Logic Circuits Unveiled: Bridging the Gap between Discrete Mathematics and Computer Science Education

Charlie Lake
c.lake@alumni.ubc.ca
UBC
Vancouver, Canada

Karina Mochetti
mochetti@cs.ubc.ca
UBC
Vancouver, Canada

ABSTRACT

Discrete mathematics is crucial for a Computer Science major, yet often challenging. We describe how to integrate Digital Circuit labs with a Discrete Math course, employing active learning methods. Our approach, featuring a logic simulator application and a digital circuit lab kit, enhances comprehension of relevance of content, and student engagement. We describe the design of our course and discuss the latest changes made to improve lab engagement and better connect them to the theory seen in lecture.

KEYWORDS

Discrete Mathematics, Logic Circuits, Computing Education

ACM Reference Format:
Charlie Lake and Karina Mochetti. 2024. Logic Circuits Unveiled: Bridging the Gap between Discrete Mathematics and Computer Science Education. In *The 26th Western Canadian Conference on Computing Education (WCCCE '24), May 2–3, 2024, Kelowna, BC, Canada.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3660650.3660653

1 INTRODUCTION

Discrete mathematics is the backbone of several computer science concepts, giving future professionals problem-solving capabilities and better knowledge of how and why their code executes as it does. Although important, it is considered a difficult topic by students, not only because of their different backgrounds in mathematical thinking but also because its application in the field is not as obviously clear [12]. This can be extra challenging for first-year students, who are equipped with algebraic math experience but may have a difficult time abstracting and visualizing the application of such topics.

Although the base of discrete mathematics is an old and steady topic, education is constantly evolving. New methods and techniques of teaching and learning are still being developed and researchers seek ways to improve this process. Pedagogical strategies should facilitate students' learning process while motivating and developing their critical vision. Therefore, the goal of each class should be well-defined, and the students should be the protagonists of their own learning, after all, as Paulo Freire once said "Banking education treats students as objects of assistance; problem-posing education makes them critical thinkers." [9].

Traditional learning is usually composed of expository lectures focusing on content rather than the student. In this method, students remain as passive actors in the learning process, seen as mere information recipients. They memorize facts and use them in evaluation exams, discarding them later. This process is superficial and has little contribution to the formation of critically and socially responsible citizens [18]. However, active learning methods [19, 20, 24] agree with Freire's theory, in which the student should be active in their learning. Recent studies indicate better results in content retention and problem-solving in classes with active learning techniques [8, 11].

Project-based learning [5, 14–16] is a popular active learning approach, commonly deployed in CS as a Hackathon, that involves coding and user-experience design. However, in our experience for more theoretical subjects, such as discrete math, it can be more challenging to use these kinds of methods due to the apparent disconnect between theory and application. This can result in computer science students being less engaged, therefore doubting the real application of that knowledge. In this work, we describe how we use digital logic projects along with a discrete math course to show students the importance of theoretical knowledge in Computer Science.

1.1 Main Goals

The main goal of this work is first to describe how we added a digital circuit lab along with a discrete mathematics course to help engage students and connect more math topics to Computer Science. we give details of the course so other instructors can follow or be inspired by our course design. We use the Project-based Learning method in a discrete mathematics course. The second goal involves the analysis of changes made to this course in order to improve the lab activities, fixing the gaps students felt between lectures and lab activities stated by their evaluations. The changes made started in 2022 and were considered stable in 2023.

2 RELATED WORK

Teaching discrete mathematics in computer science curricula has not been researched as much as teaching programming. However, one of the main topics in previous research on teaching discrete math is how the material is linked with other courses, with the main two topics being either programming or data structures. Furthermore, several teaching methods are described in the literature, such as visual tools, collaborative homework, and problem-directed approaches [23].

