# Preface

This book is unique in its treatment in that it presents digital logic design from the perspective of computer architecture, starting at the beginning with 1's and 0's, and leading through the design of a microprocessor.

We believe that building a microprocessor is a special rite of passage for engineering and computer science students. The inner workings of a processor seem almost magical to the uninitiated, yet prove to be straightforward when carefully explained. Digital design in itself is a powerful and exciting subject. Assembly language programming unveils the inner language spoken by the processor. Microarchitecture is the link that brings it all together.

The first two editions of this increasingly popular text have covered the MIPS architecture in the tradition of the widely used architecture books by Patterson and Hennessy. As one of the original Reduced Instruction Set Computing architectures, MIPS is clean and exceptionally easy to understand and build. MIPS remains an important architecture and has been infused with new energy after Imagination Technologies acquired it in 2013.

Over the past two decades, the ARM architecture has exploded in popularity because of its efficiency and rich ecosystem. More than 50 billion ARM processors have been shipped, and more than 75% of humans on the planet use products with ARM processors. At the time of this writing, nearly every cell phone and tablet sold contains one or more ARM processors. Forecasts predict tens of billions more ARM processors soon controlling the Internet of Things. Many companies are building high-performance ARM systems to challenge Intel in the server market. Because of the commercial importance and student interest, we have developed this ARM edition of this book.

Pedagogically, the learning objectives of the MIPS and ARM editions are identical. The ARM architecture has a number of features including addressing modes and conditional execution that contribute to its efficiency but add a small amount of complexity. The microarchitectures also are very similar, with conditional execution and the program counter being the largest changes. The chapter on I/O provides numerous examples using the Raspberry Pi, a very popular ARM-based embedded Linux single board computer.

We expect to offer both MIPS and ARM editions as long as the market demands.

## FEATURES

### Side-by-Side Coverage of SystemVerilog and VHDL

Hardware description languages (HDLs) are at the center of modern digital design practices. Unfortunately, designers are evenly split between the two dominant languages, SystemVerilog and VHDL. This book introduces HDLs in Chapter 4 as soon as combinational and sequential logic design has been covered. HDLs are then used in Chapters 5 and 7 to design larger building blocks and entire processors. Nevertheless, Chapter 4 can be skipped and the later chapters are still accessible for courses that choose not to cover HDLs.

This book is unique in its side-by-side presentation of SystemVerilog and VHDL, enabling the reader to learn the two languages. Chapter 4 describes principles that apply to both HDLs, and then provides language-specific syntax and examples in adjacent columns. This side-by-side treatment makes it easy for an instructor to choose either HDL, and for the reader to transition from one to the other, either in a class or in professional practice.

### ARM Architecture and Microarchitecture

Chapters 6 and 7 offer the first in-depth coverage of the ARM architecture and microarchitecture. ARM is an ideal architecture because it is a real architecture shipped in millions of products yearly, yet it is streamlined and easy to learn. Moreover, because of its popularity in the commercial and hobbyist worlds, simulation and development tools exist for the ARM architecture. All material relating to ARM® technology has been reproduced with permission from ARM Limited.

### Real-World Perspectives

In addition to the real-world perspective in discussing the ARM architecture, Chapter 6 illustrates the architecture of Intel x86 processors to offer another perspective. Chapter 9 (available as an online supplement) also describes peripherals in the context of the Raspberry Pi single-board computer, a hugely popular ARM-based platform. These real-world perspective chapters show how the concepts in the chapters relate to the chips found in many PCs and consumer electronics.

### Accessible Overview of Advanced Microarchitecture

Chapter 7 includes an overview of modern high-performance microarchitectural features including branch prediction, superscalar, and out-of-order operation, multithreading, and multicore processors. The treatment is accessible to a student in a first course and shows how the microarchitectures in the book can be extended to modern processors.

### End-of-Chapter Exercises and Interview Questions

The best way to learn digital design is to do it. Each chapter ends with numerous exercises to practice the material. The exercises are followed by a set of interview questions that our industrial colleagues have asked students who are applying for work in the field. These questions provide a helpful glimpse into the types of problems that job applicants will typically encounter during the interview process. Exercise solutions are available via the book's companion and instructor websites.

## ONLINE SUPPLEMENTS

Supplementary materials are available online at http://textbooks.elsevier.com/9780128000564. This companion site (accessible to all readers) includes the following:

- Solutions to odd-numbered exercises
- Links to professional-strength computer-aided design (CAD) tools from Altera®
- Link to Keil's ARM Microcontroller Development Kit (MDK-ARM), a tool for compiling, assembling, and simulating C and assembly code for ARM processors
- Hardware description language (HDL) code for the ARM processor
- Altera Quartus II helpful hints
- Lecture slides in PowerPoint (PPT) format
- Sample course and laboratory materials
- List of errata

The instructor site (linked to the companion site and accessible to adopters who register at http://textbooks.elsevier.com/9780128000564) includes the following:

- Solutions to all exercises
- Links to professional-strength computer-aided design (CAD) tools from Altera®
- Figures from the text in PDF and PPT formats

Additional details on using the Altera, Raspberry Pi, and MDK-ARM tools in your course are provided. Details on the sample laboratory materials are also provided here.

