Use of FPGAs in A Digital System Design Course with Computer Gaming Applications

Dr. Cheng Chih Liu, University of Wisconsin-Stout

Cheng Liu is an Associate Professor in the Computer and Electrical Engineering Program at University of Wisconsin Stout. He taught courses in computer and electrical engineering program. His teaching and research interests are FPGA based digital systems, microprocessor system design, and embedded systems.

Use of FPGAs in A Digital System Design Course with Computer Gaming Applications

This article discusses our continuing efforts to integrate the Field Programmable Gate Arrays (FPGAs) into the digital system design course. The course follows on from the first-year digital logic course. Programmable gate arrays had previously been introduced in our senior-level embedded system course, but the recent trend toward applications in industry demanded that FPGAs be introduced earlier in the Electrical and Computer Engineering curricula. In the fall semester of 2009 we began introducing FPGAs into the digital system design course. This allowed our second-year students to explore high-level circuit designs with a state-of-the-art Computer-Aid Design suite (CAD) with Xilinx's FPGAs.

In this work, we describe the revision for the digital system design course starting in 2015, including software and hardware upgrades that improved our hands-on laboratory exercises. Because Xilinx ISE 14.7 version no longer supported newer FPGAs devices, we adopted the Xilinx Artix-7 FPGAs on the Basys-3 educational board and the Xilinx Vivado design suite.

We also provide some historical context regarding to the evolution of the laboratory exercises used for this course. Two new lab exercises were developed to address student concerns from the student survey in 2015, including introducing the hierarchical design flow for FPGAs earlier in the course, as well as lack of real-world examples in the lab exercises. In this paper, we describe two new computer gaming labs added in 2016 along with evaluation data showing marked improvement in students' perception of the course.

Introduction and Literature Review

Our university offers a Bachelor of Science degree in Computer Engineering. The digital logic course is the first course in our embedded system sequence. The digital system design course is typically offered in every fall semester. The class size for the digital system design course is between 18 to 25 students. The embedded system course sequence is listed below:

- Computer Science I (introducing C and C++ languages)
- Computer Science II (introducing Java language)
- Digital Logic
- Digital System Design (introducing Xilinx FPGAs and Verilog language)
- Microprocessor System Design (introducing Atmel and ARM Cortex processors)
- Embedded System Design (Altera FPGAs and ARM Cortex-A processors)

Recently, there has been a growing interest for various teaching approaches for embedded systems education. Kumar, *et al*. [8] taught students how to use the Xilinx Spartan-3E board to develop a five-a-side soccer game system. Several papers used video games as a teaching tool for embedded system education with FPGAs. One of the earliest papers by Jamieson used a 'Star Trek' game framework in the classroom [2]. FPGAs were used in the computer architecture classes [3]. Use of FPGAs for computer gaming applications has been one of good tools to introduce students to digital designs [1], [4] – [7]. The Basys-3 FPGAs board has a 12-bit VGA output, a USB HID host for mice, keyboard, and memory stick. These features are suitable for gaming console development with FPGAs. Sensors are now a vital part of any digital systems. A variety of digital and analog sensors including microphones, accelerometers, magnetometers, and gyroscopes are incorporated into our lab exercises. These sensor components are tangible and can be used to demonstrate the analog to digital conversion and serial communication techniques with the FPGAs. Our digital system design is a laboratory-based class in which students learn coding, simulation, and troubleshooting for circuits on the FPGAs and breadboard. The students were highly motivated with the course contents through hands-on learning. The course topics that helped our lab run smoothly while maximizing students' learning are listed in the Appendix I in this paper.

We introduced two popular arcade games, the maze game and tetris game (lab 8 and lab 9) using the Artix-7 Basys-3 FPGA board, so students were able to learn and created different games based off the two lab exercises. Derived from lab exercises 8 and 9, video games Pong [8], Crossy Road [9], Frogger [10], Space Invader [11] were developed by our students in their final projects. In the conclusion, this paper documents our revised lab exercises to integrate computer gaming into our digital system design course. Appendix II contains the student evaluations and student written comments in the fall semester of 2017.

