Logic Circuits Lab – Breadboard or Verilog

Abstract:

Logic Circuit design is one of the introductory courses for Electrical Engineering and Electrical Engineering Technology students. Ideally, it introduces students to hands-on circuit building, problem solving, testing and function verification. The lab teaches them to use correct lab equipment from digital meters and oscilloscopes to digital logic analyzers. The logic design lab is a learning experience that most students enjoy, as it is their first hands-on experience with designing and building miniature systems.

Other instructors prefer introducing students to Verilog design as the lab experience. Students usually use evaluation boards to implement their designs. The evaluation boards have a Field Programmable Gate Array (FPGA) with hard-wired connections to LEDs, seven-segment displays and switches. Based on a survey conducted, many students appreciated the fact that they get introduced to learning Hardware Description Language (HDL) but miss the hands-on circuit building.

To obtain primary comparison results, we had two students run the two types of labs in two consecutive semesters. As primary results, both students agreed that (1) physically building the circuits gave them a better visual understanding of how different circuit elements were connected; (2) having the option to write Verilog behavioral coding, allowed them to drift away from the structural coding that was supposed to enforce the logic design concepts they learned in lecture.

Our goal was to find a middle ground between the two lab approaches and redesign our lab experiments to give students the benefits of the two options. As a compromise, a Complex Programmable Logic Device (CPLD) chip was added to the circuit building lab kit. The CPLD is soldered on a Printed Circuit Board (PCB) for easy mount on the breadboard. The PCB also has a voltage regulator and crystal clock generator. Students have the possibility to connect it to switches, LEDs or 7-segment displays and program it. The lab exercises have instructions on creating schematics and structural Verilog code. Behavioral coding is postponed to higher level courses. Preliminary outcomes on the modified lab structure has been obtained from the same two students who are helping us in amending the lab experiments as part of an internship experience. Having the student feedback during the lab handout design is helpful, in terms of identifying any points of confusion and complexity on the lab experiment. Students have stated that the lab with the upgraded format gives them the experience of hands-on circuit building and the learning of HDL. For the lab implementation in an academic semester, more assessment data will be collected for course evaluation. The preliminary data show that students are gratified with the fact that they can see the parallel connection between circuit design and CPLD programming to achieve the same functionality.

Introduction:

Logic Circuit design is an introductory course offered for Electrical Engineering, Electrical Engineering Technology, and Computer Engineering students. It is the first cornerstone in learning about circuits and digital logic systems. Figure 1 illustrates the digital logic hierarchy, and indicates the coverage of the logic circuit design lab.

There are different ABET outcomes that relate to their lab experience. It teaches them to use correct lab equipment and tools, from digital meter and oscilloscope to digital logic analyzers. It helps them to develop their problem-solving skills, as they are required to design logic circuits with given specifications and then verify their functions. They learn about test vectors and troubleshooting to detect any errors or misconnections that caused unexpected results. They also practice writing their lab reports and comparing theory to actual lab measurements. The logic design lab course is one of the learning experiences that most of the students have enjoyed based on a survey that we have conducted.

