

Circuits and logic in the lab: Toward a coherent picture of computation

Elizabeth Patitsas, Kimberly Voll, Mark Crowley, Steven Wolfman
University of British Columbia
Department of Computer Science
2366 Main Mall
Vancouver BC V6T 1Z4
{patitsas,kvoll,crowley,wolf}@cs.ubc.ca

ABSTRACT

We describe extensive modifications made over time to a first year computer science course at the University of British Columbia covering logic and digital circuits (among other topics). Smoothly integrating the hardware-based labs with the more theory-based lectures into a cohesive picture of computation has always been a challenge in this course. The seeming disconnect between implementation and abstraction has historically led to frustration and dissatisfaction among students. We describe changes to the lab curriculum, equipment logistics, the style of in-lab activities and evaluation. We have also made logistical changes to the management and ongoing training of teaching assistants, allowing us to better anchor our larger course story into the lab curriculum. These changes have greatly improved student and TA opinions of the lab experience, as well as the overall course.

Categories and Subject Descriptors

K.3.2 [Computers and Information Science Education]: Pedagogy, curriculum, education research, survey

1. INTRODUCTION AND MOTIVATION

The University of British Columbia Computer Science department offers a first-year, introductory course in discrete mathematics, proofs, logic, and digital circuitry. The course is unusual in spanning hardware and theory, particularly at the first-year level. In practice, lectures tend to focus on theoretical issues while labs focus on hardware. The course designers and instructors are enthusiastic about the connection between these areas, but the gap between them has caused ongoing problems. Indeed, focus groups conducted before our work began indicated that students' perception of the labs as disconnected from the remainder of the course dominated student dissatisfaction with the course [1].

In the spring of 2009, we began restructuring labs to address this issue. Our revisions spanned logistics, TA management, pedagogy, style, and even course content, aiming

at three goals: to reinforce the connection between lab learning goals and those of the overall course, to improve labs' pedagogical value for students, and to minimize student frustration with labs.

In this paper, we detail the interventions we have taken in the lab, each of which may be applicable in restructuring labs of other courses. In particular, we revised lab content and course flow to reintroduce a coherent overarching story of computer design for the circuits section of the course (Section 2); we reduced students' equipment management headaches with minimal impact on departmental support (Section 3); we altered lab materials both pedagogically, to be more engaging and experimental, and stylistically, to improve students' workflow (Section 4); we refined students' lab experience to encourage effective learning practices before and during the lab (Section 5); and we changed TA management and grading practices to improve TAs' effectiveness as facilitators (Section 6). In Section 7, we discuss positive changes in student perception of the labs exposed in course evaluations. Finally, we discuss future work and conclude that the changes have substantially improved the lab experience from the point of view of both students and teaching assistants.

1.1 Course Context

This course is a pre-requisite for our second-year theory and systems courses. Although itself has no pre-requisites, the vast majority of students take our introductory programming course either concurrently with or before this course. Roughly two years ago, we added the course as a corequisite, making this relationship formal.

Weekly contact hours include three hours of lecture, an optional one-hour tutorial, and a required, two-hour closed laboratory session. Labs and tutorials are run by teaching assistants, one per tutorial, and two per lab. About 100 students register for the course in a single autumn section, while about 200 register in two regular and one small, restricted spring section.¹ Labs are limited to 25 students and are typically close to full. The instructional staff may include as many as one dozen teaching assistants.

2. AN OVERARCHING STORY

Prior to the spring of 2009, the course was organized so that theory was taught largely in lectures and tutorials,

¹The restricted section is for second-degree students in our Bachelor of Computer Science program, outside the scope of this paper.

while hardware was taught in the labs. Originally the course was designed with a cohesive “story” for the hardware component, namely the exploration of a digital computer from ones and zeroes to a working (simulated) computer constructed from basic gates. Over time, however, this story ceased being conveyed as an initially overambitious course was pruned back to one that fits the time available.