Lab Exercise: Computer Gaming Development

Fig. 1a shows our video game development platform. Fig. 1b shows a simple maze game lab exercise. The maze game was the first video game introduced to our students. The tetris game has a more intensive algorithm using the finite state machine. Because of the games' control logic, simple operations, appropriate programming challenges, and strong entertainment, we used the maze and tetris games to convey many digital design concepts with the FPGAs, such as designing an interactive game interface to the FPGA with a 16-button keypad [13], a USB keyboard, and a LCD monitor. Students were given the high-level design specifications for the games, along with code templates with partially filled code. Code listing of video game components, in Fig. 2a, were provided as specifications that required the students to use and build a hierarchical design. Students then ran the tests and created a bit-stream file for the FPGA to verify their design functionality. Fig.1b shows the maze game and tetris game running on a standard VGA monitor.

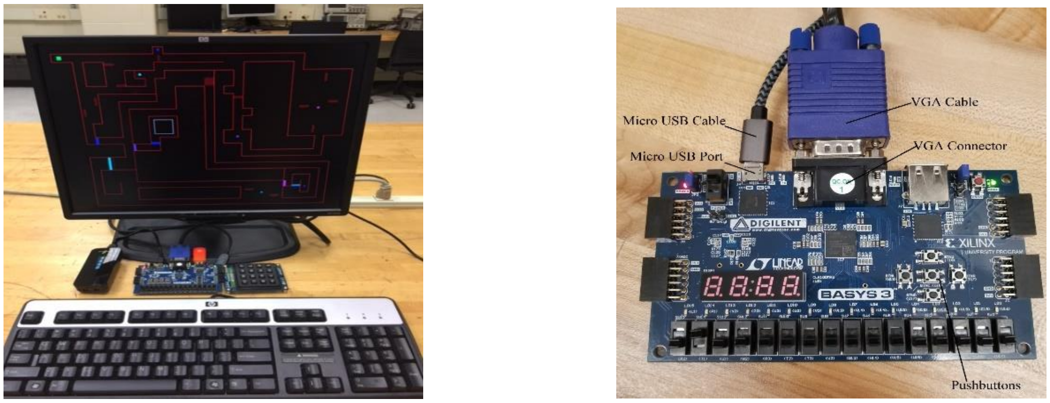


Fig. 1a video game system with Xilinx Artix-7 FPGA, Basys-3 Educational Board

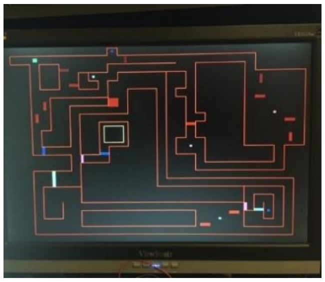


Fig. 1b the maze game lab to create moving obstacles

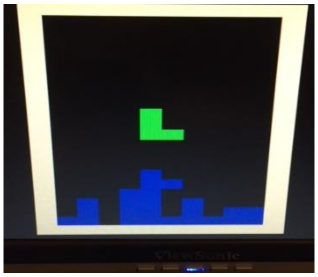


Fig. 1c the tetris game to rotate blocks

Laboratory Exercise 9 (The Tetris Game)

The following four subsections provide overviews of how the tetris game can be used to enable insightful experiments that both motivate and facilitate understanding of the core concepts of making a video game with the FPGAs.

1. Overview of Game Design Development

We developed the video game exercises so that students would be able to learn about hierarchy design concepts and use them in a different context. Tetris game is a classic television and handheld video game and has the functions of movement, speed control of blocks, rotation of blocks, randomly creating new blocks, eliminating rows, and counting scores. In our tetris game, game players can move and rotate moving blocks with four tactile push buttons on the Basys-3 FPGA board. The main body of HDL code is composed of three modules in a hierarchy design: game control module, VGA interface, and display module as illustrated in Fig. 2a.

