# USE OF A CPLD IN AN INTRODUCTORY LOGIC CIRCUITS COURSE

Krista Hill, Ying Yu
Electrical and Computer Engineering Department
University of Hartford

## Abstract

In the 2011 Fall semester we successfully adopted a complex programmable logic device (CPLD) for use in our introductory logic circuits course at the University of Hartford. While the adoption of the corresponding CAD tools is an important element, we have been convinced, from the start of our research, that such a course must also be tangible to students. We feel that in such an introductory course, students must be aware that they are dealing with real circuits and that logic signals are represented with physically measurable quantities. We found that in using a CPLD with a breadboard, the CPLD is identifiable to students and that with modest wiring they constructed demonstrative circuits that they felt were satisfying and engaging.

This paper outlines our more recent experience to further integrate our use of the CPLD in our introductory course. Given the potentially wide reaching impact on the curriculum, we are taking incremental steps, each with measurable goals. In the Fall 2012 semester, new lecture material involving hierarchy, propagation delay, and CPLD structure was developed. The tutorial was revised and new lab material was developed, to make use of these principles.

In prior semesters, propagation delay and the more analog side of logic circuits were still presented in the context of TTL devices. We have developed material to introduce these same topics in the context of our CPLD module. We have students investigate CPLD propagation delay, first with a hands-on experience. Later in the lecture portion of the course, the internal structure of a CPLD is presented along with the device timing model. The device timing model provides a means for students to better understand propagation delay within the device and how a circuit is actually implemented with a CPLD.

Medium scale integration (MSI) devices such as decoders, multiplexers, and counters are important topics. TTL MSI devices each integrate into a single chip, the functionality provided by networks of small-scale integration (SSI) parts, such as gates and flip-flops. As with TTL MSI device integration, our CAD tools support a technique called hierarchy, in which students integrate lower level functionality into their own MSI like symbols, which they can use in their own schematics.

For our initial adoption of CPLDs, few changes were made to the actual course content. For our more recent experience, we developed new lecture and course content, and our tutorial material was expanded accordingly. Finally, we used student feedback to assess our results. In addition we are most concerned that our students still have meaningful experiences in the laboratory and lecture components of the course. In this paper we also present our future plans.

## Introduction

This paper outlines our experiences from the Fall 2012 semester in adopting the complex programmable logic device (CPLD) into our introductory logic circuits course. These efforts involve integrating the technology deeper into the course and developing entirely new content for that purpose. Our research started in the Fall 2011 semester, when we successfully adopted a CPLD in our course, see [1] for details. At that time we expressed our concern that the laboratory work retains a hands-on experience, which was made possible with the CPLD adapter module that we designed, that allows for the use of a breadboard. We were also concerned that our students quickly learn to use the CAD tools, which was made possible with the tutorial [7] that we authored.

For our recent work, new lecture material involving hierarchy, propagation delay, and the presentation of a CPLD structure was developed. New laboratory material was also developed to make use of these principles. The tutorial was expanded regarding these new topics, so the tutorial is not only useful in getting our students using the CAD tools, but it now serves as an indispensable reference throughout the semester. Our new "Teaching Logic Circuits with CPLDs" page [8] provides links to our own content, as well as links to other content.

Since the beginning of our research, we have been convinced that in such an introductory course, the lab component must be tangible, demonstrating the connection between digital and analog concepts, rather than an entirely abstract notion. We feel that students must be aware that they are dealing with real circuits and that logic signals are represented with physically measurable quantities. Our main concerns with the use of a development board in such an introductory course is that students may not clearly grasp the notion of what digital logic signals are, or have a clear sense of what a PLD is, apart from the development board. The key difference in using the CPLD module described here is that it is an identifiable component and that students are using real wires to convey signals.