WCCCE '24, May 2–3, 2024, Kelowna, BC, Canada

ACM ISBN 979-8-4007-0997-5/24/05
https://doi.org/10.1145/3660650.3660653

Another topic of research has been the grading scheme, especially since these courses tend to have a large number of students enrolled, due to the rising numbers of students interested in CS majors. Specs Grading and Mastery Learning have been discussed as an alternative to traditional grading, aiming to lower levels of stress and anxiety from students and less staff time used for grading, being able to be applied to other more valuable parts of the course, such as Tutorials, Office Hours and other activities with closer contact with students and their learning process [25]. Other works in teaching discrete mathematics to CS students involved comparing an in-person and online offering of the class to assess learning outcomes, with no main differences found [12] and applying digital courseware [27]

Our course was designed over 20 years ago [2], and some work has been done focusing on the gap between lectures and labs, however, those changes were not updated and some were lost over the years. In 2010, the style of labs changed to a more Active Learning approach, discussion periods were added to labs, and lectures were changed to routinely and explicitly discuss connections to both the labs' big picture story and a corresponding theoretical story traced mainly in lecture [21].

A couple of years later, the lab structure was revised. The grading scheme was more clearly presented to students, clarifying the learning goal of labs, and eliminating obstacles that did not contribute to students learning by reducing text, and illustrating labs with more figures and animations [22].

Both works made clear how the gap between labs and lectures is a constant issue with the course, and although some solutions were proposed neither work puts their main efforts and goals into solving this. Furthermore, none of the previous work gives enough information for other instructors and faculties to be able to reproduce our course design.

Therefore, our work focuses on giving a clear description of our course and detailing the actions taken to fix the main issues found in years of students' evaluation, especially the gap students feel between labs and lectures.

3 BACKGROUND

CPSC 121 is a core course for the Computer Science major at UBC, focusing on Discrete Mathematics topics. The course has all its activities done in person and usually has around 600 students per term distributed across three sections. It is a first-year course, and mandatory for students to enter the Computer Science major. The course is usually considered hard by students. Furthermore, since it takes our university's CS1 course as a co-requisite, a lot of students have difficulty seeing the application of the math topics seen in the course and Computer Science.

3.1 Course Schedule

The course runs twice a week, with 1.5-hour lectures for 13 weeks in the Fall and Winter terms. Besides that, there are 2-hour lab sections once a week, capped at 26 students per lab run by three TAs, and 1-hour tutorial sections once a week, run by one TA.

There are 11 modules and each one should take three hours, or two lectures to be completed. All students must go through 9 different labs during the term, so the lab schedule has to be designed to consider which weeks have holidays (usually sections that fall on a holiday fall a week behind all other sections), and midterms (where there are typically no labs). Each module should also be covered by one tutorial, usually happening the week after the module was covered in lecture.

3.2 Reading

Before each module, students are required to do a reading from the textbook [7] and answer a quiz with 10 questions. The first nine questions are fairly easy multiple-choice questions, the last one is always an open-ended question that is later discussed in the lecture. The idea is to make students think beforehand about the problem that will be covered in class.

3.3 Lectures

CPSC 121 has the following 11 modules:

- **Module 01. Propositional Logic:** Translation between circuits, propositions, and truth tables.
- **Module 02. Logical Equivalences:** Proof of logical equivalence using equivalence laws.
- **Module 03. Number Representation:** Integer representation in binary using two's complement, and a superficial view of floating point numbers and ASCII table.
- **Module 04. Propositional Logic Proofs:** Proof of argument validity using inference rules.
- **Module 05. Sets and Predicate Logic:** Basic set operations and the use of the universal and existential quantifiers.
- **Module 06. Regex and DFA:** Basic Regex operations, DFA and NFA differences, superficial view of Turing Machine.
- **Module 07. Direct Proofs:** Formal direct proofs based on universal/existential quantifiers.
- **Module 08. Indirect Proofs:** Formal indirect proofs based on contrapositive and contradiction methods.
- **Module 09. Weak Induction:** Formal simple induction proofs using weak induction, and why induction works.
- **Module 10. Strong Induction:** More formal induction proofs now use strong induction and scenarios using more than one base case.
- **Module 11. A Working Computer:** Big O definition and proofs, the von Neumann architecture and relation between binaries and assembly instructions.

The active learning used during lectures are the commonly used Clickers questions and worksheets. Every module has a worksheet that students work through together in class and live clicker questions that are asked during lectures.

3.4 Labs

CPSC 121 has 9 labs during the term. Labs are taught by TAs and use a lab kit called Magic Box [2].

Each lab guides students through the design of solutions to specific problems, and for most of the activities, students implement their solutions on a simple hardware kit called The Magic Box, shown in Figure 1. The kit includes a circuit board that accepts 9-volt power from the included wall-plug adapter and provides the 5-volts required by the integrated circuit logic chips also included in the kit. It also contains twelve switches to provide inputs, and eight LEDs to display outputs.