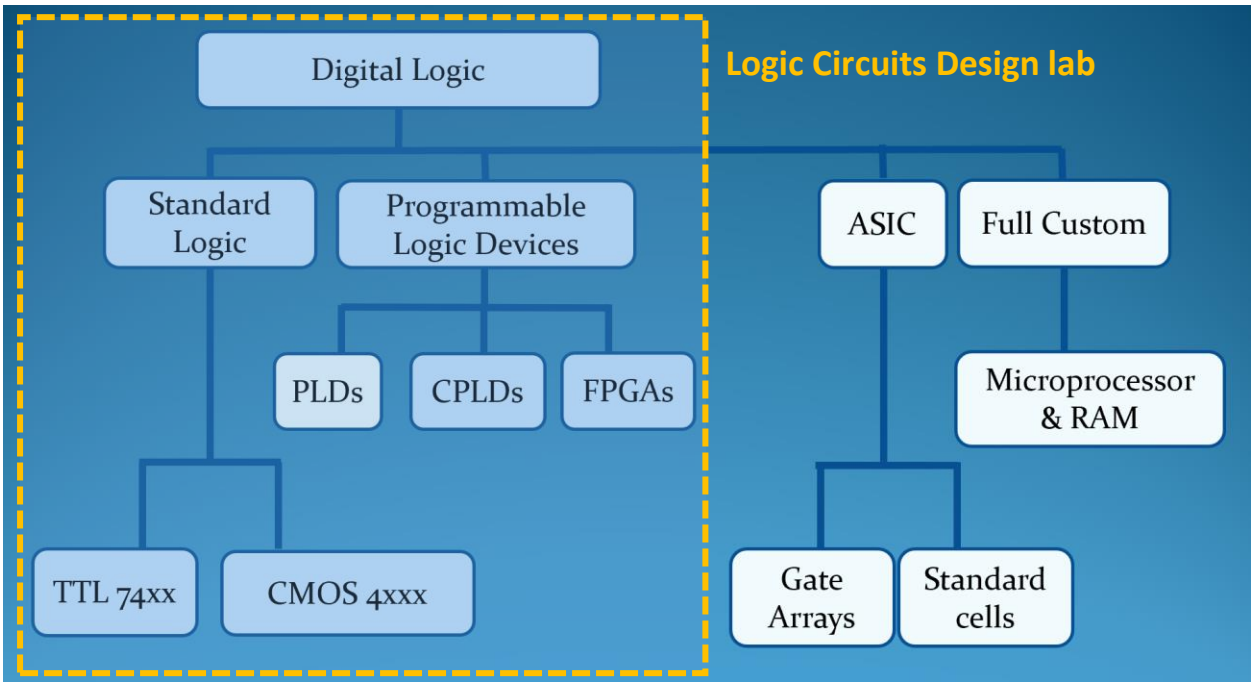

Fig. 1 Digital Logic hierarchy. The logic circuits design courses concentrate on the Standard logic or the programmable logic devices [1].

The lab exercises concentrate on using logic gate chips from the TTL 74xx and CMOS 4xxx families. Working with the standard logic gates helps in visualizing the idea of FAN-in, Fan-out and the propagation delays.

We start the lab exercises by introducing students to the data sheet parameters for voltage and current ranges that characterize a logic 'high' or a logic 'low'. Students also learn about the connection of switches to use as logic inputs and the connection of logic outputs to LEDs. In

addition they learn how to calculate the value of the resistance to use in the circuit to limit the current flowing through the gates. In this introductory lab, students also learn about FAN-in, FAN-out, and noise immunity.

In a separate lab exercise, students use the propagation delay of logic gates to build a ring oscillator, by constructing a loop of cascaded inverters. Based on the propagation delay on the data sheet and the observed signals on the oscilloscope, they verify and explain the operation of the oscillator. They also observe the rise and fall times of the output signals. This lab sets a foundation for understanding the timing analysis for a logic design. It also serves the lecture material on propagation delays for logic gates.

Design Levels	Design Description	Primitive Components	Theoretical Techniques
Algorithmic	Specifications High-level Language Math Equations	Functional Modules (black boxes)	Signal processing Control Theory Sorting Algorithm
Functional Logic Circuits Design lab	Verilog or VHDL	Registers Counters ALU	Finite State Machines Timing Analysis
Logic	Boolean equations Truth Tables Timing Diagrams	Logic Gates Flip-Flops	Boolean Algebra K-maps Boolean minimization
Circuit	Circuit equations Transistor netlist	Transistors Passive components	Linear/non-linear equations

Table 1 Levels of design abstractions in logic circuit design [1].

The logic circuits design lecture and lab concentrate on the logic and functional design levels as shown in Table 1. The circuit level is handled by electronics courses that introduce students to transistors and their application as logic gates and reinforced in courses explaining VLSI. The logic and functional levels are studied in logic circuit design courses with different methods and different focal points. The upper algorithmic levels are studied in more advanced PLD courses, or in computer architecture related courses.

For design simulation and implementation on PLDs and FPGAs, there are different design description methods that can be used:

(1) Schematic capture:
This is a schematic of the design using primitive components such as logic gates, flip-flops, decoders, encoders and wires for interconnections. There is a list of components in the software library that users can pick from.

