BYOE: Teaching PLC Programming Through Digital Logic Knowledge

Dr. Maria Javaid, Indiana State University

Dr. Maria Javaid joined Indiana State University in August 2019 as Assistant Professor. Before coming to ISU she was Assistant Professor at Jacksonville University. She received her PhD in Electrical and Computer Engineering from University of Illinois at Chicago.

William Clyburn, Indiana State University

Faculty in the Dept. of Electronics & Computer Technology; Bailey College of Engineering & Technology; at Indiana State University since August 2001. Rank: Full Professor Teaching area: Automation & Controls Engineering Technology.

31 years total teaching experience in Higher Education with emphasis in electronics, digital logic, programmable logic controller operation, and robotics.

15 years industrial experience as industrial electrician, automation tech support.

BYOE: Teaching PLC Programming Through Digital Logic Knowledge

Abstract

In this BYOE (Bring Your Own Experiment) paper, the authors present laboratory experiments that teach students Programmable Logic Controller (PLC) programming in Ladder Logic by first solving real-world automation problems at the logical level using digital logic concepts such as logic gates and Boolean algebra. Once students successfully solve the automation problems using digital logic, they are taught how logic gate circuits or Boolean equations can be translated into PLC programs in Ladder Logic. Students then develop Ladder Logic code, download it to the provided PLC, and test the functionality of the code.

This approach of solving real-world problems at the logical level using the fundamentals of digital logic can also be adopted in digital logic courses.

Keywords

Digital Logic, PLC Programming, Ladder Logic, Boolean Algebra, Logic Gates.

Background

Learning through laboratory experiments is a critical component of undergraduate STEM (Science, Technology, Engineering, and Mathematics) education [1],[2], as it promotes active learning and supports the development of practical problem-solving skills. Digital Logic is a foundational course within engineering and technology curricula, particularly in disciplines related to computer hardware and software, including Electrical Engineering Technology, Computer Engineering, and Computer Science. The course introduces students to fundamental digital design concepts, including logic gates and Boolean algebra [3], with the objective of enabling students to analyze and implement combinational and sequential logic systems.

In addition, some programs offer Programmable Logic Controllers (PLC)-based courses that emphasize industrial control applications through PLC programming in Ladder Logic. Laboratory activities in PLC programming typically require students to design, implement, and validate control solutions for real-world automation problems, with testing conducted on either physical PLC hardware or simulation platforms. These activities support ABET Student Outcome 1 by requiring students to identify, formulate, and solve complex engineering problems using principles of mathematics and engineering.

The introduction of a Digital Logic project into a first-year honors engineering course serving students from multiple engineering majors, including those not directly related to computing disciplines, was reported to be both educational and engaging [4], even though the presented project, which involved the development of a basic Arithmetic Logic Unit (ALU), is primarily focused on computer hardware. The importance of solving real-world problems through digital logic circuits has also been emphasized in the literature. Reference [5] presents three hardwired circuit projects designed for sophomore students majoring in Electrical Engineering and Computer Science; two of these projects incorporate sequential logic concepts. Logical reasoning, problem-solving, and critical thinking are important skills for STEM majors [3], [6], and instructional strategies that integrate applied digital logic design can help cultivate these competencies.

Building upon prior work, this paper proposes and evaluates a structured instructional approach that bridges foundational digital logic concepts and industrial automation applications. Specifically, the authors describe an approach in which simple real-world automation problems are first solved at the logical level using concepts of logic gates and Boolean algebra, and subsequently translated into Ladder Logic programs for PLC implementation. The primary contribution of this work is the development and classroom implementation of laboratory modules that explicitly connect Boolean logic design to PLC programming, thereby strengthening conceptual understanding while enhancing students' ability to apply theoretical knowledge to practical automation problems. The paper further discusses the pedagogical rationale, implementation details, and observed student learning benefits of this approach.

Introduction

This paper presents laboratory exercises in which several simple real-world automation problems are solved (1) first at the logical level, where the solution is obtained as a logic gate circuit and Boolean equation, and (2) subsequently converted into a Ladder Logic program. Students download the Ladder Logic program to the PLC provided in the lab and test the functionality of the code using the PLC trainer.