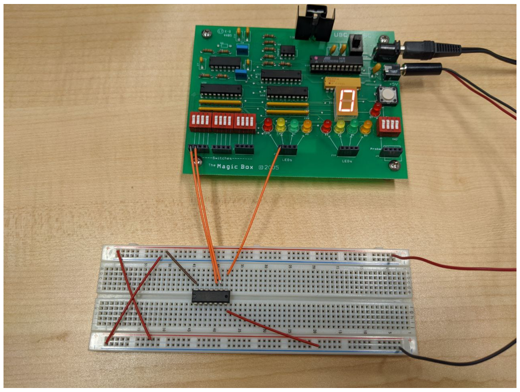

Figure 1: The Magic Box Lab Kit

Labs also consist of a pre-lab, usually three or four questions students have to answer individually before their lab section, and the lab questions which students should answer during their 2-hour sections.

Labs are designed based on Project-Based Learning since it involves students constructing hands-on solutions to computer and logic problems using a lab kit. Therefore, no lab content is taught in a traditional lecture format. The pre-lab and video aim to prepare students for the topics taught in the lab section, with the pre-lab assignment marked for correctness.

Students complete activities in the lab in groups of 2. These activities include physical wiring of circuits, designing digital logic circuits, short calculation questions, and always include a "Further Analysis" question that aims to have students think more abstractly about the concepts they explored that day. Finally, all labs end with a mandatory post-lab feedback survey to collect data and inform future improvements to the course.

Each of the 9 Labs has the following main goals:

- **Lab 01:** Icebreaker and how the Magic Box lab kit works.
- **Lab 02:** Debugging and creating truth table corresponding to a digital logic circuit.
- **Lab 03:** Multiplexers, instability and circuit efficiency.
- **Lab 04:** Adders, RAM, and ALU.
- **Lab 05:** Flip-flop and registers.
- **Lab 06:** Sequential circuits.
- **Lab 07:** Converting DFA to sequential circuits.
- **Lab 08:** Simulate running instructions in the computer.
- **Lab 09:** Putting all together to build a small computer.

Lab 9 has advanced topics that are part of future courses in the CS major so usually its content is not covered in the exams. This lab allows students to see how all the content fits together to build a working computer with the topics seen in lecture and previous labs.

3.5 Other Components

Besides regular lectures and labs, our course also has 11 Tutorial sections (one per module), which are 50-minute lectures when TAs go over questions with students. Our course also has a really active Piazza, where students can ask questions about the content and interact with TAs, colleagues, and instructors.

The grading scheme of the course includes the following parts:

- **14% - Assignments (5)**: Assignments have 5 or 6 questions, usually with a high level of difficulty, submitted by students in groups of three during the term.
- **14% - Labs (9)**: Labs grades are based on the submission of a pre-lab assignment and doing all the in-person tasks during the 2-hour lab section, as described in Section 3.4.
- **4% - Pre-class Quizzes (11)**: Students are required to read some sections of the textbook and answer a quiz with 9 simple multiple-choice questions and one open-ended question, as described in Section 3.2.
- **28% - Midterm (2)**: The course has two 1.5-hour Midterms, converting all content, including lectures and labs.
- **34% - Final Examination (1)**: The final examination follows our university rule and is a 2.5-hour exam that covers all course content, including all modules, labs, readings, and every single material covered during the term.
- **3% - Tutorials (11)**: Tutorials are 50-minute sections with TAs, in which students go over some questions based on the current module. TAs answer questions, show solutions to some questions, and engage students in how to apply the lecture knowledge to solve problems. Students are graded by participation, by just attending the section.
- **3% - Clicker questions (66)**: During lectures, multiple choice questions are asked of students using a classroom response system ("clicker") [17]. Students get marks for participation, not correctness and this is used in a contingent teaching method [1, 6]

4 IMPROVEMENTS

4.1 Enhancing Students Experience

The course was designed over 20 years ago [2], but the general feeling based on students' evaluation and previous work was that two courses were put together, with no clear connection between lectures and lab activities. Furthermore, students' evaluations and Piazza's questions show students struggle with lab content for different reasons. The first action we took was to modernize the tools used, which included Logisim, an application that has had its development suspended since 2014 [3]. A lot of students had a problem installing and using the tool on their own computers, making labs more complex than they were meant to be. After considering a few options, we chose Logisim Evolution [13], a similar but more modern application was the one picked. All course files were updated to work with the new application.