Radu [2] emphasizes the use of development boards and Coowar [3] elaborates on PLD logic devices themselves as well as the CAD tools however, students did not actually construct logic circuits. In teaching digital logic circuits, Nickels [4] provides a choice between two options, either to construct logic circuits using transistor-transistor logic (TTL) family devices on a breadboard, or use a programmable logic device on a development board. While Nickels rightly points out that the use of programmable logic eases the development of logic circuits, the use of a development board is not necessary with CPLDs. There can be no doubt that pre-wired development boards provide a great convenience in using PLDs. However, with such convenience, Nickels [4] suggests 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 our further development of CPLDs in the course, we continued with an integrated approach, which includes our use of schematic capture, CPLDs, breadboards, and now includes the concept of hierarchy. With regard to CAD tools Radu [2] reports that with the inclusion of CAD tools and FPGA development boards, they observed a statistically significant increase in student learning. Radu et al emphasizes schematics, but also introduces students to a hardware description language (HDL), in the context of code fragments and writing test benches. Wang [5] reports positive student feedback and outlines 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 suggests 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.

In furthering the integration of CPLDs in our introductory logic circuits course, we started the Fall 2012 semester with several clearly defined, achievable goals.

1. Make some significant changes to the laboratory content where we could, to replace our use of TTL devices with the CPLD. The second laboratory which was previously TTL based is now CPLD based. In this lab our students first experience logic circuit design and observe propagation delay, in a hands-on fashion.

2. Retain the lab hands-on experience and improve students’ understanding of the simulation procedure. We focused on having students perform activities that eventually leads to a functional circuit. Rather than simply producing a final working circuit, our students investigate a circuit in stages and eventually demonstrate the finished circuit along with interesting results. One of the most important stages involves investigating the simulation results, which provides insight in terms of connecting theory with actual circuits.

3. Further our students’ use of CAD tools and to implement hierarchical designs. This involved revising the tutorial and developing new content for two of the more advanced labs.

4. Develop new lecture material to introduce the internal structure of a CPLD and CPLD propagation delay. This understanding of the inner workings of a CPLD allows students to consider propagation delay in theoretical terms, beyond their hands-on experience in lab.

At the end of the Fall 2012 semester we conducted an exit-survey and interviewed individual students. Combining the analysis results of the exit-survey, the interview feedback, as well as our own observations, we conclude that our further integration of CPLDs is a success. We also made recommendations for future course offerings. First, students will start using the CAD tools much earlier, which will help them accelerate their learning curve, so they can focus more on the actual design of the circuits soon after. This requires modification of the lecture content to introduce essential analysis and design concepts at the very beginning. Second, the CAD tool will be used in the lecture component of the course. This means new homework content will be developed to make more use of the CAD tool and PLD principles. Additional technical and software support will need to be provided to students in this regard. Third, revise lecture content to introduce more about the internal structure and other principles of CPLD and at an earlier timeline. The final recommendation is to increase the number of the laboratory projects that use hierarchy concepts, and revise some projects to include more visual, realistic and tangible results that students will demonstrate.

To summarize our recent student's overall experience, from the questionnaire (see complete list of questions and the corresponding Likert scores in Appendix A) we considered four questions (questions 1, 2, 3, and 4a) which serve specifically for that purpose. The feedback from these questions indicate that the students generally felt that in the course, using CPLDs is an overall improvement to the course, the CPLD projects were interesting and educationally valuable, and experience with CPLDs and CAD tools gives them more confidence so that in the future they will be more competent as engineers. In considering our student's laboratory experience we considered three questions (7, 8, and 9), which indicate that they 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.

In the rest of this paper we present topics related to hierarchy, outline our student’s first hands-on experience with logic circuit design and propagation delay using a CPLD. Our new CPLD structure and timing document is outlined. Next, issues regarding the CPLD module and the CAD software are presented. We present our concerns with having our student's use of the CAD software outside of the class laboratory environment. We close with an outline of our future plans. We will make use of lab sessions in the first two weeks of class for so-called *lab startup* activities, where our students will have a first hands-on activity with logic circuits and learn how to use a breadboard. They will also perform the CAD tutorial and learn about our expectations for project reports. We will also provide in a lecture, a historical perspective by outlining some prior logic families. Appendix A provides a summary of the student questionnaire results and Appendix B outlines all the labs performed.

## Hierarchy

The notion of medium scale integration (MSI) logic is essentially an application of hierarchy in chip design. Hierarchy is a technique for describing logic circuits that involves conceptual layers. Each box in Figure 1 is a subsystem, like a chip, and with a CPLD is *one instance* of a *description*. Box A could represent an MSI type component such as a counter, which involves simpler, lower-level components. We use Level-0, which is also called the top level to describe how Level-1 modules are interconnected, and so on.

