A Short Course on Implementing FPGA Based Digital Systems

G.R. Jagadeesh, S.K. Lam and T. Srikanthan
Centre for High Performance Embedded Systems
Nanyang Technological University
50, Nanyang Drive, Singapore
[asgeorge, assklam, astsrikan]@ntu.edu.sg

Abstract

The rapid advances in the FPGA technology along with high-levels of system integration have made FPGAs the preferred platform not only for rapid prototyping but also for production of digital embedded systems. This paper presents the experience of a team of instructors in designing and conducting a short course on implementing FPGA-based digital systems for industry professionals. The selection of topics, course organization, the issues involved in designing effective hands-on exercises and the response of the students to the course are discussed.

1. Introduction

Over the last two decades, there has been a rapid expansion in the capabilities and use of Field Programmable Gate Arrays (FPGA). The recent advances in the FPGA technology and the incorporation of dedicated functional blocks such as arithmetic units, memory, clocking and processors are increasingly justifying FPGAs to become the central component in more and more digital systems [1]. Also, the rapid advances in Electronic Design Automation (EDA) tools for implementing FPGA based designs have made it possible to design complex digital systems in a short time-to-market window. It has therefore become indispensable for digital designers to acquaint themselves with the latest developments in FPGA technology and the EDA tools used for implementing digital systems on FPGA.

Due to their programmable nature, FPGAs have long been a preferred platform for rapid prototyping of digital systems. With the latest FPGAs achieving high levels of system integration, it has become possible to prototype large and complex digital systems on a single chip. This has caused a large number of hardware designers practicing non-FPGA based design methodologies to turn to FPGAs as an efficient means to validate their designs.

Design and development of FPGA-based embedded systems is a major focus of the Centre for High Performance Embedded Systems (CHiPES) [2] at Nanyang Technological University, Singapore. Based on requests and queries from many quarters, CHiPES decided to develop and offer a short course on implementing FPGA-based digital systems for working industry professionals. It was planned to pitch the course at the beginner to intermediate level. No prerequisites were required of the participants other than familiarity with digital design. Since it is not feasible for working professionals to be away from their workplaces for long durations, it was decided to limit the course to three days.

This paper describes the process of developing and delivering this course. The topics covered in the course and its organization are described in Section 2, followed by a coverage of the issues involved in designing suitable hands-on exercises for a course of this nature. The results and insights gained after two runs of the course are presented in Section 4. Section 5 concludes the paper.

2. Course coverage and organization

The topics to be covered in the course were decided after careful consideration of the likely backgrounds of the prospective students. To a large extent, the course design was influenced by the feedback provided by students who attended an earlier course on FPGA-based rapid prototyping conducted by CHiPES. It appeared likely that a considerable number of students attending the course would have little or no familiarity with FPGAs and hence a greater emphasis was placed on imparting the fundamentals relating to FPGA technology, architecture and standard FPGA design flow. At the same time, the need to provide an introduction to relatively advanced topics such as utilization of on-chip resources, in-circuit debugging and integration of FPGAs into board-level designs was also recognized. The course covers the following topics, each of which is organized as a separate lecture module.

Programmable Logic Technologies: The course opens with an introduction to technologies that help realize programmability and architectures of simple and complex programmable logic devices (SPLDs and CPLDs). This is followed by a coverage of basic FPGA architecture, state-of-the-art FPGA devices and FPGA trends.

Technologies for Implementing Digital Systems: This module was included with the view of comparing and contrasting the FPGA technology with the other major option available for implementing digital systems, namely
Application Specific Integrated Circuits (ASIC). This module also includes an introduction to Electronic Design Automation.

**FPGA Front-End Design Flow:** The topics covered in this module include design architecture, design entry, logic synthesis and design verification through simulation. Wherever necessary, Xilinx Integrated Software Environment (ISE) is used as a background to explain the concepts.