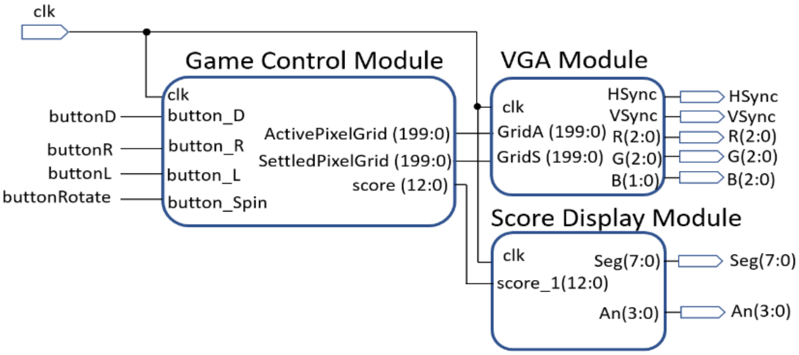
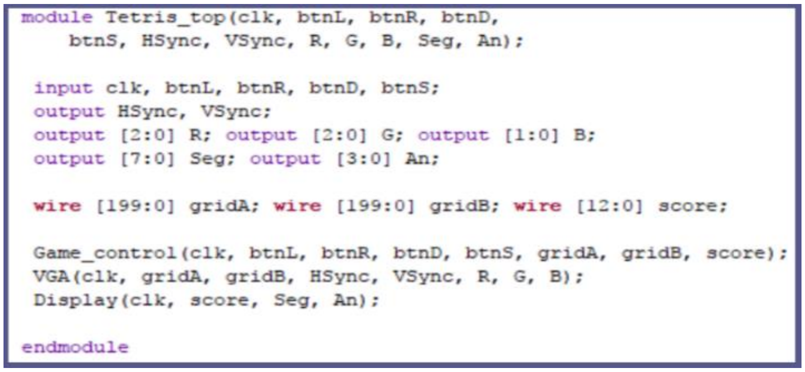


Fig. 2a Structure of the tetris game



```
module Tetris_top(clk, btnL, btnR, btnD,
    btnS, HSync, VSync, R, G, B, Seg, An);

 input clk, btnL, btnR, btnD, btnS;
 output HSync, VSync;
 output [2:0] R; output [2:0] G; output [1:0] B;
 output [7:0] Seg; output [3:0] An;

 wire [199:0] gridA; wire [199:0] gridB; wire [12:0] score;

 Game_control(clk, btnL, btnR, btnD, btnS, gridA, gridB, score);
 VGA(clk, gridA, gridB, HSync, VSync, R, G, B);
 Display(clk, score, Seg, An);

endmodule
```

Fig. 2b Hierarchy design of the Tetris game

The top-level Verilog module is illustrated in Fig.2b. Players use the keyboard module to control the game process and complete the movement and rotation of blocks with four push buttons (right, left, and down buttons) on the Basys-3 board. The user interface for the tetris game contains four tactile push buttons as shown in Table II. A USB interface keyboard or the NES Nintendo control pad can be used instead with some modifications of the HDL model in Fig. 2a and Fig. 2b. Players use four push buttons to control the movement of the game as shown in Table II.

Table II Tetris Game function table

Button (Basys-3)	Function Description
btnL	Move blocks to the left
btnR	Move blocks to the right
btnD	Move blocks to the bottom
btnS	Rotate blocks
btnC	Game reset

2. Tetris Game Lab Exercise A: Game Control Module in Fig. 2b

Fig. 3 shows the flow chart of the tetris game control module. This game control module has a total of 13 distinct states that interact with the user display module and VGA module. It keeps track of the game state and makes decisions and manipulates that game state based on its input. It produces 8-bit RGB color data for a VGA monitor to run at 640 by 480 pixels resolution. It receives five single bit input signals: the clock and four push buttons. The four buttons represent the four actions the user can take.