These laboratory exercises are the first two PLC experiments covered as a part of the Introduction to Robotics and Automation course, which is a sophomore-level course. This course enrolls students from various majors such as Mechanical Engineering Technology, Robotics and Automation Engineering Technology, and Computer Engineering Technology. Due to the diverse backgrounds of the students, the prerequisite for the course is only a basic circuits course. Since the PLC hardware is similar to computer hardware, the course covers the basics of digital logic such as numbering systems, logic gates, and basic Boolean algebra. The course introduces students to PLC programming in Ladder Logic. Students develop programs using RSLogix 500 software and download the code to a MicroLogix PLC. The code is tested using the PLC trainers provided in the laboratory, which include various buttons, switches, and lights to simulate real-world inputs and outputs. Figure 1 shows a PLC trainer available to students.

The laboratory problems presented in this paper are assigned to students after they have learned about PLC architecture and the basics of digital logic. After completing these labs, students progress to more advanced laboratory exercises, where they learn additional PLC features, such as timers and counters, and solve problems related to these features. However, in these advanced exercises, students are not required to submit the logical solutions.

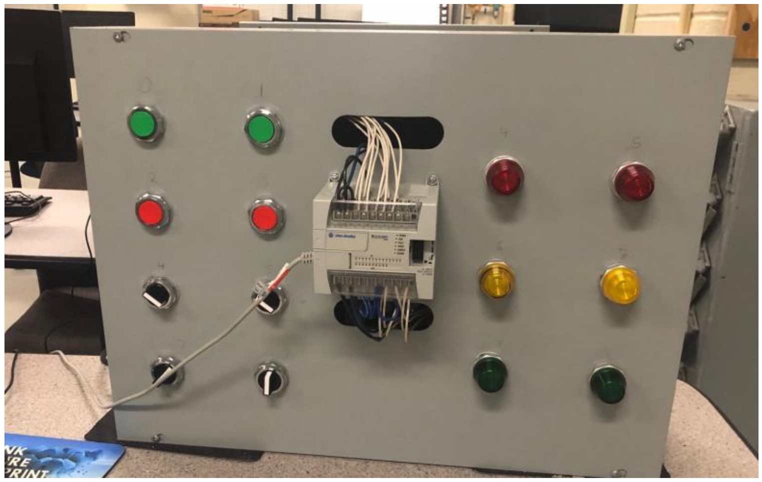

Figure 1: MicroLogix PLC Trainer

PLC Laboratory Experiment No. 1: Digital Logic Application

In this laboratory session, students work in groups of four to logically solve five real-world automation problems on paper. The logical solutions to these problems are obtained in the form of logic gate circuits and Boolean expressions. To help students understand the problem-solving process, the instructor solves the first problem in class and engages students by asking guiding questions. Students then work in their groups to solve the remaining problems through discussion and collaboration. The instructor and the Teaching Assistant (TA) provide guidance throughout the process. Students are required to present the solution to each problem to the instructor or TA before proceeding to the next problem. If a proposed solution is incorrect, the instructor or TA guides students toward the correct solution by posing thought-provoking questions rather than directly providing the answer.

Three of the five laboratory problems are presented here, along with their solutions.

1. STOP-START CONTROL:

Manual control of a conveyor belt is accomplished by the use of a standard STOP-START pushbutton station at one end of the conveyor. When the start button is pressed the conveyor motor will run until the stop button is pressed. Use output feedback by means of a "seal-in" contact.

Design a Boolean Expression and Logic Gate Circuit to solve this problem. Use the following bit symbols to represent inputs and outputs.

Inputs:

Start Button: Strt
Stop Button: Stp
Conveyor Running: Mtr

Output:

Conveyor Switch: Mtr

Note: Motor is used both as input and output.

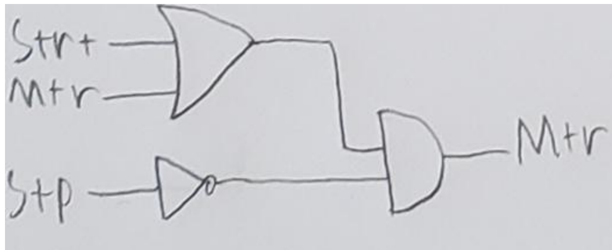

Figure 2: Logic Gate Circuit for Problem No 1

$$Mtr = (Strt + Mtr) \cdot \overline{Stp}$$

Figure 3: Boolean Expression for Problem No 1

2. PARALLEL STOP-START CONTROL:

The conveyor operation described above is modified to use two STOP-START pushbutton stations using the seal-in type of operation. Either stop or start pushbutton can operate the conveyor.
Design a Boolean Expression and Logic Gate Circuit to solve this problem. Use the following bit symbols to represent inputs and outputs.

