Robert J. Feugate, Jr.

Steven M. McIntyre

INTRODUCTION TO VLSI TESTING

Robert J. Feugate, Jr.
Northern Arizona University

Steven M. McIntyre

INTRODUCTION TO VLSI TESTING

PRENTICE HALL, *Englewood Cliffs, New Jersey 07632*

Library of Congress Cataloging-in-Publication Data

FEUGATE, ROBERT J. (date)
Introduction to VLSI testing.

Includes bibliographies and index.
1. Integrated circuits—Very large scale integration—Testing. I. McIntyre, Steven M. (date)
II. Title.
TK7874.F48 1988 621.395 87-3501
ISBN 0-13-498866-3

Editorial/production supervision
and interior design: *Barbara Marttine*
Cover design: *Photo Plus Art*
Cover photo: *Courtesy of Electroglas*
Manufacturing buyer: *Paula Benevento*

Printed in the United States of America

10 9 8 7 6 5 4 3 2 1

ISBN 0-13-498866-3 025

PRENTICE-HALL INTERNATIONAL (UK) LIMITED, *London*
PRENTICE-HALL OF AUSTRALIA PTY. LIMITED, *Sydney*
PRENTICE-HALL CANADA INC., *Toronto*
PRENTICE-HALL HISPANOAMERICANA, S.A., *Mexico*
PRENTICE-HALL OF INDIA PRIVATE LIMITED, *New Delhi*
PRENTICE-HALL OF JAPAN, INC., *Tokyo*
SIMON & SCHUSTER ASIA PTE. LTD., *Singapore*
EDITORA PRENTICE-HALL DO BRASIL, LTDA., *Rio de Janeiro*

PREFACE

Test engineering has long been a vital, recognized specialty within the broad field of electrical and computer engineering. In recent years, advances in design have been outdistancing advances in testing. New circuits can now be designed and placed into production faster than adequate automated test programs can be written for them. While some products are now being designed with built-in test features, and computer aids for test engineers are being improved, the rapid growth in application-specific integrated circuits is keeping the demand high for engineers with some background in testing. Not only do semiconductor manufacturers and high-volume users need test and product engineers who are directly concerned with automated testing, they also need designers who have an appreciation for test engineering and who can anticipate and design out test problems.

This text developed from handout notes prepared for our undergraduate classes in semiconductor device testing. Although we do not cover board-level testing, most of the concepts discussed apply directly to board testing. The book, when used in the classroom, is intended for junior or senior students in electrical and computer engineering. We have also taught the material, with considerable success, to hardware-oriented computer scientists and to electrical engineering technologists. As much as possible, we tried to make the text stand on its own, making it useful for the practicing engineer as well as the student. The only prerequisites are some experience with a high-level computer language and an elementary understanding of digital logic and electric circuits. Knowledge of microprocessor fundamentals is helpful but not essential.

There are two important points regarding the book's focus: First, there is limited coverage of the mathematics of automated test pattern generation; second, there is detailed discussion of an example of an automated tester and its control language. We strongly feel that an extensive exploration of the mathematics of automated pattern generation techniques is out of place in an introductory text. Automated test generation has, as yet, limited application to very large scale circuits, due in part to the inherent inefficiencies of the algorithms used. In addition, the user of such tools does not need to be expert in the underlying algorithms, any more than a computer programmer needs to know compiler theory. However, we believe wholeheartedly in requiring students to learn something of the architecture of testers and of the details of their programming languages. Programming test equipment is not like programming ordinary computers; there is an interrelationship between the hardware and software far closer than anything students will have encountered previously. In teaching courses in test engineering, we have found repeatedly that students who seem to have overall concepts well in hand find it difficult to apply them in specific situations. So the essence of our approach is detailed application of general knowledge. We heartily endorse this approach, which we have used with great success.

In the past, when one used a specific, detailed tester language, the class either had ready access to actual automated testers and support software or conducted purely paper exercises. Since real testers are almost never available to universities, the first approach is effectively restricted to industrial classes taught by ATE makers or users. Academics have been limited to rather unsatisfactory imaginary practice. Fortunately, the situation has changed recently. Lorimac Softwares, of Tempe, Arizona markets a tester language compiler and a tester simulator that run on IBM PC compatible microcomputers. Their language, LSTL, is derived from and similar to actual tester languages. The language example used in this book closely resembles LSTL, differing mostly in the handling of program branching. The text's program and exercises are readily translated to the Lorimac structures.

