An Active-Learning Approach to Teaching Digital Logic Design

Sami Khorbotly
Electrical & Computer Eng.
Valparaiso University
Valparaiso, IN
sami.khorbotly@valpo.edu

Eric Johnson
Dean, College of Engineering
Valparaiso University
Valparaiso, IN
eric.johnson@valpo.edu

Douglas Tougaw
Electrical & Computer Eng.
Valparaiso University
Valparaiso, IN
doug.tougaw@valpo.edu

Abstract

Digital systems, in general, and digital logic design, in particular, are essential topics in the curriculum of electrical and/or computer engineering programs. It is virtually impossible to come across an electrical/computer engineer who has not received training in the design and analysis of combinational and sequential systems. Various schools, however, use differing approaches to structure their classes and deliver their content. In our program, the class consists of two main components: the lecture and the lab. In addition to elements of a traditional, presentation-based lecture, the "think, pair, share" technique is used to ensure an adequate delivery and understanding of the course contents.

The paper describes the course educational objectives and includes a full schedule of activities. A detailed description of both the design projects and the laboratory projects is also provided. The paper is concluded with the course assessment results including both the student feedback and the faculty comments and reflections.

Introduction

Digital Logic Design is a required class in the curricula of Electrical Engineering, Computer Engineering, and most Computer Science programs around the world. It is very common for the class to have a laboratory component associated with it.

A significant amount of work has been done to prove the effectiveness and promote the use of active learning techniques [1-2], as opposed to the traditional lecture-based teaching. In our version of the class, we adopted the famous Think, Pair, Share (TPS) active learning approach [3] in a project-based setting to cover a significant part of the course materials. In this approach, students complete a total of seven laboratory projects and six design projects that are interlaced over the 14 weeks of instruction. In the first laboratory experiment, the instructor introduces the CAD software to be used in designing logic circuits. For the rest of the semester, the activities are scheduled as follows:

- Design Project – **Think** component. Each student is to work individually on an assigned project. These projects involve conceptual design as well as the use of CAD software tools, and simulating the final design.

- Laboratory Project - **Pair** component. Students are asked to team up in groups of two. Teammates will compare their individual design projects and decide on which design is the best to build in the laboratory.

- Laboratory Report - **Share** component. In this step, each student writes an individual report explaining the design and sharing the results of the project.

The advantage in this structure is that each student gets the time to think, research, and simulate a solution to the project before sharing their findings with a peer. The two peers then adopt a design to implement. The adopted design may be based on one of the two original solutions or a hybrid combination of both.

The rest of this paper is organized as follows: the next section describes the course, its catalog description and its educational objectives; the subsequent section details the design projects and the laboratory projects assigned to carry out the TPS technique; the following section includes assessment results of the last offerings of the class; and the last section includes final comments and reflections from the instructor.

Course Description

At our university, the digital logic design course is a required course for all electrical engineering and computer engineering majors. It is also listed as a technical elective for computer science majors. The class is three credit-hours, covering both the lecture and the laboratory components. The course is scheduled on a biweekly periodic rotation where the class meets for five hours of lectures and three hours of laboratory meetings every two weeks.

The catalog description of the class is as follows:

Digital Logic Design 2.5+1.5, 3 Cr.
(Also offered as CS 320.) An introduction to digital logic concepts, including the analysis and design of combinational and sequential digital circuits.

The course educational objectives are listed as follows:

- Convert values between decimal, binary, hexadecimal, and octal number systems.
- Perform arithmetic operations on binary, hexadecimal, and octal number systems.
- Design minimized combinational logic circuits using Karnaugh Maps or Espresso/Minilog given natural language descriptions or truth table representations.
- Implement combinational circuits using simple gates, complex gates, steering logic, programmable logic, or universal gates.
- Given a set of input waveforms, sketch the output waveforms for various memory types (SR, JK, D, and T), and element classifications (latch, gated latch, and edge-triggered flip flop)
- Create a minimized finite state machine given natural language descriptions or state diagram representations.
- Implement (create, verify, and debug) digital designs using integrated circuits (both standard off-the-shelf parts and PLDs).

