AC 2010-2266: INTRODUCING HYBRID DESIGN APPROACH AT THE UNDERGRADUATE LEVEL

Firas Hassan, Ohio Northern University

Dr. Firas Hassan is an assistant professor at Ohio Northern University. He finished his PhD studies at The University of Akron and worked for one year as a visiting professor. His area of research is hardware implementation of real-time embedded image processing algorithms

Srinivasa Vemuru, Ohio Northern University

Srinivasa Vemuru obtained his bachelors and masters degrees in Electrical Engineering from Indian Institute of Technology, Madras in 1984 and 1986, respectively. He received his PhD from the University of Toledo in 1991. From 1991-2001 he served as faculty member in the Department of Electrical Engineering at the City College of the City University of New York. He is currently an associate professor in the Department of Electrical & Computer Engineering and Computer Science at Ohio Northern University. His research and teaching interests are in the areas of analog and digital electronic circuit design, embedded systems, wireless sensor networks, built-in self test, and RF integrated circuits.

Introducing hybrid design approach at the undergraduate level

Abstract

Nowadays, embedded developers are designing their applications using a hybrid approach where the configurable components of the design are implemented in software, and the time critical components are implemented in hardware. Most of the universities, on the other hand, are still teaching these two design approaches separately. A typical electrical and computer engineering (ECE) program includes a class on embedded software design using microcontrollers and a class on hardware design using field programmable gate arrays (FPGAs). This paper explains a teaching plan to introduce this hybrid design approach at the undergraduate level. The plan was applied successfully in an elective class at the University of Akron. A similar approach is used in a required course for computer engineering students at the Ohio Northern University. This paper presents the teaching plans and experiences from the two course offerings.

Introduction

In today's quickly changing world, staying up-to-date is a recipe for success. Currently, embedded design companies are using a hybrid approach to implement configurable and real-time application. In such approach, the embedded developer can achieve configurability and real-time constraints on a single chip. The configurable components of the design are implemented in software, and the time critical components are implemented in hardware. Field programmable gate array (FPGA) providers such as Alter and Xilinx are offering soft cores for configurable processors[1,2]. Using these soft cores, the developer can select a processor that best fits his application without paying overhead or sacrificing performance. After building his own version of the configurable processor, the developer adds the hardware components of his design as peripheral devices surrounding the processor but on the same chip. The software part of the design runs on the processor itself. This important design concept can be easily introduced at the undergraduate level by using the teaching plan described in this paper.

There is a continuous expansion of the scope of electrical and computer engineering in the technology-oriented world. This requires the curriculum committees to look at a more efficient and effective means of covering these topics within a four-year undergraduate program to ensure that their graduates are well prepared to excel in the industrial/research world. There is a critical need to provide undergraduate electrical and computer engineering (ECE) students with an experience of studying the tradeoffs of hardware and software implementation in embedded systems[3-5]. A typical electrical and computer engineering (ECE) program includes a class on embedded software design using microcontrollers[6] and a class on hardware design using field programmable gate arrays[7]. In most universities one or both of these courses are required core courses. Hardware/software co-design approaches have been used at other universities targeting undergraduates and graduate students[8-16]. This paper explains a teaching plan to introduce this hybrid design approach at the undergraduate level that introduces both these topics in a single course. The plan was applied successfully in an elective class at the University of Akron . A similar approach is used in a required course for computer engineering students at the Ohio

Northern University. This paper presents the experiences in development of the lab plans for the two course offerings.

Approach at the University of Akron:

The teaching plan is divided into three main steps. In the first step, students will learn how to use the hardware descriptive languages at the behavioral level to describe their design totally in hardware. They also learn how to simulate their design using test benches and synthesize it on FPGAs. In the second step, they take one tutorial on system-on-programmable-chip (SOPC), which is the tool used by Altera to build a configurable processor inside an FPGA and a second tutorial on Altera Monitor program which is used to compile the software in the processor. In the third step, students will be asked to build an embedded application using the hybrid approach. The design is divided into a datapath which is implemented totally in hardware and a controller that is implemented in software inside the processor. For the instruction and data memory of the processor they use the on-chip memory within the FPGA. Describing the controller in software they will be able to configure their application. Details about each step will be described in the following section using the blackjack game as a design example.

