Active Learning in the Introduction to Digital Logic Design Laboratory Course

Jing Pang
Department of Electrical and Electronic Engineering, Computer Engineering Program, California State University, Sacramento, Sacramento, California, USA

Abstract

The introduction to digital logic design class is in general the first digital course for electrical and computer engineering undergraduate students at many universities. The related laboratory offers students hand-on experience to apply the principles of Boolean algebra and K-maps for minimizing logic expressions, to develop skills on using digital logic chips, to understand sequential circuit operations, to perform basic finite state machine design work and to get ready for more advanced digital logic design study. In order to stimulate students in active learning inside the laboratory, this paper proposes effective technology integration with laboratory instruction including using online tools for logic circuit visualization, applying Multisim for transition to hands-on breadboard hardware experiments, running simulation in Moldelsim and prototyping the digital circuit design using Field Programmable Gate Array (FPGA) device.

Keywords

Active learning, online logic circuit visulization, Multisim, Modelsim, FPGA.

Introduction

The digital logic design has been defined by IEEE and ACM societies as one of core areas[1]. The introduction to the digital logic design course is usually the first digital design course that is offered to the lower division undergraduate engineering students at many universities. It gives introduction to undergraduate students about logic gates, Boolean algebra and K-maps which are the basis for the digital design. It also teaches the structure and functionality of the clock driven sequential circuits. In addition, it covers the finite state machine concepts, and illustrates how to represent the finite state machine with state diagram and how to design the basic finite state machine.

After studying logic design concepts from formal lectures, students are generally required to attend the laboratory to acquire hands-on design skills[2]. Through the laboratory work, students are expected to develop design experience and problem-solving skills which are important for the engineering profession[3]. However, after coming out of high school, most undergraduate students may have got used to passively accepting facts provided by instructors. They have not gained skills for analyzing design problems, synthesizing and evaluating design information. Moreover, many lower division undergraduate students have little or no exposure to digital logic circuit design in high school. Therefore, it is very important for the instructor to develop effective teaching pedagogy to engage students in active learning[4-5] for the logic design laboratory class.

In order to solve the above problems, this paper presents multiple active learning methods the author has applied in teaching the introduction to logic design laboratory class. First, the online interactive logic circuit visualization can be used to get students' attention and also demonstrate the functionality of the logic circuits. Next, the professional Multisim[6] schematic capture and simulation tool can help students identify and correct logic design errors earlier in the design flow, and reduce hardware design iterations before students build real hardware circuits on breadboards. In addition, FPGA[7-8] device can be used by students to prototype digital design with the help of the Verilog hardware description language. Furthermore, Modelsim[9] simulation of the Verilog HDL design allows students to actively create simulation data, analyze the design results, and gain deeper understanding of the digital circuit functionality.

Online digital circuit visualization

Logicly is a free online tool which allows users to design basic logic circuits in drag-and-drop method. The web address for Logicly is http://logic.ly/demo. The Logicly offers input logic components of toggle switch, push button, clock pulse, constant '1' logic, and constant '0' logic. It also provides output logic components of light bulb, and 7-segment display with built-in driver. It has basic logic gates of Buffer, NOT gate, AND gate, NAND gate, OR gate, NOR gate, XOR gate, XNOR gate and Tri-state buffer, as well as SR latch, D Flip-flop, JK Flip-flop and T Flip-flop. Figure 1 shows how students can use drag and drop method to test the functionality of the basic AND gate.

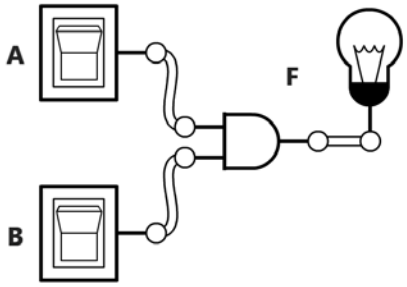

Figure 1. Drag and drop learning of AND gate

In Figure 1, students drag and drop two toggle switches, one AND gate, one bulb into the schematic panel and then wire them together in Logicly. After they create the circuit, they can click on the two input toggle switches to generate logic '1' or logic '0' inputs. When the output is logic '1', the bulb will be on with blue color. Otherwise the bulb will be off without color. Students can record the experiment data by observing different input logic combinations and the corresponding output logic values. After this exercise, the lab instructor can organize group discussion to ask students to summarize the behavior of the AND gate. Usually, after small group discussion and then the whole class discussion, students can successfully draw the conclusion that only when both inputs are logic '1's, the AND gate output will be logic '1'. Similarly, students can familiarize themselves with other basic logic gates. Afterwards, the lab instructor can organize students to work on lab experiments to actively verify the Boolean algebra equations by using Logicly.