**Design Implementation and Device Configuration:** This unit provides a detailed coverage of the concepts involved in the downstream processes in the FPGA design flow such as translation, mapping, placement, routing and device programming. In addition, setting design constraints and analyzing design performance based on the reports produced by the implementation tools are also explained.

**Efficient Utilization of FPGA Features:** The latest FPGA devices abound in terms of on-chip resources such as clock distribution resources, digital clock manager, block memory and embedded multipliers. Efficient ways of utilizing these resources to improve design performance are described in this module. The concept of FPGA-specific IP-cores is covered along with the Xilinx Core Generator tool, which delivers a library of parameterizable IP cores. Several examples are included in this unit to illustrate how knowledge of the FPGA’s architecture can help to create designs with better performance.

**Advanced Debugging Techniques:** This unit introduces in-circuit debugging and imparts a working knowledge of Xilinx ChipScope Pro, a powerful tool for on-chip debugging.

**Incorporating FPGA-based Digital Systems:** The process of integrating FPGAs into board level designs and the various considerations involved are covered in detail. The module also covers auto-configuration of FPGAs for standalone operation.

It is widely accepted that practice-oriented pedagogy is the preferred approach while designing technical courses for working professionals [3]. Having long hours of lectures, especially in the afternoon sessions when attention spans are typically low, can be taxing to the participants. Therefore, it was decided to limit the lectures to the morning sessions and devote the post-lunch sessions to laboratory-based hands-on exercises, which seek to strengthen the understanding of the concepts covered during the lectures.

### 3. Design of hands-on exercises

CHiPES is equipped with state-of-the-art FPGA design tools from all the major EDA tool manufacturers and development boards incorporating a variety of FPGA devices from leading FPGA makers Xilinx and Altera. The hands-on exercises for this course make use of EDA tools from Xilinx and Mentor Graphics. The Celoxica RC200 board [4], which incorporates a Xilinx Virtex II FPGA chip along with a variety of peripheral chips, is used as the primary development platform. The RC200 board used in the course is shown in Figure 1.

**Figure 1. The Celoxica RC200 board used in the course**

The team of instructors who designed the hands-on exercises were guided by the objective of exposing the participants to all the major steps in the FPGA design flow. The primary aim was to develop laboratory exercises corresponding to each of the blocks shown in Figure 2. However, the practical realization of this objective involved some hurdles.

**Figure 2. Standard FPGA design flow**

The FPGA design tools require the design to be represented in the form of a hardware description language (HDL) such as
This suggests that the design entry step would involve writing HDL code for a given design specification. However, most of the students are unlikely to be familiar with VHDL. Instructors who have conducted courses with FPGA-based laboratory sessions have reported that students new to VHDL find it most difficult to comprehend and make the mistake of treating it as a programming language, whereas it is actually a textual form of coding the schematics and state machines [6]. Given the short duration of the course, it is not feasible to impart the basics of VHDL-based digital design to the students. Hence, there was a need to devise a way to enable the students to perform the design entry step, without being burdened by their lack of HDL knowledge.

It was decided to set up a hands-on exercise in which the students would create a design using a tool that supports alternative design entry methods. The HDL Designer Series [7] from Mentor Graphics is one such tool which supports graphical design entry using block diagrams, state diagrams, algorithmic state machines, flowcharts, truth tables and any combination of them. The exercise requires the students to architect and graphically create a design for generating the Fibonacci number series and verify its correctness using the ModelSim simulator.

For the exercises corresponding to the other steps in the standard FPGA design flow, a different design example was used. The design incorporates an Universal Asynchronous Receiver Transmitter (UART) for serial data transfer. In its basic form, the FPGA-based design receives ASCII characters through the UART interface, performs some simple computations, and produces ASCII characters as output. The VHDL codes that describe this design are pre-written and provided to the students, who take the design through functional simulation, logic synthesis, post-synthesis gate-level simulation, constraints specification, design implementation, post-layout timing simulation and programming file generation. Subsequently, the FPGA device on the development board is configured with the generated programming file. The Xilinx ISE software is used for performing the above steps. The design is tested by connecting the board to the serial port of a PC and using the HyperTerminal program on the PC to send input characters to the FPGA and to verify the outputs.

