

Book Reviews

About the power problem

Scott Davidson

■ **SOME YEARS AGO**, the place I worked shut off the air conditioning nights and weekends. In those days, if you needed to work extra hours, it had to be *at* work, and the building got pretty hot. There was one oasis of coolness we could retreat to—the air-conditioned lab where our trusty super-minicomputer stood. Though tiny when compared to mainframes, the computer still needed lots of power and thus lots of cooling.

Today we carry far-more-powerful computers in our pockets, which not only don't require air conditioning but which are not even supposed to get warm. These low-power devices need special consideration when being tested, and that is the subject of the book under review.

The book addresses the following major issues. First, which defects are power related, and how can we test for them? Second, how do we deal with power-related issues during normal test? The importance of reducing test time encourages us to increase circuit activity during test, to do it faster, but this can get us into trouble with low-power designs. Finally, how do we test the structures we use to reduce chip power, such as clock gating?

The first chapter is on DFT basics, which most readers should be able to skip, having seen it before. Next is an excellent introduction to the basics of power in general and during test in particular. I'm not an expert in this area, and I found it readable and pitched to just the right level.

Chapter 3 is a perhaps too-detailed discussion of low-power test generation techniques. It's basically a literature survey, and doesn't attempt to compare the described techniques. I would have liked to see two tables, one summarizing the major features of

Reviewed in this issue

Power-Aware Testing and Test Strategies for Low Power Devices, Patrick Girard, Nicola Nicolici, and Xiapqing Wen, eds. (Springer, 2010, ISBN: 978-1-4419-0927-5, 353 pp.).

Patrick Girard
Nicola Nicolici
Xiapqing Wen
Editors

Power-Aware Testing
and Test Strategies for
Low Power Devices

Springer

the covered techniques and the other comparing results from the papers running experiments when the methods were applied to standard benchmarks. I noticed a few flaws. In section 3.3.2.1, shift-in power is reduced by estimating shift power use by computing weighted transitions. However, unknowns don't seem to be treated correctly. A 1X0 sequence will cause either a 110 or 100 transition, neither of which is counted. Section 3.6.1 proposes reducing memory test power consumption by modifying the order in which memory cells are tested to reduce switching on address lines. However, it's often important to test in physical-address order, so memory defect coverage might be reduced if this method is used.

Typically, changing the way we do test generation alone will not prevent power consumption problems. DFT techniques must be used—in this case, design for low-power testability. Chapter 4, which describes some of these, is a good length, and doesn't dive too deeply into any specific DFT method. Useful diagrams are plentiful. Impractical solutions, however, are mixed with practical ones. In some cases, the authors alert the reader about a problem with a proposed solution, such as methods that are strongly dependent on a specific set of patterns. I appreciate this, but is completeness in presenting proposed techniques worth the risk of misleading the reader?

With test time at a premium, and the number of defect models for which we need to generate tests increasing, we wish to do as much testing work per

test vector as possible, and test compression methods have become very popular for this reason. But compression works directly against the goals of low-power test. Chapter 5 deals with low-power compression and BIST methods. It is a high-level presentation, which is good, but still basically a literature survey.

The next level of complexity comes with multicore SoCs, where each core has a test and a power budget. How can we plan our test to stay within chip-level power constraints and still minimize test time? Chapter 6 considers this, and here we have a good comparison, for Table 6.1 compares 15 different test schedule approaches for a single benchmark design. We finish the chapter with a short section on I_{DDQ} testing. This seems out of place—the most recent paper, by one of the chapter’s authors, is 8 years old, and the others referenced are more than 12 years old. The section actually covers what used to be called design for I_{DDQ} testability, but the whole area has decreased in relevance significantly, so its utility is not clear.

Low power is not just important while testing; it is vital for many types of systems. Chapter 7 discusses low-power design methods, as well as the implications for these on test. While this chapter still has a good literature survey, it also is fairly practical, describing low-power design methods in industrial use.

For reasons of speed and power, individual chips may have voltages set to what is optimal for their particular characteristics, which vary due to process drift or wafer location. It is also often the case that a chip is divided into multiple voltage domains. Chapter 8 covers these cases, which are quite different, and I wish the chapter had made these distinctions more apparent. The first section describes how defect models change with changes in supply voltage, and how it is often necessary to test at different voltages. The second section describes the impact of multiple voltage domains on DFT techniques such as scan. These are important issues but not directly related to low power—those of us who work with very high power designs must deal with them also.

In Chapter 9, we move into how these issues are really faced in the industry—in particular, dealing with designs with gated clocks. This chapter is a tutorial on clock gating with notes on some of the less obvious but extremely important issues. It is a valuable chapter for those starting work in this area.

On-chip power management implies the existence of power-management structures. On starting Chapter 10,

I didn’t think they would pose much of a test problem, but it convinced me I was wrong. The basic problem is that the functionality of this circuitry is to turn off logic sections using power switches. How do you test something that’s off when working properly? You will get the answer, in a chapter logically organized with alternative approaches well laid out and compared. The last section, on faults in power distribution networks, is especially good and might explain some hard-to-diagnose field failures.

The final chapter concerns challenges for EDA tools caused by low-power designs, and how tools must become aware of issues such as the use of multiple power domains, many of which have not been considered previously. This chapter contained a good deal of new and interesting material.

IN SUMMARY, this book is both a good literature survey and a source of practical advice. But it could have been better, as could most books containing literature surveys. Writers of these sections should add value by critically comparing the results of these techniques run on benchmark circuits. Please, surveyors, summarize those results. Try to show the reader how concepts have developed in a field, starting with early work and showing the influences of pioneering work on later efforts. It’s unlikely that those who publish early will have the best results; we all try to improve. It would help a novice tremendously if she can see which of the methods described give the best known results, and which have been improved on. This will take a bit more work for the surveyor, but the rewards for the reader will be great.

This should not be taken as a criticism of the book under review; this is just the book where I realized why some of these types of chapters were so dissatisfying. Survey articles that do this well are rare; one is the survey of test compression by Nur Touba in the July-August 2006 issue of *Design & Test* (vol. 23, no. 4, pp. 294-303). I believe I speak for many readers in requesting that writers of survey articles go beyond abstracts and survey the field as a whole. ■

■ Direct questions and comments about this department to Scott Davidson, Oracle, M/S USCA 16-107, 4160 Network Circle, Santa Clara, CA 95054; scott.davidson@oracle.com.



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.