## HOW TO USE THE SOFTWARE TOOLS IN A COURSE

### Altera Quartus II

Quartus II Web Edition is a free version of the professional-strength Quartus™ II FPGA design tools. It allows students to enter their digital designs in schematic or using either the SystemVerilog or the VHDL hardware description language (HDL). After entering the design, students can simulate their circuits using ModelSim™-Altera Starter Edition, which is available with the Altera Quartus II Web Edition. Quartus II Web Edition also includes a built-in logic synthesis tool supporting both SystemVerilog and VHDL.

The difference between Web Edition and Subscription Edition is that Web Edition supports a subset of the most common Altera FPGAs. The difference between ModelSim-Altera Starter Edition and ModelSim commercial versions is that the Starter Edition degrades performance for simulations with more than 10,000 lines of HDL.

### Keil's ARM Microcontroller Development Kit (MDK-ARM)

Keil's MDK-ARM is a tool for developing code for an ARM processor. It is available for free download. The MDK-ARM includes a commercial ARM C compiler and a simulator that allows students to write both C and assembly programs, compile them, and then simulate them.

## LABS

The companion site includes links to a series of labs that cover topics from digital design through computer architecture. The labs teach students how to use the Quartus II tools to enter, simulate, synthesize, and implement their designs. The labs also include topics on C and assembly language programming using the MDK-ARM and Raspberry Pi development tools.

After synthesis, students can implement their designs using the Altera DE2 (or DE2-115) Development and Education Board. This powerful and competitively priced board is available from www.altera.com. The board contains an FPGA that can be programmed to implement student designs. We provide labs that describe how to implement a selection of designs on the DE2 Board using Quartus II Web Edition.

To run the labs, students will need to download and install Altera Quartus II Web Edition and either MDK-ARM or the Raspberry Pi tools. Instructors may also choose to install the tools on lab machines. The labs include instructions on how to implement the projects on the DE2 Board. The implementation step may be skipped, but we have found it of great value.

We have tested the labs on Windows, but the tools are also available for Linux.

## BUGS

As all experienced programmers know, any program of significant complexity undoubtedly contains bugs. So, too, do books. We have taken great care to find and squash the bugs in this book. However, some errors undoubtedly do remain. We will maintain a list of errata on the book's webpage.

Please send your bug reports to ddcabugs@gmail.com. The first person to report a substantive bug with a fix that we use in a future printing will be rewarded with a $1 bounty!

## ACKNOWLEDGMENTS

We appreciate the hard work of Nate McFadden, Joe Hayton, Punithavathy Govindaradjane, and the rest of the team at Morgan Kaufmann who made this book happen. We love the art of Duane Bibby, whose cartoons enliven the chapters.

We thank Matthew Watkins, who contributed the section on Heterogeneous Multiprocessors in Chapter 7. We greatly appreciate the work of Joshua Vasquez, who developed code for the Raspberry Pi in Chapter 9. We also thank Josef Spjut and Ruye Wang, who class-tested the material.

Numerous reviewers substantially improved the book. They include Boyang Wang, John Barr, Jack V. Briner, Andrew C. Brown, Carl Baumgaertner, A. Utku Diril, Jim Frenzel, Jaeha Kim, Phillip King, James Pinter-Lucke, Amir Roth, Z. Jerry Shi, James E. Stine, Luke Teyssier, Peiyi Zhao, Zach Dodds, Nathaniel Guy, Aswin Krishna, Volnei Pedroni, Karl Wang, Ricardo Jasinski, Josef Spjut, Jörgen Lien, Sameer Sharma, John Nestor, Syed Manzoor, James Hoe, Srinivasa Vemuru, K. Joseph Hass, Jayantha Herath, Robert Mullins, Bruno Quoitin, Subramaniam Ganesan, Braden Phillips, John Oliver, Yahswant K. Malaiya, Mohammad Awedh, Zachary Kurmas, Donald Hung, and an anonymous reviewer. We appreciate Khaled Benkrid and his colleagues at ARM for their careful review of the ARM-related material.

We also appreciate the students in our courses at Harvey Mudd College and UNLV who have given us helpful feedback on drafts of this textbook. Of special note are Clinton Barnes, Matt Weiner, Carl Walsh, Andrew Carter, Casey Schilling, Alice Clifton, Chris Acon, and Stephen Brawner.

And last, but not least, we both thank our families for their love and support.