PREFACE

An *SoC* (system on a chip) integrates a processor, memory modules, I/O peripherals, and custom hardware accelerators into a single integrated circuit. As the capacity of *FPGA* (field-programmable gate array) devices continues to grow, the same design methodology can be realized in an FPGA chip and is sometimes known as *SoPC* (system on a programmable chip). In a traditional embedded system, the hardware is constructed around a fixed-sized processor and off-the-shelf peripherals and the software is customized to implement the desired functionalities. The emerging SoPC-based design provides a new alternative. Because of the programmability of FPGA devices, *customized hardware* can be incorporated into the embedded system as well. We can tailor the processor, select only the needed I/O peripherals, create a custom I/O interface, and develop specialized hardware accelerators for computation-intensive tasks.

The current development of *HDL* (hardware description language) synthesis and FPGA devices and the availability of soft-core processors allow designers to quickly develop and simulate custom hardware and software, realize the entire system on a prototyping device, and verify the operation of the physical implementation. We can now use a PC and an inexpensive FPGA prototyping board to construct a sophisticated embedded system. This book uses a "learning by doing" approach and illustrates the hardware and software design and development process by a series of examples. An Altera FPGA prototyping board and its *Nios II soft-core processor* are used for this purpose.

The book is divided into four major parts. Part I covers HDL and synthesis of custom hardware. Part II provides an overview of embedded software development with the emphasis on low-level I/O access and drivers. Part III demonstrates the

design and development of hardware and software for several complex I/O peripherals, including a PS2 keyboard and mouse, a graphic video controller, an audio codec, and an SD (secure digital) card. Part IV provides several case studies of the integration of hardware accelerators, including a custom GCD (greatest common divisor) circuit, a Mandelbrot set fractal circuit, and an audio synthesizer based on DDFS (direct digital frequency synthesis) methodology. All the hardware and software examples can be synthesized, compiled, and physically tested on the prototyping board.

Focus and audience

Focus The embedded system is studied extensively and many books cover this subject. The coverage is mostly focused on software development, usually around a specific processor. The new "hardware programmability" of the SoPC platform provides a new dimension on the embedded system development. This book mainly focuses on this aspect and the relevant design issues, including the derivation of a soft-core processor and *IP* (*intellectual property*) core based system, the partition and integration of software and hardware, and the development of custom I/O peripherals and hardware accelerators.

Audience and prerequisites The intended audience is students in an advanced digital design, embedded system, or software–hardware codesign course as well as practicing engineers who wish to learn FPGA-, HDL-, and SoPC-based development. Readers need to have a basic knowledge of digital systems, usually a required course in electrical engineering and computer engineering curricula, and a working knowledge of the C language. Prior exposure to computer architecture, microcontroller, and operating system is not necessary but will be helpful.

Logistics

FPGA prototyping board This book is prepared to be used with an Altera *DE1* board (also known as the *Cyclone II FPGA Starter Development Kit*) and *DE2* board. All HDL and C codes and discussions can be applied to the two boards directly. Most peripherals discussed in this book are de facto industrial standards, and the corresponding codes can be used as long as a board contains an Altera FPGA device and provides proper analog interface circuits and connectors.

PC accessories The design examples include interfaces to several PC peripheral devices. A PS2 keyboard, a PS2 mouse, a VGA compatible monitor, a pair of earphones or powered speakers, and an SD card are required for the respective I/O peripherals. These accessories are widely available and probably can be obtained from an old PC.

Software Several Altera software packages are needed for the Nios II-based system: *Quartus II Web edition*, which performs HDL synthesis; *SOPC Builder*, which configures and creates a Nios II-based system; *Nios EDS* (*embedded design suite*), which is the integrated software development platform; and *ModelSim-Altera Starter Edition*, which performs HDL simulation. They can be downloaded from Altera's website.

Codes and tutorials The HDL and C codes of the book can be obtained from the companion website. The codes and tutorials are developed and tested with *Altera Quartus II Web Edition v10 sp1* and *Altera Nios II EDS v10 sp1*. The software packages are running under Windows 7 32-bit with administrator privileges. Minor differences in the procedure may occur for other versions and operating systems.

Book organization

The book consists of four parts plus an introductory chapter. It starts with the "big picture":

- Chapter 1 provides an overview of the embedded system and introduces the SoPC concept and development flow.