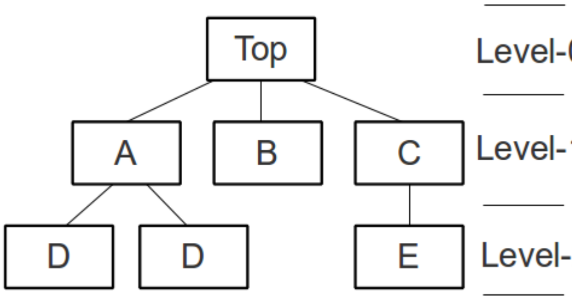


Figure 1: Layers of hierarchy.

Our CAD tutorial [7] is expanded with several new sections, to introduce our students to the use of hierarchy with Xilinx [6] ISE. The first new section outlines how instances of the full-adder in Figure 2, presented earlier in the tutorial, are used to make the ripple-carry adder in Figure 3. The following tutorial sections describe the use of hierarchy in making a state machine and a binary counter, each of which is represented as a symbol in a higher level schematic.

The use of hierarchy provides students with a means to make use of the top-down concept, using block diagrams in designing logic circuits. In discussing state machines we teach our students to think in terms of blocks that provide state memory, produce the next state, as well as the overall output. Each of these, the state memory, next state logic, and output logic, corresponds to a hierarchical circuit represented as a block in a block diagram.

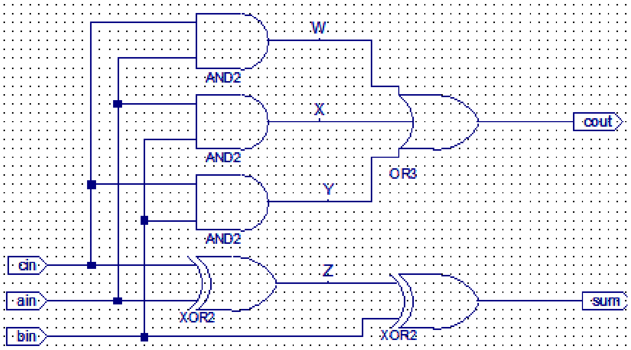


Figure 2: Full-adder (fadd) schematic.

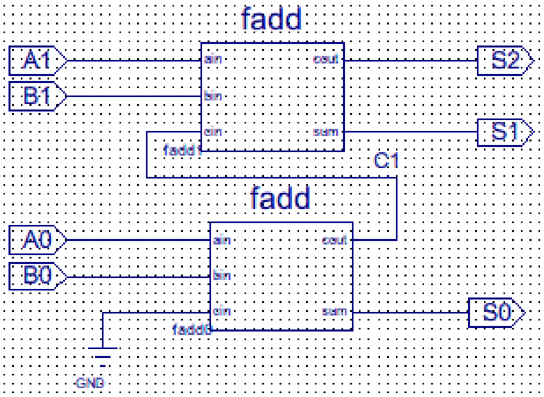


Figure 3: Ripple-carry adder.

We introduced the topic of hierarchy later in the course, along with the state machine analysis laboratory, which involved a Gray code counter. In the questionnaire, question 11b scored 2.09 which indicate moderate agreement that the tutorial helped students to learn the principle and application of hierarchy. In talking with students and the lab instructor, there was a consensus that hierarchy is a general idea and should be introduced much earlier in the course and in particular that it enables students to design their own decoders and multiplexers, which they can then use in their own designs.

The Xilinx Hierarchical Methodology Guide [10] provides advanced reading for a course instructor, but is otherwise unsuitable for an introductory logic circuits course. The use of hierarchy allows a design to be divided into packages called partitions. Such a partition can be placed into a given region of the PLD and can be assigned to an independent design organization. The use of partitions allows for partial reconfiguration of a PLD, as well as the ability to address strict security and reliability requirements.

## First Experience with Propagation Delay

Previously we relied on 74LS series TTL devices to introduce propagation delay. In the second lab experiment, students constructed a given combinational logic circuit and used an oscilloscope to measure propagation delay at points along a delay path.

