2021 ASEE ANNUAL CONFERENCE
Virtual Meeting | July 26–29, 2021 | Pacific Daylight Time
Paper ID #33020

Having it All: Infusing Parallel Computational Thinking in the Lower-level Computer Engineering Curriculum Using Extended Learning Modules

Mr. Zeran Zhu, University of Illinois at Urbana Champaign
Dr. Ujjal K. Bhowmik, University of Illinois at Urbana Champaign
Ms. Yue Wang, University of Illinois at Urbana Champaign

Graduate research assistant, Electrical & Computer Engineering, UIUC Master student, School of Labor and Employment Relations, UIUC

Dr. Zuofu Cheng, University of Illinois at Urbana Champaign
Prof. Yuting W. Chen, University of Illinois at Urbana Champaign

Dr. Yuting W. Chen received the B.S. degree from University of Illinois at Urbana-Champaign in 2007, and the M.S. and Ph.D. degrees from Rensselaer Polytechnic Institute in 2009 and 2011, all in Electrical Engineering. She is currently a Teaching Assistant Professor with the Department of Electrical and Computer Engineering at University of Illinois at Urbana-Champaign. Prior to joining Illinois ECE as a faculty, she worked at IBM Systems Group in Poughkeepsie, NY in z Systems Firmware Development. Her current interests include recruitment and retention of under-represented students in STEM, integrative training for graduate teaching assistants, and curriculum innovation for introductory computing courses.

Having it all - Infusing Parallel Computational Thinking in the Lower-Level Computer Engineering Curriculum Using Extended Learning Modules

Abstract

Modifying a course in a well-established program is often challenging due to many factors. First of all, it requires removing a significant portion of the current materials to make room for new topics. In addition, these changes must be reviewed and approved by several layers of committees, which can be a long process. Last but not least, the impact on students could vary widely, depending on their preparation and learning ability and the results would not be known until the changes have been made.

At the University of Illinois at Urbana-Champaign, a team of faculty and graduate teaching assistants is taking on the challenge of making a curriculum change in lower-level computer engineering courses to infuse parallel computational thinking using extended learning modules. The proposed changes impact three required courses in the curriculum: a 100-level digital logic course, a 200-level introductory programming course, and a 300-level digital systems course. Despite the prevalent use of multi-core and GPUs in computers and handheld devices, parallel and distributed computing education in undergraduate courses is largely absent at the lower levels. This effort identifies current topics in the three courses that could be extended into parallel computing learning modules. These modules are launched at the same time the corresponding course topics are covered, and students are given extra-credit for completing these modules.

This paper focuses on the implementation and assessment of the extended learning modules in the 100-level digital logic course. Two modules were developed and launched in the fall of 2019, one on carry-look-ahead parallel adder and the other on counter with parallel implementation. Each module contains a short-recorded video (under 12 minutes), a set of PowerPoint slides, as well as an asynchronous assessment. Each assessment contains five to six true-or-false, multiple-choice, and fill-in-the-blanks questions. Students are expected to complete each module individually within three weeks of its release date. After completing the adder module, students should be able to: 1) understand the inefficiency of a serial adder; 2) understand the concepts of generate and propagate signals as the basis of carry-look-ahead recursive formulation; 3) express the carry-out recursive expression in terms of inputs. After completing the counter module, students should be able to: 1) recognize a carry-ripple counter and explain its shortcomings; 2) understand that the same approach in carry-look-ahead adder can be used to solve the delay in carry-ripple counter; 3) understand the trade-offs among different parallel counter implementations.

In Fall 2019, 48% (n=183) of students completed the adder module and 47%(n=178) completed the counter module. The completion rate in Spring 2020 and Fall 2020 are 51% (n=144) and

60%(n=215) for the adder module, 52%(n=147) and 60%(n=216) for the counter module, respectively. Besides presenting student assessment data, we will also investigate which group of students by academic achievement are more likely to complete these extended learning opportunities and whether there is a correlation between their performance in these modules and overall performance in the course.