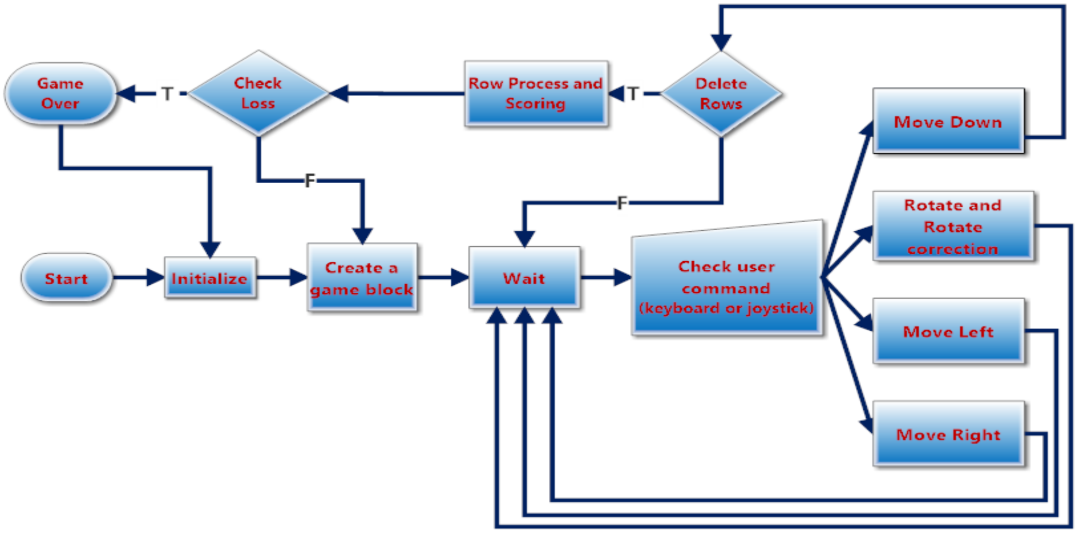


Fig. 3 flow chart of the tetris game

3. Tetris Game Lab Exercise B: Method of Displaying Blocks (VGA module in Fig. 2b)

The VGA module in the tetris game provides the coordinates and pixel locations for the settled and active blocks, then sends the information into the VGA control signal, resulting in the correct display of the image on a LCD monitor. “HSynch” and “VSync” signals are pulses used for the start of the horizonal scan line and vertical scan line of the monitor.

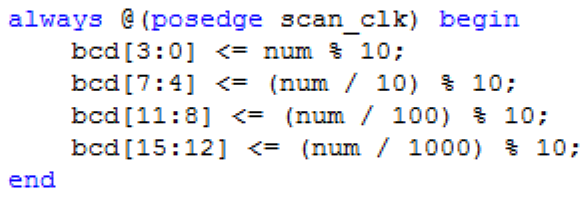
In our tetris game, the game display area consists of 640 by 480-pixel grid locations. The internal state is represented by a 200-bit value representing the grid of settled pieces, and a 230-bit value representing the active piece grid. As the active piece is moved (left, right, down), the shift amount is changed accordingly. To settle a piece to the bottom of the VGA screen, the 200 most significant bits of the active piece grid are combined with the settled piece grid using bitwise OR operator. To detect collisions, the bitwise AND operator can be used.

The state machine begins in the “Start” state and moves into the “Initialize” state when the user presses the “rotate block” button to start a new game. Then a new piece is generated by moving into the “creating a new piece” state. When moving into the “wait” state, it is necessary to add a delay before accepting any other input. This also acts as a way of de-bouncing input push button, so de-bouncing circuits aren’t needed on the input signals. When the user rotates a piece, it is possible for the active piece to rotate into a wall or overlap with an already settled piece. The “rotate correction” state is necessary to detect and avoid overlaps between blocks and walls.