We developed new course content using CPLDs for this purpose. For their first experience with propagation delay, we continue to provide a hands-on approach. Students hand wire a CPLD configured with NAND gates and inverters, as shown in Figure 4.

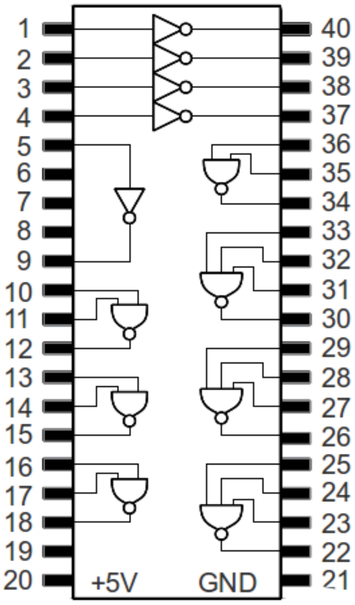


Figure 4: CPLD module configuration.

The timing analysis for this CPLD configuration is eased as the delay for such simple functions is characterized as the pin-to-pin delay. As before, our students use an oscilloscope to measure propagation delay. In the questionnaire, question 5b scored 1.64 which indicates slight to moderate agreement that students were able to learn about propagation delay by using CPLDs.

Later in the course, in discussing the internal structure of a CPLD, detailed timing models are presented that outline propagation delay for more complicated CPLD circuits.

## CPLD Structure and Timing

After reviewing the literature, we perceived a need for material outlining the structure and timing of CPLDs in a way that is accessible to students. To address this need, one of our authors wrote, “Xilinx XC9536 Structure and Timing” which provides an explanation and also includes study exercises. The document [9] is freely available for your use.

Figure 5 is a block diagram of the contents of a XC9536 CPLD, though the programming logic is not shown. The chip pins to the left connect through the Input/Output block (IOB) to the Switch Matrix (SM), which interconnects with the function blocks (FBs). The SM provides each FB with 36 inputs and receives 18 outputs. Three Global Clock (GCK) capable pins and a Global Set/Reset capable pin (GSR) also connects directly to each FB. The GTS pins can provide control of three-state buffers in the IOB.

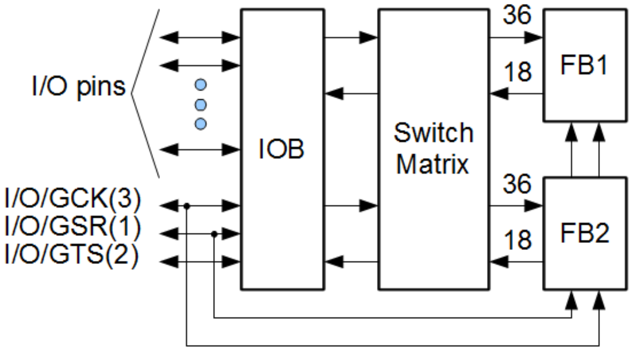


Figure 5: XC9536 block diagram.

Each FB is organized into 18 so-called macrocells (MCs), as shown in Figure 6. The buffers receive FB inputs B1 to B36 and produce the values and the inverted values as well, which are connected to all the MCs in the FB and used in wire-and type combinational logic. Each MC has five direct product terms that it can use in realizing combinational logic, an OR plane, as well as one flip-flop. The MCs are adjoined and can make use of product terms from neighbor MCs. The MC outputs the signal busses F and PTOE connect back to the SM.

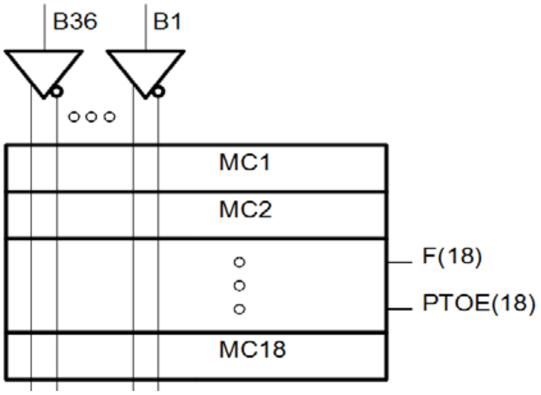
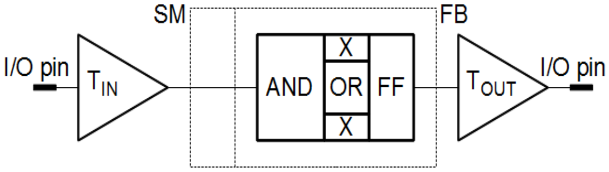


Figure 6: Macrocells in a function block.