Teaching plan

As mentioned before, the teaching plan was applied successfully in an elective class in the ECE department at the University of Akron. Although, the class contained lecture notes and lab session, the concentration in this paper will be on the lab sessions. The lab sessions were designed as one large project that ran through the whole semester. At the end of the semester students were able to build a configurable full version of the blackjack game using a hybrid design approach. All lab sessions were implemented on Altera's T-Rex C1 development board which is equipped with a Cyclone EP1C6Q2440C8 FPGA. The first four lab sessions were designed to help the students understand the first step of the teaching plan. In the first lab, they built the dealer's version of the blackjack game on the development board. The value of the hand and the flags were displayed on a seven segment display and light emitting diodes (LEDs), respectively. In the second lab session they learned how to use the FPGA and the development board as a graphics generator for a VGA monitor. In the third lab session they again implemented the dealer's version of the blackjack game but this time using the VGA monitor to display the value of the hand and flags. In the fourth lab, they created a full version of the blackjack game with one dealer and one player. The hand value and flags were again displayed on the VGA monitor. Next, students took two tutorials that helped them understand how to build a soft core processor inside the FPGA, integrate the soft core processor with their design project, and run a software application on the processor. In the fifth lab session, they repeated lab one using a hybrid design approach. The controller of the game was built as software inside the soft core processor that was integrated with their design project. For the remaining lab sessions, they were asked to use all the skills they learned to build a configurable full version of the blackjack game with multiple players and displayed on the VGA monitor.

Lab session one

In blackjack, the cards are valued as follows: The cards from 2 through 9 are valued as indicated. The 10, Jack, Queen, and King are all valued at 10. The suits of the cards do not have any meaning in the game. The value of a hand is simply the sum of the point counts of each card in the hand. For example, a hand containing (5, 7, 9) has the value of 21. The Ace can be counted as either 1 or 11. You need not specify which value the Ace has. It's assumed to always have the value that makes the best hand. The dealer must continue to take cards ("hit") until his total is 17 or greater. The player can choose to stop drawing cards (stand) at any value. If the hand is above 21, it is a bust.

In the first lab session, students were asked to implement the blackjack game on a development board applying only the dealer's rules. The block diagram of the design is shown in figure 1. The card values were entered using dip switches. One push button was used to start a new game and a second push button was used to read the card value from the dip switches. The value of the hand was displayed on a seven segment display. The flags including bust, hit, stand and isAce were displayed on led diodes.

From the beginning, the student learned how to split the design between datapath and controller components. The datapath should generate the sum of the hand and the four different flags. The controller should decide when to reset the datapath, when to read the card, and when to subtract ten from the card's total.

Students started by designing the different components of the game. For this lab session, the controller was implemented using a regular state machine. Next, they translated the design using a hardware descriptive language at the behavioral level. Then, they built a testbench to check the different scenarios of the game. After that, they used the testbench to simulate the design using a simulation tool such as Modelsim. Finally, they synthesized their design and download it to the FPGA on the development board. The game was tested in real-time using different sets of hands.

Lab session two

In this lab students learned how to build a graphics generator for a VGA monitor. The block diagram of the design is shown in figure 2. The only input to the design is the VGA pixel clock which is derived from the 25.200 MHz clock in the Cyclone phase locked loop. The horizontal sync pulse and vertical sync pulse are provided to the monitor so that it may position the 640 by 480 pixel image in the center of the monitor screen. Each horizontal sync pulse causes the monitor to retrace the color gun from the right side of the screen to the left side of the screen. The monitor then automatically moves the next trace of color gun down one line and scans for left to right on the screen. The horizontal information to the VGA monitor is based on a total of 800 clock pulses. A horizontal counter is used to keep track of these clock pulses. The vertical information is based on 525 total horizontal scan lines. A vertical counter is used to keep track of these scan lines. The vertical sync pulse causes the monitor to retrace the color gun from the bottom to the top of the monitor screen.

