Use of a CPLD in an Introductory Logic Circuits Course with Software and Hardware Upgrade

Dr. Ying Yu, University of Hartford

Dr. Ying Yu received her B.Eng. from Fudan University, Shanghai, China, in 2000. She received her M.Eng. and Ph.D. in Electrical Engineering from Brown University, R.I., USA, in 2003 and 2007, respectively. Since 2008, she has been teaching as an assistant professor of the Department of Electrical and Computer Engineering at University of Hartford. Her current research interests are audio and speech signal processing, acoustic scene classification, speaker identification and verification, and teaching with new educational methods, including peer instruction, personal response systems, video games, and state-of-the-art CAD tools.

Dr. Krista M Hill, University of Hartford

Dr. Krista M. Hill is an associate professor in Electrical and Computer Engineering at the University of Hartford in Connecticut. PhD and MSEE from Worcester Polytechnic Inst. in Worcester MA, and previously a project engineer at Digital Equipment Corp. She instructs graduate and undergraduate computer engineering computer courses, directs graduate research, and performs research involving embedded microprocessor based systems. Her current projects involve small system design, signal processing, and intelligent instrumentation.

Use of a CPLD in an Introductory Logic Circuits Course with Software and Hardware Upgrade

Abstract

This paper documents our continued efforts to integrate the use of complex programmable logic devices (CPLD) into our introductory logic circuits course at the University of Hartford. Although programmable logic devices (PLDs) have been long introduced in our advanced courses, the widespread acceptance demands that PLDs be introduced earlier in the electrical and computer engineering curriculum. In the fall semester of 2011, we selected a CPLD as the choice of PLDs for the introductory logic circuits course, because CPLDs allows for an experience that includes modern design tools and devices, as well as hands-on activities. Our CPLD module allows such a device to be used with a classic breadboard. In prior research we found that in using this module, students can easily identify the CPLD and with modest wiring they can construct circuits that they feel are both satisfying and engaging.

In this paper, the most recent developments, which include both software and hardware upgrades, along with student feedback, are documented. With the Xilinx XC9536 CPLDs now obsolete we adopted a newer CPLD, and designed a new logic circuits trainer. The newer XC9536XL CPLDs require 3.3V signals and power, which is not compatible with any commercially available trainer that we are aware of. Facing this inevitable trend to lower Voltage logic, we decided to adopt the 3.3V CPLD devices and design our own trainer that provides 3.3V power and I/O signal. The artwork for the module and trainer are available at our webpage under free software license for your use.

On the software front, we revised our tutorial and started having our students working with test bench files. The CAD software used in our labs was upgraded from Xilinx ISE Version 10.1 to Version 13.2. In the past, we specifically chose Xilinx ISE 10.1 32-bit version for its graphical test bench generator which is very convenient for students to use when performing simulation. Unfortunately, this feature is absent in the ISE 10.1 64-bit version as well as the subsequent versions of Xilinx ISE, including 13.2. On the other hand, version 13.2 is much more stable in comparison to version 10.1. In adopting version 13.2, we had concerns regarding how students would generate the simulation test bench, which involves modifying VHDL codes, since students do not learn to write hardware description language (HDL) in our introductory logic circuits course. Xilinx ISE provides an aid in generating a skeleton which our students are able to modify for their own use following the instructions in our revised tutorial.

New lecture material was developed to help students understand these upgrades. Based on student feedback, we also provided some historical context with regard to the current state of the art in logic circuits. New lab content was developed to address some concerns from our previous experience, which include: a) start-up activities to help students master the CAD software better and earlier in the course; b) incorporating the use of hierarchical design earlier and in more experiments. The students' experience and feedback, as well as the instructors' observations are presented concerning both the hardware and software upgrades along with other changes made. In closing, we present our future plans.

Introduction and Literature Review

This paper documents our most recent efforts to integrate the Complex Programmable Logic Device (CPLD) into our introductory logic circuits course during the Fall 2013 semester. These efforts involve: a complete hardware upgrade and a major software upgrade; integrating the concept of hierarchical design in progressive steps and deeper into the course; developing entirely new lab content for that purpose.

Our university offers Bachelor of Science degrees in both Electrical Engineering (B.S.E.E.) and Computer Engineering (B.S.Comp.E). The introductory logic circuits course (lecture and lab) is the first course in the digital systems sequence and is required in both Electrical Engineering and Computer Engineering curricula. It is usually taken during the students' first semester of their sophomore year. Although the majority of the students in this course comes from Electrical and Computer Engineering majors, there are frequently students from Computer Science, Math, Mechanical Engineering majors, etc. The courses that follow in the digital systems sequence are listed below:

- ECE 234 Digital design using CPLD*
- ECE 332 Microprocessor Applications**
- ECE 335 Computer Architecture *
- ECE 336 Computer Systems Laboratory*
- ECE 534 VHDL and Applications***
- ECE 532 Embedded Microprocessor***

  * required only by B.S.Comp.E,
  ** required by both B.S.Comp.E. and B.S.E.E
  *** electives for both B.S.Comp.E. and B.S.E.E

Our research started in the Fall 2011 semester, when we successfully adopted a CPLD in the lab component of our introductory logic circuits course, see [1] for details. Our main focus was that the laboratory work must retain a hands-on experience. This was made possible with the CPLD adapter module that we designed, which allows for the use of a breadboard. Our second focus was that our students quickly learn to use the Computer-Aided Design (CAD) tools, which was made possible with the tutorial [2] that we authored.