When the user pushes the down button, the game enters the “Move Down” state. If the bitwise AND operator between this 200-bit sequence and the settled grid returns a value equal to the original sequence, then a filled row has been detected. This filled row can then be removed using the “exclusive OR” bitwise operator. This is repeated until all rows have been checked. A counter is used to check each row individually, from top to bottom. If the game should continue, then the game generates a new piece by entering the “Generate New Piece” state.

4. Tetris Game Lab Exercise C: Display Module

The display module is responsible for displaying the player’s score in the tetris game. The 7-segment display on the Basys-3 board is a four-digit common anode LED display. The common anode signals are available as four separate enable signals, and the seven cathodes are available as seven enable signals. This enables displaying multiple digits by scanning through each digit and illuminating the correct LEDs for that digit. To display a 13-bit binary number as a four-digit decimal number, it is required to calculate the value of each digit separately. Each digit can be stored in a four bit register and can be calculated using integer division and the modulus operator. The Verilog code for doing this is shown here:



```
always @(posedge scan_clk) begin
    bcd[3:0] <= num % 10;
    bcd[7:4] <= (num / 10) % 10;
    bcd[11:8] <= (num / 100) % 10;
    bcd[15:12] <= (num / 1000) % 10;
end
```

Student Final Project

The final project is the capstone for this class. Table III is a list of all video game projects in 2017. Table IV shows whether the lab assignments and student projects have a focus on gaming from 2015 to 2017. Each project group was formed by students and had between one to three students per group. Students can choose their project with a gaming focus or a project of other focus using the Basys-3 board. Table V shows the number of student projects with a gaming focus and other design focus, as well as the project evaluation outcome.

Table III List of video game projects in digital system design course in 2017

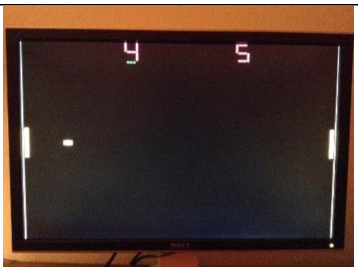
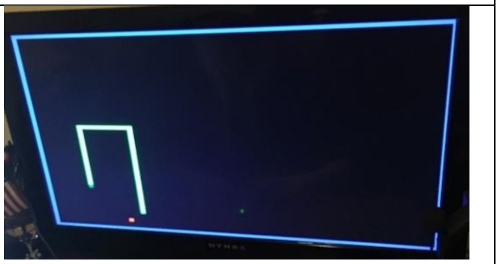
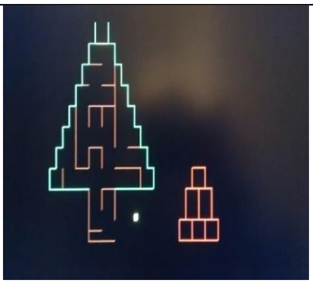
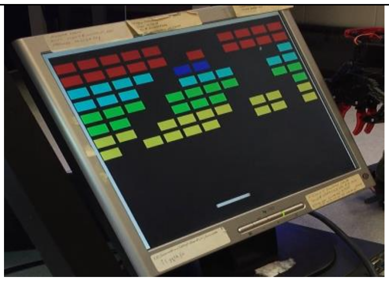
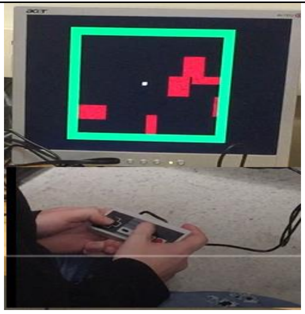
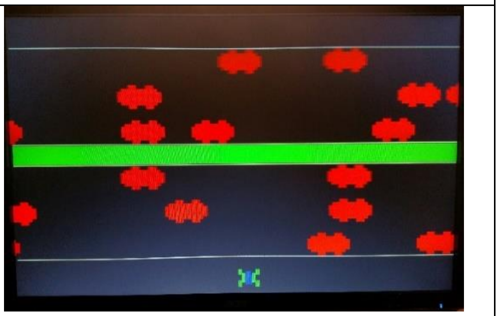
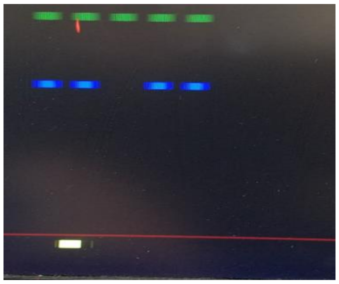
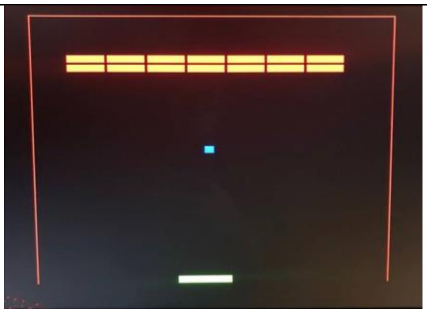
Project	Video Game Screen	Project	Video Game Screen
Pong Game – two player modes		Snake Game	
Christmas Maze		Atari Breakout Game	
Obstacle Avoidance Game with the Nintendo keypad controller		Crossy Road	
Space Invaders		Brick Breaker	