A one-semester course should be adequate to cover the entire text. If additional outside material is introduced (for instance, to expand coverage of test pattern generation algorithms beyond that in appendix C), some of the material in chapter 5 may be deleted. We do not recommend deletion of the section on guardbands, because an introduction to guardbanding will expose the student to brand new ideas about tolerances and margins beyond tolerances. The first five chapters alone will provide enough material for a good short course in testing fundamentals, suitable for a two-year technology program or an in-house industrial course in test concepts.

As countless others have pointed out, producing a textbook involves assistance from a great many sources. The book certainly benefited from the valuable suggestions of many people. We would especially like to thank Jesse Wilkinson of

Teradyne Corp., whose painstaking review and thoughtful comments were of great help; and our wives, Alisa and Lori, whose monumental patience, understanding, and encouragement were essential to the completion of this book.

BOB FEUGATE
STEVE MCINTYRE

CONTENTS

1

AUTOMATED TESTING OVERVIEW

1.1 INTRODUCTION

During the 1970s and 1980s, integrated circuits (ICs) reached incredible levels of sophistication, with manufacturers fabricating circuits containing many tens of thousands of logic gates on a single chip. Figure 1.1 illustrates the remarkably rapid growth of circuit complexity during the last two decades. To design such complex circuits, an array of computer-aided design tools was developed. These tools dramatically reduced the time required to design new circuits, giving rise to a multitude of individual large-scale part types.

However, it is not sufficient just to design and manufacture integrated circuits; semiconductor manufacturers and users must also verify that the circuits work as intended. Semiconductor manufacturing processes are so complex that this verification cannot be done on a sampling basis; rather, each VLSI circuit must be individually tested. In the fast-paced production environment, the slow speed and high labor costs of manual testing dictate automated testing of ICs. A sizable industry specializing in automated test equipment and ancillary support equipment has emerged to serve the needs of circuit manufacturers and users.

As the use of automated test equipment (ATE) has grown, so has the demand for engineers and programmers to develop the programs that control these testers. The problem of program development is aggravated not only by a shortage of test

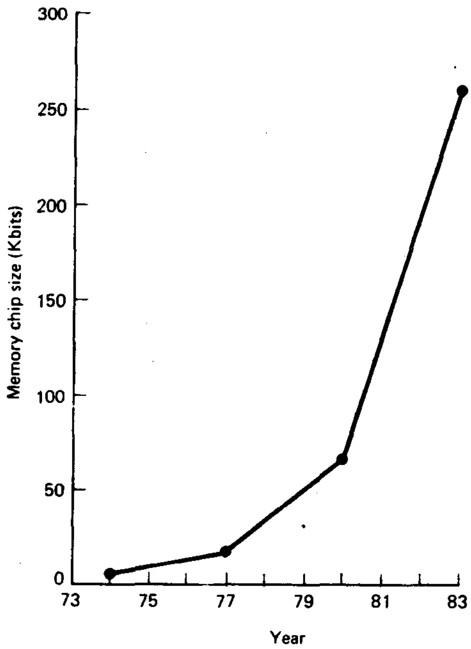

Figure 1.1 Integrated Circuit Complexity versus Time.

engineers but also by the difficulty of developing the programs themselves. In many ways, testing a very large scale integrated circuit is more difficult than designing it. It is both possible and likely that a large integrated circuit will contain embedded elements that cannot be tested externally, even by forcing the circuit through every conceivable operating state. For complex circuits, such exhaustive testing becomes impossible. As an example, Feuer has shown that an exhaustive test of the 8080 microcomputer, only modestly complicated by today's standards, would take over 10^{20} years, at one million tests a second.[1] Thus, the test engineer's task is to create adequate tests that run in limited time. (Incidentally, the definition of what constitutes an "adequate" test is not simple. We explore this point in some depth in chapter 7.) The test engineer's problem is complicated by having few computer aids that approach the usefulness of the designer's CAD systems.