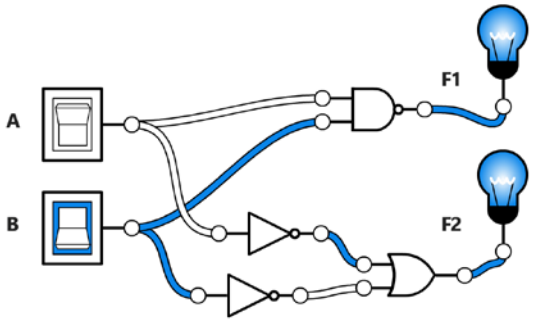

Figure 2. Logicly circuit experiment for verifying DeMorgan's Law

For example, in order to verify DeMorgan's law defined by Boolean algebra, students can build circuit in Figure 2 using Logicly. The Boolean logic expression for the circuit in Figure 2 is described by the following equations.

$$F1 = \overline{A \bullet B} \qquad (1)$$

$$F2 = \overline{A} + \overline{B} \qquad (2)$$

In Figure 2, both the F1 and F2 outputs share the same inputs of A and B. When students toggle A or B switch, they can easily tell the outputs F1 and F2 turn on or turn off both bulbs in the same way. As a result, this experiment can prove that F1 and F2 logic equations are equivalent. After this experiment, the lab instructor can organize the discussion of the behavior of the Figure 2 circuit. The conclusion is that the NAND operation of two inputs is equivalent to the OR operation of the inverse logic of each input.

The circuit diagram in Logicly is color coded. In Figure 2, the blue color connections indicate logic '1's, and the white color connections indicate logic '0's. The color-coded logic level on the circuit diagram helps engage students interactively, assists students to verify the logic values of the circuit and makes learning more interesting.

In general, Logicly is a handy online tool which allows students to actively learn logic fundamentals, and also provides students free design environment to verify the functionality of more complex logic designs at the later stage of their logic laboratory study. The laboratory instructor can use Logicly to organize logic design assignments and discussions to stimulate student interest in logic circuit design and help students get better understanding of the logic circuit functionality.

Digital hardware schematic capture and visualization

In the introduction to the logic design laboratory, students are usually required to learn how to build the real hardware circuits. They start with using the logic gate chips on solderless breadboards. First, they need to place the hardware components on the breadboards. Next, they need to wire them together. It usually takes beginners some time to complete the whole process.

In case the final logic circuit outputs do not work as expected, they will need to use Multimeters to troubleshoot the circuit. Such experiments help students gain hands-on skills with building and testing digital logic circuits.

However, once logic circuits become more complicated, it will become cumbersome for students to troubleshoot circuits with design problems. Especially, the beginning learners will feel very frustrated with failure after they try to fix problems multiple times. Therefore, proper support is needed to help students gain confidence and also improve problem solving skills in active learning of the real logic hardware design work.

Multisim is one of the industry tools for designers to work on electronic schematic capture and simulation. As a computer aided design tool, it allows students to check logic design and identify problems at the early design stage. Students can change the size and resolution of the logic circuit schematics to view design details on computers. They can also change the schematics to improve design quality. Furthermore, they can experiment with different hardware parameters to see their impact on the circuit performance. Figure 3 shows one laboratory circuit design example by using the Multisim tool.