Introduction

The current undergraduate computer engineering curriculum at the University of Illinois at Urbana-Champaign, which has been in place since 2015, consists of a three-course sequence. The first course, ECE 120, is taken Freshman year for most computer engineering students. Topics consist of Boolean algebra, combinational and sequential logic design, state machine design, and basic computer architecture and machine data representation. ECE 120 is also augmented by a hardware lab component consisting of the design and construction of several discrete logic based circuits. The next course in the sequence, ECE 220, focuses on LC-3 assembly and C programming, fundamental data structures and algorithms. ECE 220 has a laboratory component as well, in the form of software lab assignments. The final course in the sequence, ECE 385, consists of a series of hardware design challenges on various digital logic topics, followed by a final project of the students' own choosing involving digital logic. These design challenge span a range of technologies, from discrete logic in the earlier labs to FPGA logic and then to system-on-chip hardware-software co-design.

In response to the paradigm shift in machine capabilities (especially, with multicore, many core, and GPU computing capabilities), a group of faculty along with the teaching assistants at University X have been integrating the concepts of parallel and distributive computing (PDC) in the courses mentioned above. Our goals is to introduce the concepts of PDC as small modules, programming assignments and advanced timing analysis labs as extra credits along with the existing materials in ECE 120, ECE 220, and ECE 385, before students get in to the specialized parallel computing course, ECE 408, designed for seniors.

Integrating parallel computing in the undergraduate curriculum has started in many universities in the United States. Several universities have attempted and incorporated the parallel and distributive computing concepts in the early level of their curriculum[1][2][3][4][5]. When should we introduce parallel concepts into the curriculum is a debatable question. Lesson learned and insights provided by the several authors suggests that some educators and scientists argued that because of the intricacy of parallel computing it should be introduced as early as in the freshmen level. Some argued that it should be introduced slowly as small modules in the existing traditional courses. They believe that by introducing the parallel concepts early in the curriculum will enable students to look at a problem from different perspectives and will consider the parallel option as a solution to problems that they want to solve. Some argued that it should be offered as a senior level required course. The last option will be harder to implement at our program, as it would require elimination of another core course in the existing curriculum. We believe that parallelism and concurrency are fundamental topics in computer engineering and should permeate the curriculum. By adding the concepts slowly as modules to existing courses, we hope to encourage and motivate more students to pursue parallel and distributive computing in the senior level and choose it as a future career. In 2019, we obtained an internal grant from the college to help

facilitate our efforts. The remaining of the paper presents the implementation and assessment of PDC modules integrated in the ECE 120 course.

Module Selection and Development Process

After evaluating multiple sources, we have identified two primarily focused topics to cover in our entry-level digital course - ECE 120. These topics include parallel adder from the book "Fundamentals of Logic Design" by Jr. Charles H. Roth[6] and parallel counter from Chapter 5 of the book "Topics in Parallel and Distributed Computing: Introducing Concurrency in Undergraduate Courses" by Vaidyanathan et al.[7] We believe these are excellent topics for entry-level students because they are relatively easy to understand and closely related to the pre-existing course materials covered in the course.

In the original ECE 120 course planning, the adders and counters covered are the simplest ones. We teach students the concepts of full-adders (FA) and carry-ripple adders, as well as the concept of ripple-counters. In order to train the students' minds with the parallelism methodology, we decided to introduce them to the Carry-Lookahead Adders (CLA) and Parallel Counters. The development of these modules went through an iterative process. The graduate teaching assistant prepares the initial presentation draft for both modules, which will then be discussed during the weekly course staff meetings for feedback. Faculty members then give their suggestion for improvement and the graduate teaching assistant will then revise accordingly. Next, when the slide contents are ready, a short video that provides a walk-through of the slides will be produced for each of the modules. Finally, the course staff evaluate the key learning objectives within each module and work together on preparing a list of questions to be included in the evaluation quizzes.