In the spring of 2009, we first attempted to reintroduce the “big picture” of a working computer via a brief unit in lecture that sought to more explicitly tie the theory to the hardware, and a “capstone” lab during which students traced through machine instructions in a simulated CPU.

In the autumn of 2009, we further integrated this “big picture” into course, taking care to ensure that the labs and lectures clearly motivated each other as different perspectives of the same story, while keeping the story manageable in its scope. It was made clear that the labs were an instantiation of the ideas taught in lecture, and a motivation for the theory work (and vice versa).

In lab, we made two significant changes. First, we introduced the story of the computer at a unifying theme at the start of the term, referring explicitly to it throughout the term. In their first lab, students explored a simplified computer, examining the components inside, and speculating what some of the readouts might signify. The TAs and the lab writeup both emphasize that students should not expect to understand the computer at this point, but that they *should*, by the end of term, expect to understand each component they discover and be able to trace the interaction among the components that give rise to the computer operations. Students are naturally skeptical in the face of the system’s apparent complexity, but later labs returned to the computer again and again as students became familiar with key components such as multiplexers, buses, the ALU, registers, et cetera.

Second, we restructured the capstone lab to target a modest yet profound learning goal, namely to “trace execution of an instruction through our computer...down to individual wires and gates”. The capstone lab thus returned to the full simulated CPU and required students to follow machine code instructions bit by bit to work out how high level functionality arises from low-level manipulations. Our goal is to convince students that the functioning of a computer is fundamentally understandable and that given the compositional nature of the course, students can, at the end, trace any functionality within a simple computer to the underlying circuitry that gives rise to that functionality. We explicitly leave to later courses any exploration of the larger architectural issues in computer design.

Meanwhile, lectures now routinely and explicitly discuss connections to both the labs’ big picture story and a corresponding theoretical story traced mainly in lecture. While this takes relatively little time, we hope that it counteracts students’ tunnel-vision as they run up against new and difficult concepts.

3. SUPPORTING LAB EQUIPMENT

Roughly four to six of the approximately nine labs in the term involve actual hardware, with students creating their own beginner circuitry using an in-house-created kit called “The Magic Box.” Prior to the spring of 2009, students had to purchase and manage their own kits (at a cost of \$80 CAD per pair of students). The intention was that

ownership would give students interest and opportunity to experiment with them. Unfortunately, anecdotal and open-ended survey feedback both suggested that students instead saw the boxes as a frustrating and unnecessary expense.

In light of this, in the spring of 2009 we eliminated the requirement to purchase a kit and shifted to having TAs oversee the management and distribution of our university-owned kits during lab sessions. Our department’s educational technology coordinator handles repairs and replacements for components. Although working with the hardware is (inevitably) often frustrating, the kits now draw very few complaints in end-of-term evaluations; students seem to view “The Magic Box” as fun lab equipment rather than as a financial burden from which they need “their money’s worth”. Additionally, per term maintenance costs have proven modest², and neither theft nor intentional damage to equipment has occurred. Some students even enjoy the kits so much that they elect to buy their own, an option that remains available to them.

4. IMPROVING LAB MATERIALS

Central to our efforts to improve the labs has been the modification of lab materials. From a pedagogical standpoint, we changed the style of labs from confirmatory activities (in which students verify what they are expected to already know from readings and prior work) to exploratory and experimental activities that engage the skills they are learning to solve problems. From a logistical standpoint, we have streamlined students’ workflow so that they are more likely to complete necessary pre-lab work and can complete and learn from labs more efficiently.

4.1 Promoting Exploratory Activities

Prior to this work, labs often emphasized confirmation rather than exploration. For example, in one lab, students chose a logic gate, wired it up in their hardware kit, and verified its known behaviour. Such activities directly address learning goals, yet lack motivation from students’ perspective.