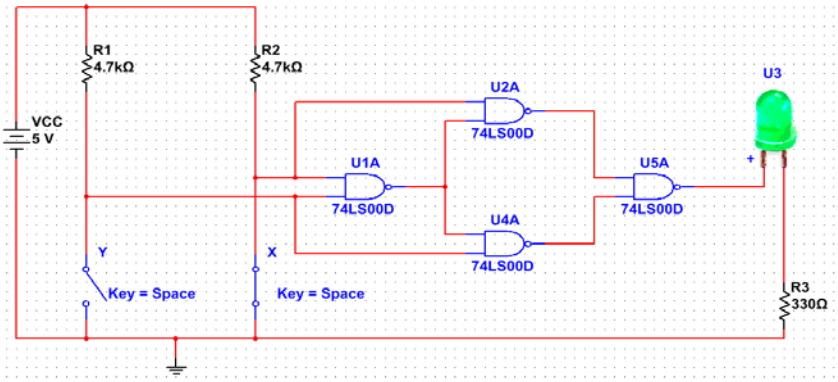

Figure 3. Multisim schematic design example

After students complete schematic design in Figure 3 using Multisim, they can run simulations to change the X, Y input logic values and observe the output logic. At the same time, they record the experimental data as shown in Table 1.

Table 1. Experimental data recording

Inputs		Output
X	Y	F
0	0	0
0	1	1
1	0	1
1	1	0

The above Multisim schematic experiment can guide students in building the real hardware circuit on the breadboard. Figure 4 shows the sample of student laboratory design work following the Multisim design in Figure 3. The left hand side of the Figure 4 has 8 switches and 10-pin bussed pull-up resistors. Only two input switches with two resistors are used to provide two logic inputs to the circuit. One quad 2-input NAND gate 74LS00 chip is inserted into the breadboard. The LED bars and 10-pin bussed pull down resistors are used for the output logic indication. The power supply of 5V is marked as "Vcc" and the ground is marked as "Gnd". Although in this experiment, only one output LED with one output pull down resistor is used, students can use LED bars and the bussed resistors to support multiple outputs in different laboratory experiments. After completing hardware placement and wiring on the breadboard, students can press the input switches, observe output LED logic and also record the breadboard experimental data. And then they can compare the results with the data in Table 1 to evaluate if their real hardware design work is correct or not.

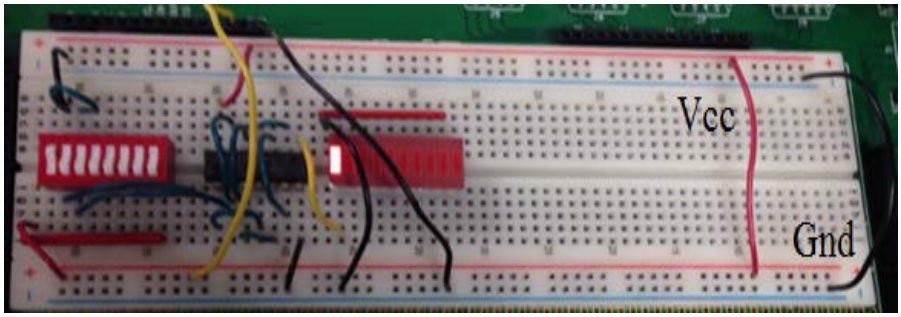

Figure 4. Breadboard design work

The Multisim schematic capture and simulation help students increase the productivity and accuracy for constructing real hardware work on breadboards. In addition, the Multisim tool also allows students to record simulation data, analyze data, and think more critically about the functionality of the circuit. The Multisim tool empowers students to build logic circuit design skills and engage student in deep learning.

Verilog hardware description language based FPGA design

Besides schematic capture method for logic circuit design, the Verilog hardware description language (HDL) is another modern digital design method for programming the structure, design and operation of digital circuits. In addition, the HDL designs can be downloaded on the FPGA devices, and students can directly check the design results on FPGA hardware.

At CSUS, the DE0-Nano development board[10] with Altera Cyclone IV EP4CE22F17C6N FPGA device[11] is used in the introduction to the logic design laboratory. The DE0-Nano has two external GPIO headers to connect FPGA designs with external digital components. One interesting laboratory assignment is to use FPGA to control external seven-segment display device.