Figure 7 shows the model of the simplest pin-to-pin propagation delay path $T_{PD}$. The delay here is the sum of $T_{IN}$ which is the combined input buffer and SM delays, the MC AND-OR logic delay $T_{LOGI}$, the flip-flop bypass delay $T_{PDI}$, as well as the output buffer delay $T_{OUT}$. For more complicated circuits that require more resources than that of a single MC, resources in adjoining MCs can be used or the outputs in different FBs can be routed through the SM. For each such case, the corresponding timing model provides an estimate of the delay.



$$T_{PD} = T_{IN} + T_{LOGI} + T_{PDI} + T_{OUT}$$

Figure 7: Pin to pin propagation delay.

## CPLD Module Issues and Plans

In selecting the CPLD we were concerned that it be used with our existing breadboard kits. Our existing breadboard kits are a significant investment and we wanted the ability to use TTL parts in conjunction with the CPLD, if we so choose. The XC9536 device that we selected, using the module in Figure 8, is compatible with the 5 Volt power and 5 Volt logic signals, used with traditional 74LS type TTL logic devices. The XMOD module in Figure 8 is our in-house design, it is similar to the C-Mod[11] from Digilent, which has a different Voltage CPLD. The artwork[12] for our XMOD module is available under free software license.

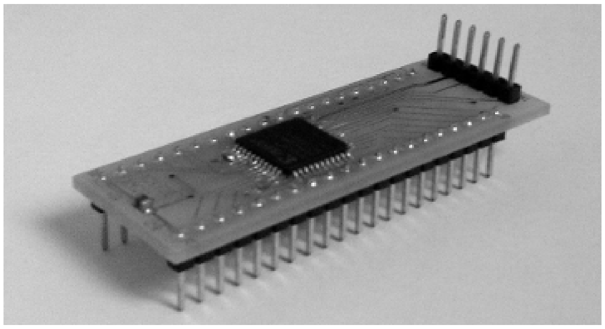


Figure 8: Example CPLD module.

For our purposes we regard the XC9536 based XMOD as transitional as the XC9500 series logic family has recently become obsolete as Xilinx has discontinued the manufacture and sales of these products. The evolving situation is causing more pressure for us to move forward with our long term goal, to eliminate our use of devices that use 5 Volt power and 5 Volt logic signals. To allow for our continued use of discrete logic, we will replace our 74LS parts with 74HC or similar components that are compatible with 3.3 Volt power and 3.3 Volt signals.

## CAD Software Issues and Plans

We have some issues with regard to the Xilinx ISE CAD tools. In reviewing feedback, question 12a in the questionnaire scored 1.28, indicating more than a slight agreement that the CAD tools were useful and effective, however the standard deviation in question 12a is 2.14 which is large and indicates significant disagreement between students. This means that there is a significant population in the course that is approving of Xilinx ISE, as well as another group that was less than satisfied. Of the latter group, some stated in their opinion, that ISE 10.1 is difficult to use, and is unstable.

Regarding software versions, we are continuing to rely on the 32-bit version of Xilinx ISE 10.1. Our dependency is that, this is the last version that includes a graphical tool for making test benches. Newer versions, including the 64-bit version of Xilinx ISE 10.1 do not include this useful feature, which is unfortunate, as newer versions may be more stable and more reliable. To move to a newer version of ISE we will either have to obtain such a graphical tool, or expose our students to how to manually write a test bench.

As with any tool, there is a significant learning curve to overcome. In our first investigation we found the CAD tutorial to be instrumental in getting students started. In our last feedback, questions 11a and 11c scored 2.00 and 1.73, indicating moderate and less than moderate agreement that the tutorial is still helpful in getting students started, and was a useful reference later in the course, respectively.