Based on the number of Piazza questions and lab questions grades on exams, students previously struggled with tested lab content due to the lack of support on lab content outside of the lab session, "Lab Notes" were developed, which were more traditional study materials. These notes described learning objectives for the lab, summarized the content covered in the lab (and walked students through TODOs that they didn't understand), and provided practice problems to show how this content might be tested on exams. Students therefore had these study notes and solutions of the labs as study material for exams, helping them to direct studying to more relevant concepts.

Changes were made to lab content to improve cohesiveness. To illustrate this, one example of a lab that was reworked significantly was Lab 4. The lab previously provided disjointed exercises involving multiplexers and analyzing a simple ALU. To improve it, we kept the ALU and replaced the less relevant tasks with others connected to Module 03 content about binary numbers and data representation on the computer. The new tasks involved binary adders, and analyzing a RAM module, to tie into computer architecture with the ALU. This also lets us address questions about number representation and lead into Lab 05 which focuses on memory components.

Overall, changes like these allowed us to create labs with more consistent themes (here, a lab about computation in circuits and computers) that were more cohesive with lectures and allowed labs to flow better from one to another.

Furthermore, we also developed 9 screencasts, which are videos of around 8 minutes in a traditional lecture format on the main topic for each lab. The purpose of these is to ease stress for the students who may not be used to Active Learning. One common comment we heard about labs was how students prefer if there was a more traditional lecture component, and although we understand the importance of using active learning, we understand how using purely Active Learning can be uncomfortable as a huge paradigm change for first-year students.

4.2 Connecting Labs and Lectures

The main addition to the course to better show the connection between lecture content and lab activities was a concept map. This map shows the relation between all modules' topics with all labs' topics. The concept map was introduced to students at the first lecture, and before each new module starts, a review of that concept is done, showing how the topics of that module are connected to other modules and labs. The concept map can be seen in Figure 2. Note that since the connections are not easily described in a small sentence, the linking between all topics is not on the map, but it is made clear by the instructor at the start of the lecture for each module. Furthermore, a color-code visual is used to express which types of skills students will practice during each module. This gives the course a purpose deeper than the content itself, focusing on skills students can apply not only in their professional lives but beyond.

Finally, to make the connection between lectures and labs explicit, most modules would show images of the Lab Kit and how some of the topics seen could be applied to the Magic Box. Module 03, for example, covers binary and decimal number conversion. In lecture, an image showing the LED outputs connected to the 7-segment display is shown to students, so they see that that binary numbers are used by the Magic Box. Not only that, but only numbers up to 9 are shown in lecture, and students are encouraged to find out during Lab sections what happens if the number is greater than 9, which in decimal would need two digits to be represented. Figure 3 shows a sample slide.

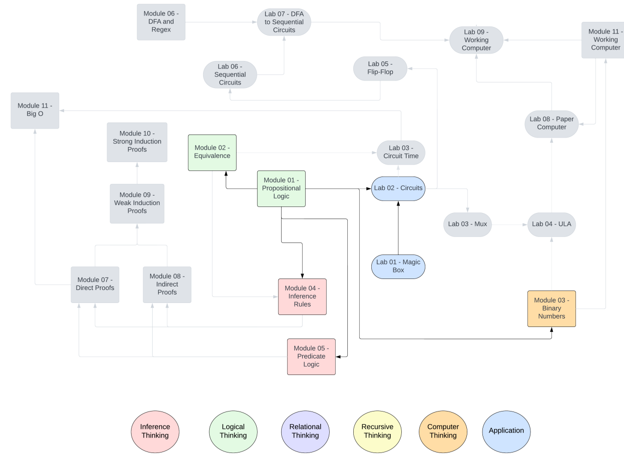

Figure 2: Concept Map using on CPSC 121.