The common cathode seven segment display device has internal diode LEDs with external seven segment connectors from segA to segG. When one segment gets logic '1' input, the associated diode LED will be on. In Table 2, the truth table below is used for displaying characters A, b, C, d, E, F, G and also blank on seven-segment display device by using three input control switches Sw1, Sw2 and Sw3.

Table 2. Truth table of character display on seven segment display device

Inputs			Outputs							Functionality
Sw1	Sw2	Sw3	segA	segB	segC	segD	segE	segF	segG	
0	0	0	1	1	1	0	1	1	1	Display A
0	0	1	0	0	1	1	1	1	1	Display b
0	1	0	1	0	0	1	1	1	0	Display C
0	1	1	0	1	1	1	1	0	1	Display d
1	0	0	1	0	0	1	1	1	1	Display E
1	0	1	1	0	0	0	1	1	1	Display F
1	1	0	1	0	1	1	1	1	1	Display G
1	1	1	0	0	0	0	0	0	0	No display

With the truth table defined above, the seven segment display controller can be designed by using three inputs and seven outputs. The K-map for one of the output segE is shown in Figure 5.

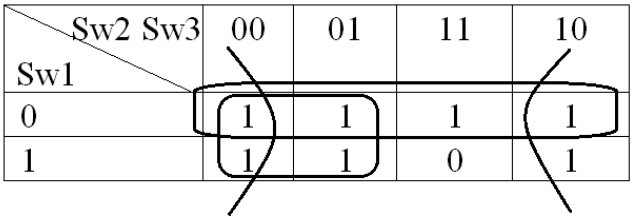

Figure 5. K-map for the seven segment display control of the output E segment signal segE

According to the K-map above, the segE logic can be derived in the logic equation (3).

$$segE = \overline{Sw1} + \overline{Sw2} + \overline{Sw3} \qquad (3)$$

The above logic equation can be designed in Verilog as following:

assign segE = (~Sw1) | (~Sw2) | (~Sw3);

Furthermore, the logics for segA, segB, segC, segD, segF, and segG can also be derived in the similarly way. The Figure 6 shows one sample of student work of using FPGA to control the external seven-segment display device. Usually, students feel very rewarding after they see the

working character display on the hardware device and this stimulate them more for active learning about logic design.

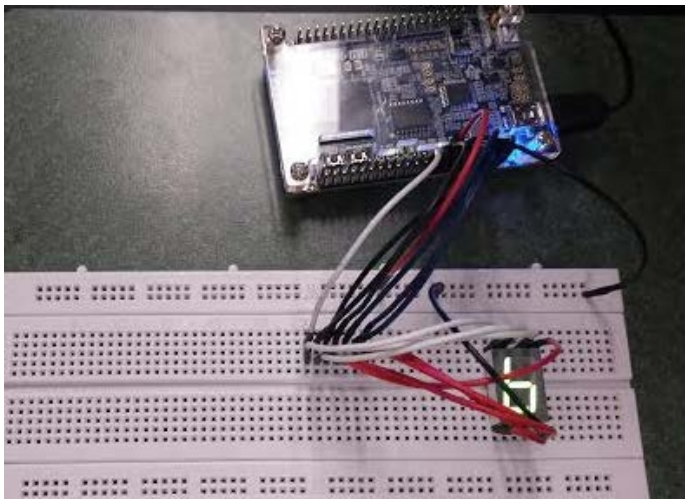

Figure 6. FPGA design of character display on seven-segment display device

Students can be further challenged to design all seven segment outputs with particular number of AND, NAND, NOR, and NOT logic chip resources on breadboards defined by the instructor. They have to work hard to manage the logic conversions necessary for all segment outputs to fit into the hardware resource requirement. Both Multisim and FPGA device can be used to verify the design schematics before they build the final hardware circuit on breadboards. This gives students opportunities to apply and reflect on what is being learned about Boolean algebra. Such type of laboratory exercise will stimulate students more on active learning.