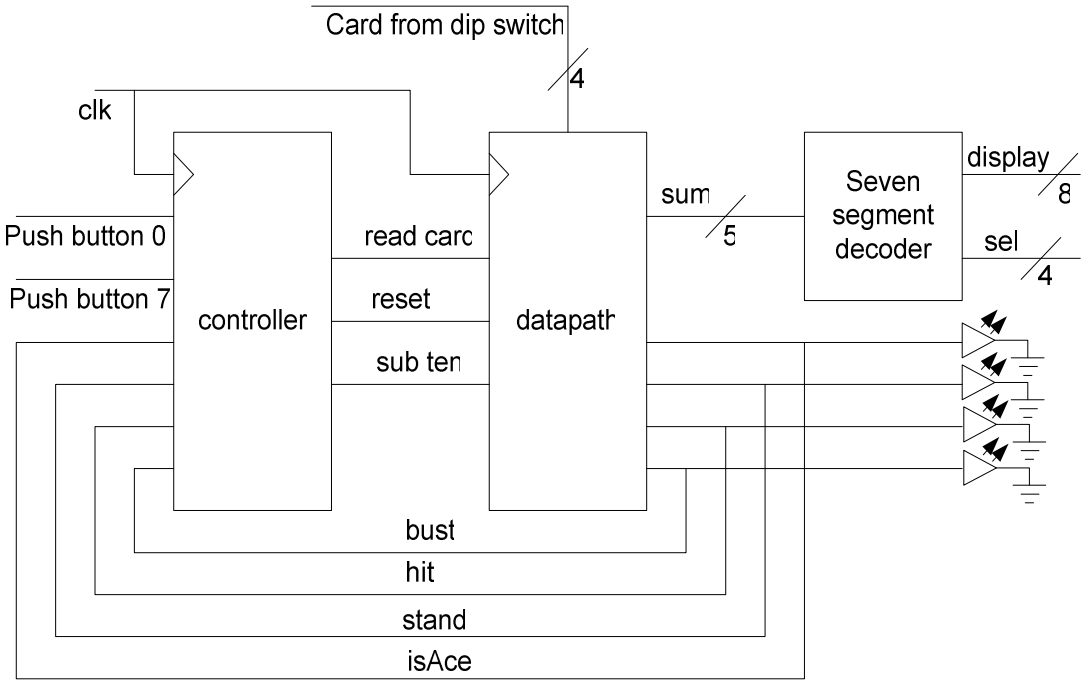

Figure 1 Block diagram of the blackjack game

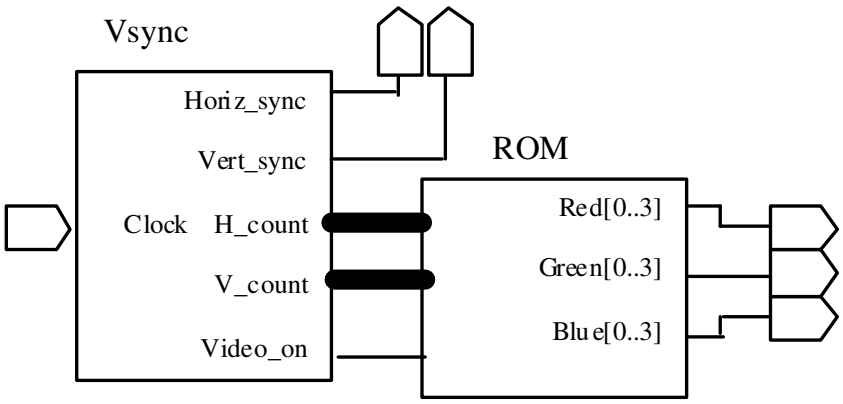

Figure 2 Block diagram of the graphics generator for the VGA display

Picture elements or pixels are generated on the screen by controlling the value of the red, blue and green signals. One turns on the appropriate pixel color and zero turns it off.

Students were asked to create a color image of their choice of 64×32 pixels. They converted the image to a constant array of 12 bit RGB values. These values were used as a 2^{11}×12 bit read only memory (ROM) inside their design. The address of the memory was created using different combination of the vertical and horizontal counter. At the end of this project, students were able to display their image at different positions in the screen.