Parallel Adder (Carry-lookahead Adder) Module

Carry-lookahead Adders (CLA) is a standard component of the 300-level digital systems course. Thus, this module is a great candidate for evaluating the outcome of this project because we can test whether the students still remember the principles of CLAs at the beginning of ECE 385 using short in-class quizzes. We designed the Parallel Adder module in the following format. An animated slide deck[8] that includes a self-paced learning guide for the students, as well as a short 10-minute narrated video[9] that walks through the entire slide deck.

In the presentation slides, we first give an example of a four-bit carry-ripple adder. In the example, we use two inputs 0b0101 + 0b0111, and walk the students through the process of each carry bit between the single-bit adders, as well as the final output. These two input numbers are carefully selected because we can see all of "propagate" (P), "generate" (G) as well as "kill" (K) situations. In the next few slides, we direct the students to focus on the bits where the Carry signal can be generated, killed, or propagated without knowing the C_{in} signal. After getting familiar with the potential speed-up opportunities, the students will revisit the C_{out} truth table for the simple adder, and learn to identify the P and G signals from the four input combination of a one-bit simple adder. At this point, the student should have a solid understanding of how the P and G signals work and what they represent. We then introduce the recursive formula for calculating the carry signals within an adder. To put it all together, we show an example of a carry-lookahead adder

and walk through an example with the students. To demonstrate the performance improvement, we implemented a carry-ripple adder as well as a carry-lookahead adder, both 16-bit wide, in SystemVerilog, and demonstrated the difference in achievable F_{max} in Intel Quartus.

Parallel Counter Module

Parallel Counter is a completely new concept to students at University X because it never showed up on any level of courses. We believe that the Parallel Counter module is a natural extension of the existing ripple-counter concept in ECE 120, and there are a lot of interesting parallel programming ideas that go beyond the counter itself. The Parallel Counter module is also in the form of self-paced slides[10] with a narrated video[11].

In the presentation slides, we first show the students a ripple counter design using D Flipflops. In this design, each bit's output signal is connected to the inverted clock input of the next bit. We then show a clock-by-clock demonstration of the basic operation of this design. Naturally, this leads us to the discussion of the downside of the asynchronous design: every bit is not updated at the same time. To solve this issue, we present the next synchronous design where we use XOR gates to precompute the next state for all bit positions and update all bits together in the next rising edge of a common clock. Here we briefly mention the selective inversion capability of an XOR gate. The precompute unit contains a chain of AND gates, which is analogous to a chain of simple adders in a carry-ripple adder. At this step, we pause and remind the students what problem did carry-ripple adders have and how we solved the problem. With a solid understanding of the previous module, the students should be able to come up with a connection to carry-lookahead adders. Then we present a "faster" design by replacing the AND gate chain (inherently recursive) with a group of multiple-input AND gates (flattened). After we demonstrate how this design works by analyzing the precompute unit, we show that some of the AND logic is redundant, and this circuit can be further improved to reduce the cost. We present the Kogge-Stone circuit as well as the Ladner-Fischer circuit that further parallelize the adder group computation. We inform the students that similar techniques will be introduced in ECE 408, where they will learn GPU programming with multiple threads.

Learning Objectives

Two sets of assessments are created based on the following key learning objectives. Details of assessment questions are shown in the next section.

Parallel Adder

- Understand the inefficiency of a serial adder
- Understand the concepts of generate and propagate signals as the basis of carry-look-ahead recursive formulation
- Express the carry-out recursive expression in terms of inputs

Parallel Counter

- Recognize a carry-ripple counter and explain its shortcomings
- Understand that the same approach in carry-look-ahead adder can be used to solve the delay in carry-ripple counter
- Understand the trade-offs among different parallel counter implementations

Assessment Results

The extended learning modules discussed above were deployed in Fall 2019, Spring 2020 and Fall 2020 in ECE 120. In Fall 2019, 49% (n=183) of enrolled students completed the assessment on the parallel adder module and 48% (n=179) completed the assessment of the parallel counter module. In Spring 2020, the completion rates are 52% (n=145) and 53% (n=148) for the parallel adder module and the parallel counter module, respectively. Completion rates are 60% in Fall 2020 for both modules. Student participation has grown steadily throughout the three semesters, as shown in Table 1 and Table 2. Besides higher completion rates, students in Spring 2020 and Fall 2020 also performed better on both modules than those in Fall 2019.