We are convinced that the lab component of such an introductory course must be tangible, demonstrating the connection between digital and analog concepts. We believe that students must be aware that logic signals are represented with physically measurable quantities. Our main concerns with the use of a development board in such an introductory logic circuits course were that it may prevent students from clearly grasping the notion of what digital logic signals are, or having a clear concept of what a Programmable Logic Device (PLD) is, apart from the development board.

The key difference in using the CPLD module described in this paper is that it is an identifiable component and that students are using real wires to convey signals. During the Fall 2012 semester, new lecture material involving hierarchy, propagation delay, and the presentation of a

CPLD structure was developed. New laboratory material was developed to make use of these principles. The tutorial was expanded regarding the new topics as well. See [3] for details.

Radu [4] emphasized the use of development boards and Coowar [5] elaborated on PLD themselves as well as the CAD tools; however students did not actually construct logic circuits. In teaching digital logic circuits, Nickels [6] provided a choice between two options, either to construct logic circuits using Transistor-Transistor Logic (TTL) family devices on a breadboard, or use a PLD on a development board. Nickels rightly pointed out that the use of programmable logic eases the development of logic circuits, while the use of a development board is not necessary with CPLDs. There can be no doubt that using pre-wired development boards makes it extremely convenient to use PLDs. However, with such convenience, Nickels [6] suggested that electrical and computer engineering students may not have a suitable hands-on laboratory experience. As such, our use of a PLD with a classic breadboard is a very different choice.

In using CPLDs in our course, we use an integrated approach, which includes the use of schematic capture, CPLDs, breadboards, and the concept of hierarchy. With regard to CAD tools Radu [4] reported that with the inclusion of CAD tools and Field-Programmable Gate Array (FPGA) development boards, they observed a statistically significant increase in student learning. Radu et al emphasized schematics, but also introduced students to a Hardware Description Language (HDL) in the context of code fragments and writing test benches. Wang [7] reported positive student feedback and outlined the controversy regarding the use of schematics versus an HDL, expressing a concern that emphasis on an HDL may distract students from the fundamentals of digital logic systems. Wang suggested an integrated approach incorporating breadboard debugging techniques, design and simulation with CAD tools, and verification on a development board, and that an HDL be taught later at the junior level.

For the Fall 2013 semester, we started with several clearly defined, achievable goals in furthering the integration of CPLDs in our introductory logic circuits course:

1. Upgrade the CAD software from Xilinx ISE Version 10.1 to Version 13.2, and revise our tutorial accordingly. In the past, we specifically chose Xilinx ISE 10.1 32-bit version for its graphical test bench generator which is very convenient for students to use when performing simulation. Unfortunately, this feature is absent in the ISE 10.1 64-bit version as well as the subsequent versions of Xilinx ISE, including 13.2. On the other hand, version 13.2 is much more stable. In adopting version 13.2, we had concerns regarding how students would generate the simulation test bench, since students do not learn to write HDL code in our introductory logic circuits course. Xilinx ISE provides an aid in generating a skeleton, which our students then modify. We revised the tutorial so that students can learn to modify the skeleton for their own use.

2. Upgrade the CPLD from the obsolete 5V compatible XC9536 to the 3.3V compatible XC9636XL, and adopt our new trainer that provides 3.3V power and I/O signals. The newer device requires 3.3V signals and power, which is not compatible with any commercially available trainer that we are aware of. Facing this inevitable trend to lower Voltage logic, we decided to adopt the 3.3V CPLD devices and design our own trainer that provides 3.3V power and I/O signals. The transition was implemented during the latter half of the semester,

which allowed students to have experience with both old and new trainers, and thus be able to make comparisons. The artwork for the module and trainer are available at our webpage [8] under free software license for your use.

3. Integrate the concept of hierarchical design in progressive steps deeper into the course and develop entirely new lab content for that purpose. According to our past research, students didn't grasp the concept of hierarchy very well as it was introduced late in the course and used only in the last lab. We developed two new labs so that students learn about hierarchical design earlier and use it in different contexts before applying it in the last culminating design.

Both an exit-survey and a focus group were conducted at the closing of the semester. The exit-survey was anonymous and had a total of 17 responders. The focus group involved four volunteering students, two lab instructors, and the authors. Combining the analysis results of the exit-survey, the focus group feedback, as well as lab instructors' observations, we conclude that the software and hardware upgrades are both successful and that our attempt to further integrate the concept of hierarchy was effective. We also made several recommendations for future course offerings. First, the tutorial will be expanded to include a trouble-shooting section to improve students' learning experience and help them better master the CAD software. Second, we will expand the tutorial discussion on test benches, and possibly provide a script file that produces better test-bench files than the skeleton files that are generated by the ISE tools. Third, revise lab contents to include more demonstrative designs that are related to real-life problems. One option is to have two-stage design problems where students can choose to complete additional circuits when lab time allows. This makes it possible to increase the level of design complexity without risking overwhelming some other students. The final recommendation is to use the newly designed 3.3V compatible trainers throughout the course of the labs, which means replacing the traditional TTL 74LS family with the newer 3.3V compatible 74HC family in the first two labs.

The complete exit-survey questionnaire and the corresponding average Likert scores are listed in Appendix A. The Likert scores range from -3 to 3, with -3 indicating strong disagreement, -2 moderate disagreement, -1 slight disagreement, 0 indicating neutral, 1 slight agreement, 2 moderate agreement, and 3 indicating strong agreement. We considered four questions (No. 1, 2, 3, and 4a) in the exit-survey questionnaire in summarizing the students' overall learning experience. The feedback from these questions (averaged Likert scores being 2.11, 2.17, 2.00, 2.23 respectively) indicate that the students generally felt that: using CPLDs was an overall improvement to the course; the CPLD projects were interesting and educationally valuable; and experience with CPLDs and CAD tools made them feel more confident that in the future they would be more competent as engineers.