Modelsim waveform simulation

The Modelsim tool is one of the industry level verification and simulation tools for testing Verilog HDL designs. The Figure 7 below shows the Verilog design and the related Modelsim testing commands for 3-stage right shift registers. When the parallel load signal PL is active high, after rising edge of clock, the 3-bit data vector p will be loaded into the 3-bit D Flip-Flop type of register bank q. When PL is not active, the data inside the 3-bit registers will shift from the left to the right.

```
// Verilog codes:
module sfreg3 ( clock, SI, PL, p, q);
 input clock, SI, PL;
 input  [2:0]  p;
 output [2:0]  q;
 reg    [2:0]  q;

 always@(posedge clock)
 begin
  if (PL)
    q <= p;
  else
  begin
    q[2] <= SI;
    q[1] <= q[2];
    q[0] <= q[1];
  end
 end
endmodule
```

Modelsim Commands:

```
force clock 0 0us, 1 1us -r 2us
force PL  0 0us, 1 1.8us, 0 6.2us
force SI  0 0us, 1 2.2us, 0 4.6us, 1 8us, 0 12us
force p "011" 0us
run 20us
```

Figure 7. The three stage shift register Verilog design and the Modelsim commands for testing

In Figure 7, the Modelsim commands create a clock with 50% duty cycle and 2 us period, as well as the SI, PI, p testing data values associated with the different testing time intervals. The resulting simulation waveform is shown in Figure 8.

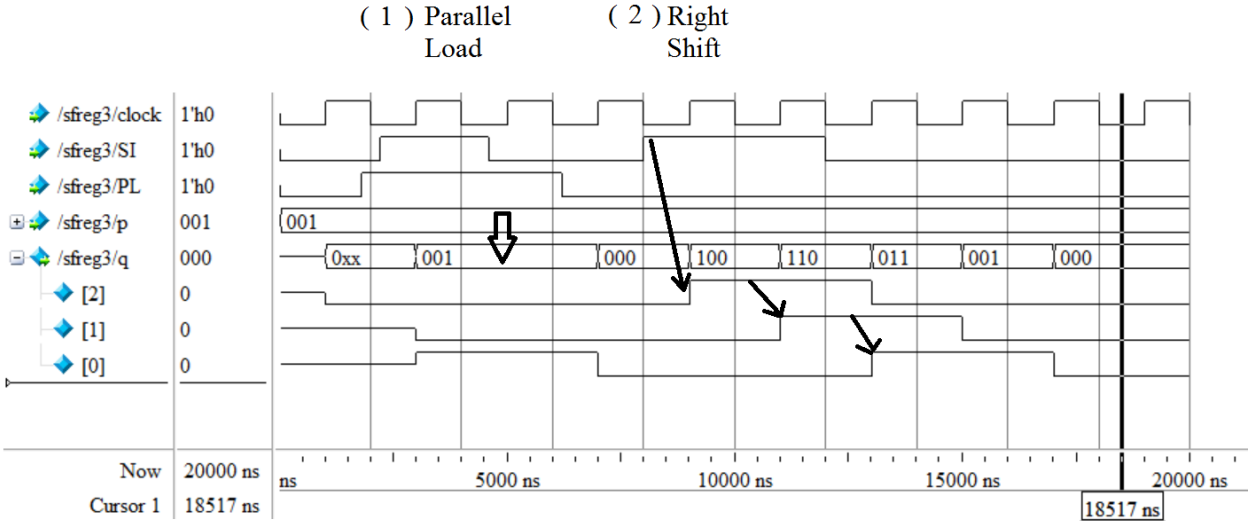

Figure 8. Modelsim simulation waveform of the 3-stage shift register design

From the Figure 8 simulation waveform, students can clearly identify the parallel load functionality, and also the right shift functionality. The simulation waveform helps students visualize the circuit functionality and develop deeper insight of the shift registers. After simulation study of the design, students can download the final work on the FPGA board. Three

LEDs located on the FPGA board can be associated with the three shift register outputs. Students can compare the LED display data with the waveform simulation data to further confirm their analysis results.

Next, after additional study of the finite state machine state diagram and the state table, students can use sequential D Flip-Flops to design finite state machine. They can also learn how to use behavior style to describe finite state machine in Verilog HDL. With all of the above skills acquired, students will feel more comfortable to move forward to the more advanced logic design experiments.