Inputs:

Start Button No. 1: Strt1
Start Button No. 2: Strt2
Stop Button No. 1: Stp1
Stop Button No. 2: Stp2
Conveyor Running: Mtr

Output:

Conveyor Switch: Mtr

Note: Motor is used both as input and output.

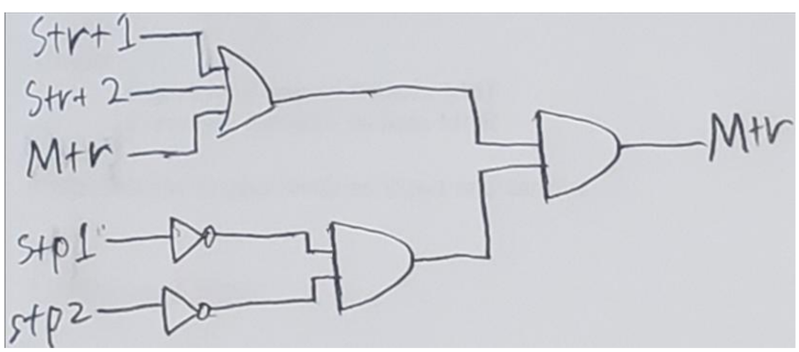

Figure 4: Logic Gate Circuit for Problem No 2

$$Mtr = (Str1 + Str2 + Mtr)\cdot(\overline{Stp1}\cdot\overline{Stp2})$$

Figure 5:Boolean Expression for Problem No 2

3. REVERSING MOTOR STARTER

The conveyor operation is modified by the addition of a reversing motor starter with two control coils. These reverse the phases and make the conveyor motor turn in the opposite direction. This gives us the ability to have the conveyor move forward when a forward start button is pressed, and seal-in, and to move backwards when a reverse start button is pressed, and seal-in. Pressing a stop button stops the conveyor regardless of which direction it is moving.

For safety reasons the forward and reverse operation must be "interlocked" so that the phases are not shorted together. When the conveyor is running forward pressing the reverse button has no effect until a stop button is pressed first. This also applies to when the conveyor is running in reverse.

Design a Boolean Expression and Logic Gate Circuit to solve this problem. Use the following bit symbols to represent inputs and outputs.

Inputs:

- Forward Button: Fwd
- Reverse Button: Rvrs
- Stop Button: Stp
- Conveyor Forward Running: MtrF
- Conveyor Reverse Running: MtrR

Output:

- Conveyor Forward Switch: MtrF
- Conveyor Reverse Switch: MtrR

Note: Motor is used both as input and output.