(2) Hardware Description language:
This is a script describing the logic design. The code may have 3 main different styles:

(a) Structural design: This style is mainly a word description of the schematic capture. It handles each primitive component or building block as a module defined by its name, inputs and outputs. Top modules are constructed by connecting smaller modules using wires. *and and1 (in1, in2, out0)*

(b) Switching functions: Wires and outputs are assigned a switching function using inputs or other wires. *assign wire1 = inputA & inputB;*

(c) Behavioral design: The behavioral design describes the function of the module using if statements, case statements … etc.

Schematic Capture	Hardware Description Languages
• Good for multiple data-flow • Gives an overview picture • Relates to hardware better • High information density • Possible delay tracking	• Flexible & Parametrizable • Excellent for Optimization & Synthesis • Direct mapping to algorithm • Excellent for data-path • Readily interfaced for optimizers
• Not good for algorithms • Not good for data-path • Does not interface well for optimizers • Not good for synthesis software • Difficult to reuse • Not parametrizable	• Serial presentation • May not show the overall circuit picture • Often needs good programming skills • Does not show the physical hardware • Needs special software such as Intel Quartus®.

Table 2 Design Description [1]. A comparison between schematic capture, and HDL showing the advantages and disadvantages of both design description methods.

Aside from the advantages and disadvantages of each design description listed in table 2, we notice that the schematic capture, the HDL structural design and HDL switching functions require that the students apply truth tables, Boolean equations, Boolean minimization, and Karnaugh maps to come up with an optimum design to satisfy a system functionality. For the HDL behavioral design, they can describe the behavior of the system using Verilog or VHDL syntax and let the optimizer design the schematic.

For campuses that prefer introducing students to Verilog design as the lab experience for the logic design course, students usually implement their designs on FPGAs mounted on evaluation boards. The evaluation boards have switches, LEDs and 7-segment displays connected to specific pins of the FPGA. With correct pin assignments, students can control the input and observe the output for testing. Based on a preliminary survey, some students appreciated the

fact that they get introduced to learning HDL; but they missed the circuit building part and mentioned that they sometimes felt that the lab was not flowing in parallel with the lecture.

To obtain primary comparison results, we had two students run the two types of labs in two consecutive semesters. As primary results, both students agreed that:

(1) Physically building the circuits gave them a better visual understanding of how different circuit elements were connected. As all the logic design concepts were new to them, having this hands-on circuit building aspect helped them get a deeper understanding of the material covered in lecture.

(2) Having the option to write Verilog behavioral coding, allowed them to drift away from the structural coding that was supposed to enforce the logic design concepts they learned in lecture. The higher-level behavioral description coding is easier to write based on the functional requirements of the system.

(3) They have appreciated the importance of the proper documentation of the code in terms of descriptive comments and meaningful wire names to help them track any errors.

One of the main ideas that have been raised by faculty members was that when students work with the full evaluation board, such as the one shown in figure 2, they think that this is how it would be integrated in any system. With the bulky form, they don't tend to think of it as a controller option for their capstone projects for example. They tend to prefer to work with more compact microprocessor boards such as Arduino.