In considering our student's overall laboratory experience we considered two questions (No. 6 and 7, averaged Likert scores being 2.17 and 1.82 respectively). The feedback from these questions indicates that the students generally felt that a laboratory involving actual construction of circuits and investigating the behavior of components helped them to better learn and retain the material. For the hardware upgrade, we considered question No. 5b, which indicates that students generally felt that the new trainer was more convenient and easy to use. For the software upgrade, we considered question No. 10, which was also an identical question asked in last year's survey. The average Likert score improved significantly from 1.28 of last year to 2.29 of

this year. This validated our initial belief and intention that upgrading the software to Xilinx 13.2, a much more stable version, improved students' overall experience with the CAD tool.
In the rest of the paper we present topics related to: 1) CAD software upgrade to Xilinx ISE 13.2 and tutorial expansion; 2) hardware upgrade with the new 3.3V-compatible CPLD module and our new trainer; 3) new lab contents that integrated the concept of hierarchy in progressive steps. We close with an outline of our future plans. Appendix A provides a summary of the student questionnaire results and Appendix B outlines all the labs performed.

Xilinx ISE CAD Software Upgrade

In the past, the CAD software we used in the introductory logic circuits lab was Xilinx ISE 10.1 32-bit version. We specifically chose that version for its graphical test bench generator which is very convenient for students to use when performing simulation. Unfortunately, this feature is absent in the 64-bit version as well as the subsequent versions of Xilinx ISE, including 13.2. During the Fall 2012 semester, we observed some issues with Xilinx ISE 10.1, which significantly affected students' overall experience with the CAD tool. One of the Fall 2012 exit-survey questions, "The CAD software used to draw schematics and configure the CPLDs was useful and effective", received an average Likert score of 1.28, indicating just a bit more than a slight agreement overall. However, the standard deviation was 2.14, which is large and indicated significant disagreement between students. Combining the students' interview feedback and the lab instructor's observations, we determined that Xilinx ISE 10.1 was unstable and in the Fall 2013 semester we would upgrade to a newer and more stable version, 13.2.
In adopting version 13.2, we have concerns regarding how students would generate the simulation test bench, which involves modifying VHDL codes, since students do not learn to write HDL in our introductory logic circuits course. Xilinx ISE provides an aid in generating a skeleton. We revised the simulation section of the tutorial [9] so that students can learn to modify the skeleton for their own use.

To judge the effectiveness of the software upgrade, we considered question No. 10 in the exit-survey questionnaire, which is also an identical question asked in last year's survey. The average Likert score improved significantly from 1.28 of last year to 2.29 of this year. This validated our initial belief and intention that upgrading the software to Xilinx 13.2, a much more stable version, improved students' overall experience with the CAD tool. Question 9c received an average Likert score of 1.53, which indicated that the students generally agreed that the online tutorial was helpful for them to learn to write the simulation test bench. Question 12 received an average Likert score of 2.06, which indicated that the students generally agreed that the skeleton test bench generated by the Xilinx test bench generator provided a convenient means for them to produce a test bench, without learning the VHDL language. But the standard deviations of the two questions were 1.84 and 1.43, respectively, which were relatively large. This indicated that there were significant disagreements among the students.

According to the focus group feedback and observations of the lab instructors, we learned that there were a small number of students who had difficulties generating their own test bench by modifying the skeleton produced by the ISE CAD tool. Most other students also found it a very challenging task, especially at the beginning. During our focus group meeting, our students first asked that the tutorial discussion on test benches be expanded. They also expressed an interest in

having a script file that would produce a better test-bench file than the skeleton files currently provided by the ISE tools. We are proposing a script file that could be menu driven, allowing students to enter certain input values and select an input to convey a clock signal. Such a script could better aid in producing test bench files.

Hardware Upgrade: CPLD Module and Logic Trainer

Our newly designed logic trainer enables students to construct logic circuits that use 3.3 Volt signals and power. Such logic circuits can be constructed with complex programmable logic devices (CPLDs) and/or discrete logic devices. As shown in Figure 1, the trainer is constructed with a small PC board that attaches to the top region of a standard breadboard. Power is provided by the external power cube shown on top of the breadboard. The trainer includes the following features:

- Eight switches providing logic signals used as inputs;
- A clock generator providing 1Hz, 10Hz, or 100 Hz clocking signal;
- A push-button pulse generator;
- Eight LEDs serving as logic-high signal indicators;
- A logic probe indicating logic-high, logic-low, clock, and non-driven signals.

We found that mounting toggle switches to a PC board with three wiring tabs was far more secure than switches having two wiring tabs. We also recommend the use of hot-melt glue under the switches to make the switch mount most securely.