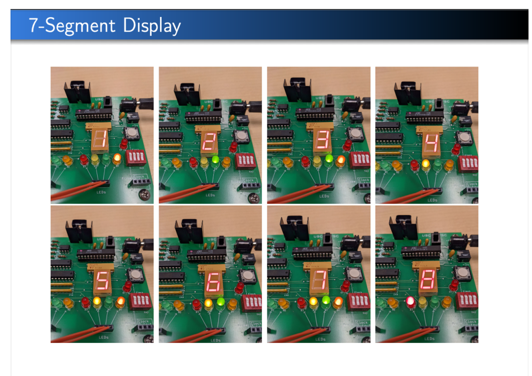

Figure 3: Sample slides of Module 03 showing images of the Magic Box in lecture.

5 RESULTS AND LESSONS LEARNED

The changes made started in 2022 and were finalized in 2023. Survey data was collected from students during this period to inform improvements to the course. A mandatory post-lab feedback survey is commonly provided to students at the end of every lab session (results of that survey are in Figures 6 and 8), and an additional end-of-term survey was also provided at the end of the first winter term (results of that survey are in Figures 4, and 5, 7), asking about the effectiveness and cohesion of various course components. These terms are named 2021W1, 2022W1, and 2023W1 respectively [1]. Data from before changes were implemented (2021W1) was compared to when changes were implemented (2022W1 and 2023W1).

Figures 4 and 5 show students' answers to different questions asked in a 5-point Likert Scale. Students rated lecture and lab contents as better integrated over time and also found it more helpful

[1]Winter Term 1 (W1) occurs from September to December, and Winter Term 2 (W2) occurs from January to April.

to their overall course learning. Furthermore, content within labs was rated as more cohesive after lab content was rearranged (see Figure 6) [2]

This implies that changes made to connect labs and lectures had a significant impact on how students viewed labs. Providing explicit connections in lecture between theory (shown in lectures) and the corresponding application (shown in labs) helped students understand the relevance of lab exercises. Furthermore, students viewed labs as more helpful to their learning of the course, implying that students more strongly associated lab content as a core part of the course material, rather than viewing lecture and lab content separately. Finally, students viewed the content within labs as more cohesive. The changes made to labs including rearranging of activities to better align with course modules and topics led to improved cohesiveness within each individual lab. A better understanding of the connection between lab and lecture likely influenced student perceptions of how each individual lab connected to the next, as they better understood the purpose of the lab component of the course as a whole. Overall, these methods of connecting labs and lectures worked very effectively in creating a more cohesive course.

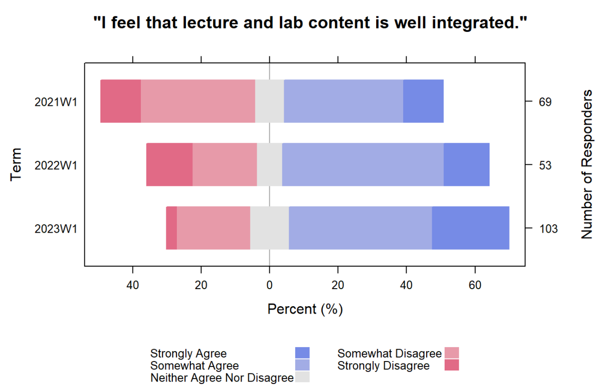

Figure 4: Student Responses to question "I feel that lecture and lab content is well integrated" in terms 2021W1 (n = 69), 2022W1 (n = 53), 2023W1 (n = 103)

Students, however, did not rate the lab section of the course overall as a more effective learning experience (see Figure 7), even though we believe the content was improved and rearranged within labs to be more cohesive and relevant to course topics. The fundamental learning objectives and Project-Based Learning approaches used within labs did not change. Even if labs were found to be more cohesive and taught better to students, the overall approach to labs was similar enough to 2021W1 to not change the effectiveness of the exercises. Individual labs, though, were still shown to be more effective learning exercises when made more cohesive with the rest of the course content. Figure 8 shows the effectiveness ratings for Lab 4. One possible reason for that is that the sense of effectiveness may be more tied to the enjoyment of the lab component as a whole. On the questions open to comments, it was common for students

[2]Note that data for this plot was taken from 2022W1, 2022W2, and 2023W1, as data was not collected for this question before these terms.

