Preface

Why publish yet another book on digital design and computer architecture? There are dozens of good books in print on digital design. There are also several good books about computer architecture, especially the classic texts of Patterson and Hennessy. 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 students through the design of a MIPS microprocessor.

We have used several editions of Patterson and Hennessy's *Computer Organization and Design* (*COD*) for many years at Harvey Mudd College. We particularly like their coverage of the MIPS architecture and microarchitecture because MIPS is a commercially successful microprocessor architecture, yet it is simple enough to clearly explain and build in an introductory class. Because our class has no prerequisites, the first half of the semester is dedicated to digital design, which is not covered by *COD*. Other universities have indicated a need for a book that combines digital design and computer architecture. We have undertaken to prepare such a book.

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.

This book is suitable for a rapid-paced, single-semester introduction to digital design and computer architecture or for a two-quarter or two-semester sequence giving more time to digest the material and experiment in the lab. The course can be taught without prerequisites. The material is usually taught at the sophomore- or junior-year level, but may also be accessible to bright freshmen.

FEATURES

This book offers a number of special 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 applying to both HDLs, 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.

Classic MIPS Architecture and Microarchitecture

Chapters 6 and 7 focus on the MIPS architecture adapted from the treatment of Patterson and Hennessy. MIPS 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, hundreds of universities around the world have developed pedagogy, labs, and tools around the MIPS architecture.

Real-World Perspectives

Chapters 6, 7, and 8 illustrate the architecture, microarchitecture, and memory hierarchy of Intel x86 processors. Chapter 8 also describes peripherals in the context of Microchip's PIC32 microcontroller. 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 applying for work in the field. These questions provide a helpful glimpse into the types of problems job applicants will typically encounter during the interview process. (Exercise solutions are available via the book's companion and instructor webpages. For more details, see the next section, Online Supplements.)

ONLINE SUPPLEMENTS

Supplementary materials are available online at *textbooks.elsevier.com/9780123944245*. This companion site (accessible to all readers) includes:

- Solutions to odd-numbered exercises
- Links to professional-strength computer-aided design (CAD) tools from Altera® and Synopsys®
- Link to QtSpim (referred to generically as SPIM), a MIPS simulator
- Hardware description language (HDL) code for the MIPS processor
- Altera Quartus II helpful hints
- Microchip MPLAB IDE helpful hints
- Lecture slides in PowerPoint (PPT) format
- Sample course and lab materials
- List of errata

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

- Solutions to all exercises
- Links to professional-strength computer-aided design (CAD) tools from Altera® and Synopsys®. (Synopsys offers Synplify® Premier to qualified universities in a package of 50 licenses. For more information on the Synopsys University program, go to the instructor site for this book.)
- Figures from the text in JPG and PPT formats

Additional details on using the Altera, Synopsys, Microchip, and QtSpim tools in your course are provided in the next section. Details on the sample lab 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 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 Starter Edition degrades performance for simulations with more than 10,000 lines of HDL.

Microchip MPLAB IDE

Microchip MPLAB Integrated Development Environment (IDE) is a tool for programming PIC microcontrollers and is available for free download. MPLAB integrates program writing, compiling, simulating, and debugging into a single interface. It includes a C compiler and debugger, allowing the students to develop C and assembly programs, compile them, and optionally program them onto a PIC microcontroller.

Optional Tools: Synplify Premier and QtSpim

Synplify Premier and QtSpim are optional tools that can be used with this material.

The Synplify Premier product is a synthesis and debug environment for FPGA and CPLD design. Included is HDL Analyst, a unique graphical HDL analysis tool that automatically generates schematic views of the design with cross-probing back to the HDL source code. This is immensely useful in the learning and debugging process.

Synopsys offers Synplify Premier to qualified universities in a package of 50 licenses. For more information on the Synopsys University program or the Synopsys FPGA design software, visit the instructor site for this book (*textbooks.elsevier.com/9780123944245*).

QtSpim, also called simply SPIM, is a MIPS simulator that runs MIPS assembly code. Students enter their MIPS assembly code into a text file and run it using QtSpim. QtSpim displays the instructions, memory, and register values. Links to the user's manual and example files are available at the companion site (*textbooks.elsevier.com/9780123944245*).

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 Microchip MPLAB IDE.

After synthesis, students can implement their designs using the Altera DE2 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 Cyclone II Web Edition.

To run the labs, students will need to download and install Altera Quartus II Web Edition and Microchip MPLAB IDE. 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@onehotlogic.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

First and foremost, we thank David Patterson and John Hennessy for their pioneering MIPS microarchitectures described in their *Computer Organization and Design* textbook. We have taught from various editions of their book for many years. We appreciate their gracious support of this book and their permission to build on their microarchitectures.

Duane Bibby, our favorite cartoonist, labored long and hard to illustrate the fun and adventure of digital design. We also appreciate the enthusiasm of Nate McFadden, Todd Green, Danielle Miller, Robyn Day, and the rest of the team at Morgan Kaufmann who made this book happen.

We'd like to thank Matthew Watkins who contributed the section on Heterogeneous Multiprocessors in Chapter 7. We also appreciate the

work of Chris Parks, Carl Pearson, and Johnathan Chai who tested code and developed content for the second edition.

Numerous reviewers substantially improved the book. They include 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, and an anonymous reviewer.

We also appreciate the students in our course at Harvey Mudd College who have given us helpful feedback on drafts of this textbook. Of special note are 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.