Reflections on Implementing and Teaching an Advanced Undergraduate Course in Embedded Systems

D. Helvick, R. Mercado, Z. Zhang, and D. T. Rover
Department of Electrical and Computer Engineering, Iowa State University, Ames, IA
{dhelvick, rmercado, zzhang, drover}@iastate.edu

Abstract

Many universities offer introductory courses that focus on microcontroller-based systems and embedded programming. Advanced course offerings vary, and are often not available until the graduate level, leaving a gap in training undergraduates. Courses are emerging that take advantage of new embedded development platforms that support hardware-software codesign. At Iowa State University, the Department of Electrical and Computer Engineering has developed a senior-level course on embedded systems design (CPRE 488) that sits between the introductory course on microcontrollers (CPRE 211) and a graduate course on system-level design (CPRE 588). The course has now been offered twice, during the fall 2005 and 2006 semesters. Reflections on course design and delivery are presented in this paper.

1. Introduction

Undergraduates are often introduced to microcontroller software design in an introductory course taken during the second year. The course typically deals with software and hardware issues, but leaves the two relatively separate. Because of the growing importance of system-level design methodologies, there is an increasing demand to incorporate aspects of hardware-software codesign into the curriculum. Indeed, several books are available which are geared towards undergraduate system-level embedded design courses [2, 3]. System-level design languages (SLDLs) such as SpecC [5] or SystemC [6] have also been used in undergraduate courses (e.g. [4]). However, advanced system-level design based on real hardware and software platforms is often taught only in graduate-level classes. Thus, there is a gap between the graduate level codesign course and the undergraduate introductory course relative to system-level design techniques. Such a course is needed by students pursuing both industrial career and graduate education.

We created a new course, CPRE 488 Embedded Systems Design, at Iowa State University, along with corresponding labs that strive to give students practical problems in a real-world environment while preparing them for future study in codesign methodologies. The FPGA-based laboratory is equipped with Xilinx Virtex II Pro boards, specifically the XC2VP7. The labs are designed to emphasize the issues involved in codesign, such as performance tradeoffs, timing/power constraints and communication between system components. These topics are explored in the realm of two main applications, a digital camera and an MP3 player. Additional details on the course and lab development are described in an earlier paper [1]. In this paper, based on two years of experience, we reflect on the course design and delivery with particular focus on student reactions, the laboratory experience, the course project, and lecture-lab integration. This paper, together with the previous one, should assist educators in offering embedded systems design courses.

2. Student Reaction to Course

We obtained student feedback through several channels: interactions with students in the classroom and labs, WebCT-based informal surveys, and a formal course evaluation. Generally, the students reacted very positively. The overall interest level of the course was high, particularly for the lab exercises and projects. Most students agreed that the learning objectives were met. All students are either juniors or seniors, and they felt the difficulty level was appropriate. A student who had the best overall performance in the class commented that the labs are best done in the junior or senior year because of their complexity.

Some students felt the lecture coverage was somewhat over-diversified. In addition, in the first offering in fall 2005, students dealt with bugs in our template code and the Xilinx XPS software, and most of those issues were solved before the second offering.

3. Laboratory Experiences

The laboratory exercise sequence for CPRE 488 was presented in detail in the earlier paper [1]. The labs are also available at the course website [8]. Several changes have been made to the content of the labs, particularly labs 7-9. Lab 7 provides a detailed introduction to the VxWorks Real-Time Operating System (RTOS), including topics such as task scheduling, interrupts, and intertask communication. Lab 8 involves the design of an MP3 system. The system runs on VxWorks and is provided to the students in a working form, though it does not meet performance requirements. The students are then asked to analyze the complex system, and to suggest and implement performance improvements to the code in an RTOS environment. Lab 9 focuses on networking in VxWorks with an exercise involving the design of a web server running on VxWorks.

The students generally reacted very favorably to the laboratory experience, largely due to their interest in implementing "real-world" applications such as the MP3 player and the digital camera. Though occasional problems arose with the software tools used in the laboratory, the students generally took this in stride, and often assisted in creating solutions. In some sense, this added to the real-life nature of the lab.