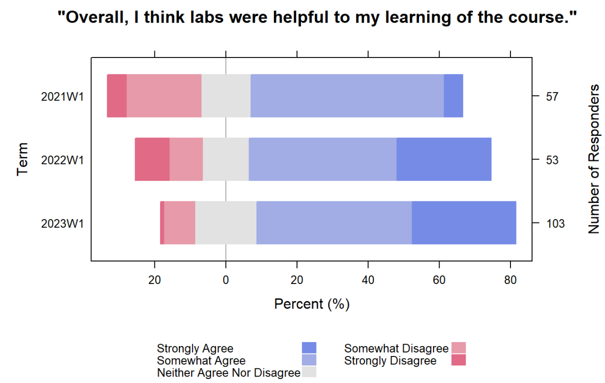

Figure 5: Student Responses to question "Overall, I think labs were helpful to my learning of the course" in terms 2021W1 (n = 57), 2022W1 (n = 53), 2023W1 (n = 103)

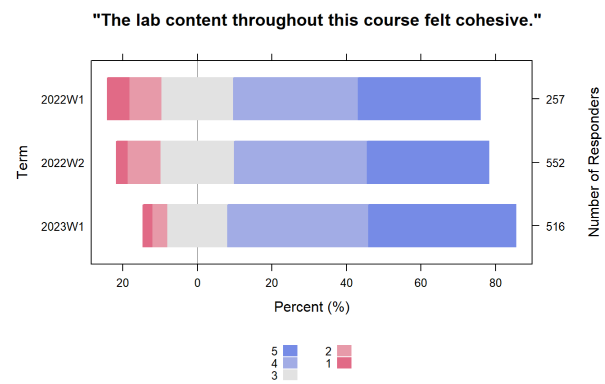

Figure 6: Student Responses to question "The lab content throughout this course felt cohesive" in terms 2022W1 (n= 257), 2022W2 (n = 552), 2023W1 (n = 516)

who rated this higher to refer to their lab TAs by name as contributing positively to their lab experience, for example, showing that other factors can influence the student's feeling of how effective a lab section is for them.

The same surveys had open-ended questions for students to comment on topics of the course.

Starting in the terms when the lab notes were introduced, students commented that the lab notes were incredibly beneficial to their studying, though often requested more practice problems provided in the lab notes (which typically contained 3-5 extra practice problems per lab session). While more practice problems would surely be beneficial to students, we expect this to be a common student request. One way of addressing this issue may be using a platform that allows auto-generation and auto-grading of questions such as PrairieLearn [26], to give students semi-unlimited practice material, which is not the focus of this work.

Although grade averages for the labs were high (often in the mid-90s), students often requested more time for labs so that they were more stressed, citing that they were more focused on completing the

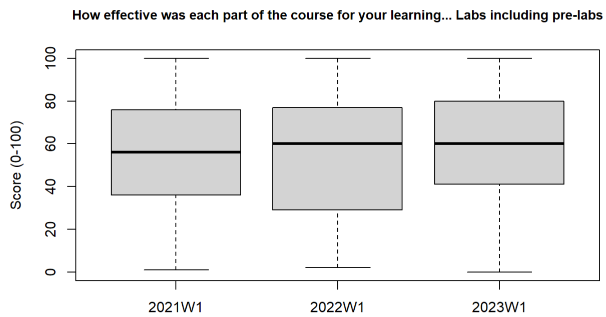

Figure 7: Student Responses to question "How effective was each part of the course for your learning... Labs including pre-labs" on a 0-100 scale in terms 2021W1 (n = 67), 2022W1 (n = 53), 2023W1 (n = 103)

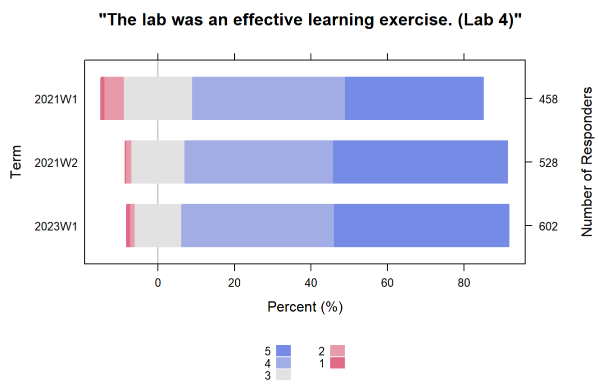

Figure 8: Student Responses to question "The lab was an effective learning exercise" for Lab 4 in terms 2021W1 (n = 458), 2021W2 (n = 528), and 2023W1 (n = 602)

lab activities over understanding. This is likely due to the grading structure of labs causing student mentality to focus on achieving a high grade over learning [4]. Although additional time may reduce student stress, it's expected that to fully focus students on learning, the grading scheme must be reviewed, not only for lab sections themselves but for the whole course.