Content Description

As explained earlier, the various learning modules are each organized into an individual Design Project (DP) for the students to “Think” about the problem and try to individually devise a

solution, a Laboratory Project (LP) where two students are "Paired" together to discuss their ideas and implement a solution, and an individual Laboratory Report for each student to "Share" his/her solution with the rest of the world.

In an introductory Laboratory meeting (Lab 0), the students are introduced to OrCAD PSpice [4], the design tool that will be used in a significant part of the class. Then, a total of six learning modules are implemented to cover the course contents. Each module runs over a period of two weeks. This section describes the contents of each of these modules.

Module # 1

DP: In this DP, students are asked to use their newly acquired knowledge of universal gate sets to design a half adder using only one type of gates, typically two-input NOR or two-input NAND.

LP: In this LP, students are asked to implement the half adder developed in the DP using small scale integration (SSI) circuits. The main goal when using SSI circuits is to create a design that uses the minimum number of Integrated Circuits (ICs).

Module # 2

DP: In this DP, students are asked to design a code converter that converts ***Excess-3*** to ***2 of 5*** codes. Students are required to create two different designs of the same code converter. The first design shall minimize the number of gates in the circuit while the other one minimizes the number of ICs in it.

LP: In this LP, students are asked to implement an "Excess-3" to "2 of 5" Code converter using small scale integration (SSI) circuits.

Module # 3

DP: This DP focuses on iterative design practices to create larger designs. Students are asked to use 4 copies of a full-adder/subtractor to create a two's complement 4-bit adder/subtractor circuit.

LP: In this LP, the task is to implement the 4-Bit Adder/Subtractor using Altera's Development and Education 2 (DE2) board [5]. This requires students to get familiar with Altera's Quartus II CAD environment as well as the board.

Module # 4

DP: In this DP, students study the main binary memory unit, the D edge-triggered flip flop. They also use it to build a provided sequential logic circuit design. They also analyze the provided sequential system and explain its functionality.

LP: In this LP, students are asked to implement the sequential system from the DP on the DE2 board and analyze it. This requires the students to create a clock signal and connect it to the system.

Module # 5

DP: This DP is very similar to the one in the previous module except the students will design a sequential system rather than building a provided one. The sequential system to be designed and implemented is a parallel-input, parallel-output (PIPO) shift left/load register.

LP: The objective of this lab is to implement the PIPO Shift Left/Load register designed in the last DP as well as a modulo-16 up counter.

Module # 6

DP: This DP is a culmination of most of the topics covered in the course. It involves the design and verification of a vending machine controller. The vending machine controller is a Finite State Machine (sequential circuit) that controls the operation of a vending machine. This includes the control of coins inserted, merchandize released, and possibly change returned.

LP: The objective of this LP is to implement the vending machine controller on the DE2 board and verify its functionality.

Course Assessment

At the end of the course, students were asked to complete a course evaluation survey to provide feedback on their learning experience and make suggestions for improvements. Only a subset of the questions asked in the survey are included in this paper. The included questions (shown in Table I) are selected because of their relevance to the content of this work. For each of these questions, students were asked to select a number out of five possibilities in a standard Likert scale as follows:

1. Strongly disagree (or poor)
2. Disagree (or below average)
3. Neutral (or average)
4. Agree (or good)
5. Strongly agree (or excellent)

The assessment results include two groups of students (section A and section B) taking the same course at different times in the same semester. The number of participants in the survey was 22 students from section A and 26 students from section B for a total of 48 students. Table I shows the results related to the selected questions. The results show that students were pleased with the class and the quality of their learning. Students also expressed an overwhelming satisfaction with the course organization and the order of occurrence of different subjects. Moreover, it was important for the class designers to get the positive reassurance that the learning modules covered all the subjects listed in the course educational objectives.

On a closely related issue, students usually believe that active learning techniques make the teacher's job easier while creating a larger workload on the student side. One question in the student evaluation survey helped assess this particular issue by asking: How would you rate the workload of this class? The student responses were collected using the same standard Likert

scale with 5 indicating "too much work" and 1 indicating "too little work." The responses are shown in Table II. The results show that only three out of 48 students thought that the class required too much work while more than half of the students (25/48) thought that the class required just the right amount of work.