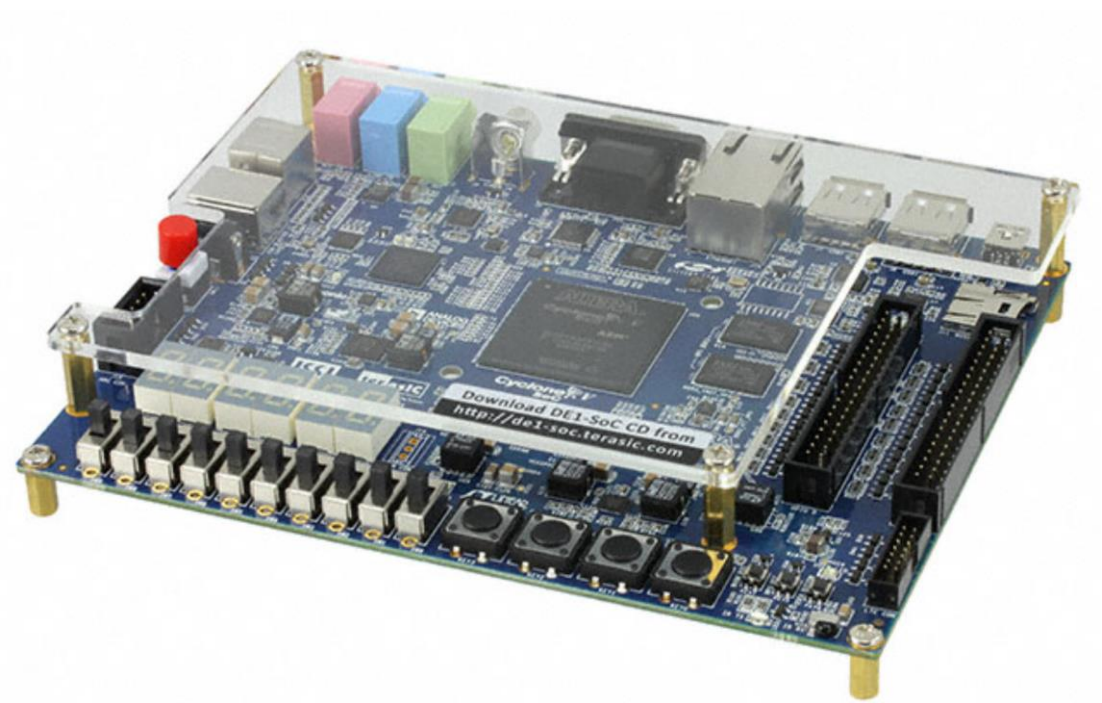

Fig. 2 Intel-Altera DE1-SoC board, one of the commonly used evaluation boards for digital design courses. It offers multiple applications with hardwired peripherals.

Our goal was to find a middle ground between the two lab approaches and redesign our lab experiments to give students the benefits of the two options keeping in mind the feedback and concerns that were reported. As a compromise, a Complex Programmable Logic Device (CPLD) chip was added to the circuit building lab kit. The CPLD is soldered on a small Printed Circuit Board (PCB) that can be mount on the breadboard. The PCB also has a voltage regulator and crystal clock generator. Students have the possibility to add external connections and visualize the integration of the CPLD as a controller in a larger system.

Lab kit addition and modification of lab experiments:

The new addition to the breadboard lab kit for the logic design course is the PCB with mounted MAXV CPLD chip as shown in Figure 3. The board can be easily mounted on a breadboard to integrate in any design.