Semester	Participation Rate	Mean	SD
Fall 2019	48% (n=183)	81.42%	23.48
Spring 2020	51% (n=144)	88.72%	19.51
Fall 2020	60% (n=215)	88.66%	18.32

Table 1: Assessment statistics of the parallel adder module.

Semester	Participation Rate	Mean	SD
Fall 2019	47% (n=178)	86.70%	21.55
Spring 2020	52% (n=147)	96.60%	11.32
Fall 2020	60% (n=216)	95.19%	12.46

Table 2: Assessment statistics of the parallel counter module.

To understand learning outcomes of the parallel computational topics, question by question scores are examined for both extended learning modules across three semesters. Table 3 shows the statistics for the parallel adder module and Table 4 the parallel counter module. The statistics for both modules are also plotted out in Figure 1 and Figure 2. For the parallel adder module, the overall trend is similar for all three semesters despite the difference in scores. Students performed well on questions that ask them to calculate the propagate, generate and kill signals based on a given truth table. The formula for these calculations are specifically covered in the module or provided in the question. Therefore, it aligns with our expectation that most students should be able to answer these questions correctly. For the other three questions, which require them to

derive the answers themselves, fewer students were able to arrive at the correct answers, leading to significantly lower average scores.

The question by question scores in the parallel counter module are much higher than the parallel adder module, indicating that majority of the students arrived at the correct answers. The most difficult question, which consistently has the lowest score in all three semesters, assessed the understanding of a given parallel counter design. The most common mistake is assuming that the given design has no obvious downside, failing to realize that it will be quite slow with large number of inputs.

Questions	Question type		FA19	SP20	FA20
Q1.Given a truth table, determine Generate Signal	True/False	Mean	8.75	9.06	9.13
		SD	3.33	2.94	2.84
Q2.Given a truth table, determine Propagate Signal	True/False	Mean	8.97	9.26	8.85
		SD	3.06	2.64	3.21
Q3.Match K, G, P signals to blanks in the CLA adder	Fill in the Blanks	Mean	7.87	8.42	8.98
		SD	3.80	3.43	2.75
Q4.Express P in terms of K and G	Multiple Choice	Mean	6.94	8.05	8.16
		SD	4.63	3.99	3.89
Q5.Given a truth table, based on Input/Output/Carry, determine P, G, K	Multiple Choice	Mean	8.97	9.19	9.31
		SD	3.06	2.74	2.55
Q6.Use recursive formula to calculate C_i	Multiple Choice	Mean	7.38	8.25	8.30
		SD	4.42	3.82	3.77

Table 3: Means and Standard deviations of parallel adder assessment question for Fall 2019, Spring 2020, and Fall 2020. Each question is worth 10 points.

Questions	Question type		FA19	SP20	FA20
Q1.Understand XOR gate as "selective inverter"	True/False	Mean	9.56	9.73	9.73
		SD	2.08	1.63	1.65
Q2.Determine the clock inputs of a ripple counter	True/False	Mean	8.94	9.74	9.63
		SD	3.09	1.63	1.90
Q3.Given a serial counter design, identify its major disadvantage	Multiple Choice	Mean	8.38	9.60	9.31
		SD	3.70	1.98	2.55
Q4.Given a ripple counter design, identify its major disadvantage	Multiple Choice	Mean	8.61	9.80	9.87
		SD	3.48	1.42	1.18
Q5.Given a parallel counter design, identify its major disadvantage	Multiple Choice	Mean	7.88	9.53	9.08
		SD	4.11	2.13	2.91

Table 4: Means and standard deviation of parallel counter question for Fall 2019, Spring 2020, and Fall 2020. Each question is worth 10 points.