Likewise, with any tool it takes time to develop a sense of mastery. This last semester, students were introduced to the Xilinx software in the third laboratory experience. We are dissatisfied with devoting a lab session to a tutorial as this is not an investigation that leads to a project report. Rather, we will have our students perform the CAD tutorial earlier in the semester, during a lab startup activity session. This will help our students accelerate their learning curve so that later they can focus more on the actual design of logic circuits.

## Using CAD Tools Outside of the Class Laboratory

In the questionnaire, questions 4b and 10b scored 2.13 and 1.60, indicating slightly more than moderate agreement that more topics related to the CAD tools should be presented in lecture, and slight to moderate agreement that exercises involving CAD tools and CPLD topics should be incorporated into the homework.

Using ISE outside of lab would provide students with more practice using the tools and would benefit our students. We only have plans to use the actual CPLDs in lab, so that work performed at home or in an open lab would only involve design and simulation. While we do have Xilinx ISE installed in open PC labs, it is still desirable to provide students with more support and also have students install ISE on their own personal computers.

There is a growing population of Macintosh users in our college. A rough estimate is that one quarter of our logic circuits students each own a Macintosh. The issue is that there is no version of ISE for the Macintosh. There are versions of ISE for Windows as well as Linux. There are at least two possible solutions to this dilemma:

- Installing Linux or Windows as an additional operating system on the Macintosh
- Remotely accessing a Linux or Windows computer that has ISE installed

The advantage of remotely accessing a Linux computer is that it allows more options than a single desktop, allowing for simultaneous login activity. But in either case, students will need support in some measure, and in the past with other software, it has proven useful to organize the Macintosh users into a self-supporting group.

## Lab Start-up Session Plans

Normally, our first laboratory project experience takes place in the third week of classes, leaving the first two weeks of lab sessions unused. We have future plans to use the first two weeks of lab sessions for so-called *lab start-up* sessions. There are several key topics that will be addressed during start-up sessions.

- Learning to use or practice in using a breadboard.

- Having a first hands-on experience with logic devices, in the form of discrete logic.
- Perform the CAD tutorial.
- Outline our expectations in a project report and present an example report.

A start-up lab activity is markedly different from that of a normal lab in that the start-up lab activities do not involve the writing of a project report. Having students use a breadboard to construct an arbitrary logic circuit using discrete logic devices is useful in that students gain an experience with actual gates. The devices will be from a TTL device family that is compatible with our trainers. There is also a need for our students to perform the CAD tutorial before the first actual lab session, and students need to know about what we expect a laboratory report to be like.

## Recent Logic Circuits History

Our discussions with students helped us to understand the apparent contradiction in the response to questions 5a and 6, which consider our student's opinion of TTL logic gates, which they used to a small degree, along with CPLDs. Question 5a, which asked them about their opinion of TTL devices suggests a moderate to strong agreement that our use of TTL devices in the first lab is educationally relevant and a good use of their time. But question 6 indicates a nearly moderate agreement that students learned more with CPLDs than with TTL.

Our discussions with students helped us to understand that while students do prefer their experience with CPLDs, they also appreciate the history of the technology and to some degree feel that knowledge of a prior technology provides a foundation to better understand modern devices. The point is that our students value knowledge of TTL over its use.

Given that PLDs are simply a next step in the progression of logic technologies, we have made plans to satisfy the need for such historical perspective by first providing a hands-on experience with discrete logic devices during a start-up lab session, and also in lecture we will consider and contrast several historically significant logic families. Short of mechanical, electro-mechanical, or vacuum tube based logic, perhaps diode-transistor logic (DTL), the original 74 series TTL, basic NMOS and CMOS logic can be considered in a meaningful way, while remaining within the scope of the course.

## Summary of Recommendations for Future Plans

Our adoption of start-up lab sessions will allow us to introduce the CAD tools very early in the course and will requires a modification of the lecture content so that logic gates and essential analysis and design concepts will be presented at the very beginning of the course. We also have plans to develop new content that will incorporate the CAD tools and CPLDs deeper into the lecture component of the course as well as homework. Their actual use of the CAD tools in this regard will call for additional technical and software support for our students.