Table IV Use of gaming lab assignments and gaming project focus

Year	Game labs used	Number of student project with gaming focus
2015	No	None
2016	Yes	8
2017	Yes	8

Table V Number of student project and project evaluation

Year	Student project with gaming focus		
	Successful	Marginal (partial functionality)	Failed
2015	Not applicable	Not applicable	Not applicable
2016	7	1	0
2017	8	0	0
Year	Student project evaluation with other focus (student choice)		
	Successful	Marginal (partial functionality)	Failed
2015	9	4	3
2016	5	1	0
2017	5	2	0

Conclusion

Overall, introducing digital designs with gaming applications was instrumental for successful student projects that produced a better student learning outcome. The student survey in 2017 about the revised digital system design course is in the Appendix II. Evaluation scores on all surveyed aspects improved after the new gaming exercises were introduced. Although the group size was too small to assess true statistical significance, we were encouraged to see the evidence of the intended outcome. Ratings on a Linkert scale were 1(extremely well) to 5(extremely poorly). The ratings averaged between 2 (well) and 3 (acceptable) on all 6 survey questions in 2015. Mean ratings on all 6 questions improved to between 1 (extremely well) and 2 (well) in 2017. Of the interest to use were the substantial increases in the response to two survey items: “I would recommend this course to another student” (2.63 in 2015 versus 1.22 in 2017) and “The presentation of lab materials was clear, logical, effective and easily understood” (2.88 in 2015 versus 1.56 in 2017).

The best part of the course was student projects that showed a wide variety of uses for FPGAs. Students enjoyed building their own video games on the Basys-3 FPGAs boards, especially the fact that they could take it away when completing the course. The overall instructor evaluation in fall semester of 2017 and 2015 showed that the computer gaming labs had improved the effectiveness of teaching the digital system design course. The students provided feedback in their project reports that they learned the most from working their project and the project will help them in the future on FPGA designs.

Appendix I: List of laboratory exercises performed by the students

There are eleven laboratory exercises, where * denotes the new lab exercises added for 2017.
Lab 1 and lab 2: introduction to Vivado design flow. Topics include creation of test-bench files and constraint files, simulations and synthesis for simple combination circuits: NAND gate and

one-bit full adder. Bill of materials for the lab: a Basys-3 board with its onboard switches and LEDs [12].

Lab 3 and lab 4: introduction to hierarchy design using Vivado. Design of 16 and 32-bit ripple carry adders and look-ahead carry adders. Design of an arithmetic logic unit (ALU) that performs simple logic and arithmetic operations of two 32-bit numbers, including AND, OR, addition, and subtraction operations. Bill of Materials for the lab: Basys-3 board and testbench files.