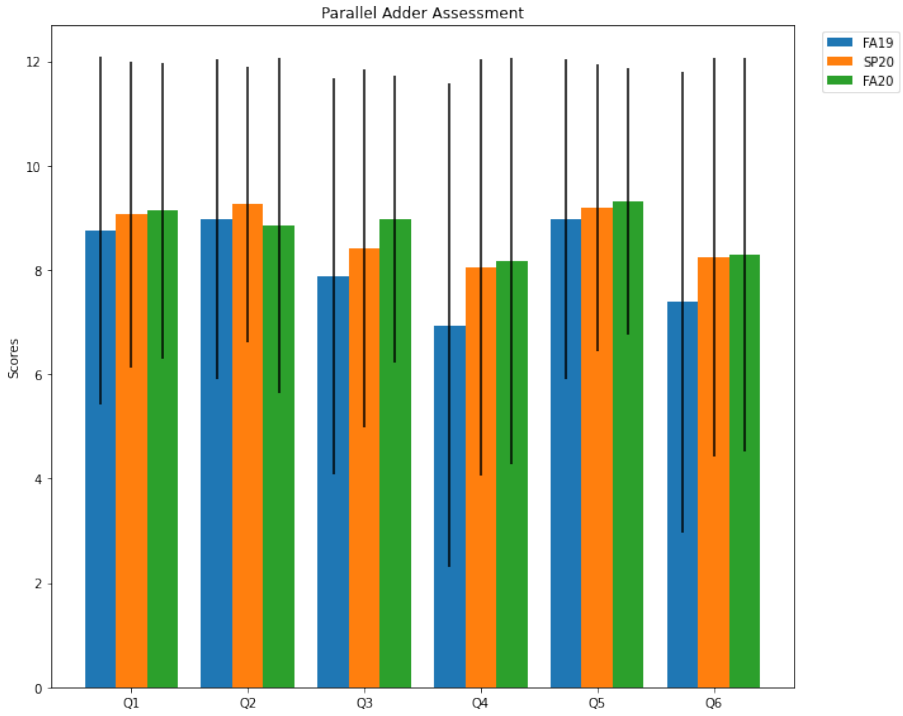

Figure 1: Question by Question Results of Parallel Adder Assessment.

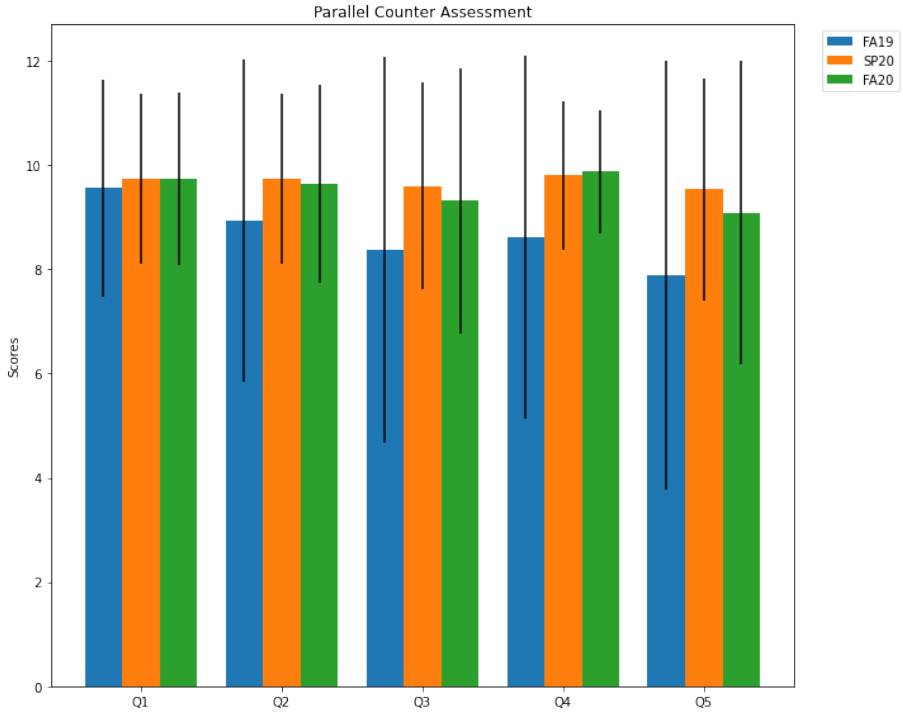

Figure 2: Question by Question Results of Parallel Counter Assessment.