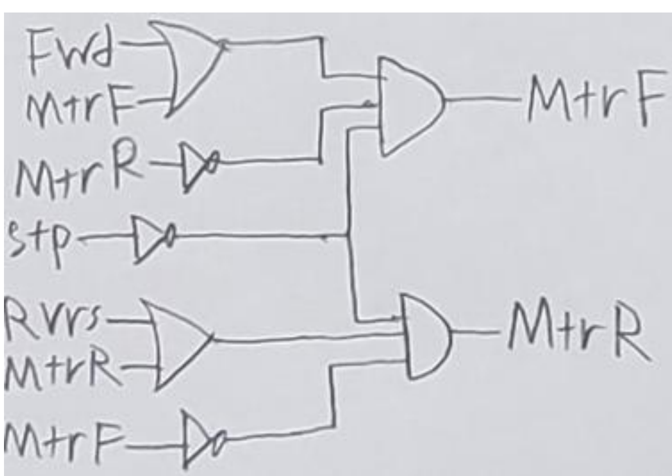

Figure 6: Logic Gate Circuit for Problem No 3

$$MtrF = (Frwd + MtrF) \cdot \overline{MtrR} \cdot \overline{Stp}$$

$$MtrR = (Rvrs + MtrR) \cdot \overline{MtrF} \cdot \overline{Stp}$$

Figure 7: Boolean Expression for Problem No 3

PLC Laboratory Experiment No. 2: PLC Programming

After Lab No 1 students are taught basics of PLC programming in theory class. They then perform Lab No 2 which involves solving real-world automation problems through a PLC program coded in ladder logic. The problems are same as covered in Lab No 1 so students have the logical solution available from previous lab and in this lab they convert those solutions in ladder logic program. Students code in RSLogix 500 software and download the code on MicroLogix PLC. The code is tested using the PLC trainers provided in the laboratory with various buttons, switches, and lights to simulate real-world inputs and outputs. In this laboratory session, students work in groups of two students. To help students learn how to code in RSLogix 500, verify the code for syntax errors, download and run the code on MicroLogix PLC and test various inputs and outputs scenario through the PLC trainer, instructor demonstrate by solving the first problem. Students then start working in their groups to solve other problems in the same manner through discussion with group members. Instructor and Teaching Assistant (TA) help students in the process. Students are required to show the solution to each problem to Instructor or TA before moving to the next problem. After each student is checked and approved by instructor or TA, students take screenshot of it and include it in their lab report.

Three of the five laboratory problems are mentioned here along with the solutions.

<u>1. STOP-START CONTROL:</u>

Manual control of a conveyor belt is accomplished by the use of a standard STOP-START pushbutton station at one end of the conveyor. When the start button at I:0/0 is pressed the conveyor motor will run until the stop button at I:0/2 is pressed. Use the lamp at O:0/4. Design a PLC program to operate the conveyor by each means. Write your program in Ladder Logic and run it on the PLC.
Note that the STOP pushbutton (I:0/2) is a normally closed pushbutton, which means that when the button is not pressed, the input I:0/2 is at logic level 1, and when the button is pressed, I:0/2 changes to logic level 0.

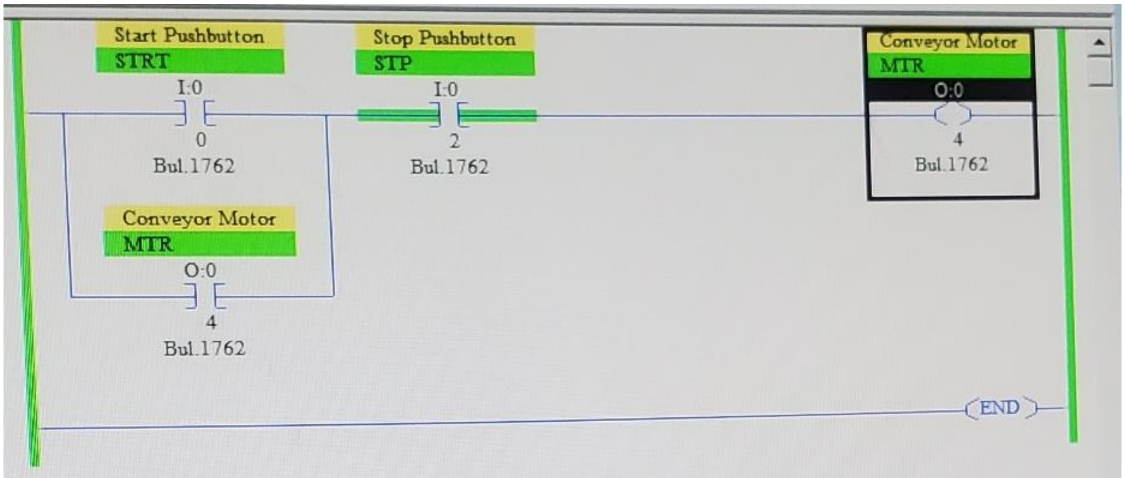

Figure 8: PLC Ladder Logic for Problem No 1

2. PARALLEL STOP-START CONTROL:

The conveyor operation described above is modified to use two STOP-START pushbutton stations using the seal-in type of operation. Either stop or start pushbutton can operate the conveyor. Design a PLC program to operate the conveyor.

Use I:0/0 for Strt1, I:0/1 for Strt2, I:0/2 for Stp1, I:0/3 for Stp2 and O:0/4 for Mtr.