Lab 5 and lab 6 Introduction to clock/frequency divider circuit, switch de-bounced circuit, knight-rider scanning LED light circuit, and two-digital counter with LED display. Bills of materials for the lab: Basys-3 board with its onboard push buttons and eight LEDs.

Lab 7 Introduction to a 16-button keypad and USB keyboard and their interfaces to the FPGAs. This lab involves with a keypad speed control for scanning LED lights. Bills of Materials for the lab: 16-button keypad [13] and basys-3 boards.

*Lab 8 (new lab in 2017) Introduction to the design of a VGA controller. A simple maze video game is introduced. Bill of materials for the lab: Basys-3 board, LCD monitor, and a video cable.

*Lab 9 (new lab in 2017) Introduction to the tetris game. The details on the tetris game is discussed in the laboratory exercise section. Bill of materials for the lab: Basys-3 board, LCD monitor, and a video cable.

Lab 10 introduction to motor technologies, discussion of stepper [14], servo [15], and DC motors [16] and their interfaces to the FPGAs. Bill of the materials for the lab:16-button keypad lock, motors, and Basys-3 board.

Lab 11 introduction to digital sensors and their communication protocols: Inter-Integrated circuit (I^2C) and Serial Peripheral Interface (SPI) protocols are introduced. Bill of materials for the lab: digital microphone [17], accelerometer [18], bluetooth [19], and Basys-3 board. A volume-meter [20] experiment is part of lab exercise performed by students. Audio signal strength received by the microphone can be displayed by the eight LEDs on the Basys-3 board.

Appendix II: Course Assessment and Instructor Evaluation

Two surveys were sent out to students in the fall semester 2015 and 2017 respectively. Lab 8 and lab 9 were added in the fall semester 2017. The student ratings were based on the Likert scale: 1 (extremely well), 2 (well), 3 (acceptable), 4 (rather poorly), 5(extremely poorly). The survey question table provides the mean and standard deviation for 2015 and 2017. Student survey for 2016 digital system design course was not conducted due to the time constraint. Summary of the results for 2015 and 2017 is presented in the abstract, introduction, as well as the conclusion sections.

#	**Survey Question**
1	The use of class time was effective
2	The lab materials were interesting and stimulating
3	The presentation of lab materials was clear, logical, effective and easily understood
4	The use of laboratory equipment, the FPGAs in the lab exercises were valuable, and helped me to better learn the materials

5	I would recommend this course to another student
6	Project evaluation system was always fair

The student response rates for the 2015 and 2017 were 49% and 48% respectively.

Survey question	Year	Mean	Std. dev
1	2015	2.38	0.52
	2017	1.56	0.68
2	2015	2.75	0.71
	2017	1.67	0.82
3	2015	**2.88**	0.35
	2017	**1.56**	0.50
4	2015	2.13	0.35
	2017	1.11	0.31
5	2015	**2.63**	0.74
	2017	**1.22**	0.42
6	2015	2.25	0.46
	2017	1.11	0.31

Questions 7 and 8 are open-ended survey questions. Each student uses their own words to answer the questions.

7	Based on this course, describe the instructor's strengths
8	Based on the course, describe the instructor's weakness

For question 7 in the fall 2017 survey, one student expressed "we learned a large variety of cool labs."; one student expressed concern regarding the reliability of the Vivado design suite with computer crushed several times during the semester. One student expressed "the lab was both interesting and challenging, I really learned a lot and all of it can be directly applied to industry."; one student expressed "the instructor values in class time and uses it to have the student perform hands-on laboratory experiments."; One student expressed "These experiments have the students understand the outcome and see the piece of technology in action. This is valuable experience considering we are here to work in the outside world of industry, and we gain bits of pieces of what we will be using in the future."; One student expressed "The projects we receive are always fun and interesting that really tests what we learned in class; a very enjoyable class; a decent teaching plan throughout the semester". One student expressed "class went at a good speed and gave enough time for projects to allow them to be completely understood before moving on". One student commented "discussing problems outside of class time can be extremely helpful".