Furthermore, students often commented on lab content being more difficult than lecture content. This is likely because less course time is dedicated to the content, therefore students spend proportionally less time studying and learning this content. There were fewer comments on this in recent terms (2023W1) after lab notes had been introduced.

One common suggestion made by students is to address lab content explicitly in a lecture session following the lab. As a result, students would be forced to recall the content and review it in a traditional lecture-style format. However, it's not necessarily true that this would be beneficial; given that lab, content is taught using Project-Based Learning, we believe that providing a lecture on the theoretical background behind the circuits would not help the majority of students who understand the theoretical background but are more concerned about problem-solving on exams. We believe this is common as first-year students are used to traditional lecture-style classes from high school.

Instead, we believe a more appropriate way to revisit lab content is through office hours, where one-on-one support can be provided for working through a problem. However, office hours are often underutilized by students [10], especially as our course is primarily taken by first-year students still adjusting to university teaching. A strategy we have used to moderate success is whenever students request additional support for labs, we unofficially designate one of the office hour sessions for a week as a "Lab Office Hour" and direct them to that session. This allows us to directly encourage students to seek out office hours, providing them with a dedicated space to revisit lab content.

Finally, students who rated labs as less effective exercises often commented on equipment issues, mentioned having to wait for TAs or stated issues with not meshing well with their partner. On the other hand, students who rated labs highly effective often mentioned their section's TAs by name and credited them with making it a great experience. This was consistent over time, not changing even in terms when cohesiveness was improved. Therefore, one key takeaway is that the effectiveness of these lab activities was highly connected to student comments on their enjoyment of the labs, directly influenced by equipment issues, TA engagement, and their partner. Given that effectiveness scores did not improve with cohesiveness, focusing on improving these other factors may allow students to engage more effectively in labs.

6 CONCLUSION AND FUTURE WORK

This paper describes how a Digital Circuit lab is being used at UBC to help students understand the applications in Computer Science of some of the Discrete Mathematical topics covered in CPSC 121. We have shown a detailed scheme of the course and the updates made to make the course more concise and connected.

Our future work includes finding a more modern and updated lab kit than the Magic Box used now. We believe that by changing the Magic Box we can solve some technical issues problems with chips not properly working that can be really frustrating for students. Not only that, but with a better, more complete lab kit, we may be able to connect labs with more theoretical modules such as direct, indirect, and induction proofs.

Additional future work is studying how to use auto-graded and randomized questions and a new grading scheme, so it is possible to better manage a course as large as 600 students course. In this way less time is spent grading, and better options can be offered to students (such as two attempts at exams).

REFERENCES

[1] Ian D Beatty, William J Gerace, William J Leonard, and Robert J Dufresne. 2006. Designing Effective Questions for Classroom Response System Teaching. *American Journal of Physics* 74, 1 (2006).
[2] Patrice Belleville. 2004. Integrating Introductory Discrete Mathematics and Logic Design. In *Western Canadian Conference on Computing Education (WCCCE)* (Kelowna, Canada). Association for Computing Machinery.
[3] Carl Burch. 2002. Logisim: A Graphical System for Logic Circuit Design and Simulation. *J. Educ. Resour. Comput.* 2, 1 (2002).
[4] Kelsey Chamberlin, Maï Yasué, and I-Chant A Chiang. 2023. The impact of grades on student motivation. *Active Learning in Higher Education* 24, 2 (2023).
[5] Heather Coffey. 2008. Project-based learning. *Learn NC* 1, 1 (2008).

[6] Stephen W Draper and Margaret I Brown. 2004. Increasing interactivity in lectures using an electronic voting system. *Journal of Computer Assisted Learning* 20, 2 (2004).

[7] Susanna Samuels Epp. 2019. *Discrete Mathematics with Applications* (fifth ed.). Cengage Learning.

[8] Scott Freeman, Sarah L Eddy, Miles McDonough, Michelle K Smith, Nnadozie Okoroafor, Hannah Jordt, and Mary Pat Wenderoth. 2014. Active learning increases student performance in science, engineering, and mathematics. *Proceedings of the national academy of sciences* 111, 23 (2014).