The primary roadblocks in VLSI circuit development are test program development and verification, not design and fabrication. There are several approaches to integrated circuit design using computer-stored building blocks that make it possible to develop chips tailored to individual purposes. Depending on the situation, these building blocks may range from generalized logic gates (gate-arrays) to complex microcomputers. Regardless of the techniques used, the result is an *application-specific integrated circuit* (ASIC) that has functional characteristics different from any other IC. The IEEE's *Spectrum* magazine has examined the possibility of creating a ''one-month'' VLSI circuit—an IC that could be brought from conception to production in 30 days using ASIC development tools. The experts surveyed for the magazine's articles concluded that, while the design and fabrication of the chip could be done in a month, ''it is clear that, in the immediate future, VLSI chips will not be exhaustively tested, and certainly not in one month.''[2] Clearly, testing large and very large scale integrated circuits is of great importance to semiconductor users and manufacturers. Great emphasis continues to be placed on ways to improve test programmer productivity and to design ICs that are easier to test.

In this text, we explore test engineering in some depth to give readers an idea of the scope and complexity of test engineering. The first two chapters provide the background necessary to develop a concept of where testing fits into the overall semiconductor manufacturing sequence and to understand what circuit parameters are actually tested. In the four following chapters, we examine in great detail a hypothetical automated tester architecture and the associated test programming language. During this examination, we prepare an actual test program for a typical IC. After completing these chapters, readers should understand ATE programming basics and have a good concept of the challenges of test engineering.

After chapter six, the material is more general, as we present an overview of automated test programming aids and discussions of some of the solutions to classic VLSI testing situations, such as testing memories or microcomputers. We conclude by presenting some methods that designers can use to ease the test burden by including test capabilities in their original circuit designs.

1.2 THE IC MANUFACTURING SEQUENCE

Integrated circuits are tested at least twice during the manufacturing sequence: first before the wafer is separated into individual circuit dice and again after the dice have been packaged into completed circuits. In the next chapter, we present an overview of the processes by which the circuits are initially created on the wafer. Now we focus on the manufacturing steps that take place after wafer fabrication is complete. (Shown in figure 1.2.) First, completed wafers are subjected to a wafer sort test, during which faulty devices are identified and marked with dots of ink. The marked wafers are then separated into circuit dice by a *scribe and break* process. The circuits are *scribed* or partially sawn apart with a diamond saw or laser scriber, then subjected to pressure with a roller. The wafers break cleanly along the saw cuts, separating into the individual rectangular circuit dice. Ink-marked rejects are culled out and discarded.

Each good die is mounted on a supporting base with epoxy or a gold, silver, or copper eutectic solder paste. Next, fine gold wires are used to connect bonding pads on the die to lead pins that will be used for external circuit connections. When all bonding wires are attached, the rest of the circuit's external package is completed. The packaged but unmarked circuits then undergo final testing to find and remove circuits that were damaged during assembly or that somehow slipped through the wafer sort test. In many cases, good parts are further separated into different performance categories, based on operating speed, temperature performance, and so on, during final testing. Circuits failing final test may be discarded immediately but are usually held for analysis to determine the causes of the failures.

After passing their electrical tests, circuits intended for high-reliability applications undergo still further tests, this time mechanical. They are immersed in heated baths of an inert chemical such as Freon to check that their packages are hermetically sealed. Air bubbles escape from faulty packages and are easily seen

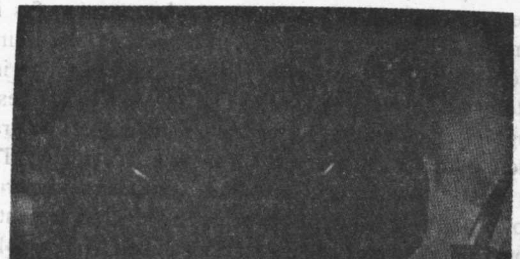

Figure 1.2 Post-Fabrication Semiconductor Manufacturing Steps.

a. IC probing during wafer sort testing (courtesy of Electroglas).

b. Scribing wafers after testing (courtesy of Kulicke and Soffa Industries, Inc.; photo by Leon Oboler).

c. Bonding wires to connect package leads to chip (courtesy of Kulicke and Soffa Industries, Inc.; photo by Michael Denese).