Relationship Between Module Performance and Course Performance

In ECE 120, students' original final scores did not include the extra credits earned from completing the modules. Therefore, the original final score is the reflection of their overall academic performance in the course. Based on their original final score, students were assigned letter grades from A+, A, A-, B+, B, B-, C+, C, C-, D+, D, D-, to F. We use the standard cut-offs of 97 / 93 / 90 / 87 / 83 / 80 / 77 / 73 / 70 / 67 / 63 / 60 when assigning letter grades. To simplify our analysis, students were divided into three groups based on their original final letter grade to represent three different levels of course performance. Students who had original final letter grades of A+, A, and A- belong to Group 1; those with original final letter grades of B+, B, and B- belong to Group 2; and the rest (with original final letter grades of C+, C, C-, D+, D, D-, and F) belong to Group 3. All statistical analysis presented in this paper is performed using IBM SPSS [12].

To explore the difference in module performance among groups in each semester, ANOVA was conducted to inspect whether there is any significant difference between at least two of the three groups. If there exists a significant difference, a Post-Hoc Test (Scheffe test when variances are equal, Games-Howell test when variances are not equal) will be conducted to identify the pair and their specific difference. Results of the ANOVA tests are shown in Table 5.

In Fall 2019, since the variances of parallel-adder-module scores of the three groups are not equal, Welch's ANOVA was conducted, yielding a p-value smaller than 0.05. Therefore, a significant difference in parallel adder module performance exists between at least two groups. The follow-up Post-Hoc Test revealed that significant difference exists between Group 1 and Group 2 ($p=0.021$), Group 1 and Group 3 ($p=0.000$), and Group 2 and Group 3 ($p=0.043$). By examining the means of each group for the parallel adder module, we concluded that students in Group 1 performed significantly better than those in Group 2 and Group 3; students in Group 2 also performed significantly better than those in Group 3. One-way ANOVA was conducted on the parallel counter module data because of the homogeneity of variances of the three groups. No significant difference in performance is shown between any two groups ($F=0.52$, $p=0.595$).

One-way ANOVA was conducted on scores of both modules for Spring 2020. It showed no significant difference in performance on the parallel adder module and the parallel counter module between any two groups with $F= 2.65$, $p=0.074$ and $F=1.96$, $p=0.144$, respectively.

For Fall 2020, since the variances of parallel adder module scores of the three groups are not equal, Welch's ANOVA was conducted and indicated a significant difference in performance between at least two groups ($F= 5.40$, $p=0.007$). The follow-up Post-Hoc Test showed that students in Group 1 performed significantly better than those in Group 3 ($p=0.019$). For the parallel counter module, with unequal variances of the three groups, Welch's ANOVA was conducted and indicated a significant difference in performance among at least two groups ($F=3.35$, $p=0.041$). Although the follow-up Post Hoc test showed no difference among group means in the 0.05 significant level, the p value in the comparison between Group 1 and Group 3 is very close to 0.05. Therefore, we treated the mean difference between Group 1 and Group 3 as significant and reached the conclusion that students in Group 1 performed significantly better that those in Group 3.

To further explore whether there is any correlation between module performance and course performance, Pearson correlation analysis was conducted on the module scores and the original final scores in each semester. The cut-off points we used for the judgement of weak, moderate, strong correlations are 0.3 and 0.7[13]. According to the results shown in Table 6, the parallel adder module score in Fall 2019 is positively associated with the original final score (r=0.309, p<0.001); the correlation strength is moderate. In Spring 2020, the parallel adder module score is positively associated with the original final score (r=0.199, p<0.05); the correlation strength is weak. In Fall 2020, both parallel adder module score and parallel counter module score are positively associated with the original final score, r=0.382, p <0.001 and r=0.308, p<0.001 respectively; the correlation strength is moderate.