[9] Paulo Freire. 1970. *Pedagogy of the Oppressed* (thirtieth anniversary edition ed.). New York; London: Bloomsbury.

[10] Mario Guerrero and Alisa Beth Rod. 2013. Engaging in Office Hours: A Study of Student-Faculty Interaction and Academic Performance. *Journal of Political Science Education* 9, 4 (2013).

[11] Qiang Hao, Bradley Barnes, Ewan Wright, and Eunjung Kim. 2018. Effects of Active Learning Environments and Instructional Methods in Computer Science Education. In *Technical Symposium on Computer Science Education (SIGCSE)* (Baltimore, Ireland). Association for Computing Machinery.

[12] Sandy Irani and Kameryn Denaro. 2020. Incorporating Active Learning Strategies and Instructor Presence into an Online Discrete Mathematics Class. In *ACM Technical Symposium on Computer Science Education (SIGCSE)* (Portland, OR, USA). Association for Computing Machinery.

[13] Mehmet Kayaalp. 2021. Using Logisim-evolution for Teaching Datapath and Control. In *ACM/IEEE Workshop on Computer Architecture Education (WCAE)* (Raleigh, NC, USA). Institute of Electrical and Electronics Engineers.

[14] Dimitra Kokotsaki, Victoria Menzies, and Andy Wiggins. 2016. Project-based learning: A review of the literature. *Improving schools* 19, 3 (2016).

[15] Joseph S Krajcik and Phyllis C Blumenfeld. 2006. *The Cambridge Handbook of the Learning Sciences* (first ed.). Cambridge University Press.

[16] Thom Markham. 2015. *Project Based Learning* (first ed.). Blurb, Incorporated.

[17] Margie Martyn. 2007. Clickers in the classroom: An active learning approach. *Educause quarterly* 30, 2 (2007).

[18] J. Patrick McCarthy and Liam Anderson. 2000. Active Learning Techniques Versus Traditional Teaching Styles: Two Experiments from History and Political Science. *Innovative Higher Education* 24, 1 (2000).

[19] Chet Meyers and Thomas B. Jones. 1993. *Promoting Active Learning. Strategies for the College Classroom* (first ed.). Jossey-Bass.

[20] Joel Michael. 2006. Where's the evidence that active learning works? *Advances in Physiology Education* 30, 4 (2006).

[21] Elizabeth Patitsas, Kimberly Voll, Mark Crowley, and Steven Wolfman. 2010. Circuits and Logic in the Lab: Toward a Coherent Picture of Computation. In *Proceedings of the 15th Western Canadian Conference on Computing Education (WCCCE)* (Kelowna, British Columbia, Canada). Association for Computing Machinery.

[22] Elizabeth Ann Patitsas and Steven Andrew Wolfman. 2012. Effective Closed Labs in Early CS Courses: Lessons from Eight Terms of Action Research. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE)* (Raleigh, North Carolina, USA). Association for Computing Machinery.

[23] James F. Power, Thomas Whelan, and Susan Bergin. 2011. Teaching Discrete Structures: A Systematic Review of the Literature. In *Proceedings of the 43rd ACM Technical Symposium on Computer Science Education (SIGCSE)* (Dallas, TX, USA). Association for Computing Machinery.

[24] Michael Prince. 2004. Does active learning work? A review of the research. *Journal of Engineering Education* 93, 3 (2004).

[25] Ella Tuson and Tim Hickey. 2022. Mastery Learning and Specs Grading in Discrete Math. In *ACM Conference on Innovation and Technology in Computer Science Education (ITiCSE)* (Dublin, Ireland). Association for Computing Machinery.

[26] Matthew West, Geoffrey L. Herman, and Craig Zilles. 2015. PrairieLearn: Mastery-based online problem solving with adaptive scoring and recommendations driven by machine learning. In *American Society for Engineering Education*. Association for Computing Machinery.

[27] Yuni Xia, Shiaofen Fang, and Kathy E. Johnson. 2023. An Exploration into Adaptive Teaching of Discrete Mathematics for Computer. In *Technical Symposium on Computer Science Education (SIGCSE)* (Toronto ON, Canada). Association for Computing Machinery.