Finding a solution to resolve our dependence on Xilinx ISE 10.1 is a critical issue that we will attempt to address. We will also develop a migration plan to a newer CPLD technology such as the Xilinx[6] Coolrunner-II family of devices. We will also write a tutorial that outlines solutions for our Macintosh users to be able to use Xilinx ISE.

Finally, the new content involving hierarchical design as well as the structure of CPLDs and timing will be presented earlier and further integrated into the course. In particular, we will revise our labs to increase the number involving the use of hierarchy and also revise our lab projects to include more visual, realistic, and tangible results for students to demonstrate.

## Conclusion

In closing, this document outlines our latest experience in adopting the complex programmable logic device (CPLD) into an introductory logic circuits course. For our recent work, new lecture material involving hierarchy, propagation delay, and CPLD structure was developed, the CAD tutorial[7] was expanded, and new laboratory material was also developed to make use of these principles. We developed and presented the material involving propagation delay and hierarchy to directly replace content previously based on the use of traditional TTL devices. In lab our students use an oscilloscope to measure the timing of a signal propagating through a delay path. In addition, a new document[9] was written that presents the internal structure of a CPLD along with the timing models used in estimating the delay of signals through more complicated circuitry in a CPLD. Finally, we also collected student feedback and made future plans for the course. Appendix A provides a summary of the student questionnaire results and Appendix B outlines all the labs performed. We consider our efforts to integrate CPLDs and CAD tools into our introductory logic circuits course to be a continued success.

## 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. 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.

3. R. Coowar, "The Development of a Course in Programmable Digital Devices," ASEE National Convention, 1996, session 2648.

4. K. Nickels, "Pros and Cons of replacing discrete logic with programmable logic in introductory digital logic courses," ASEE National Convention, 2000, session 2532.

5. G. Wang, "Teaching Digital Logic Using CAD Tools in a Teaching-Oriented University," ASEE Annual conference, 2005.

6. Xilinx Corp., Homepage, http://www.xilinx.com.

7. K. Hill, "Schematic Capture – ISE 10.1," http://uhaweb.hartford.edu/kmhill/suppnotes/isetut/index.htm.

8. K. Hill, "Teaching Logic Circuits with CPLDs," http://uhaweb.hartford.edu/kmhill/suppnotes/teach_cpld/index.htm.

9. K. Hill, "Xilinx XC9536 Structure and Timing," version 1.0, Nov. 25, 2012, http://uhaweb.hartford.edu/kmhill/suppnotes/teach_cpld/CPLD_XC9536.pdf.

10. Xilinx Corp., "Hierarchical Design Methodology Guide," UG748 (v12.3) Sept. 21, 2010.

11. Digilent Inc., "Digilent C-Mod," http://www.digilentinc.com.

12. K. Hill, "XMOD Project – Univ. Hartford, CETA", http://uhaweb.hartford.edu/kmhill/projects/xmod/index.htm.

## Appendix A: Survey Descriptions and Results Summary

Our students answered questions 1 through 12b 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 right-most columns in the summary table provide the mean or 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.

|  | Question | Mean | S.D. |
|---|---|---|---|
| 1 | Using CPLDs in the Logic Circuits course is an overall improvement | 2.27 | 0.90 |
| 2 | The lab projects using CPLDs were interesting and educationally valuable | 1.82 | 1.78 |
| 3 | My experience with CPLDs makes me more confident and I foresee that in the future I will be more competent as an engineer | 1.73 | 1.01 |
| 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 | 1.56 | 1.13 |
| 4b | There should be more topics related to CAD tools presented in lecture | 2.13 | 1.13 |
| 5a | Having some experience also with TTL devices, in the first two labs, is educationally relevant and is a good use of my time | 2.27 | 0.90 |
| 5b | I was able to learn about propagation delay by using CPLDs | 1.64 | 1.80 |
| 6 | In recalling my experiences with discrete TTL devices and CPLDs, I found that I learned more with CPLDs than with TTL devices | 1.91 | 0.94 |
| 7 | I feel that a laboratory experience in which I construct circuits and investigate signals helps me to better learn the material | 2.18 | 0.75 |
| 8 | I found that in our use of CPLDs in the laboratory, the hands-on experience was retained, and helped me to better learn the material | 1.73 | 1.19 |
| 9 | Activities such as investigating the behavior of components in a design helps to retain the hands-on experience, and helped me to better learn the material | 1.91 | 1.14 |
| 10a | There should be more use of CPLD topics in the lecture portion of the course | 2.36 | 0.67 |
| 10b | It would be a benefit to incorporate exercises involving CAD tools and CPLD topics into the homework | 1.60 | 1.51 |
| 11a | The online tutorial was helpful in getting me started using CPLDs | 2.00 | 1.00 |
| 11b | The online tutorial helped me learn the principle and application of hierarchy | 2.09 | 0.83 |
| 11c | The online tutorial served as a useful reference to me, later in the course | 1.73 | 1.85 |
| 12a | The CAD software used to draw schematics and configure the CPLDs was useful and effective | 1.28 | 2.14 |
| 12b | The CAD software helped me make use of and understand hierarchy principles | 1.33 | 1.32 |