We have restructured and sometimes replaced labs to retain a focus on core learning goals yet achieve those goals through experimentation, encouraging analysis, exploration, construction and creativity. For instance, the example lab above was replaced with a task to wire up and identify “mystery” integrated circuits whose labels had been painted over. We also introduced several new activities where students are tasked with finding glitches and errors in circuits, such as asynchrony glitches in multiplexers and faulty wirings in circuit simulations. The students identify the causes of the bugs and propose solutions. For example, an old lab on adders that considered alternate one-bit adder designs has been replaced by one in which students design an eight-bit adder hierarchically and apply their design to explore Caesar ciphers. Adders are also addressed more abstractly in lecture, further solidifying the lab-lecture connection.

4.2 Streamlining Workflow

Students learn little from frustrating or overambitious labs that they are unable to complete on time, though workflow

²In more than three terms, none of the communal pool of 15 kits has been damaged beyond local repair, and a stock of one hundred extras remains untapped.

in the labs can contribute to such difficulties. Unfortunately, some of the changes from the previous section can actually make labs more challenging (if more satisfying) for students.

Thus, we have taken care to manage the clarity and length of the labs as we make changes. Our goal has been lab exercises that take about 100 minutes to complete (of 110 minutes available). As we reduce the length of labs, students and TAs feel less time pressure, which results in a more relaxed and effective learning experience.

As we prune for length, we eliminate irrelevant commentary, extraneous details, and even “colour text” that may seem pedagogically valuable but doesn’t contribute to the students’ activities. (Instead, we add “colour” to lessons by having students solve problems of interest rather than reading about them.) For a sense of scope, Lab #2 from the autumn of 2008 had 2400 words, while the revised edition has only 1800 words [2]. The version used in spring 2010 has 1700 words. To make students’ workflow clear, instructions evaluated by the TAs are consistently labeled with an emboldened “TODO”. Pre-lab work—particularly important to achieving the relaxed, effective atmosphere we aim for in lab—is now summarized in the introduction and consistently labeled “TODO (pre-lab)”. Between these workflow changes and changes to the marking scheme discussed below, pre-lab completion rates have dramatically improved.

5. REFINING THE IN-LAB FORMAT

5.1 Open-Ended Discussion Periods

Also new to the current term, short, ten-to-thirty minute discussion periods have been added to the lab curriculum. Divided into groups of four-to-seven students at the start of lab, TAs lead free discussions in the context of open-ended questions relevant to the main lab activities, and with the purpose of establishing a clearer bridge between lab and lecture.

Example questions include, “What is a computer?”, asked during the first lab of the term (introducing the CPU simulation), or “What problems could arise from packing too many multiplexers together on a chip of RAM?” (for a lab on multiplexers).

A second aim of grouping students for the discussions is to encourage student interaction, allowing students to meet and find partners for working on the labs. This is discussed in the next section.

5.2 Encouraging Student Collaboration

The labs have been reformatted to strongly encourage students to work in pairs. Students are afforded an opportunity to interact with and support one other, creating a more team-based and enjoyable environment for both students and TAs. As suggested in the literature on pair programming, the overall quality of student work has increased [3]. Additionally, since students can rely upon each other more for questions and ideas, the demand upon the lab TAs has decreased to more manageable levels. TAs now report enjoying the labs, which in turn improves the quality of teaching and the overall student experience.

One hitch with the pair work has been in the collaboration policies. In the autumn of 2009, students were required to register with their partners at the beginning of the term, completing all labs and assignments with this individual. This was found to be problematic, and was relaxed

in the spring of 2010 semester. Students now may work with whomever they wish in the labs from week to week. Changing policies has caused some confusion; most important is to assign a collaboration policy at the start of class and abide by it throughout the term to avoid ambiguity. In future terms we may look at more pro-active assigning of partners, in light of the success other implementors of pair programming have had [5].

6. TEACHING ASSISTANT TRAINING

This course requires a varied skill set that challenges the TAs, who receive no course-specific training beforehand. Increased attention to continual TA training during the term has been shown to enhance the student learning experience [4]. Often TAs in our course, hired for their knowledge of discrete mathematics, were lacking confidence with digital circuits. In addition, the TAs were given minimal instructions with respect to the labs and instead were instructed to independently prepare for them. As part of our revisions, the course was modified to support a stronger team-based approach, and to accommodate more thorough and ongoing TA training through weekly Lab TA meetings, which has proven tremendously beneficial in the form of improved teaching and overall TA commitment.