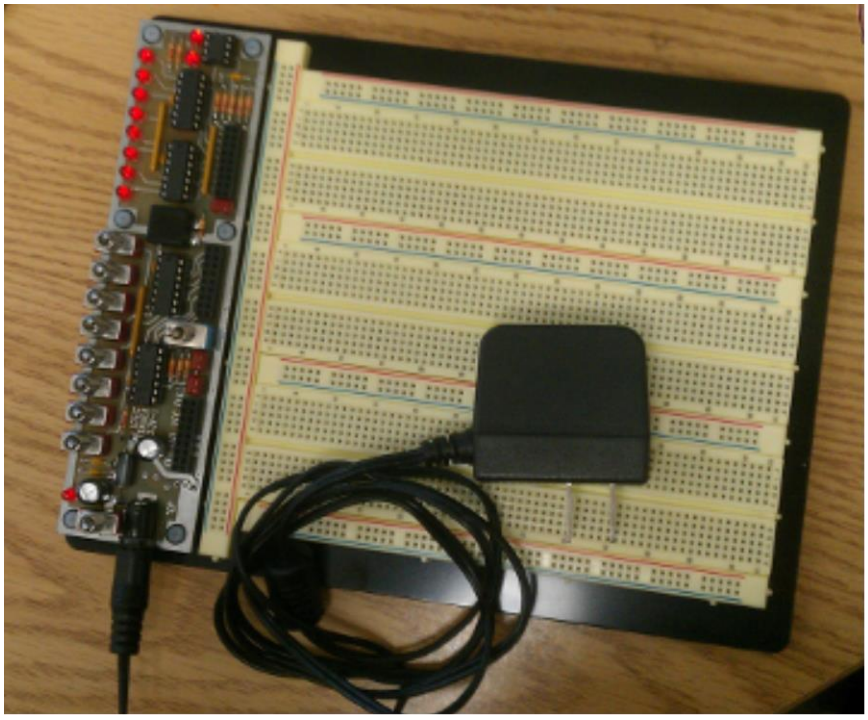

Figure 1: Completed logic trainer

The schematic is shown in Figure 2. The logic switches are to the upper left. The pulser and clock signal generator are to the middle left. Logic indicators are to the upper right. The logic probe is to the lower right. Power filtering is to the lower left. The regulator shown below the power supply filtering is not currently used, leaving those PC board locations empty for now.

The pulse generator is essentially a debounced push-button circuit. The phenomenon of switch-bounce occurs in time on the order of milliseconds. The push-button switch resistance is less than 0.10 Ohm, so that the discharge time constant T1 is much less than that of switch-bounce, and any switch closure will discharge C9 and cause the logic to produce a high pulse output. With the switch open, the pulse output remains high until C9 slowly charges to half the power supply Voltage, which is approximately 33msec with the selected values.

With sequential logic circuits it is important to have a clock signal generator that produces a rectangular waveform. A three-way switch allows the clock generator to operate at a selected clock frequency. Based on our experience we find that 1Hz is appropriate as a slow clock for circuits like the Gray Code counter (lab 7); 10 Hz is an appropriate middle frequency; and 100 Hz is appropriate as a fast clock for circuits like the "roll the dice" (Lab 9) circuit, which models the rolling of a six-sided die.

