentors for setting me on the course to becoming the design-for-test professional that I turned out to be, Greg Young and Andy Halliday. The knowledge and data gathered for many of the test techniques and trade-offs described in this book would not have been possible without the work being accomplished to implement these techniques in real chips. To this end, I would like to recognize the hard work, support, and commitment of the Motorola ColdFire® DFT team, and other DFT professionals that I work with and have worked with at Motorola: Renny Eisele, Teresa McLaurin, John Potter, Michael Mateja, Dat Tran, Jennifer McKeown, Jim Johnson, Matthew Pressley, Clark Shepard, Bill Underwood, and Jason Doege. I would like to thank Janusz Rajski and Pat Scoggin of Mentor Graphics for convincing me that this information would make a good book (and that I could write it). And I would also like to thank Grady Giles for all his years of invaluable guidance and assistance. I would also like to thank my reviewers, Elizabeth Rudnick of the University of Illinois, and Nur Touba of the University of Texas.

The person I need to thank the most, however, is my wife, Kelly, for allowing me to work my "second" job of book writing for so long.

For my sister, Monika,
who works harder just
staying alive every day
than I can ever hope
to match.

Introduction

There is a lot of diverse information in this book, and although I would love to have everyone voraciously devour the book from the front cover to the back cover, many readers will be interested in reading only the pieces and parts that pertain to their needs. Since the practical application of design-for-test is a very interleaved and interrelated subject, none of the test and design-for-test techniques can exist in a total vacuum without some consideration for the other related test topics. For example, the application of a scan test architecture to a simple device with logic and memory, requires that some consideration be given to the interaction of the scan architecture with the memory array and the memory test architecture. So the study of scan will also require the study of some memory techniques. I have resisted the temptation to repeat information in several places, I have created a guide of sorts to the use of the book. This means that an individual who is interested only in applying scan test techniques to a chip-level design does need to read information in several sections to complete a course of study on the topic of scan. I will attempt to provide that type of outline here.

Also, I wish to apologize right here for writing in much the same manner as I speak. I have a habit of making my point by using slang and jargon. Since this book is based on the "corporate" or "business" point of view, the language I use is the language I learned at work, which may not line up exactly with the prescribed academic terms. To be completely honest, I learned most of what is in this book by the seat of my pants, applying DFT techniques to designs while my hair was on fire from schedule pressure—I did not spend a lot of time reading academic texts to learn my art (or science, as the case may be).

To begin with, the book contains five chapters, which can be thought of from a content point of view as lining up with the following main headings: Test, ATPG, Scan, Memory Test, and Cores. A quick synopsis of each chapter follows:

Chapter 1: Test and Design-for-Test Fundamentals: this chapter contains the terms, definitions, and the information involved with test to provide a basic understanding of just what test and design-for-test (DFT) are, how they are done, why they are done, what is being tested, what test is measuring and accomplishing, what equipment is used for test, and what the engineering and cost trade-off drivers are. It is a very basic chapter that can be used by a novice or a junior test or DFT professional to come up to speed on the requirements, actions, and language of test.

Chapter 2: Automatic Test Pattern Generation Fundamentals: this chapter describes the process of automating the onerous task of vector generation and reducing the time-to-volume by supporting an automatic test pattern generation (ATPG) methodology. Some of the analyses and techniques used in vector generation for both AC (dynamic) and DC (static) fault models are described to give an understanding of why certain rules must be applied to the hardware design and of how to reduce the size of the vector set or the time involved in generating the vector set. Also discussed are the measurables and trade-offs involved with the ATPG process so that an evaluation, or benchmark comparison, can be done between various ATPG tools to assist in the selection of the methodology and tool that is right for the application.

Chapter 3: Scan Architectures and Techniques: this chapter is about the scan test methodology and begins with the fundamentals of scan design and operation, and the design concerns and trade-offs involved with adopting a scan design. Also included are some techniques on the installation of scan into a design, how to deal with common problems such as safe-shifting, contention-free vectors, shift timing, and clock skew. Finally, some information is included on reducing test time by shifting the scan architecture at the rated functional frequency (at-speed scan), and on using scan to test for AC goals by operating the scan sample at the rated frequency (AC scan) based on using critical paths extracted from a design's timing analysis.