4. Digital Camera Laboratory Sequence

The digital camera laboratory sequence comprises three labs. To challenge the students through this sequence, time requirements that the system must meet are specified. The goal of Lab 4 is to profile a software-only implementation of a JPEG camera, which reads an uncompressed image from a camera, encodes it, and saves it to a Compact Flash (CF) card. A variety of profiling methods is used to identify candidate functions to be moved to hardware. In Lab 5, the students apply several optimization techniques to the candidate functions found previously. Examples of optimization techniques implemented by students include integer refinement and look-up tables. At the end of Lab 5, the students have not yet met the time requirements for the digital camera, even after implementing all optimizations. The next step, covered in Lab 6, is hardware acceleration. Students first learn hardware acceleration in Lab 3; in Lab 6 they apply it to the candidate function to meet timing requirements. After identifying the appropriate function to move to hardware, adding custom hardware, and implementing the design, students measure the system performance. When finished, the camera is fully functional, and capable of acquiring, compressing, and storing a picture in less than 1.5 seconds. Further details on the digital camera labs are available at the course website [8].

5. Course Project Experiences

The course project in CPRE 488 requires students to either expand on the laboratory exercises they have completed thus far in the course or to complete a project from scratch. Teams of 3-5 students are given approximately one month to complete a project. Several ideas are presented to students; for example, developing enhancements to the MP3 player previously developed in lab, modifying the digital camera to function as a web camera, using the board as a miniature recording studio, and using the camera as a security camera with motion-detection capabilities. The success of a project often depended on solving a few key problems.

The students reacted enthusiastically to the challenge of designing an advanced embedded system as the final course project, as well as the relative open-endedness of choosing their project topic. Most groups went beyond the expectations of the instructors by designing innovative systems within the time constraints of the course. The project was one of the most successful parts of the laboratory experience in the course.

6. Lecture-Laboratory Connection

In CPRE 488, like other embedded system courses that use contemporary toolsets and conventional textbooks, it is a challenge to synchronize lecture and lab content. The textbook [7] emphasizes a diverse array of subject matter, from computer architecture and compilers, to scheduling and testing. The labs require knowledge on topics such as hardware components and performance profiling. The labs also require an in-depth understanding of one or more applications. We reorganized the lectures so that the subjects referenced by the labs are introduced early in the semester. Within the first six weeks, embedded CPUs and memory components, common I/O devices, FPGA basics, hardware accelerators and compiler techniques are covered. The coverage of several topics was divided into two parts. For the OS and multiprocessor topics, we used the lab application examples whenever possible. We also connected the development and testing methods in the lectures with the project.

Acknowledgements

This work was partially supported under NSF grant No. 0431924, through support from Rockwell Collins Foundation, Xilinx, and GAANN grant from Dept. of Education to Information Infrastructure Institute at ISU. We gratefully acknowledge the contributions of J. Schneider, A. Larson, J. Boyd, and R. Walstrom to development of the laboratory.

References

[1] Schneider, J., Bezdek, M., Zhang, Z., Rover, D. T., *A* Platform FPGA-Based Hardware-Software Undergraduate Laboratory. *Proc. Microelectronic Sys. Educ*, 2005. pp 53-54

[2] J. Staunstrup and W. Wolf, *Hardware/Software Co-Design: Principles and Practice*, Kluwer Academic Publishers, 1997.

[3] F. Vahid and T. Givargis, *Embedded System Design: A Unified Hardware/Software Introduction*, Wiley & Sons, 2002.

[4] K. Sakiyama, P. Schaumont, D. Hwang, I. Verbauwhede, *Teaching trade-offs in system-level design Methodologies*. Proc Microelectronic Sys. Educ., 2003. pp. 62- 63.

[5] *SpecC website*, Available: http://www.specc.org

[6] *SystemC website*, Available: http://www.systemc.org

[7] Wayne Wolf, *Computers as Components,* Morgan Kaufmann Publishers, 2001.

[8] Cpr E 488 Website, Available: http://class.ece.iastate.edu/cpre488/06Fall