

Book Reviews

Making a list...checking it twice

Grant Martin

Tensilica

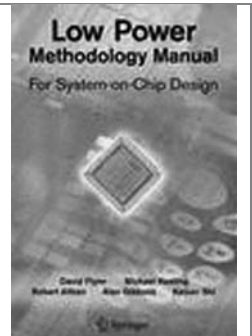
■ **IN THE BEGINNING**, there was the *Reuse Methodology Manual*—the first edition of which was published almost a decade ago (in 1998) and is now on its third edition. Then came the *Verification Methodology Manual for System Verilog* (in 2005). Now, *Low Power Methodology Manual: For System-on-Chip Design* has arrived. When we look at these three methodology manuals, we see several threads of continuity: First, all three books have been published by Springer or Kluwer Academic Publishers (which merged with Springer a few years ago). Second, all three have coauthors from Synopsys. Third, the first and third books were coauthored by Michael Keating. Fourth, the last two books have coauthors from ARM.

The nature of a methodology manual is often realized as a set of recommendations and guidelines—a kind of checklist, organized into various thematic chapters, with at most a few pages of text illustrating each recommendation, guideline, or option. Design groups wanting an update on best practices and insight into experts' recommendations will find this kind of checklist extremely valuable. Indeed, such works can act as references for design companies, groups, and individual designers—both to audit their own practices in different design and verification topic areas and to become familiar with new design practices as they seek to upgrade their methodologies and improve productivity and design quality.

The *Low Power Methodology Manual* is no exception to this general approach, and this makes the text all the more valuable. This book does not teach all there is to know about power structure and low-power design from the ground up, nor does it cover all the

Reviewed in this issue

Low Power Methodology Manual: For System-on-Chip Design, by Michael Keating, David Flynn, Rob Aitken, Alan Gibbons, and Kaijian Shi (Springer, 2007, ISBN 978-0-387-71818-7, 304 pp., \$129.00).



research that people are doing on future low-power methods, tools, and design approaches. Rather, it provides a comprehensive inventory of methods of low-power IC design, along with discussions of the various options available to design groups for implementing both basic and advanced techniques into their SoC designs. It also contains some chapter references and a reasonably sized bibliography that point to some of the significant background and fundamental work underlying these recommendations.

In addition, three characteristics make the book even more useful and practical: First, the authors have many years of design and electronic design automation (EDA) tool experience in the design of complex SoC devices, including processor-centric design. Second, they have collectively developed significant reference designs incorporating ARM processors, buses, other IP, and Synopsys IP as demonstration vehicles for many of the low-power design techniques they recommend. Third, they incorporate some of the latest tool-based approaches to low-power design by giving several examples using the Unified Power Format (UPF) syntax, which allows low-power design intent and constraints to be expressed in a form that EDA tools can work with to guide their operations. Nothing beats practical demonstrations on quasirealistic designs as a way of making guidelines real and actionable.

The book is organized to cover the theme thoroughly. After an overview and introduction in the first chapter, Chapters 2 and 3 briefly discuss well-

known, standard low-power methods, such as clock gating, gate-level power optimization, and multisupply and threshold-voltage-based methods. Chapter 4 gets to the heart of the matter: the newer methods that some design groups have tried but that are still more on the cutting edge than standard design practice. These can be summed up as two key methods: power gating, and dynamic scaling of frequency and voltage. Indeed, the primary purpose of this low-power methodology manual is to give design groups enough information, recommendations, and pointers so that they can incorporate newer as well as standard methods in their next SoCs.

With this in mind, Chapters 4 through 8 provide a rather exhaustive explanation of power gating. This topic is first given an overview, and then treated at the register-transfer, architecture, and IP-related levels. Interspersed is an RTL power-gating example, the SALT (Synopsys ARM Low-Power Technology) project, with measured results demonstrating the energy savings possible from applying these methods. These chapters also give several practical examples of the use of UPF and discuss how to apply the techniques in a non-UPF-based RTL flow.

Chapters 9 and 10 discuss a set of guidelines and approaches for dynamic scaling of frequency and voltage, followed by two examples for a 130-nm process and a 65-nm process that illustrate energy savings based on actual silicon measurements and analysis. Here, the authors really show that they have “been there, done that.”

The final four chapters cover a range of methodology and design issues in implementing multivoltage and power-gated designs, discuss options for appropriate physical libraries, outline the important topic of retention register design, and review various implementation options for power-switching networks. In addition, two appendices provide details on sleep transistor design and review the UPF syntax. These final chapters and appendices round out the book and provide much additional practical guidance for design teams.

The experiential evidence based on the authors’ design experiments gives significant pragmatic context to their recommendations. However, there are a few areas where the book could at least touch on related topics. As those familiar with the latest EDA standards

wars know, Cadence and Si2 have advocated a power specification format, CPF (Common Power Format), in competition with the Accelera-promoted UPF. Although there would be no need to give details on it, the authors might have at least mentioned the existence of CPF to give a more complete background to the reader. (Experts on both formats have indicated that they overlap in semantics by about 85% to 90%, which gives the community some hope that after a suitable period, a truly unified, common power format will emerge that will be supported in all tool flows and be pragmatically interoperable. We can always hope!)

Moreover, although the authors from ARM are no doubt most familiar with fixed-instruction-set processors, as discussed in the examples given, there are other techniques for processor-centric design that have a radical impact on energy consumption. As someone who works for Tensilica, which is one of several companies offering configurable and extensible processor cores, I can attest to the powerful energy savings possible from using application-specific instruction-set processors (ASIPs). Again, this book would have presented a more balanced picture if it had at least mentioned such possibilities.

BUT THESE ARE SMALL quibbles. This book stands on its own merits and can be well recommended. Any design team should be able to benchmark its low-power design practices using this book. For those who are already using the most advanced techniques, there will be little to learn here, but the book will still stand as a mark against which they can measure themselves. For those wishing to move closer to the cutting edge, this is an extremely valuable checklist of recommendations. It is well worth reading, and is a tool well worth using. ■

■ Direct questions and comments about this department to Grant Martin, 2424 Raven Road, Pleasanton, CA 94566; gmartin@ieee.org.

For further information on this or any other computing topic, please visit our Digital Library at <http://www.computer.org/csdl>.