Table I. Results of the student evaluations of the course

	Section	5	4	3	2	1	Avg.
Overall, I would rate this course	A	11	10	1	0	0	
	B	16	9	1	0	0	
	Total	27	19	2	0	0	4.52
The course is well organized and logically arranged	A	11	9	2	0	0	
	B	19	7	0	0	0	
	Total	30	16	2	0	0	4.58
The topics covered matched the learning objectives	A	11	9	1	1	0	
	B	21	4	1	0	0	
	Total	32	13	2	1	0	4.58
Overall Quality of the course	A	11	9	2	0	0	
	B	18	7	1	0	0	
	Total	29	16	3	0	0	4.54

Table II. Student's perception of the course workload

	Section	TMW				TLW
How would you rate the workload of this class?	A	2	8	11	1	0
	B	1	9	14	2	0
	Total	3	17	25	3	0

On a related note, different questions in tests, quizzes, and homework assignments are used to assess the students' achievement of the course educational objectives. Only the students who finished the class and sat for the final examination are included in this assessment. For each of the course educational objectives, following the assessment system in [6], the performance of each student is classified as Excellent (E), Adequate (A), Minimal (M), or Unacceptable (U). The assessment results are shown in Table III. The numbers show that course educational objectives were achieved with a great rate of success. In only one of the seven objectives, 2/47 students performed at an unacceptable level while in another objective, 1/47 student had an unacceptable performance.

Reflections and Conclusion

This paper described our experience in developing and teaching an active-learning-based digital logic design course for undergraduate electrical and computer engineering students. The Think, Pair, Share learning method was adopted because of the way it enables students to be in charge of their own learning experience without overloading their already busy schedules. The paper described the class objectives as well as the design projects and laboratory projects that were developed to achieve those objectives. The paper also presented the student feedback received at the end of the course.

From a course designer's perspective, the main concern is to design the right modules to effectively cover all the learning objectives within the time restrictions. From an instructor's perspective, the main concern is the way students received the new learning methods. Early in the semester, some students were not shy to express their dissatisfaction and express some resistance. The main complaint was the excessive workload associated with having to submit too much coursework. However, at the end of the course, the feedback received from the student evaluation surveys was overwhelmingly positive and approving of all course aspects.
Due to both the positive student feedback and the great success in achieving the course educational objectives, the plan is to continue teaching the class in the suggested format for the foreseeable future.

Table III. Student's performance on the course educational objectives

Course Educational Objective	E	A	M	U
Convert values between decimal, binary, hexadecimal, and octal number systems	21	16	10	0
Perform arithmetic operations on binary, hexadecimal, and octal number systems	19	17	11	0
Design minimized combinational logic circuits using Karnaugh Maps or Espresso/Minilog given natural language descriptions or truth table representations	17	24	6	0
Implement combinational circuits using simple gates, complex gates, steering logic, programmable logic, or universal gates	21	20	4	2
Given a set of input waveforms, sketch the output waveforms for various memory types (SR, JK, D, and T), and element classifications (latch, gated latch, and edge-triggered flip flop)	23	22	2	0
Create a minimized finite state machine given natural language descriptions or state diagram representations	11	19	16	1
Implement (create, verify, and debug) digital designs using integrated circuits (both standard off-the-shelf parts and PLDs)	24	22	1	0

References

[1] Laws, P., Sokoloff, D., & Thornton, R., "Promoting active learning using the results of physics education research." *UniServe Science News*, *13*, 14-19. 1999.
[2] Richards, L., Gorman, M., Scherer, W., & Landel, R., "Promoting active learning with cases and instructional modules." *Journal of engineering education*, *84*(4), 375-381. 1995.
[3] Kaddoura, M., "Think pair share: a teaching learning strategy to enhance students' critical thinking." *Educational Research Quarterly*, *36*(4), 3. 2013.
[4] http://www.orcad.com/products/orcad-ee-pspice-designer/overview
[5] http://www.altera.com/education/univ/materials/boards/de2/unv-de2-board.html
[6] Estell J., Yoder J-D., Morrison B., and Mak F., "Improving upon best practices: FCAR 2.0." *Proceedings of the American Society of Engineering Education annual conference*, 2012.