Lab session three

In this lab session, students combined the different components they created in the first two sessions to implement the dealer's version of the blackjack game and display it on the VGA monitor. Each group of students was asked to divide the available screen space to be able to display all the different flags and the hand value simultaneously. They created their own images of the different numbers and flags. Then, they convert their images into constant arrays that were installed inside the FPGA as ROMs. The most difficult part of this project was to build the controller that should generate all the different addresses for the ROMs to be able to display all the images on the screen.

Lab session four

The purpose of this lab was to build a full version of the blackjack game with one dealer and one player displayed on the VGA monitor. As described before, the player's rules are slightly different from the dealer's rules. Each set of rules was implemented using a separate controller. A third controller was built to generate the different images to be displayed on the screen similar to the previous lab session. Finally, a main controller was built to switch between the player and the dealer and decide the winner. At the end of this lab session, students should understand how difficult it is to write all these different controllers using the regular state machines. They would appreciate a hybrid approach, where they can write these controllers as software that runs on a regular processor. All the datapath components of the design including counters, memories, comparators, etc… can still be implemented in hardware.

Tutorials on hybrid design approach

Next, students were asked to complete the necessary tutorials to understand the hybrid design concept. A good tutorial provided by Altera is, "*Introduction to the Altera SOPC Builder Using VHDL Design*"[17]. This tutorial is divided into four sections. The first section gives a necessary summary of a soft processor, defined in a hardware description language, which can be implemented in Altera's FPGA devices by using their computer aided design (CAD) software. The second section describes Altera's SOPC Builder. This tool allows the user to easily create his own version of the soft processor, by simply selecting the desired functional units and specifying their parameters. In the third section, the students learn how to integrate the soft processor into their design project. In the fourth section, the students learn how to run a very simple application program on the processor. Altera provides monitor software, Altera *Monitor Program*, that provides a simple means for compiling, assembling and downloading of programs into the soft processor. It also makes it possible for the user to perform debugging tasks. A description of this software is available in the *Altera Monitor Program* tutorial[18]. After completing this lab session, students learned how to configure their own soft processor in the FPGA and run a simple application. An example of this simple application would be lighting a light emitting diode (LED) using one of the push button on the development board.

Lab session five

In this lab session, students redesigned the dealer's version of the blackjack game using a hybrid approach. The block diagram of the design is similar to figure 1. However, in this case the controller was built as software inside the soft processer and integrated with the datapath component of the design. Students started by using the SOPC builder to create their own version of the soft processor. For the instruction and data memory of the processor they used the on-chip memory within the FPGA. The processor should have at least six inputs and three outputs. They selected to use eight bit input port and eight bit output port. After this step, the processor would be available as a regular component that can be added easily to the design. Next, students described the high level entity that connects the datapath component from lab one to the soft processor. Students then synthesize the hardware on the FPGA. The processor now can be accessed from the host computer through the JTAG interface. Finally, students wrote a C program that describes the controller and use the Altera monitor software to run the program on the processor. The design was tested in real-time using different sets of hands.

The main project

For the main project, students started by redesigning lab session four in a hybrid approach. All four controllers were described in software. Describing the controller in software they were able to configure the rules of the game and the number of players. Real-time components such as the VGA display were implemented using hardware components surrounding the processor. This was a good practical example for them to understand the beauty of the hybrid design approach.

Approach 2 at Ohio Northern University:

The Embedded Computing Systems course at Ohio Northern University is a required course for computer engineering majors and is an elective for electrical engineering and computer science majors. Students enter this class with knowledge of VHDL from sophomore level introductory digital logic course. The course focuses on getting a hardware/software codesign experience through a combination of lectures and laboratory sessions. The laboratory component will be the focus of the paper. The course has the following learning outcomes:

Laboratory learning outcomes:

L1. Students will develop a capability of embedded systems design by programming a microcontroller board using C with modern development tools.
L2. Students will develop a custom reconfigurable embedded platform using FPGA based development boards including appropriate drivers for peripherals.

