Testing is no longer good enough to ensure the quality of systems that must never fail. A highly reliable system must be able to detect and usually correct faults arising from test escapes, from aging or from the environment. For the most part, this has been done using techniques such as parity, error correcting codes, and redundancy. In particular, memories are designed with technologies so small as to make errors caused by cosmic ray hits inevitable, and memories with concurrent error detection and correction are now standard.

Concern has existed for some time that random logic will fall prey to soft errors, and thus require online error checking and correction. The book under review examines precisely this area, presenting micro-level techniques for concurrent checking, which should be less expensive than high-level redundancy. Concurrent checking for logic alone is covered, which is an area in which a lot of new research is needed.

First, a caveat: if you are new to concurrent checking, this is not the book for you. Although the authors spend some time on background, most of the book is devoted to highly specific solutions to the problem of concurrent checking, developed largely by various subsets of the authors in a considerable number of papers. Some background is assumed, and this is clearly stated.

This volume consists of but four chapters, the first of which is a brief introduction. Most of the second chapter is an introduction to faults: stuck-at, stuck-open, bridging, transient, and delay. Finally, there is an introduction to the concepts of self-testing and self-checking, and a definition of output dependencies. Two outputs are independent under a fault if the effect of that fault can show up on only one of them for any set of inputs; they are weakly independent under a fault if an input that causes the effect to show up on one output but not on another.

The chapter is well written, but little would be new to most test researchers.

Chapter 3, approximately half of the book, describes specific methods of concurrent checking developed by the authors. They start with simple duplication and checking, then move on to parity checking. The principle here is to add a generator, which computes the parity of a combinational circuit output, and a predictor, which computes what the output parity should be, on the basis of the inputs. This prediction is done by synthesizing a duplicate version of the circuit, with no outputs except the predicted parity, which, the authors claim, will be reduced in size. The generated output parity and predicted parity are then compared. If outputs are independent, further improvements can be made.

Chapter 3 exemplifies both the strengths and weaknesses of the book. The writing here, though complex, is clear, and a good example of concurrent checking is set forth. The authors give an excellent description of how this method works. However, because the book focuses on their work, we learn nothing at all of how other researchers have approached this problem. References are plentiful, but a large number are the authors’ work; many others are for general background. We do not get a sense of the history of this problem, nor of how the
solutions proposed are better than or different from those of others.

A second problem arises only if you are interested in the practical aspects of concurrent checking. We get no idea of the predictor circuit’s overhead, though we do see that it would be smaller than simple duplication. No thought is given to performance implications. Those are outside the book’s scope, but a reader who might be interested in implementing some of these methods should be warned that a feasibility study would be required.

The chapter continues with another method of complementary checking: a complementary circuit, designed so that the exclusive-OR of the circuit and its complement are a code word of a chosen code. These can be more efficient than duplication, since any code word can be used, giving more freedom to a synthesis tool, although it appears that synthesis tools are not up to the task of creating efficient complementary circuits.

Yet another innovative method described is the use of self-dual circuits, designed so that \( y = f(x), \overline{y} = \overline{f(x)} \). For applications that do not push performance, such as controllers for mechanical systems, the dual version of the input can be applied after the regular version and the outputs compared. A method is described to convert a circuit into a self-dual one.

The last chapter, Chapter 4, features several ways to design concurrent checking adders. It begins with a brief tutorial on different types of adders and then applies the techniques from Chapter 3 to them. Each example includes a reasonably complete design, an idea of the expense involved, and a list of faults covered. Once more, there is no survey of previous work, but a quick search of the literature using the IEEE Digital Library reveals that most recent papers on self-checking adders are by at least one of the four authors of this book.

This is not the book from which to learn the basics of concurrent checking techniques, and it does not claim to be. New Methods is an excellent resource for understanding some of the most recent work in concurrent checking, for a seminar on this subject, or for use by someone needing to implement self-checking logic for a highly reliable system; additional research, however, would be required. The writing is quite dense but clear, given the subject’s complexity. If this subject is of interest to you, and you would like a convenient survey of some advanced work in the area of concurrent checking, this book is worth obtaining.

Finally, one feature of this book is particularly noteworthy: The authors begin each chapter, and each chapter subsection, with a short description of what is going to be explained, and they end each chapter and subsection with a summary of what was explained. This is accomplished using more detail than in other summaries I’ve seen. I found this approach extremely useful in solidifying my understanding of each topic area. I went back and reread all of these sections before writing this review. Such summaries amount to a digest of the material, and are very helpful for establishing context for a section that readers might want to return to later. I encourage all authors to do something similar; it is a gift to the reader and to the reviewer.