6.1 Weekly Lab TA Meetings

Beginning in the 2009 spring term, a weekly TA meeting was established to prepare for the upcoming labs. Meetings were led by the course’s “Lab Coordinator”, a TA whose role was to develop the lab curriculum, manage the equipment, liaise with the instructor(s), and run these meetings.

During the meetings, lab TAs worked through the entire upcoming lab, beta testing the new activities and providing feedback for the Lab Coordinator who could then make modifications in time for releasing the lab to students[2]. During this meeting, the lab TAs also received training on how to lead lab activities.

While there has been a net positive effect on TA confidence and teaching quality in the labs, attendance of these meetings has been an issue. Central to this issue seems to be the perceived seniority of the Lab Coordinator, as well as returning TAs who question the efficacy of such a meeting given their experience level. To address this issue, instructors must make clear the expectations of the teaching team, and the importance of these meetings for addressing lab edits, mentoring new TAs, and discussing strategies for the current term.

6.2 Posting Labs in Advance

The weekly TA meetings have brought about a major improvement in the lab workflow. Labs are now consistently made available to students one full week in advance of their lab. This gives students an entire week, beginning the moment they finish the current lab, to complete the pre-lab activities for the next week’s lab. During the weekly TA meeting the lab discussed is not the week’s immediately pending lab, but the lab following. This allows time for edits, syncing with lecture, and publishing in enough time to maintain the one-week lead time for students. This also benefits TAs by ensuring they are familiar with where the labs are going.

6.3 Tying Grading Scheme to Learning Goals

Unclear marking schemes are known to negatively affect student views of science courses [4]. As of the autumn of 2009, the grading scheme used for the labs has been substantially changed in light of previously ambiguous instructions. To promote clarity, all labs were now marked out of ten points, and the grading scheme was posted clearly on the website for the students to read.

The new marking scheme stressed analysis of the lab material that was consistent with student observations and experimentation. Students are graded on how well their analyses fit what is observed in the software or hardware portions of the lab, rather than solely whether their answer was correct or not.

A key addition to the marking scheme was the inclusion of two dedicated marks for prelab exercises, motivating the students to do the exercises and show us the work.

The labs were graded out of ten marks, with an optional bonus mark reserved for applicable labs. In practice we found that most marks ranged from six to nine. To determine the requirements for each lab mark, the TAs complete each lab problem, discussing the calibre of answer required for full points, partial points, and, if reasonable, a bonus point.

7. SURVEYING STUDENTS

Surveying students via online evaluation form in the spring of 2009 and in the spring of 2008, students were asked on a 1-5 Likert scale whether, “The labs so far tie into the rest of the course material” (with a 5 being strongly agree). Before the changes, in the spring of 2008, the mean was 3.4 and the median was 3 ($n = 38$). Halfway through the spring 2009 term, however, this had already improved to a mean of 3.5 and median of 4 ($n = 56$), suggesting that our interventions were well placed, and serving to provide a baseline for comparison for our continued efforts.

Students were also asked whether the pre-lab work contributed to their understanding of the material. In 2008, the mean response was a 2.7 with a median of 3. In contrast, in 2009, the mean response was a 3.3 with a median of 4.

On the question of whether “the labs are contributing to my understanding of the course material”, students in 2009 responded on average with 3.6 and a median of 4 – unchanged from the 2008 results. It is important to reiterate, however, that these results were obtained only halfway through the term and that our efforts were ongoing.

Anecdotally, some teaching assistants have expressed their satisfaction with the new lab materials and format, and have reported seemingly increased student satisfaction as well.

8. FUTURE WORK

At present, we are continuing to rework many of the lab instructions, as outlined in this paper. Creating labs with high educational value that motivate and enhance the lecture material continues to be a challenge. The lack of an obvious hands-on lab component for some units, such as proof techniques, often forces us to look for other course material to cover in lab, causing a break in the timing of material presented in lab versus lecture. Developing discussion periods to better tie back to the lectures is a current direction in the effort to better bridge the lectures and labs.