Note that the STOP pushbuttons, Stp1 (I:0/2) and Stp2 (I:0/3), are normally closed pushbuttons, which means that when a pushbutton is not pressed, the corresponding input is at logic level 1, and when the pushbutton is pressed, the input changes to logic level 0.

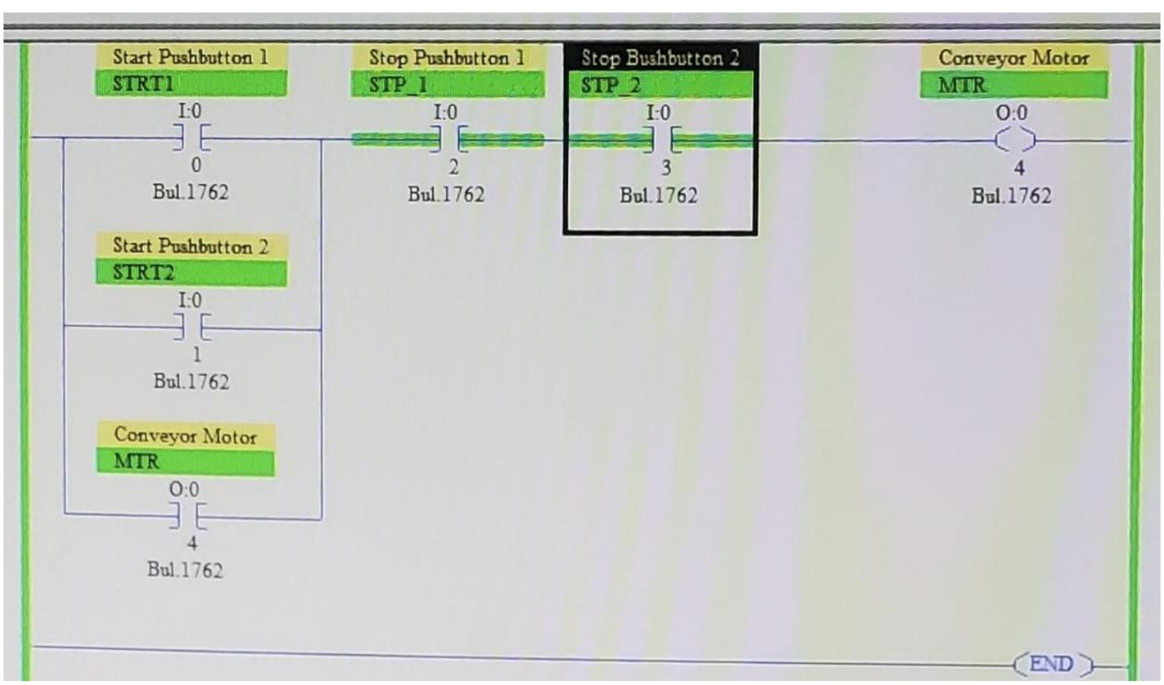

Figure 9: PLC Ladder Logic for Problem No 2

3. REVERSING MOTOR STARTER

The conveyor operation is modified by the addition of a reversing motor starter with two control coils. These reverse the phases and make the conveyor motor turn in the opposite direction. This gives us the ability to have the conveyor move forward (O:0/4) when a forward start button (I:0/0) is pressed, and seal-in, and move backwards (O:0/5) when a reverse start button (I:0/1) is pressed, and seal-in. Pressing a stop button (I:0/2) stops the conveyor regardless of which direction it is moving. Use one of the red lights to show the motor rotating "forward" (O:0/4) and use the second red light to show the motor rotating in reverse (O:0/5). Use the 2 green pushbuttons as your forward (I:0/0) and reverse buttons (I:0/1), use a red Pushbuttons as your stop.

Note that the STOP pushbutton (I:0/2) is a normally closed pushbutton, which means that when the button is not pressed, the input I:0/2 is at logic level 1, and when the button is pressed, I:0/2 changes to logic level 0.

For safety reasons the forward and reverse operation must be "interlocked" so that the phases are not shorted together. When the conveyor is running forward pressing the reverse button has no effect until a stop button is pressed first. This also applies to when the conveyor is running in reverse. Design a PLC program to operate the conveyor.