Semester	Module	Group	N	Mean	SD	Test of Homogeneity of Variances	One-way/Welch's ANOVA		Post Hoc Test		
							F	p	Pairs	Mean Difference	p
FA19	Parallel adder module	1	62	90.46	15.36	p=0.000	12.75***	0.000	Group1&2	9.23*	0.021
		2	75	81.22	24.29				Group1&3	20.89***	0.000
		3	46	69.57	26.13				Group2&3	11.66*	0.043
	Parallel counter module	1	53	89.06	16.90	p=0.595	0.52	0.595	-	-	-
		2	80	85.50	22.72				-	-	-
		3	45	88.00	20.63				-	-	-
SP20	Parallel adder module	1	64	91.02	17.19	p=0.074	2.65	0.074	-	-	-
		2	47	90.34	17.38				-	-	-
		3	33	81.94	24.97				-	-	-
	Parallel counter module	1	69	97.68	7.31	p=0.144	1.96	0.144	-	-	-
		2	46	97.39	9.99				-	-	-
		3	32	93.13	18.04				-	-	-
FA20	Parallel adder module	1	123	92.21	14.40	p=0.000	5.40**	0.007	Group1&2	5.60	0.114
		2	61	86.61	19.19				Group1&3	13.58*	0.019
		3	31	78.63	25.50				Group2&3	7.98	0.283
	Parallel counter module	1	124	97.10	9.09	p=0.001	3.35*	0.041	Group1&2	2.02	0.451
		2	61	95.08	11.35				Group1&3	9.36[a]	0.055
		3	31	87.74	21.09				Group2&3	7.34	0.180

Table 5: Results of ANOVA test on module scores in Fall 2019, Spring 2020, and Fall 2020. * denotes p <0.05, ** denotes p<0.01, *** denotes p <0.001, [a] denotes treated as significant.

Semester	Variables	N	Mean	SD	Pearson Correlation Coefficient
FA19	Parallel adder module score	183	81.42	23.48	0.309***
	Original final score	183	84.67	9.54	(p= 0.000)
FA19	Parallel counter module score	178	87.19	20.56	0.052
	Original final score	178	84.59	8.92	(p= 0.487)
SP20	Parallel adder module score	144	88.72	19.51	0.199*
	Original final Score	144	86.22	9.16	(p= 0.017)
SP20	Parallel counter Module score	147	96.60	11.32	0.111
	Original final score	147	86.72	8.95	(p= 0.182)
FA20	Parallel adder module score	215	88.66	18.32	0.382***
	Original final score	215	88.49	8.19	(p= 0.000)
FA20	Parallel counter Module score	216	95.19	12.46	0.308***
	Original final score	216	88.50	8.17	(p= 0.000)

Table 6: Results of Pearson correlation analysis on module scores and original final scores in Fall 2019, Spring 2020, and Fall 2020.* denotes $p<0.05$, *** denotes $p<0.001$.

Participation Rate Among Groups and Impact on Course Performance

To understand the likeliness to participate in extended-learning among students in the three groups defined earlier, we examine the participation rate among these groups from Fall 2019 to Fall 2020, as shown in Table 7. During Fall 2019, which was a normal semester with all in-person instruction, participation rates are at similar levels among all three groups. As instruction transitioned to half-online (Spring 2020) and then fully online (Fall 2020), we saw an increase in participation rate in Group 1 (over-performing) and decrease in Group 3 (under-performing). This corresponds to what we have observed in other parts of the course, in which excelling students were able to manage their time better and more engaged than who were already struggling.

	FA19	SP20	FA20
Group 1	48.25%	59.20%	71.51%
Group 2	59.15%	61.54%	57.27%
Group 3	50.00%	44.87%	43.06%
Overall	52.76%	55.87%	61.50%

Table 7: The participation rate in extra-credit by each group in FA19, SP20, and FA20