Chapter 4: Memory Test Architectures and Techniques: this chapter is about memory testing, memory interaction with the scan test architecture, and the adoption of memory built-in self-test (MBIST). This chapter begins with the fundamentals of memory test. and expands into the architectures involved with the coexistence with scan, and eventually describes the architectures and integration of using built-in self-test. The delivery of memories as cores with BIST (BIST'ed memories) includes information on how to deal with the integration of large numbers of memory cores, and how to minimize the routing problem, the power problem, the extraction of characterization or debug data problem, and the coordinated data retention problem.

Chapter 5: Embedded Core Test Fundamentals: this chapter is about creating testable designs with testable and accessible embedded cores. This chapter begins with the terms, definitions, issues, and trade-offs involved with the new style of device design known as "embedded IP" (intellectual property), embedded core, or core-based design. The embedded core design process is addressed in two main aspects, creating a testable reuse core, and implementing a chip design with embedded reusable core products. This chapter relies greatly on understanding the information from chapters 1, 3, and 4.

Guide to the Book

At first glance, this book may seem to be formatted a bit strangely. The beginning of each main subject heading within a chapter is a graphic that is then followed by several sub-sections of descriptive and teaching text. The graphics are the slides that I use when I teach this information as a course (all the graphics are available in full color with the included CD-ROM). You will notice that sometimes the graphic and the text are interrelated and the graphic supports the text; however, sometimes the graphic does not seem to be directly linked to the text, and sometimes the graphic will be an alternate description of the text. This is the nature of presentation material—the graphic must not be so complicated and busy that it confuses the viewer during a presentation or class—however, in the context of a reference book, readers sometimes enjoy a complicated graphic to trace through and analyze as they are learning the information. Since this text was designed to be an introductory and practical guide to the subject of Test and DFT, I decided to leave the presentation material as the graphics instead of replacing them with complicated diagrams.

The flow of information in this text also derives from the way I teach my courses in DFT. When I teach an entire comprehensive class, I naturally start with the basic information and language of test and then I move on to test pattern generation. With these two subjects as a basic foundation, I then move on to scan and then memory test, and I use scan and memory testing as a foundation to begin teaching core test. However, sometimes I am asked to teach just ATPG, just scan, just memory testing, or just core testing.

I could never hope to categorize every possible iteration of use for this book (and I don't mean its use as a doorstop or paperweight). However, I have taught several different courses of study with the information contained within this book, and to this end, I will describe what I think are the most common courses of study and outline the "paths" that must be taken through the book. These courses naturally fall into the "applied" and "management" categories, and then by the subject matter of test basics, cores, scan, built-in self-test, ATPG tool selection, and memory testing.

For the design or DFT engineer, the most important information is the items that describe how to establish a test methodology or implement particular DFT techniques, how the techniques affect the design budgets, and why they are needed (in many cases, the DFT engineer will need a course in "how to argue effectively"). These topics can be divided into the following courses of study: test basics, scan, AC scan, ATPG, memory test, built-in self-test, and core test.

For the manager, the most important information is the items that deal with trade-offs, tasks, schedules, and cost. These items also fall into similar categories of implementing scan or AC scan, adopting an ATPG methodology, implementing a BIST-based test methodology, and developing or using cores.

Test Fundamentals: If you wish to study just the basic fundamentals of testing, with the goal of learning about the test process, understanding the creation of a test program, applying the test program to a tester, and assessing the quality metric, then I would recommend reading all of Chapter 1, Sections 2.15.2, 2.16.1, 3.15.1, 5.4.1, 5.18.1, and 5.25.1, in addition to whatever training material is at hand for the target tester.

Scan Techniques: If you wish to develop a course of study in the understanding and application of scan techniques, then I would recommend reading all of Chapters 1 and 2 for a fundamental background into testing, faults, and ATPG, and then Sections 3.1 through 3.19 for a background in basic DC scan, and Sections 4.14 through 4.18 to explore the interaction of scan with memory architectures. If memory architecture needs to be learned to understand the "scan interaction with memory", then I also recommend reading Sections 4.1 through 4.5.