Course Outcomes:

After successfully completing this course the student should be able to:
C1. Understand bottom-up and top-down design principles in embedded system design.
C2. Be able to compare and contrast distinct instruction set architectures.
C3. Understand the input/output mechanisms in CPUs.
C4. Understand how architecture affects program performance and power consumption.

C5. Understand the operation of CPU buses, I/O devices and interfacing to design an embedded computer platform.
C6. Understand program design and models of programs.
C7. Optimize programs for speed, size and power consumption
C8. Understand principles of real-time operating systems

The laboratory component of the course comprises of two parts:

1. Introduction to embedded C programming
2. Development of embedded platform to study hardware/software tradeoffs.

Laboratories 1-3:

The focus of these laboratory sessions is to provide experience with embedded C programming. In the three laboratories students are introduced to Codewarrior development environment targeting Freescale HCS12 microcontrollers. Using an existing microcontroller platform allows students to focus on programming features. In the three laboratory sessions students develop

- Waveform generation using output compare features of microcontroller
- Interrupt driven I/O
- Interface a keypad and LCD display to the microcontroller board.

This provides the students with software development experience for the embedded platform to be developed to study hardware/software codesign.

Laboratories 4-6:

In the fourth laboratory students develop an embedded platform using Xilinx EDK development suite targeting FPGA (Spartan 3) development boards. The embedded platform comprises of the Microblaze soft processor along with necessary peripherals obtained from the intellectual property (IP) cores provided as a part of the development system. This platform is used to communicate with the serial port communications on the monitor.

The fifth laboratory involves interfacing LCD display to the FPGA board. This involves the development of custom hardware IP written in VHDL and device drivers to communicate with the LCD display. The students also use the Platform Studio software development kit (SDK) that comes a part of the Xilinx EDK package[19].

In the sixth laboratory, hardware and software debugging is performed through the use of Xilinx Microprocessor Debugger and Chipscope utility[20]. The students complete the interface of the keyboard to the embedded platform and display the entered values on the LCD display.

Student surveys were conducted during the end of the course offering to assess their opinion of learning course outcomes. The results of the survey are summarized in table below questions on a scale of 1-5.

TABLE 1. Summary of Course Learning Outcome

Outcome	Very Unsatisfied (1)	Unsatisfied (2)	Neutral (3)	Satisfied (4)	Very Satisfied (5)	Average
L1	0	0	0	6	16	4.72
L2	0	0	2	5	15	4.59
C1	0	0	1	5	15	4.66
C2	0	0	6	8	8	4.09
C3	0	0	0	8	14	4.63
C4	0	0	8	8	6	3.90
C5	0	0	1	5	16	4.68
C6	0	2	6	9	5	3.77
C7	0	0	10	8	4	3.72
C8	0	0	4	10	8	4.18

Faculty Course Assessment Reports (FCAR) are used at Ohio Northern University to assess the effectiveness of the course outcomes based on multiple assessment points used throughout the course[21,22]. The summary is based on the categorization of student performance on a four scale approach as given in Table II. The assessment results over the past two years are presented in the table.

TABLE 2. FCAR results over a two year period

Outcome	Effective (3)	Acceptable (2)	Minimal (1)	Unsatisfactory (0)	Average
L1	14	0	0	0	3.00
L2	6	8	0	0	2.42
C1	8	4	1	1	2.35
C2	7	6	0	1	2.35
C3	7	7	0	0	2.50
C4	8	5	1	0	2.50
C5	10	3	0	1	2.57
C6	10	3	1	0	2.64
C7	11	2	0	1	2.64
C8	8	4	1	1	2.35

Assessment from the student survey clearly indicates that all the students are overall more than satisfied with the course and laboratory learning outcomes. FCAR report averages also indicate that students have better than acceptable performance in meeting the course outcomes. Further enhancements in the laboratory experience is to integrate a model-driven embedded system design experience using UML (Rational Rhapsody) and enforcing real-time constraints through the use of real-time operating systems[23].

Conclusion

In this paper, we described teaching plans to introduce the hybrid design approach at the undergraduate level from two different universities. These plans were successfully applied in a

required/elective classes in the ECE department at The University of Akron and Ohio Northern University. The paper includes a detailed description of the laboratory plans from these courses.