In addition to the above design exercise which covers all aspects of the standard FPGA design flow, the following lab assignments were included.

**Design Enhancement with On-Chip Features:** In this exercise, more functionality is added to the earlier UART-based design in order to enable it to function as a calculator, which responds to character sequences such as "7 + 5 =" received through the serial interface and outputs the result on a pair of seven-segment displays present on the board. This exercise requires the students to make use of the dedicated block memory and embedded multiplier present on the Virtex II FPGA. It also involves generating IP cores using the Xilinx Core Generator software and integrating the cores with the rest of the design.

**In-circuit Debugging with ChipScope Pro:** This lab assignment was included to impart a working knowledge of the Xilinx ChipScope Pro technology used for in-circuit verification. The participants are required to generate the Integrated Controller (ICON) and Integrated Logic Analyzer (ILA) cores and incorporate them into the design. With this setup, the ChipScope Pro Analyzer is used to monitor internal signals in the design and to set triggers on them.

**Standalone Deployment:** The purpose of this unit is to provide insights on how an FPGA can be integrated into a custom board-level design for standalone operation. For this lab assignment, a custom-made board that consists of a Xilinx Spartan FPGA, a Flash PROM and a DC power source is used. The FPGA receives character inputs from a keyboard connected to an on-board PS/2 connector, performs some basic computations, and sends output characters for display on a monitor connected to an on-board VGA connector. The exercise involves implementing the given design, programming the Flash PROM with the FPGA configuration data and verifying the standalone operation of the FPGA.

### 4. Results and insights

The course was launched in August 2007 and rerun in January 2008. All the students were working professionals from Singapore-based companies and research institutions. Most of the participants did not have any significant background in FPGA design but considered themselves very likely to work with FPGAs in the near future. A small number of students were already working on FPGA-based projects but enrolled for the course as they felt disadvantaged due to the lack of formal training. The students were largely interactive and involved.

It was decided to limit the number of students in the class to sixteen such that during the hands-on sessions at most two students will share one design station. Trainers well experienced in designing FPGA-based systems were present throughout the hands-on sessions to assist the students and answer any questions. Most of the groups took a total of about ten hours spanning three afternoon sessions to complete the laboratory assignments, while some groups with relatively experienced students were able to complete them considerably faster.

At the end of the course, the students were asked to fill in a course evaluation form. Although the lectures lasted for three hours on each morning session, most of the students found them interesting. On a scale of 1 (poor) to 5 (excellent), the students gave an average rating of 4.0 for the instructors’ ability to maintain class interest and attention. Regarding the relevance of the course contents to industry needs, the average rating was 3.8.
A session was organized at the end of the course, where students were invited to offer their feedback and to provide suggestions for improvement. While the feedback was generally positive, the suggestions received for improving the course reinforced the view that it is challenging to decide the right level at which the course should be taught for a class of students with very diverse backgrounds. For instance, while the beginner-level students found the exercises on the standard FPGA design flow helpful, those with some exposure to the FPGA design felt that the cookbook exercises should be replaced with more open-ended design problems. The dominant view that emerged was that the students largely found the course to be a good learning experience.

The requirements expressed by the students have led the instructors to create new short courses on other related topics. A number of students wished to learn VHDL. Some of them subsequently enrolled for a course on VHDL offered by CHiPES.

5. Conclusions

This paper has described the process of creating and delivering a three-day course on implementing FPGA-based digital systems for working professionals. The selection of topics to be taught in the course and the process of designing suitable hands-on exercises were presented in detail. The feedback received from the participants justifies the need for a course of this nature and confirms its relevance to the needs of the industry. The course is periodically refined based on student feedback and another run of the course is scheduled in the near future. It is hoped that the ideas expressed in this paper are of use to instructors wishing to conduct a similar course on FPGA design elsewhere.

6. References