The first official lab of the course remain as yet unsatisfac-

tory. Introducing the unifying CPU simulation in a friendlier format for this lab is an important task as without it, the central theme of the course is potentially lost. The overall breadth of the simulation is intimidating for students, and thus it must be presented carefully, ensuring students understand that they they are currently only exploring, and that a step-by-step analysis will evolve over the course of the term. Part of this involves training TAs in how best to convey this message.

There is also the matter of “Lab #0”, a quasi-lab where students get their login information, activate their email, and perform a few other key Unix errands. The lab instructions are out-of-date and the lab has been identified as impeding student interaction and providing a negative impression of the labs. A key lesson for us has been to establish the exploratory style and collaborative nature of labs on day one [2].

9. CONCLUSIONS

The ongoing work to revise our introductory course in discrete mathematics, proofs, logic and digital circuitry has been significantly positive. In particular we have worked hard to better integrate the course along the lines of a coherent and over-arching story of computer design, streamline student work flow, and improve the overall pedagogy within the labs themselves.

Specifically we have sought to solidify the connection with lab and lecture, adding in discussion periods to the labs themselves that emphasize lecture material, and tying together labs along the lines of the the journey from zeroes and ones to a fully functional (albeit simulated) computer, a message that is further driven home in lectures. Further changes in the lab in the form of improving lab equipment operations and clarifying lab instructions have greatly improved the efficiency with which students are able to work. Incorporating weekly TA meetings with a Lab Coordinator and a concentration on ongoing TA preparation has additionally increased the calibre of the teaching within labs. Finally, making lab instructions clear, obvious and concise has been dramatically helpful for both students and TAs.

We have also learnt that challenging the students increases their engagement and enjoyment of the material, and that providing equipment for them results in a fundamentally more positive view of the hardware-based labs.

In closing, we have presented a series of revisions for a digital-circuitry and logic course that other institutinos may wish to use as a basis for a new course, or to update an existing course. Furthermore, the changes brought about in this particular course could in fact be useful in a wide variety of courses in which a lab-based component seeks to solidify the more abstract presentations in lecture, or more generally for a course that seeks to improve its TA and lab management processes.

10. ACKNOWLEDGMENTS

We would like to acknowledge the instructional staff and students of CPSC 121 over the past several years, without whom these efforts could not have proceeded. In particular, we would like to thank Mark Greenstreet and Bob Woodham for their help and inspiration in re-introducing the working computer lab, Anthony Winstanley for his work repairing and supplying the circuit kits, Michele Ng for running focus

groups, Ben Yu for his work on the evaluations and surveys, and Carl Wieman for his input to the department's educational technology coordinator the marking scheme.

11. REFERENCES

- [1] D. Acton, K. Voll, S. Wolfman, and B. Yu. Pedagogical transformations in the UBC CS science education initiative. In *WCCCE '09: Proceedings of the 14th Western Canadian Conference on Computing Education*, pages 3–8, New York, NY, USA, 2009. ACM.
- [2] G. Echlin, P. Kiarostami, E. Patitsas, and S. Wolfman. Revising an introductory computer science course: Exploratory labs, interactive lectures, and just-in-time teaching, 2009.
- [3] C. McDowell, L. Werner, H. E. Bullock, and J. Fernald. Pair programming improves student retention, confidence, and program quality. *Commun. ACM*, 49(8):90–95, 2006.
- [4] C. O'Neal, M. Wright, C. Cook, T. Perorazio, and J. Purkiss. The impact of teaching assistants on student retention in the sciences: Lessons for TA training. *Journal of College Science Teaching*, 36(5):24–29, 2007.
- [5] L. Williams. Lessons learned from seven years of pair programming at North Carolina State University. *ACM SIGCSE Bulletin.*, 39(4):79–83, December 2007.