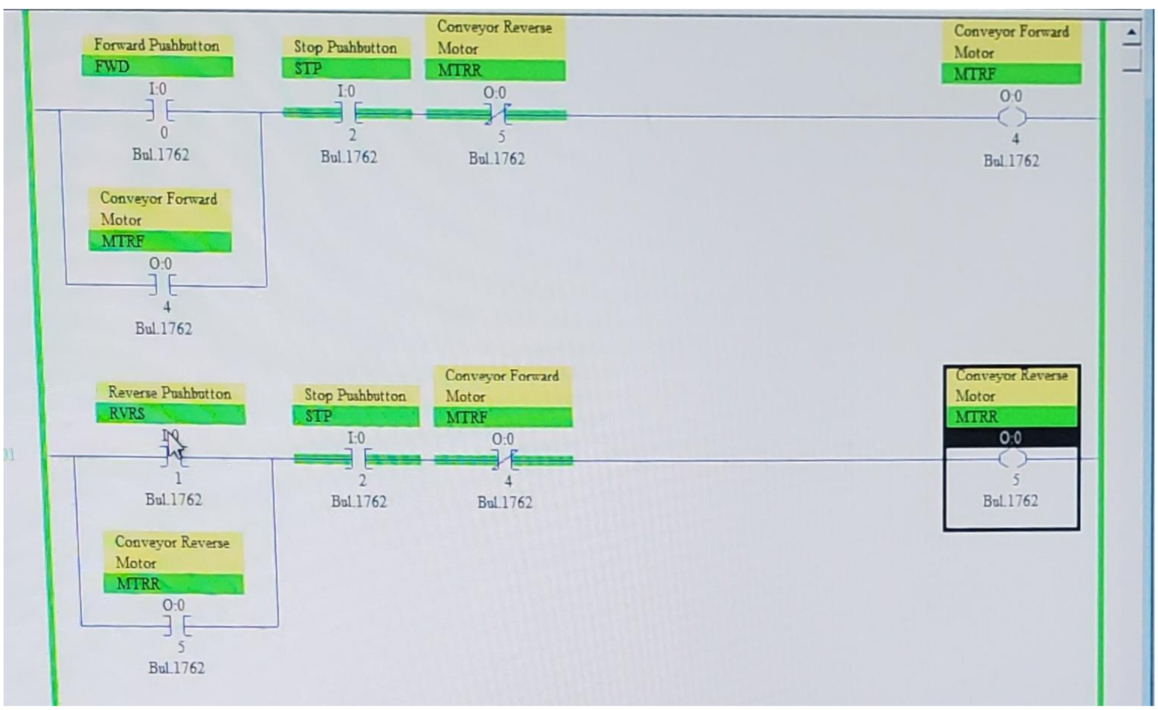

Figure 10: PLC Ladder Logic for Problem No 3

Observations

The authors have found this approach to teaching PLC programming, by explicitly connecting it to students' prior knowledge of digital logic, to be effective. The authors observe that students not only successfully learn PLC programming using this approach, but also find the laboratory experiments engaging, as indicated by informal in-class feedback and end-of-semester anonymous course evaluations.

Indiana State University allows instructors to include custom questions in the end-of-semester anonymous evaluations in addition to the standard questions. The lead author of this paper included two laboratory-related statements in the evaluation instrument. The statements were as follows:

1. Laboratory experiments help students understand theoretical concepts.

2. Laboratory experiments are interesting.

Students responded to these statements using a five-level Likert scale, where 1 indicates strongly disagree and 5 indicates strongly agree. Figure 11 presents student ratings for these two statements in the Introduction to Robotics and Automation course over a three-year period.