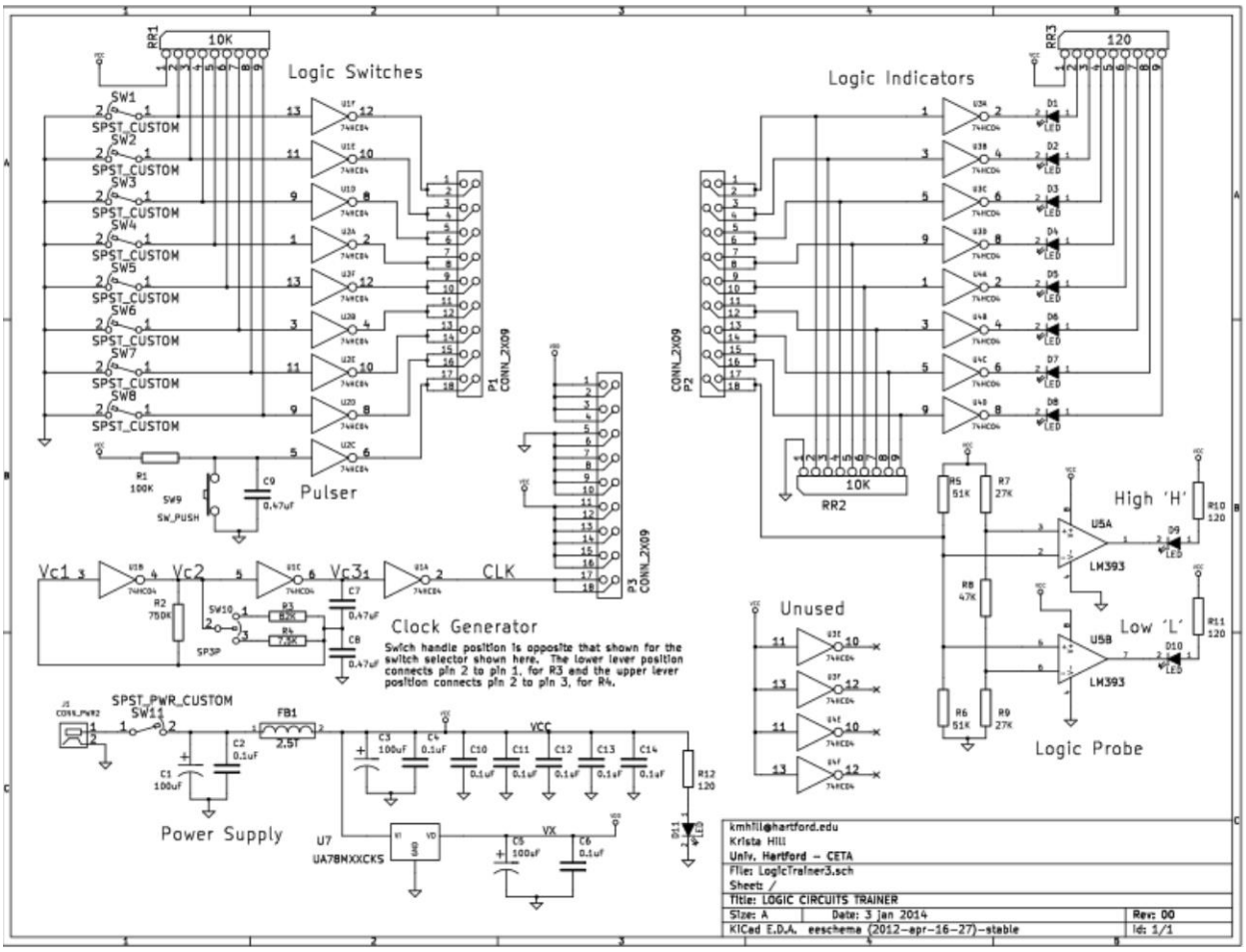

Figure 2: Schematic for logic trainer

In our planning, the first significant decision was to ease the construction and maintenance of the trainer by using only through-hole parts. We wanted the trainer to be inexpensive, so we chose to simplify the construction by using a generic solderless breadboard. We also decided that the trainer would include only basic resources. In our electronics lab each station has a multi-meter, oscilloscope, and a personal computer.

The transition from the old trainers to the new ones happened during the second half of the semester. The students conducted Lab 7, 8, and 9 using the new trainer. This allowed them to compare their experience with the old and new trainers. For the effectiveness of the trainer upgrade, we considered question No. 5b. It received an average Likert score of 1.82, which indicated that students generally agreed that the new trainer was more convenient and easy to use. During the focus group, our students expressed their appreciation for the new trainers which were easier for them to use than the old trainers.

CPLD Module

For our initial introduction of CPLDs to our course, we designed an adapter module for the XC9536 device that we call the XMOD, shown in Figure 3. The downward pins are arranged in a 40-pin dual in-line package (DIP) arrangement. The upward pins to the right are used to configure the CPLD. Changing from the XC9536 to the XC9536XL involved only changing the markings on the CPLD module, as the devices have the same pin-outs. Another option is the Digilent, Inc. C-Mod [10] which uses a CoolRunner II CPLD, but it has a different pin-out. Due to size constraints, our adapter module uses surface mount parts.

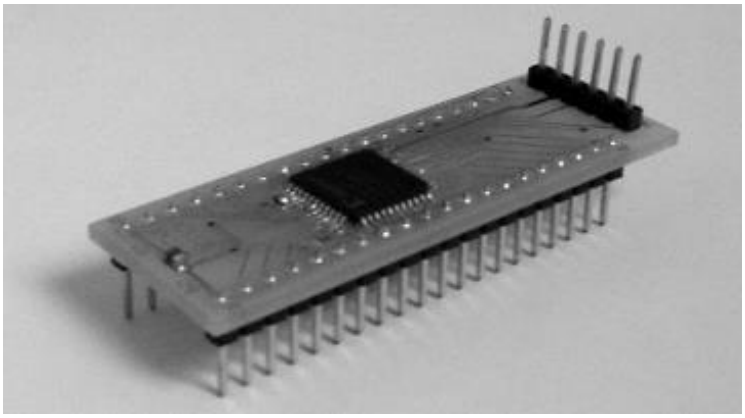

Figure 3: XMOD adapter module

Discrete Logic Devices

With regard to discrete logic, we cannot use 5 Volt 74 and 74LS type TTL devices with the new trainers, rather we choose to use 74HC type CMOS devices. These devices have very low power consumption, high input noise immunity and wide operating Voltage range, from 2.0 to 6.0 Volts. The logic trainer itself uses 74HC04 inverter chips. Given the reduced emphasis on discrete logic, the list of 74HC parts Table 1 is suggested. The column labeled “Dev.Count”

refers to the number of devices that a single given chip contains. This list is sufficient for students to investigate logic device characteristics, simple combinational logic circuits as well as modest state machines. These 74HC devices are inexpensive and available in dual in-line packages.

Table 1: Selected Discrete Logic Devices

Part Num.	Description	Dev.Count
74HC00	Dual-input NAND	4
74HC04	Inverter	6
74HC08	Dual-input AND	4
74HC10	Triple-input NAND	3
74HC20	Quad-input NAND	2
74HC32	Dual-input OR	4
74HC74	D-Type Flip-flop	2
74HC86	Dual-input Exclusive OR	4

Integrating Hierarchical Design in Three Progressive Steps

During the Fall 2013 semester, we decided to integrate the concept of hierarchical design deeper into the course in progressive steps and develop entirely new lab content for that purpose. According to our past research, students didn't grasp the concept of hierarchy very well as it was introduced late in the course and used only in the last lab. We developed two new labs so that students would be able to learn about hierarchical design earlier and use it in different contexts before applying it in the last culminating design.

The students used hierarchy in three different labs: Lab 6, Lab7, and Lab 9. Lab 6 involved a combinational circuit, while Lab 7 and 9 involved sequential circuits. Lab 6 and Lab 7 used one-level hierarchy, while Lab 9 used two-level hierarchy. In Lab 6, the students used a schematic to create a symbol for a full-adder component; in Lab 7, the students used the VHDL code provided in the tutorial to create a symbol for a flip-flop component; in Lab 9, students can choose to use either methods to create a 3-bit parallel-load counter.

In Lab 6, the students first created their own full-adder component using a schematic; then they built a 4-bit two's-complement add-subtract circuit using four instances of the full-adder component. Figure 4 shows an example of student work. By incorporating exclusive-OR gates and making use of the carry-in provided by the first full-adder, the circuit is able to perform subtraction or addition.