Space was provided in questions 13 through 17 to allow each to be answered with a statement. Question 13 asked our students about their largest concern for improving the course. Of the 11 responses, three students expressed dissatisfaction with Xilinx ISE, indicating that the software is unstable and difficult to use. One student expressed concern over non-functional hardware, and one student was concerned by the perceived fast pace of the laboratory.

Two students each provided an answer to question 14, which asked them to propose an activity involving CPLDs that helps retain the hands-on experience. One student suggested the design of a circuit that produces LED light patterns and another provided the notion of having more hands-on experience with CPLDs. By introducing the CAD tools earlier, we will satisfy this request.

Question 15 asked students what their favorite lab was, but some students expressed a preference for more than one lab. This last semester the "Elevator Controller" state machine was the most popular with four students expressing their preference. The "Roll the Dice" lab was the next most popular with three students expressing their interest. The "Roll the Dice" lab involves the use of a counter in the design of a state machine and of all the labs involves the most wiring. Our students generally indicated that of the labs, these were the most challenging and realistic.

Our students reported in question 16 that the decoder and multiplexer lab was their least favorite, which is just as well. We already have plans to revise this lab, to take advantage of hierarchy, which is a new topic just introduced into the course. Finally, two students used question 17 as an opportunity to repeat their dissatisfaction with the Xilinx ISE software.

**Questions:**

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?

## Appendix B List of Laboratory Projects Performed

There were eight laboratory projects. The first lab was TTL based and required students to use three TTL chips (74LS04 hex inverter, 74LS08 quad AND, 74LS32 quad OR) to construct a simple combinational circuit. Our students analyzed the circuit, generated a truth table, and tested the circuit using switches and LEDs. The second lab used a CPLD configured with NAND gates to design and implement a combinational circuit according to verbal descriptions of the circuit behavior. 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. Our 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, test bench generation, simulation, pin assignment, synthesis, and configuring the CPLD. In the fourth lab students used the CAD tool and CPLD to design, implement, and test a combinational circuit involving don't-care conditions. This second CPLD lab is intended to reinforce students' skills and their familiarity with the CPLD and Xilinx software.

Lab five made use of the CPLD by introducing medium scale integration (MSI) like combinational logic components. students investigated the function of a decoder and a multiplexer, then constructed and tested a circuit involving a decoder, a multiplexer, a flip-fop, and an external OR gate provided by a TTL chip.

The last three labs involved state machines and used the CPLD module. In lab six, students designed, constructed, and tested a simplistic four-floor elevator controller. In lab seven students analyzed, constructed, and tested a state machine that generated a Gray code sequence. Lab eight is called “roll the dice”, and is the highlight of all the labs. In this lab, students used hierarchy with a 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 stops in a randomly selected state.

## Biographical Information

Dr. Krista M. Hill is an associate professor in Electrical and Computer Engineering at the University of Hartford in Connecticut with Ph.D. and MSEE from Worcester Polytechnic Institute 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.

Dr. Ying Yu received the B.Eng. degree from Fudan University in Shanghai, China, in 2000. She received her M.Eng. and Ph.D. in Electrical Engineering from Brown University in 2003 and 2007, respectively. Since 2008, she has been teaching as an assistant professor in the Department of Electrical and Computer Engineering at University of Hartford. Her current research interests are audio and speech signal processing, bowel sound detection, 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.