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 only prerequisite is basic familiarity with a high-level programming language such as C, C++, or Java. The material is usually taught at the sophomore- or junior-year level, but may also be accessible to bright freshmen who have some programming experience.

FEATURES

This book offers a number of special features.

Side-by-Side Coverage of Verilog 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, Verilog 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 Verilog and VHDL, enabling the reader to quickly compare and contrast 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 IA-32 processors. These real-world perspective chapters show how the concepts in the chapter relate to the chips found in most PCs.

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 Web pages. For more details, see the next section, Online Supplements.)

ONLINE SUPPLEMENTS

Supplementary materials are available online at *textbooks.elsevier.com/9780123704979*. This companion site (accessible to all readers) includes:

- Solutions to odd-numbered exercises
- Links to professional-strength computer-aided design (CAD) tools from Xilinx® and Synplicity®
- Link to PCSPIM, a Windows-based MIPS simulator
- Hardware description language (HDL) code for the MIPS processor
- Xilinx Project Navigator 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 even-numbered exercises
- Links to professional-strength computer-aided design (CAD) tools from Xilinx® and Synplicity®. (Instructors from qualified universities can access *free* Synplicity tools for use in their classroom and laboratories. More details are available at the instructor site.)
- Figures from the text in JPG and PPT formats

Additional details on using the Xilinx, Synplicity, and PCSPIM 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

Xilinx ISE WebPACK

Xilinx ISE WebPACK is a free version of the professional-strength Xilinx ISE Foundation FPGA design tools. It allows students to enter their digital designs in schematic or using either the Verilog or VHDL hardware description language (HDL). After entering the design, students can simulate their circuits using ModelSim MXE III Starter, which is included in the Xilinx WebPACK. Xilinx WebPACK also includes XST, a logic synthesis tool supporting both Verilog and VHDL.

The difference between WebPACK and Foundation is that WebPACK supports a subset of the most common Xilinx FPGAs. The difference between ModelSim MXE III Starter and ModelSim commercial versions is that Starter degrades performance for simulations with more than 10,000 lines of HDL.

Synplify Pro

Synplify Pro® is a high-performance, sophisticated logic synthesis engine for FPGA and CPLD designs. Synplify Pro also contains HDL Analyst, a graphical interface tool that generates schematic views of the HDL source code. We have found that this is immensely useful in the learning and debugging process.

Synplicity has generously agreed to donate Synplify Pro to qualified universities and will provide as many licenses as needed to fill university labs. Instructors should visit the instructor Web page for this text for more information on how to request Synplify Pro licenses. For additional information on Synplicity and its other software, visit *www.synplicity.com/university*.

PCSPIM

PCSPIM, also called simply SPIM, is a Windows-based MIPS simulator that runs MIPS assembly code. Students enter their MIPS assembly code into a text file and run it using PCSPIM. PCSPIM displays the instructions, memory, and register values. Links to the user's manual and an example file are available at the companion site (textbooks.elsevier.com/9780123704979).

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 Xilinx WebPACK or Foundation tools to enter, simulate, synthesize, and implement their designs. The labs also include topics on assembly language programming using the PCSPIM simulator.

After synthesis, students can implement their designs using the Digilent Spartan 3 Starter Board or the XUP-Virtex 2 Pro (V2Pro) Board. Both of these powerful and competitively priced boards are available from *www.digilentinc.com*. The boards contain FPGAs that can be programmed to implement student designs. We provide labs that describe how to implement a selection of designs using Digilent's Spartan 3 Board using

WebPACK. Unfortunately, Xilinx WebPACK does not support the huge FPGA on the V2Pro board. Qualified universities may contact the Xilinx University Program to request a donation of the full Foundation tools.

To run the labs, students will need to download and install the Xilinx WebPACK, PCSPIM, and possibly Synplify Pro. Instructors may also choose to install the tools on lab machines. The labs include instructions on how to implement the projects on the Digilent's Spartan 3 Starter Board. The implementation step may be skipped, but we have found it of great value. The labs will also work with the XST synthesis tool, but we recommend using Synplify Pro because the schematics it produces give students invaluable feedback.

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 Web page.

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! (Be sure to include your mailing address.)

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 Denise Penrose, Nate McFadden, and the rest of the team at Morgan Kaufmann who made this book happen. Jeff Somers at Graphic World Publishing Services has ably guided the book through production.

Numerous reviewers have substantially improved the book. They include John Barr (Ithaca College), Jack V. Briner (Charleston Southern University), Andrew C. Brown (SK Communications), Carl Baumgaertner (Harvey Mudd College), A. Utku Diril (Nvidia Corporation), Jim Frenzel (University of Idaho), Jaeha Kim (Rambus, Inc.), Phillip King

(ShotSpotter, Inc.), James Pinter-Lucke (Claremont McKenna College), Amir Roth, Z. Jerry Shi (University of Connecticut), James E. Stine (Oklahoma State University), Luke Teyssier, Peiyi Zhao (Chapman University), and an anonymous reviewer. Simon Moore was a wonderful host during David's sabbatical visit to Cambridge University, where major sections of this book were written.

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 Casey Schilling, Alice Clifton, Chris Acon, and Stephen Brawner.

I, David, particularly thank my wife, Jennifer, who gave birth to our son Abraham at the beginning of the project. I appreciate her patience and loving support through yet another project at a busy time in our lives.