Part I introduces the basic HDL constructs and synthesis procedure and demonstrates the construction of custom digital circuits. It consists of seven chapters:

- Chapter 2 describes the skeleton of an HDL program, basic language syntax, and logical operators. Gate-level combinational circuits are derived with these language constructs.
- Chapter 3 provides an overview of an FPGA device, prototyping board, and development flow. The development process is demonstrated by a tutorial of the Altera Quartus II synthesis software.
- Chapter 4 introduces HDL's relational and arithmetic operators and routing constructs. These correspond to medium-sized components, such as comparators, adders, and multiplexers. Module-level combinational circuits are derived with these language constructs.
- Chapter 5 presents the description of memory elements and the construction of "regular" sequential circuits, such as counters and shift registers, in which the state transitions exhibit a regular pattern, as well as a discussion of the use and inference of Cyclone II device's internal memory modules.
- Chapter 6 discusses the construction of a finite state machine (FSM), which is a sequential circuit whose state transitions do not exhibit a simple, regular pattern.
- Chapter 7 presents the construction of an FSM with data path (FSMD). The FSMD is used to implement register transfer (RT) methodology, in which the system operation is described by data transfers and manipulations among registers.
- Chapter 8 discusses several more advanced topics on language constructs and coding techniques and introduces the development of more sophisticated testbenches. This chapter can be skipped without affecting the remaining chapters.

Part II introduces the construction of a Nios II-based system and the development of embedded software. A simple flashing-LED design is used to illustrate the key concepts of this process. It consists of five chapters:

- Chapter 9 provides an overview of the Nios II soft-core processor and examines its key components.
- Chapter 10 introduces the construction of a Nios II-based system and the basic coding techniques to access low-level I/O peripherals. The derivation of

hardware and software is demonstrated by a tutorial of Altera SOPC Builder and Nios II EDS, respectively.

- Chapter 11 examines the structure and use of several IP cores (i.e., predesigned I/O peripherals) of SOPC Builder and covers the development of ad hoc I/O driver software routines.
- Chapter 12 provides an overview of the Altera *HAL* (*hardware abstraction layer*) run-time environment and illustrates its usage.
- Chapter 13 discusses the interrupt structure, including the operation of Nios II's interrupt controller and the development of software interrupt service routines.

Part III applies the techniques from Parts I and II to design an array of peripheral modules on the prototyping board. Each module consists of custom hardware and a basic software driver. These can be considered primitive IP cores and can be incorporated into a larger project. Part III consists of seven chapters:

- Chapter 14 demonstrates the I/O interfacing with PIO IP cores. This scheme can be used for simple I/O peripherals and can avoid the overhead of creating a new SoPC component.
- Chapter 15 gives an overview of Altera's *Avalon interface*, which functions as a "bus structure" for a Nios II processor to connect memory and I/O modules, and demonstrates the procedure of creating a customized IP core.
- Chapter 16 covers the interface to the external SRAM (static RAM) and SDRAM (synchronous dynamic RAM) devices and the basic testing procedure.
- Chapter 17 covers the design of the PS2 interface. The hardware portion consists of a PS2 controller to generate and process the PS2 clock and data signals. The software portion is composed of two sets of drivers: one for the PS2 keyboard, which reads and decodes scan codes from a keyboard, and one for the PS2 mouse, which obtains and processes the button and movement information from a mouse.
- Chapter 18 presents the design and implementation of a graphic video controller. The hardware portion covers the generation of video synchronization signals and the construction and interface of a custom SRAM-based video memory module. The software portion covers the basic driver routines to draw pixels and to display and process bitmap images and texts.
- Chapter 19 discusses the design of the audio codec chip interface. The hardware portion consists of an I^2C bus controller for codec configuration and a serial bus controller to transmit and receive digitalized audio data streams. The software portion is composed of routines to set codec parameters and to generate and record the audio data.
- Chapter 20 presents the design of the SD card interface. The hardware portion is done by an SPI bus controller and the software portion consists of driver routines for card initialization and basic file read and write operations.

Part IV presents three case studies of hardware accelerators, which utilize custom hardware to perform computation-intensive tasks. It includes three chapters:

- Chapter 21 shows the design of a custom GCD (greatest common divisor) accelerator based on the binary Euclid algorithm. Its performance is compared with software-based implementation.

- Chapter 22 illustrates the construction and integration of a Mandelbrot set fractal accelerator, which can select any portion of the set and displays the fractal on a VGA screen.
- Chapter 23 discusses the implementation of a direct digital frequency synthesis and modulation circuit. The circuit is used for an audio synthesizer with adjustable envelops.

Companion Website

An accompanying website (http://academic.csuohio.edu/chu_p/rtl) provides additional information, including the following materials:

- Errata
- Code listing and relevant files
- Links to Altera software
- Links to referenced materials
- Additional project ideas

Errata The book is self-prepared, which means that the author has produced all aspects of the text, including illustrations, tables, code listings, indexing, and formatting. As errors are always bound to happen, the accompanying web site provides an updated errata sheet and a place to report errors.

P. P. Chu

Cleveland, Ohio
January, 2012