PREFACE

Changes to the SOPC Edition

Rapid Prototyping of Digital Systems provides an exciting and challenging laboratory component for undergraduate digital logic and computer design courses using FPGAs and CAD tools for simulation and hardware implementation. The more advanced topics and exercises also make this text useful for upper level courses in digital logic, programmable logic, and embedded systems. The SOPC edition includes Altera's new Quartus II CAD tool and includes laboratory projects for Altera's DE2 and the new DE1 FPGA boards. Student laboratory projects provided on the book's DVD include video graphics and text, mouse and keyboard input, and several computer designs.

Rapid Prototyping of Digital Systems includes four tutorials on the Altera Quartus II and Nios II tool environment, an overview of programmable logic, and IP cores with several easy-to-use input and output functions. These features were developed to help students get started quickly. Early design examples use schematic capture and IP cores developed for the Altera UP and DE FPGA boards. VHDL is used for more complex designs after a short introduction to VHDL-based synthesis. Verilog is also now supported as an option for the student projects.

New chapters in this edition provide an overview of System-On-a-Programmable Chip (SOPC) technology and SOPC design examples for the DE1 & 2 boards using Altera's new Nios II Processor hardware, the C software development tools, an overview of OS support for SOPC, and the uClinux operating system. A full set of Altera's FPGA CAD tools is included on the book's DVD.

Intended Audience

This text is intended to provide an exciting and challenging laboratory component for an undergraduate digital logic design class. The more advanced topics and exercises are also appropriate for consideration at schools that have an upper level course in digital logic or programmable logic. There are a number of excellent texts on digital logic design. For the most part, these texts do not include or fully integrate modern CAD tools, logic simulation, logic synthesis using hardware description languages, design hierarchy, current generation field programmable gate array (FPGA) technology and SOPC design. The goal of this text is to introduce these topics in the laboratory portion of the course. Even student laboratory projects can now implement entire digital and computer systems with hundreds of thousands of gates.

Over the past eight years, we have developed a number of interesting and challenging laboratory projects involving serial communications, state machines with video output, video games and graphics, simple computers, keyboard and mouse interfaces, robotics, and pipelined RISC processor cores.

Source files and additional example files are available on the DVD for all designs presented in the text. The student version of the PC based CAD tool on the DVD can be freely distributed to students. Students can purchase their own FPGA board for little more than the price of a contemporary textbook. As an alternative, a few of the low-cost FPGA boards can be shared among students in a laboratory. Course instructors should contact the Altera University Program for detailed information on obtaining full versions of the CAD tools for laboratory PCs and educational FPGA boards for student laboratories.

Topic Selection and Organization

Chapter 1 is a short CAD tool tutorial that covers design entry, simulation, and hardware implementation using an FPGA. The majority of students can enter the design, simulate, and have the design successfully running on the FPGA board in less than thirty minutes. After working through the tutorial and becoming familiar with the process, similar designs can be accomplished in less than 10 minutes.

Chapter 2 provides an overview of the various FPGA development boards. The features of each board are briefly described. Several tables listing pin connections of various I/O devices serve as an essential reference whenever a hardware design is implemented on the DE1, DE2, UP3, or UP 2 FPGA boards.

Chapter 3 is an introduction to programmable logic technology. The capabilities and internal architectures of the most popular CPLDs and FPGAs are described. These include the Cyclone FPGA used on the FPGA board, and the Xilinx 4000 family FPGAs.

Chapter 4 is a short CAD tool tutorial that serves as both a hierarchical and sequential design example. A counter is clocked by a pushbutton and the output is displayed in the seven-segment LEDs. The design is downloaded to the FPGA board and some real world timing issues arising from switch contact bounce are resolved. It uses several functions from the FPGAcore library which greatly simplify use of the FPGA's input and output capabilities.

Chapter 5 describes the available FPGAcore library I/O functions. The I/O devices include switches, the LCD, a decoder for seven segment LEDs, a multiple output clock divider, VGA output, keyboard input, and mouse input.

Chapter 6 is an introduction to the use of VHDL for the synthesis of digital hardware. Rather than a lengthy description of syntax details, models of the commonly used digital hardware devices are developed and presented. Most VHDL textbooks use models developed only for simulation and frequently use language features not supported in synthesis tools. Our easy to understand synthesis examples were developed and tested on FPGAs using the Altera CAD tools.

Chapter 7 is an introduction to the use of Verilog for the synthesis of digital hardware. The same hardware designs as Chapter 6 as modeled in Verilog. It is optional, but is included for those who would like an introduction to Verilog.

Chapter 8 is a state machine design example. The state machine controls a virtual electric train simulation with video output generated directly by the FPGA. Using track sensor input, students must control two trains and three track switches to avoid collisions. An actual model train layout can also built using the new digital DCC trains interfaced to an FPGA board.

Chapter 9 develops a model of a simple computer. The fetch, decode, and execute cycle is introduced and a brief model of the computer is developed using VHDL. A short assembly language program can be entered in the FPGA's internal memory and executed in the simulator.

Chapter 10 describes how to design an FPGA-based digital system to output VGA video. Numerous design examples are presented containing video with both text and graphics. Fundamental design issues in writing simple video games and graphics using an FPGA board are examined.

