PREFACE

In the two decades since field-programmable gate arrays (FPGAs) were introduced, they have radically changed the way digital logic is designed and deployed. By marrying the high performance of application-specific integrated circuits (ASICs) and the flexibility of microprocessors, FPGAs have made possible entirely new types of applications. This has helped FPGAs supplant both ASICs and digital signal processors (DSPs) in some traditional roles.

To make the most of this unique combination of performance and flexibility, designers need to be aware of both hardware and software issues. Thus, an FPGA user must think not only about the gates needed to perform a computation but also about the software flow that supports the design process. The goal of this book is to help designers become comfortable with these issues, and thus be able to exploit the vast opportunities possible with reconfigurable logic.

We have written *Reconfigurable Computing* as a tutorial and as a reference on the wide range of concepts that designers must understand to make the best use of FPGAs and related reconfigurable chips—including FPGA architectures, FPGA logic applications, and FPGA CAD tools—and the skills they must have for optimizing a computation. It is targeted particularly toward those who view FPGAs not just as cheap, slow ASIC gates or as a means of prototyping before the "real" hardware is created, but are interested in evaluating or embracing the substantial advantages reprogrammable devices offer over other technologies. However, readers who focus primarily on ASIC- or CPU-based implementations will learn how FPGAs can be a useful addition to their normal skill set. For some traditional designers this book may even serve as an entry point into a completely new way of handling their design problems.

Because we focus on both hardware and software systems, we expect readers to have a certain level of familiarity with each technology. On the hardware side, we assume that readers have a basic knowledge of digital logic design, including understanding concepts such as gates (including multiplexers, flip-flops, and RAM), binary number systems, and simple logic optimization. Knowledge of hardware description languages, such as Verilog or VHDL, is also helpful. We also assume that readers have basic knowledge of computer programming, including simple data structures and algorithms. In sum, this book is appropriate for most readers with a background in electrical engineering, computer science, or computer engineering. It can also be used as a text in an upper-level undergraduate or introductory graduate course within any of these disciplines.

No one book can hope to cover every possible aspect of FPGAs exhaustively. Entire books could be (and have been) written about each of the concepts that are discussed in the individual chapters here. Our goal is to provide a good working knowledge of these concepts, as well as abundant references for those who wish to dig deeper.

Reconfigurable Computing: The Theory and Practice of FPGA-Based Computation is divided into six major parts—hardware, programming, compilation/mapping, application development, case studies, and future trends. Once the introduction has been read, the parts can be covered in any order. Alternatively, readers can pick and choose which parts they wish to cover. For example, a reader who wants to focus on CAD for FPGAs might skip hardware and application development, while a reader who is interested mostly in the use of FPGAs might focus primarily on application development.

Part V is made up of self-contained overviews of specific, important applications, which can be covered in any order or can be sprinkled throughout a course syllabus. The part introduction lists the chapters and concepts relevant to each case study and so can be used as a guide for the reader or instructor in selecting relevant examples.

One final consideration is an explanation of how this book was written. Some books are created by a single author or a set of coauthors who must stretch to cover all aspects of a given topic. Alternatively, an edited text can bring together contributors from each of the topic areas, typically by bundling together standalone research papers. Our book is a bit of a hybrid. It was constructed from an overall outline developed by the primary authors, Scott Hauck and André DeHon. The chapters on the chosen topics were then written by noted experts in these areas, and were carefully edited to ensure their integration into a cohesive whole. Our hope is that this brings the benefits of both styles of traditional texts, with the reader learning from the main experts on each topic, yet still delivering a well-integrated text.

Acknowledgments

While Scott and André handled the technical editing, this book also benefited from the careful help from the team at Elsevier/Morgan Kaufmann. Wayne Wolf first proposed the concept of this book to us. Chuck Glaser, ably assisted by Michele Cronin and Matthew Cater, was instrumental in resurrecting the project after it had languished in the concept stage for several years and in pushing it through to completion. Just as important were the efforts of the production group at Elsevier/Morgan Kaufmann who did an excellent job of copyediting, proofreading, integrating text and graphics, laying out, and all the hundreds of little details crucial to bringing a book together into a polished whole. This was especially true for a book like this, with such a large list of contributors. Specifically, Marilyn E. Rash helped drive the whole production process and was supported by Dianne Wood, Jodie Allen, and Steve Rath. Without their help there is no way this monumental task ever would have been finished. A big thank you to all.

Scott Hauck
André DeHon