Conclusion

In this paper, the active learning pedagogy to engage students in the introduction to the logic design laboratory proposed by the author is discussed. The online interactive logic circuit visualization is cost-effective tool which can stimulate students to actively experiment and learn the functionality of the logic circuits. The professional computer aided Multisim design tool is user friendly and it allows students to simulate and verify the digital circuits before creating the real hardware on the breadboard. Such practice supports students in active learning, improving design productivity and accuracy, and reducing the design troubleshooting time. Moreover, the Verilog HDL based FPGA design approach allows students to use the modern FPGA technology to implement the digital design. Students can modify FPGA designs by changing the Verilog HDL codes. The reprogrammable feature of FPGA allows students to save hardware cost, reduce design time, and actively participate in the laboratory design work. Furthermore, the commands based Modelsim simulation allows students to simulate the Veirlog design to gain deeper understanding of the functionality of different digital logic circuits.

References

1 Vladimir Pavlovic, Zarko Stanisavljevic, Bosko Nikolic and Jovan Đordevic, “Digital Logic Simulator”, 2011 2nd Eastern European Regional Conference on the Engineering of Computer Based Systems, Bratislava, Slovakia, 2011, pp. 155 – 156.

2 P.W.C. Prasad, Abeer Alsadoon, Azam Beg, and Anthony Chan, “Incorporating Simulation Tools in the Teaching of Digital Logic Design”, 2014 IEEE International Conference on Control System, Computing and Engineering, Penang, Malaysia, 2014, pp. 18 – 22.

3 Lyle D. Feisel and Albert J. Rosa, “The Role of the Laboratory in Undergraduate Engineering Education”, Journal of Engineering Education, Vol. 94, Issue 1, John Wiley & Sons, Inc., January 2005, pp. 121 – 130.

4 Merrill Harmin and Melanie Toth, “Inspiring Active Learning: A Complete Handbook for Today’s Teachers”, Association for Supervision & Curriculum Deve, Alexandria, Virginia, 2007.

5 Sagini Jared Keengwe, “Handbook of Research on Educational Technology Integration and Active Learning”, IGI Global Book Series: Advances in Educational Technologies and Instructional Design, Hershey, Pennsylvania, 2015.

6 Guoping Wang, “Principles and Practices: Multisim in Teaching Digital Systems Design”, Proceedings of the 2006 ASEE Illinois/Indiana and North Central Conference, Fort Wayne, Indiana, 2006.

7 Shawki Areibi, “A First Course in Digital Design Using VHDL and Programmable Logic”, Proceedings on 31st ASEE/IEEE Frontiers in Education Conference, 2001, Vol. 1, Reno, Nevada, October 10-13, 2001, pp. TIC-19–TIC-23.

8 Sujittra Sothong and Pornpimon Chayratsami, "Design of Combinational Logic Training System Using FPGA", 2010 40th ASEE/IEEE Frontiers in Education Conference, Washington, D. C., October 2010, F4F-1 - F4F-4.
9 Guoping Wang, "Teaching Digital Logic Design Using CAD Tools in a Teaching-Oriented University", Proceedings of the 2005 American Society for Engineering Education Annual Conference & Exposition, Session 3532, Portland, Oregon, 2005.
10 Terasic Technologies Inc., "DE0-Nano User Manual", Spanish Fork, Utah, 2012, pp. 1 - 154.
11 Altera Corporation, "Cyclone IV Device Datasheet", San Jose, California, 2014, pp. 1 - 44.

Jing Pang

Jing Pang has been Associate Professor in Department of and Electrical and Electronic Engineering, and also Computer Engineering Program at CSUS since 2009. She got her Ph. D. in Electrical Engineering from Ohio University in 2003 and she became an Assistant Professor at California State University, Sacramento in 2003. At California State University, Sacramento, she taught courses including introduction to logic design, advanced logic design, microprocessor, computer architecture, advanced static timing analysis, audio and video coding and data compression, and digital speech analysis. Her research interests are digital design, FPGA, digital signal processing, digital image processing, and ASIC.