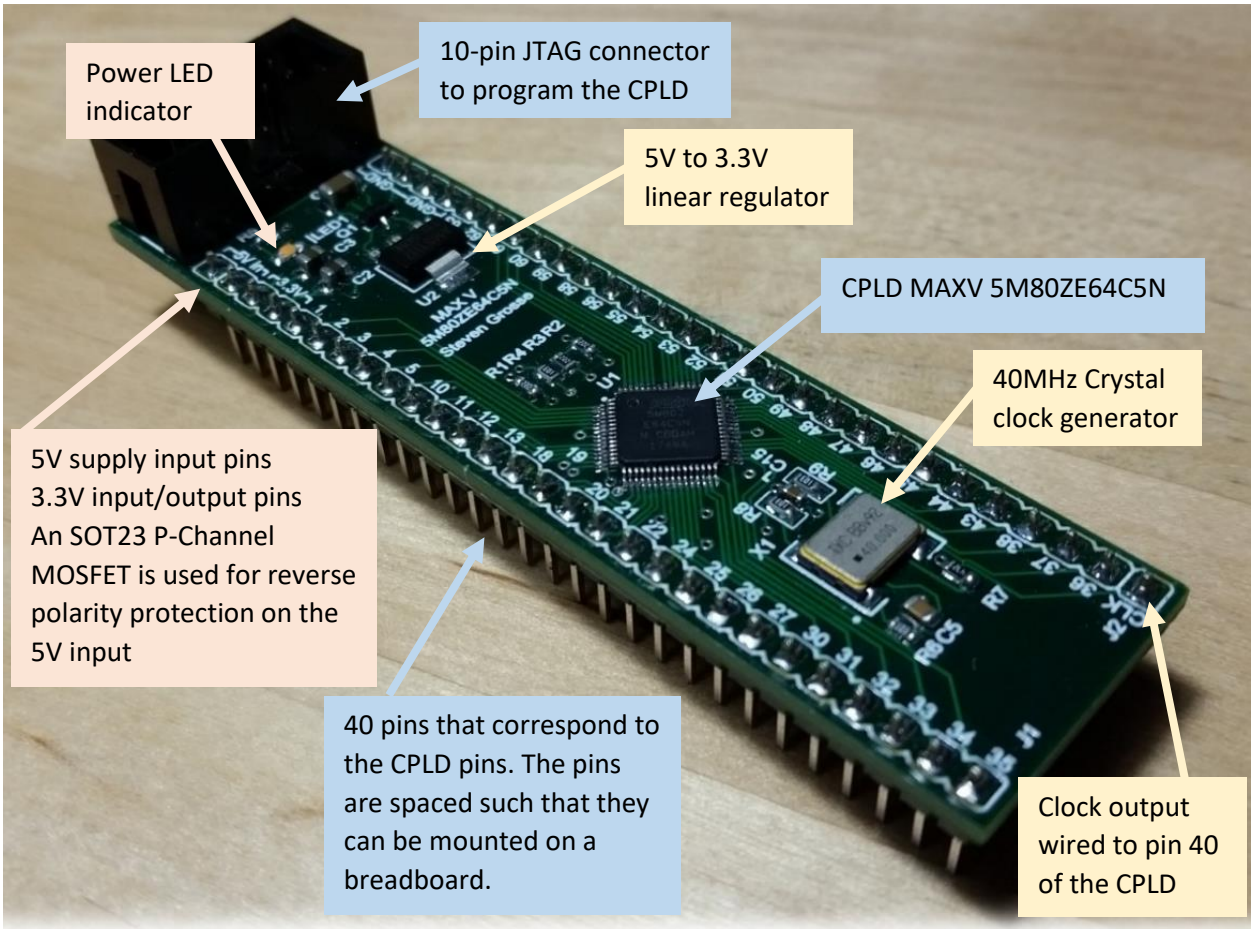

Fig. 3 PCB board with mounted CPLD.

The lab experiments are modified to include some information on the application of CPLD in terms of external connection, pin assignment using Altera Quartus®, Verilog syntax and coding best practices. The learning curve for applying CPLDs is spread over all labs, so that we give

students a steady practice every lab and to make sure that the lab can be still completed over the three-hour lab period.

Experiment 1: **Logic Inputs and Outputs**. An introduction to the data sheet parameters for voltage and current ranges that characterize a logic 'high' or a logic 'low'. Students also learn about the connection of switches to use as logic inputs and the connection of logic outputs to LEDs. Students are introduced to the terms FAN-in, FAN-out, and noise immunity. A simple inverter circuit is used for standard logic chips. In addition, we review similar connections for the CPLD chip and highlight the difference in power supply requirements.

Experiment 2: **Introduction to Altera Quartus®**. This lab uses a step-by-step guide through the application of Quartus® to generate a schematic of a simple logic circuit, simulate it using a timing diagram and then implement it in hardware using the MAXV chip. The inputs are controlled using switches and the outputs are displayed on LEDs.

Experiment 3: **Verify a Boolean equation.** Students use truth tables to verify a Boolean identity such as the distributive property. They build the circuit and test the output for all possible input combinations. For this lab, students are introduced to Verilog coding. They will assign one output to (A+B·C) and another output to (A+B)·(A+C). The inputs will be connected to switches and the two outputs will be displayed on LEDs. Going through all the possible input combinations, they verify that the two outputs are the same.

Experiment 4: **Binary Code Converter**. Using Karnaugh maps, binary numbers from 0-9 will be converted to Excess-3 code. Students will verify the design by simulation and implementation on the MAX V chip. Students would simulate the schematic design and the Verilog code specified by the switching function for each output.