For question 8 in the fall 2017 survey, one student expressed "sometimes I didn't understand fully what some of the code did and labs were not interesting", which we interpreted it as a request for providing more detailed descriptions on what the code does in our lab exercises; one student expressed "one problem was that most coding work to the lab was handed to students in a word document ", which we interpreted it as a request that the student wish to create his/her coding work from scratch, so he/she can learn how the Verilog programming language is structured, which is vital to any FPGA designs.

For question 8 in the fall 2015 survey, one student expressed “large jump in difficulty from the average class project to the final project”, which we interpreted it as a request for providing some intermediate steps to help completing their final project. One student expressed “someone who is used to teaching programming courses would probably be a better fit for this class”, which we interpreted it as a request to reallocate class time to teach basic programming in Verilog or have a separate programming class for Verilog programming language.

References

1. Cheng Liu, “Teaching Digital Designs by Building Small Autonomous Robotic Vehicles Using an FPGA Platform, *Proceedings of the 122nd ASEE Annual Conference and Exposition*, Seattle, WA. June 2015.
2. Peter Jamieson, “Early Project Based Learning Improvements via a “Star Trek Engineering Room” game framework, and competition,” *Proceedings of 41th ASEE/IEEE Frontiers in Education Conference*, October 12-15, 2011
3. Tyson S. Hall, *et al*., “System-on-a-Programmable-Chip Development Platforms in the Classroom,” *IEEE Transactions on Education*, Vol. 47, no. 4, August 2004.
4. José Nelson Amaral, “Teaching Digital Design to Computing Science Students in a Single Academic Term,” *IEEE Transactions on Education*, Vol. 48, no. 1, August 2005.
5. Hiroyasu, *et al*., “Use of Student Experiments for Teaching Embedded Software Including HW/SW Co-Design,” *IEEE Transactions on Education*, Vol. 52, no. 3, August 2009.
6. Akash Kumar, *et al*., “Project-Based Learning in Embedded Systems Education Using an FPGA Platform,” *IEEE Transactions on Education*, Vol. 56, no. 4, August 2013.
7. Christos Ttofis, *et al*., “FPGA-Based Laboratory Assignments for NoC-Based Many Core Systems, “*IEEE Transactions on Education*, Vol. 55, no. 2, August 2012.
8. “Pong game”, https://en.wikipedia.org/wiki/Pong
9. “Crossy road game”, https://en.wikipedia.org/wiki/Crossy_Road
10. “Frogger game”, https://en.wikipedia.org/wiki/Frogger
11. “Space Invaders”, https://en.wikipedia.org/wiki/Space_Invaders
12. Digilent Basys-3 Artix-7 FPGA board: https://store.digilentinc.com/basys-3-artix-7-fpga-trainer-board-recommended-for-introductory-users/
13. PMOD 16 button keypad: https://store.digilentinc.com/pmod-kypd-16-button-keypad/
14. Unipolar stepper motor, https://www.jameco.com/z/42BYGH404-R-Unipolar-Stepper-Motor-12VDC-0-4A_238538.html
15. Parallax servo motor, https://www.digikey.com/products/en?mpart=900-00008&v=149
16. DC motor, https://store.digilentinc.com/motor-gearbox-1-19-gear-ratio-custom-12v-motor-designed-for-digilent-robot-kits/
17. Digital microphone, https://store.digilentinc.com/pmod-mic3-mems-microphone-with-adjustable-gain/
18. Accelerometer, https://store.digilentinc.com/pmod-acl2-3-axis-mems-accelerometer/
19. Bluetooth, https://store.digilentinc.com/pmod-bt2-bluetooth-interface/
20. Volume meter, https://en.wikipedia.org/wiki/VU_meter