Chapter 11 describes the PS/2 keyboard and mouse operation and presents interface examples for integrating designs on an FPGA board. Keyboard scan code tables, mouse data packets, commands, status codes, and the serial communications protocol are included. VHDL code for a keyboard and mouse interface is also presented.

Chapter 12 describes several of the common I/O standards that are likely to be encountered in FPGA systems. Parallel, RS232 serial, SPI, and I^2C standards and interfacing are discussed.

Chapter 13 develops a design for an adaptable mobile robot using an FPGA board as the controller. Servo motors and several sensor technologies for a low cost mobile robot are described. A sample servo driver design is presented. Commercially available parts to construct the robot described can be obtained for as little as $60. Several robots can be built for use in the laboratory. Students with their own FPGA board may choose to build their own robot following the detailed instructions found in section 13.6.

Chapter 14 describes a single clock cycle model of the MIPS RISC processor based on the hardware implementation presented in the widely used Patterson and Hennessy textbook, *Computer Organization and Design the Hardware/Software Interface*. Laboratory exercises that add new instructions, features, and pipelining are included at the end of the chapter.

Chapters 15, 16, and 17 introduce students to SOPC design using the Nios II RISC processor core. Chapter 15 is an overview of the SOPC design approach. Chapter 16 contains a tutorial for the Nios II IDE software development tool and examples using the Nios II C/C++ compiler. Chapter 17 contains a tutorial on the processor core hardware configuration tool, SOPC builder. A DE2, DE1, or FPGA board is required for this new material since it is not supported on the UP2 or UP1's smaller FPGA.

Chapter 18 is new to the fourth edition and introduces students to a Linux based Real-Time Operating System (RTOS). A tutorial shows how the μClinux OS can be ported to the DE2 and DE1 FPGA boards.

We anticipate that some schools will still choose to begin with TTL designs on a small protoboard for the first few labs. The first chapter can be started at this time since only OR and NOT logic functions are used to introduce the CAD tool environment. The CAD tool can also be used for simulation of TTL labs, since a TTL parts library is included.

Even though VHDL and Verilog are complex languages, we have found after several years of experimentation that students can write HDL models to synthesize hardware designs after a short overview with a few basic hardware design examples. The use of HDL templates and online help files in the CAD tool make this process easier. After the initial experience with HDL synthesis, students dislike the use of schematic capture on larger designs since it can be time consuming. Experience in industry has been much the same since large productivity gains have been achieved using HDL based synthesis tools for FPGAs and Application Specific Integrated Circuits (ASICs).

Most digital logic classes include a simple computer design such as the one presented in Chapter 9 or a RISC processor such as the one presented in Chapter 14. If this is not covered in the first digital logic course, it could be used as a lab component for a subsequent computer architecture class.

A typical quarter or semester length course could not cover all of the topics presented. The material in Chapters 7 through 17 can be used on a selective basis. The keyboard and mouse are supported by FPGAcore library functions, and the material presented in Chapter 11 is not required to use these library functions for keyboard or mouse input. A DE1, DE2, or FPGA board is required for the SOPC Nios designs in Chapters 16 and 17.

A video game based on the material in Chapter 10 can serve as the basis for a final design project. We use robots with sensors from Chapter 13 that are controlled by the simple computer in Chapter 9. Students really enjoy working with the robot, and it presents almost infinite possibilities for an exciting design competition. More advanced classes might want to develop projects based on the Nios II processor reference design in Chapter 16 and 17 using C/C++ code or use the uClinux material in Chapter 18 to develop more complex application programs for embedded devices.

Software and Hardware Packages

We recommend the use of the new 7.1 SP1 web version of Quartus II FPGA CAD included with this book; all exercises were tested using this version. FPGA boards are available from the Altera University Program at special student pricing. Although boards can be easily shared among several students in a lab setting, pricing makes it possible for students who would like to purchase their own to do so.

Details and suggestions for additional cables that may be required for a laboratory setup can be found in Section 2.4. Source files for all designs presented in the text are available on the DVD.

Additional Web Material and Resources

There is a web site for the text with additional course materials, slides, text errata, and software updates at:

http://www.ece.gatech.edu/users/hamblen/book/book4e.htm

Acknowledgments

Over three thousand students and several hundred teaching assistants have contributed to this work during the past eight years. In particular, we would like to acknowledge Doug McAlister, Michael Sugg, Jurgen Vogel, Greg Ruhl, Eric Van Heest, Mitch Kispet, Evan Anderson, Zachary Folkerts, and Nick Clark for their help in testing and developing several of the laboratory assignments and tools. Stephen Brown, Mike Phipps, Joe Hanson, Tawfiq Mossadak, and Eric Shiflet at Altera provided software, hardware, helpful advice, and encouragement.

Springer

http://www.springer.com/978-0-387-72670-0

Rapid Prototyping of Digital Systems
SOPC Edition
Hamblen, J.O.; Hall, T.S.; Furman, M.D.
2008, XVII, 411 p. With online files/update., Softcover
ISBN: 978-0-387-72670-0