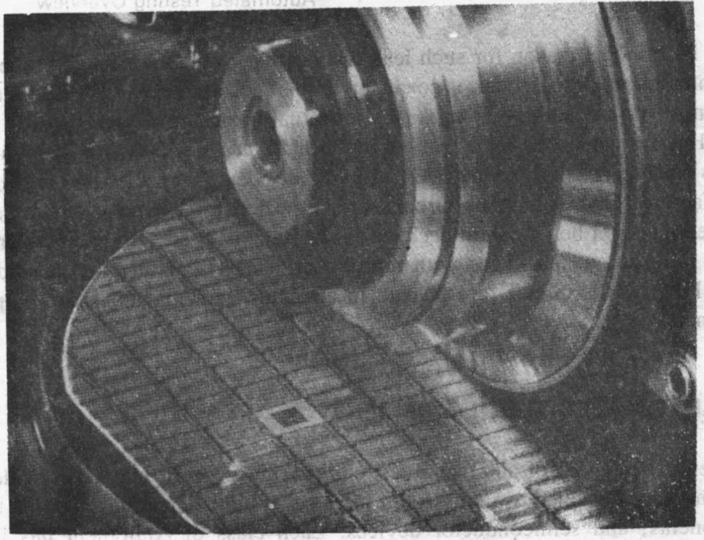

Figure 1.2 (b)

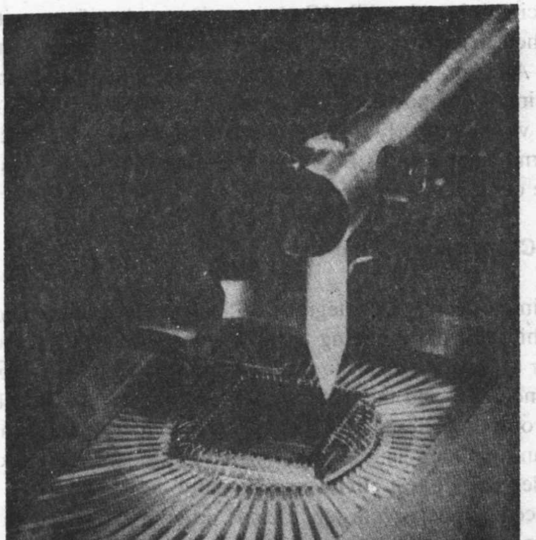

Figure 1.2 (c)

in the Freon. The reason for such leak tests is that over long periods the moisture that enters through leaks causes corrosion at the interfaces between different metals used in the circuits and their packages, eventually causing device failure.

Having passed all their tests, good circuits are marked with their manufacturer's trademark, the part number, and a code identifying the time and place of manufacture. Once again the circuits are inspected for cosmetic defects such as smeared marks or bent or short package leads. Bent leads are straightened, smudged marks redone, and packages with nonstandard leads separated to be sold at cut rates. Finally, the circuits are packed in antistatic carrying tubes and shipped to customers.

1.3 AUTOMATED TEST EQUIPMENT

The general category of automated test equipment, or ATE, includes a wide range of computer-controlled systems for testing subsystems, modules, circuit boards, components, and semiconductor devices. Each class of equipment has its own hardware and software characteristics. In this book, we restrict ourselves to computer-controlled systems for testing integrated circuits. Although many of the concepts we present also apply to board-level testers, the two fields are really somewhat distinct. Board-level testing attempts to detect and identify failed components on a complete circuit board, while IC testing tries to identify circuits that fail to meet their published performance specifications. For complex ICs, this task becomes formidable. Attempting to deduce, purely on the basis of terminal measurements, whether a circuit composed of more than 100,000 transistors and as many interconnections works properly in all possible situations is, for practical purposes, an almost insurmountable task. Board-level testing is marginally simpler in that the intermediate circuits and interconnections are available to the tester.

1.3a Comparison Testers

The simplest form of integrated circuit tester is the *comparison* tester, by means of which the device being tested is compared to a "golden device" which is known (or at least assumed) to meet all specifications. As shown in figure 1.3a, the circuit under test and the "golden device" are simultaneously stimulated with a sequence of test patterns (also called test *vectors*). The resulting outputs are compared, and, as long as the output of the unit under test matches the output of the golden device, the test circuit is assumed to be good.

Advanced comparison testers include programmable input drive voltages and output comparison voltages, so that testing can be done at data sheet limits (we discuss data sheet specifications thoroughly in chapter 2). Figure 1.3b shows a representative comparison tester. The golden device is mounted on a circuit board and inserted into the tester. Comparison testers are relatively small, tabletop pieces of equipment. Because of their simplicity, comparison testers are relatively low in