AC Scan Techniques: If you wish to extend the study of scan into the AC and at-speed scan realms, then I recommend the additional reading of Sections 3.19 through 3.26 to explore the differences between DC scan and adding the extra dimension of timing assessment.

ATPG: If you wish to understand automatic test pattern generation (ATPG) and perhaps wish to understand the development of an ATPG methodology, then I recommend reading all of Chapters 1 and 2 for a basic background of the test process and ATPG, and in addition, Sections 3.1 through 3.12 and Section 3.16, since understanding scan, partial scan, and scan design rules is fundamentally interrelated to the development and application of an ATPG methodology.

Memory Test: If you wish to develop a course of study in the understanding and application of memory testing, then I recommend reading Sections 4.1 through 4.13, and if the study should include a built-in self-test methodology, then 4.19 through 4.28 as well.

BIST: If you wish to develop a course of study in the theory, application, and use of built-in self-test (BIST), both logic and memory, then I would recommend reading Sections 3.3 through 3.11 for a scan background, Section 3.16 on scan rules, Section 3.22 through 3.25 on at-speed scan concerns, and 3.27 on logic BIST. In addition, I would recommend reading Sections 4.1 through 4.13 for a background on memory architecture and testing, and then read Sections 4.19 through 4.28 on memory BIST.

Embedded Core Test: If you wish to understand embedded core testing, either from the development of a testable reusable core, or from the integration standpoint, then I recommend reading all of Chapter 1 for a fundamental background in testing and test application; Sections 2.1 through 2.17 for a basic background in vector generation, vector optimization, and controllability and observability; Sections 3.1 through 3.26 to understand AC, DC, and at-speed scan and scan issues; Sections 4.19 through 4.28 to understand memory BIST and memory BIST issues; and the entirety of Chapter 5 on cores and core-based design.

Management: If you are concerned with the cost, schedules, tasks, and trade-offs of the various topics contained within this book, then a "management" course can be outlined that covers just the sections needed for quick understanding of the issues, costs, and so on. For example, a course of study in the trade-offs involved with DFT would include Sections 1.1 and 1.2 for an understanding of what is tested and why it is accomplished, 1.4 and 1.6 for an understanding of what is measured and with what type of testing, 1.8 to explore the cost trade-offs of test equipment, and 1.10.2 to understand the cost trade-offs involved with test programs.

To understand the selection and development of an ATPG methodology would require Sections 2.1 through 2.3 to understand what an ATPG methodology is and why it should be adopted, Section 2.14 to understand the difference between combinational and sequential ATPG,

Section 2.16 to understand ATPG vectors, Section 2.17 to understand the design rules that will be imposed on the design if ATPG is adopted, Section 2.18 on the selection of a tool, and Section 2.19 for a summary overview of all the issues.

To understand the adoption of a scan methodology would require reading the identified ATPG Sections (2.1-2.3, 2.14, 2.16-2.19) and scan Sections 3.1 through 3.3 to understand scan testing versus functional testing, Section 3.12 to understand full-scan versus partial scan, Section 3.16 to understand the design rules that will be imposed on the design for scan, Sections 3.19 and 3.20 to understand AC scan versus DC scan, Section 3.25 to understand the tasks involved with scan insertion, Section 3.26 to understand the selection of critical paths for AC scan, and Section 3.28 for a summary overview of all the issues.

To understand the adoption of a memory BIST methodology would require reading Section 4.1, which introduces memory testing; Sections 4.4 and 4.5, which address memory design and integration concerns; Section 4.6, which outlines the trade-offs between different memory test methods; Sections 4.19 and 4.20, which introduces memory BIST testing and requirements; Sections 4.22 through 4.24, which outline chip issues and concerns in applying BIST; and Section 4.28, which provides a summary overview of all of the issues.

To understand the issues, concerns, trade-offs, cost factors, goals, and tasks involved with developing a reuse core, or adopting an embedded core-based design style, I would recommend understanding the scan and memory BIST information, and then reading Chapter 5 in its entirety.

Contents