In Lab 7, the students analyzed, constructed, and tested a state machine that generated a Gray code sequence. They used hierarchy a second time when they constructed a two-bit Gray code counter. Our students first produced a schematic symbol to represent the D-type flip-flop described in a VHDL file from the tutorial. Our students next used the resulting symbol in a schematic to draw the counter circuit.

Lab 9 is called “roll the dice”, which models the rolling of a six-sided die and is the highlight of all the labs. In this lab, students used hierarchy for the third time. First, they built three different symbols using the VHDL code provided in the tutorial: a 2 by 1 MUX, a D-type flip-flop, and a half-adder. Second, they built a 3-bit parallel-load counter component with the previous created symbols using either a schematic or a VHDL file provided in the tutorial. Finally, they used the 3-bit counter component and additional logic gates to create a sequential circuit that provides outputs to actual resistors and LEDs that modeled a six-sided digital die. By manually asserting a signal called 'roll' for a brief moment the counter counts quickly and then stops in a randomly selected state. Figure 5 shows an example of a student team’s completed circuit. This lab was also voted as the favorite lab according to the exit-survey. A student wrote in his lab report: “This lab was a very fun lab and was a great way to end our semester of labs. I really enjoyed how we were able to see a physical representation of everything that we've done in lecture and prove that all of the concepts we've learn actually hold truth to them.”

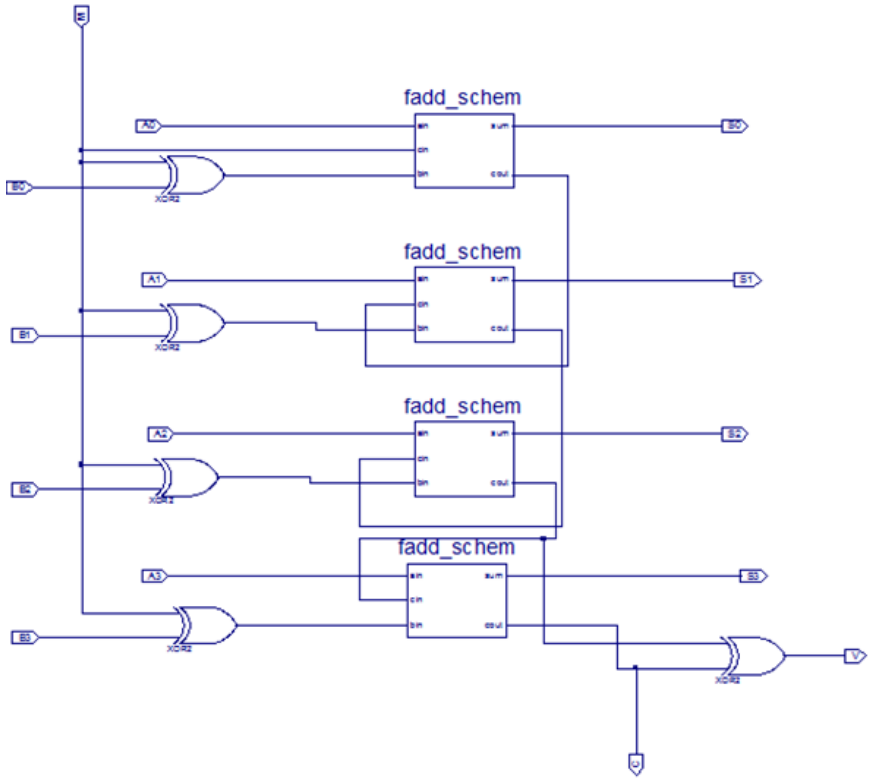

Figure 4: Lab 6 “Hierarchical 4-bit two’s-complement add-subtract circuit” example schematic by a student team

In judging the results of our efforts to further integrate the hierarchy concept, we consider exit-survey question 11: “The CAD software helped me make use of and understand hierarchy principles.” The identical question was also included in the Fall 2012 exit-survey. The average Likert score improved from 1.33 of 2012 to 1.76 of 2013, a significant improvement. Combining the student feedback from the focus group and the observations of the lab instructors, we conclude that our efforts to integrate the concept of hierarchical design deeper into the course in

progressive steps were successful. But there is still room for improvement. In the future, we will split the tutorial section on hierarchy into two parts. The first part, which will be mainly for use earlier in the course, will expand on contents related to medium scale integrated (MSI) circuits. The second part, which is for later in the course, will expand on the use of hierarchy with state machines and counters.