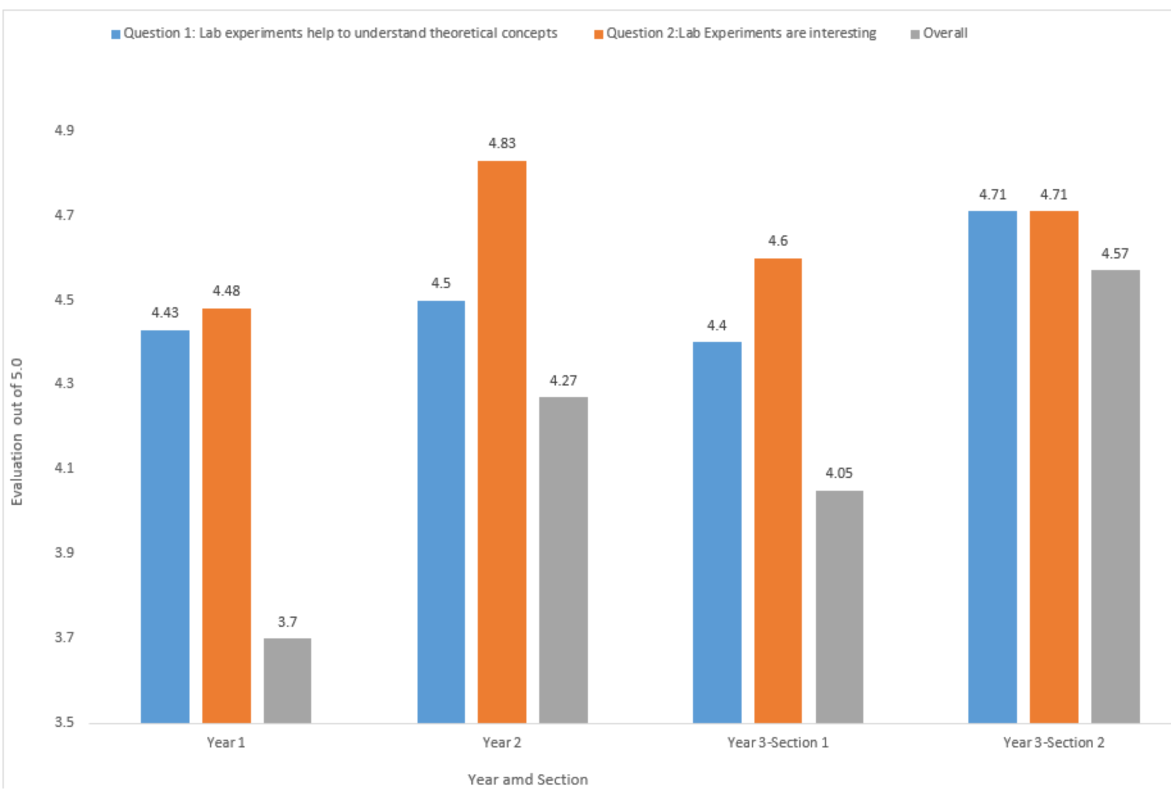

Figure 11: Laboratory Evaluations based on Question No. 1 and Question No. 2 vs Overall Course Evaluation

The general end-of-semester evaluation instrument included the following question:
"What were the best things about this course?"

The percentage of responses mentioning laboratory experiments as the best part of the course is reported in Table 2 for a five-year period. The overall average of these percentages is 68%.

Table 1: Percentage of Answers Mentioning Laboratory as Best Part of the Course

Year	Percentage of General Comments Mentioning Laboratory as Best Part of Course
Year 1	68.75 %
Year 2	90 %
Year 3 – Section 1	33.33 %
Year 3 – Section 2	66.67 %
Year 4	66.67 %
Year 5	83.3 %

References

[1] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *Journal of engineering Education*, vol. 94, no. 1, pp. 121–130, 2005.

[2] R. V. Krivickas and J. Krivickas, "Laboratory instruction in engineering education," *Global J. Eng. Educ*, vol. 11, no. 2, pp. 191–196, 2007.

[3] P. Mayer and R. Baraniuk, "The importance of teaching logic to computer scientists and electrical engineers," *ACM Transactions on Computing Education*, vol. 25, no. 2, pp. 1–12, 2025.

[4] D. G. de la Rosa-Pohl, S. A. Long, and C. Goodwin, "Introduction of a digital logic project in a first-year honors engineering course," in *2013 ASEE Annual Conference & Exposition*, 2013.

[5] J. P. Hoffbeck, "Using practical examples in teaching digital logic design," in *2014 ASEE Annual Conference & Exposition*, 2014.

[6] N. Attridge, A. Aberdein, and M. Inglis, "Does studying logic improve logical reasoning?" 2016.