Future plans include introducing other projects that make use of the interface capabilities of the configurable processors especially with external memory devices. Frame grabbers for example are usually built in hardware because of the real-time video constraints. However, it is very difficult to describe these devices in hardware without making use of the external memory interface provided with the soft cores of the configurable processors.

References

1. NIOS Embedded Design Tools, http://www.altera.com/products/ip/processors/nios2/tools/ni2-development_tools.html
2. Xilinx EDK, www.xilinx.com/ise/embedded/edk_docs.htm
3. D. Jackson and P. Caspi, Embedded Systems education: Future directions, initiatives and cooperation, ACM SIGBED Proceeedings of 2005 Workshop on Embedded Systems Education
4. T. Henzinger and J. Sifakis, The Discipline of Embedded Systems Design, IEEE Computer, vol. 40, no. 10, October 2010, pp. 32-40.
5. P. Koopman, H. Choset, R. Gandhi, B. Krogh, et al, Undergraduate embedded system education at Carnegie Mellon, ACM Transactions on Embedded Computing Systems, vol 5, no. 3, August 2005, pp. 500-528.
6. H. Broberg and E. Thompson, Selection of processor, language, and labs in introductory microprocessor/microcontroller courses, Proceedings of the American Society for Engineering Education & Exposition, 2006-275, 2005.
7. R. Hayne, VHDL Projects to reinforce computer architecture classtroom instruction, Computers in Education Journal, vol XVIII, no. 1, Apr-Jun 2008, pp 98-112.
8. T. Hall and J. Hamblen, System-on-a-programmable-chip development platforms in the classroom, IEEE Transactions on Education, vol. 47, no. 4, November 2004, pp. 502-507.
9. M. G. Morrow, Creating a realistic embedded systems design experience for computer engineers, Proceedings of the American Society for Engineering Education & Exposition, Session 1532, 2005.
10. A. L. Sangiovanni-Vincentelli and A. Pinto, An overview of embedded system design education at Berkeley, ACM Transactions on Embedded Computing Systems, vol. 4, no.3, August 2005, pp. 472-499.
11. S. M. Loo, On the use of a soft processor core in computer engineering education, Proceedings of the American Society for Engineering Education & Exposition, 2006-275, 2006.
12. R. Blaine, C. Grecu, A. Ivanov, and R. Turner, "An FPGA Design Project: Creating a PowerPC subsystem plus user logic, IEEE Transactions on Education, vol. 51, no. 3, pp 32-318, August 2008.
13. J. Bowles and G. Quan, An FPGA-based embedded-system design laboratory for the undergraduate computer engineering curriculum, Proceedings of the American Society for Engineering Education & Exposition, Paper 2009-761, 2009.
14. C. Korpela, and R. McTasney, An FPGA multiprocessor system for undergraduate study, Proceedings of the American Society for Engineering Education & Exposition, Paper 2009-2481, 2009.
15. L. Slivovsky and A. Liddicoat, Future pedagogical trends in the microprocessor course – the soft core processor, Proceedings of the 36th Frontier in Education Conference, paper M4D 1-2, 2006.
16. M. McDermott, J. Abraham, and M. Ravel, Balancing cirtual and physical prototyping across a multicourse VLSI/Embedded-systems/SOC design curriculum, Proceedings of the American Society for Engineering Education & Exposition, Session 2009-2349, 2009.
17. ftp://ftp.altera.com/up/pub/Tutorials/DE2/Computer_Organization/tut_sopc_introduction_vhdl.pdf
18. ftp://ftp.altera.com/up/pub/altera_monitor_program/tut_Altera_Monitor_Program.pdf
19. http://www.xilinx.com/tools/feature/csi/sdk.htm
20. Xilinx Chipscope, http://www.xilinx.com/tools/cspro.htm
21. J. Estell, The faculty course assessment report, IEEE Frontiers in Education Conference, 2003, pp. T4B1-T4B8.

22. J. Estell, Streamling the assessment process using the faculty course assessment report, The Internaltional Journal of Engineering Education, vol 25, no. 5, pp 941-951
23. IBM Rational Rhapsody, www.ibm.com/software/awdtools/rhapsody/