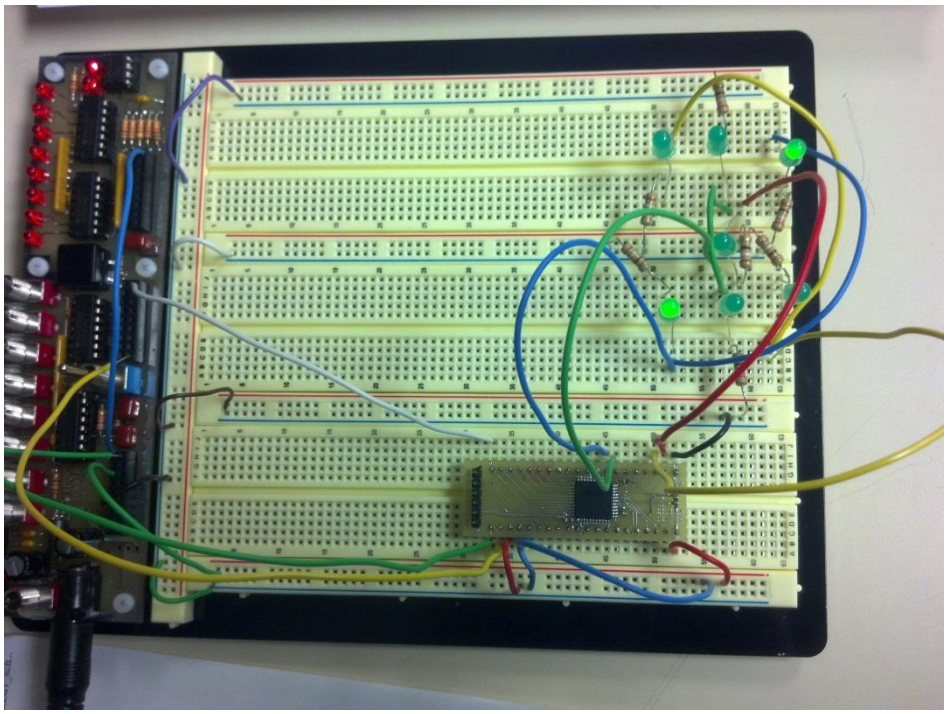

Figure 5: Lab 9 “Roll the dice”
example of a completed circuit by a student team

Conclusion and Future Work

Both an exit-survey and a focus group were conducted at the closing of the semester. Combining the analysis results of the exit-survey, the focus group feedback, as well as lab instructors’ observations, we concluded that the software and hardware upgrades were both successful and that our attempt to further integrate the concept of hierarchy was effective. We also continue to believe that CPLDs provide a great way for students to learn CAD tools and retain the hands-on experience at the same time.

We also made several recommendations for future course offerings. First, the tutorial will be expanded to include a trouble-shooting section to improve students’ learning experience and help them better master the CAD software. Second, we will expand the tutorial discussion on test benches, and possibly provide a script file that produces better test-bench files than the skeleton files that the ISE tools currently produce. Third, revise lab contents to include more demonstrative designs that are related to real-life problems. As indicated by the exit-surveys and focus group feedback, the favorite labs chosen by the students always relate to real-life problems or have very demonstrative results. One possible strategy is to have two-stage design problems

where students can choose to complete additional circuits when lab time allows. This makes it possible to increase the level of design complexity without risking overwhelming some other students. The final recommendation is to use the newly designed 3.3V compatible trainers throughout the course of the labs, which means replacing the traditional TTL 74LS family with the newer 3.3V compatible 74HC family.

References:

1. J. Hill, Y. Yu, "The CPLD Provides a Third Option in Introductory Logic Circuits Course," AC2012-5302, ASEE National Convention, 2012, session W516
2. K. Hill, "Schematic Capture – ISE 13.x", http://uhaweb.hartford.edu/kmhill/suppnotes/isetut/ise13x1/schem.htm
3. K. Hill, Y. Yu, "Use of a CPLD in an Introductory Logic Circuits Course," AC2013-7987, ASEE National Convention, 2013, session M416
4. M. Radu, C. Cole, M. Dabacan, and S. Sexton, "Extensive Use of Advanced FPGA technology in Digital Design Education," ASEE National Convention, 2008, session 2139
5. R. Coowar, "The Development of a Course in Programmable Digital Devices," ASEE National Convention, 1996, session 2648
6. K. Nickels, "Pros and Cons of replacing discrete logic with programmable logic in introductory digital logic courses," ASEE National Convention, 2000, session 2532
7. G. Wang, "Teaching Digital Logic Using CAD Tools in a Teaching-Oriented University," ASEE Annual conference, 2005
8. K. Hill, "XC9536XL CPLD module and 3.3V compatible logic trainer design artwork": http://uhaweb.hartford.edu/kmhill/projects/xmod/
9. K. Hill, "Simulation of the Full Adder Circuit with ISE13.x", http://uhaweb.hartford.edu/kmhill/suppnotes/isetut/ise13x1/simulation_fadd.htm
10. Digilent Inc., "Digilent C-Mod," http://www.digilentinc.com

Appendix A: Exit-survey Questionnaire and Results Summary

A total of 17 students answered questions 1 through 12 using the following scale: The value +3 indicates strong agreement, +2 moderate agreement, +1 slight agreement, 0 indifference, -1 slight disagreement, -2 moderate disagreement, and -3 indicating strong disagreement. The two right-most columns in the summary table provide the average as well as the standard deviation which summarizes the level of disagreement between students. Discussion of the results is presented in the main part of this paper, along with a summary in the introduction.

	Likert Rating Question Description	Average	Standard Deviation
1	Using CPLDs in the Logic Circuits course is an overall improvement.	2.11	1.17
2	The lab projects using CPLDs were interesting and educationally valuable.	2.17	1.13
3	My experience with CPLDs makes me more confident and I foresee that in the future I will be more competent as an engineer.	2	1.37
4a	My experience with the CAD tools in logic circuits lab makes me more confident so that in the future I will be more competent as an engineer.	2.23	0.83
4b	There should be more topics related to CAD software tools presented in lecture.	1.47	1.17