Experiment 5: **Logic Decoder**. Using Karnaugh maps for every LED on the seven-segment display, students convert a 4-bit binary input into the decimal value from 0-9. The mapping is simplified by the don't cares for input values between 10 and 15. Students can write the switching function for each input using Verilog and implement on the MAX V CPLD. The inputs are controlled using switches and the output pins are connected to a seven-segment display.

Experiment 6: **Propagation Delay – Ring Oscillator**. Student use the propagation delay of logic gates to build a ring oscillator, by constructing a loop of cascaded inverters. Based on the propagation delay on the data sheet and the observed signals on the oscilloscope, they verify and explain the operation of the oscillator, and why the number of cascaded inverters play a role in the oscillation perpetuation. They also observe the rise and fall times of the output signals. This lab sets a foundation for understanding the timing analysis for a logic design. It also serves the lecture material on propagation delays for logic gates.

Experiment 7: **Multiplexers and Decoders**. The design specifications are given as a list of minterms. Students will be applying decoders and multiplexers from the library of primitive

elements given in Quartus® and will build the circuit using schematics and Verilog code. They will create a testbench using the timing diagram to test the functionality of their design.

Experiment 8: **Full Adders & Subtractors**. Students will design full adders and full subtractors using binary inputs. As extra credit they are given the option to incorporate their logic decoder design from experiment 5 to display the input and output on a seven-segment display.

Experiment 9: **Flip-Flop Conversion**. Here, students convert the operation of a D flip-flop into a JK flip-flop. This is the first lab that introduces them to sequential logic.

Experiment 10: **Finite State Machine**. Students construct two Mealy FSMs that have two states and verify the functionality of the system. They will build it using JK flip flops and then also implement it on the MAX V chip. Students will be introduced to the case statement in Verilog.

Experiment 11: **Counters**. This lab provides students a hands-on experience of designing synchronous counters to count using a predefined sequence. For example, if an input x=1, the counter should output 2753 2753 … etc and if x=0 it should count 3572 3572…etc. Students use a 74112 JK flip-flop chip for the design and use a function generator to create the clock signal. The output will be observed using a logic analyzer. For the CPLD implementation, a Verilog code will be applied. The 40MHz clock on the board will be used as the input clock to the system.

Experiment 12: **Frequency Divider**. As an application for the counter lab, students are required to design a digital logic system that would divide a clock frequency by powers of two. Students will write Verilog code and come up with a way to test their system.

Preliminary outcomes and future work:

All of the lab experiments described have been well tested over several years in our program. The main addition to the lab kit is the CPLD board along with the accompanied HDL code to program it. This is one form of combining the advantages of using breadboard circuit building methods to reinforce the logic design concepts and at the same time, introduce students to HDL and PLDs.

Pilot outcomes have been obtained from two students who have implemented the two different forms of labs and who are helping us in modifying the lab experiments as part of their internship experience. Having the student feedback during the lab handout design is helpful.

As a generalizable insight into the paper, the goal was to integrate new tools to an introductory course that teaches basic concepts. Giving students hands-on experience on structuring the same design using basic building blocks and applying more advanced tools that they will see in industry was the main objective. Having the two options run in parallel in lab experiments should help them visualize the analogy in the two methods for a more profound understanding and easier prospect implementation.

Our future plan is to design an online version for the logic circuit design lab. We are planning to incorporate a Digilent Analog Discovery 2 USB Oscilloscope and Multi-function Instrument shown in Figure 4.

Fig.4 Digilent Analog Discovery 2 USB Oscilloscope and Multi-function Instrument.

References:

1. P. Cheung, “Digital System Design” www.ee.ic.ac.uk/pcheung/teaching/ee3_DSD/index, Imperial College London, 2008

2. P. Chayratsami, "Supplementary laboratory in digital circuit and logic design course for pre-service vocational teacher in Thailand," *2013 IEEE Global Engineering Education Conference (EDUCON)*, Berlin, 2013, pp. 612-617.

3. Bachnak, B., Elaraby, N.; "Developing Lab Exercises for Logic Circuit Design using FPGAs." *2018 ASEE Mid-Atlantic Section Spring Conference, University of the District of Columbia, Washington DC, April 7th, 2018*