5a	Having some experience also with TTL devices, in the first two labs, is educationally relevant and is a good use of my time.	1.29	1.21
5b	In comparing my experience with both trainers, the second trainer was more convenient and easy to use.	1.82	1.23
5c	The documentation provided for the new trainer was clear and helpful to me in starting to use the new trainer.	1.94	1.19
6	I feel that a laboratory experience in which I construct circuits and investigate signals helps me to better learn the material.	2.17	0.88
7	I found that in our use of CPLD in the laboratory, the hands-on experience was retained, and helped me to better learn the material.	1.82	0.81
8a	There should be more use of CPLD hardware topics in the lecture portion of the course.	1	1.17
8b	It would be a benefit to incorporate exercises involving CAD tools and CPLD topics into the homework.	0.41	1.37
9a	The online tutorial was helpful in getting me started using CPLDs.	2.06	0.83
9b	The online tutorial helped me learn the principle and application of hierarchy.	1.76	0.9
9c	The online tutorial helped me learn to write the simulation test bench.	1.53	1.84
9d	The online tutorial served as a useful reference to me, later in the course.	2	1.22
10	The CAD software used to draw schematics and configure the CPLD was useful and effective.	2.29	0.85
11	The CAD software helped me make use of and understand hierarchy principles.	1.76	0.9
12	The skeleton test bench generated by the Xilinx test bench generator provides a convenient means for me to produce a test bench, without having learned VHDL language.	2.06	1.43

Questions 13 through 17 are open-ended, which allow each student to answer with their own words.

13	What is your largest concern in improving the course? Please elaborate.
14	Suggest a laboratory activity involving CPLDs that helps retain the hands-on experience.
15	What was your favorite laboratory and explain why?
16	What was your least favorite laboratory and explain why?
17	Do you have any other comments?

For Q13, four students expressed concerns regarding trouble-shooting; one student expressed desire to learn more about the internal structure of the breadboard; one student expressed concern regarding the reliability of the CPLD; one student expressed concern about learning the VHDL

code; one student suggested making the lab twice a week; one student suggested having more labs like lab 8 and 9; two students expressed their satisfaction with the course instead.

For Q14, one student suggested the control of a car's signal lights as a lab topic; one student listed "more resistors", which we interpreted as a suggestion for circuits involving more wiring, One student suggested more labs like lab 8 and 9.

For Q15, ten students listed Lab 9 as their favorite; six students listed Lab 8 as their favorite; one student listed lab 6 as the favorite. Most listed reasons are "real-life design", "complicated circuits", "most interesting hardware build".

For Q16, five students listed Lab 7 as their least favorite; two students listed Lab 1; two students listed Lab 3, the Xilinx tutorial lab; one student listed Lab 2. Most of the students listed the lab where they, "felt like it wasn't obvious", "could not trouble shoot" or "simulation did not work". Lab 7 was the first lab where student started to use a clock signal and flip-flops, thus the simulation test bench also became more complicated than previous labs.

For Q17, four students responded with "more labs", "more of real life experiment", "get newer/better equipment", and "great course".

Appendix B: List of Laboratory Projects Performed

There were nine laboratory projects. The first two labs were TTL based and required students to use three TTL chips (74LS04 hex inverter, 74LS08 quad AND, 74LS32 quad OR) to construct a simple combinational circuits. Our students analyzed the circuit, generated a truth table, and tested the circuit using switches and LEDs. A second aspect is that students used an oscilloscope to measure gate propagation.

The next two labs provided the necessary transition to using CAD software and a CPLD. In lab three, the students performed every stage of the design and implementation process for a circuit described in the tutorial. The steps include making a new project, schematic capture, making a test bench, simulation, pin assignment, synthesizing the circuit, and configuring the CPLD. In the fourth lab, students used the CAD tool and CPLD to design, implement, and test a combinational circuit.

Lab five and six made use of the CPLD by introducing Medium-Scale-Integration (MSI) like combinational logic components. In Lab five, we introduced a 4 by 16 decoder. The students used the CAD tool and CPLD to design, implement, and test a combinational circuit with don't-care conditions. They used both the K-map simplified sum-of-products results realized with purely NAND gates and the sum-of-minterms results realized through a decoder. In comparing the results, they learned that the two designs were equivalent. This lab is only possible to be completed within the time frame of a two-and-a-half-hour lab with the CAD tool. Lab six is the first lab we introduced the concept of hierarchy design. The students used the CAD tool and CPLD to design, implement, and test a 4-bit signed 2's-complement adder/subtractor. They first created their own full-adder component using a schematic and then used four instances of the component along with extra logic gates to create the ultimate circuit.

The last three labs involved state machines and used the CPLD module. These three labs were also the ones that the students conducted using the new trainers and new CPLD modules. In lab seven, students analyzed, constructed, and tested a state machine that generated a Gray code sequence. They also used hierarchy for the second time by making a symbol for D-type flip-flop component from the VHDL code in the tutorial. In lab eight, students designed, constructed, and tested a simplistic four-floor elevator controller. Lab nine is called “roll the dice”, which is the highlight of all the labs. In this lab, students used hierarchy for the third time, building a 3-bit counter circuit and additional logic to model the rolling of a six-sided die. By manually asserting a signal called 'roll' for a brief moment the counter counts quickly and then stops in a randomly selected state.

Lab 1 Digital Gates - TTL
Lab 2 Verifying the DeMorgan's Law and the Distributive law - TTL
Lab 3 Xilinx ISE 13.2 Tutorial Practice
- CAD software and CPLD module

Lab 4 Combinational Circuit Design using Xilinx
- CAD software and CPLD module

Lab 5 Combinational Circuit Design with Don't-Care Conditions and a Decoder
- CAD software and CPLD module

Lab 6 4-bit Signed 2’s-Complement Adder/Subtractor
- CAD software and CPLD module
- using Hierarchical Design

Lab 7 Gray Code Counter
- CAD software and CPLD module
- using Hierarchical Design
- using new trainer

Lab 8 Elevator Controller Design
- CAD software and CPLD module
- using new trainer

Lab 9 Roll the Dice
- CAD software and CPLD module
- using a 3-bit Counter and Multi-level Hierarchical Design
- using new trainer