Students who took ECE 120 from Fall 2019 to Fall 2020 had final scores which consists of the original final score and possible extra credits from the two extended learning modules. For participation in the parallel adder module, students can earn extra credits toward homework (1% of overall grades in Fall 2019 and 1.5% of overall grades in Spring 2020 and Fall 2020 without going over the 15% homework cap). For participation in parallel counter module, students can earn extra credits toward Labs (1% of overall grades in Fall 2019 and 1.5% of overall grades in Spring 2020 and Fall 2020 without going over the 15% Labs cap). In some cases, especially with

high achieving students, participation in modules will not influence their final scores, because they already achieved full credits in their homework and lab sections of the course. We increased the extra credit points in Spring 2020 and Fall 2020 to encourage more participation in the extra credit modules. To explore to what extent the module score can impact a student's overall course performance, we calculated how many points of extra credit from modules have been added to each student's original final score in the three semesters, and then looked into the letter grade improvement caused by extra credits from modules in three semesters.

In Fall 2019, 162 out of 201 students (80.60%) had a final score increase due to completion of at least one extended learning modules. The range of score increase is between 0.015622 and 1.250001. In Spring 20, 150 out of 157 students (95.54%) had a final score increase and the range is between 0.0125 and 3. In Fall 2020, 192 out of 222 students (86.49%) had a final score increase and the change is between 0.0679 and 3.

Despite the increase in final score for most students who participated in the modules, only a fraction of them would see an actual impact on their final letter grade. Table 7 shows the number and percentage of students whose letter grade improved after extra credits from modules were added in Fall 2019, Spring 2020, and Fall 2020. In each grade level, the percentage was calculated by number of students whose letter grade moved up to that level divided by the total number of students in the same level who took part in at least one of the modules (n) in that semester. In Fall 2019, among the 201 students who took part in at least one of the modules, 16 (7.96%) students' grades changed into higher ones after extra credits from the modules were added to the original final score. In comparison, 55 (35.03%) students received a higher letter grade in Spring 2020 and 52 (23.42%) in Fall 2020.

	FA19 (n=201)		SP20 (n=157)		FA20 (n=222)	
Grade Change	Number	%	Number	%	Number	%
A to A+	1	0.50	5	3.18	0	0
A- to A	0	0	9	5.73	12	5.41
B+ to A-	1	0.50	8	5.10	9	4.05
B to B+	3	1.49	9	5.73	7	3.15
B- to B	2	1	8	5.10	4	1.80
C+ to B-	3	1.49	1	0.64	9	4.05
C to C+	2	1	4	2.55	4	1.80
C- to C	3	1.49	3	1.91	1	0.45
D+ to C-	0	0	4	2.55	4	1.80
D to D+	1	0.50	3	1.91	0	0
D- to D	0	0	1	0.64	1	0.45
F to D-	0	0	0	0	1	0.45
Sum	16	7.96	55	35.03	52	23.42

Table 8: Number and percentage of students whose letter grade improved after extra credits from modules were added in Fall 2019, Spring 2020, and Fall 2020. n is the total number of students who took part in at least one of the modules.

Conclusion and Future Work

In summary, two optional extended learning modules (parallel adder and parallel counter) were developed and deployed in ECE 120 at University x. Participation rate was around 48% in Fall 2019 and grew to 60% in Fall 2020. Corresponding assessment results indicated a high level of understanding in the topics introduced, in which the assessment means were above 80% for the parallel adder module and above 88% for the parallel counter module. In general, students in Spring 2020 and Fall 2020 performed better than those in Fall 2019.

Assessment data of these modules was analyzed by categorizing students into three groups by their original course score. Those who would have received an A- or above are in Group 1, those between B+ and B- (inclusive) are in Group 2, and the rest are in Group 3. ANOVA tests were performed to examine whether there is a significant difference in scores between any two groups. We found that the students in Group 1 performed significantly better than those in Group 3 on the parallel adder module in Fall 2019 and Fall 2020, as well as the parallel counter module in Fall 2020. Significant difference is also found between Group 1 and Group 2, Group 2 and Group 3 for the parallel adder module in Fall 2019. We think the significant differences observed between groups in the fall semesters are due to the fact that most of the students in the fall semesters are freshmen. They are less prepared for their first computer engineering course than those who took it in the spring semester, who are generally sophomores. This is also supported by the Pearson correlation analysis, which shows that parallel adder module scores in the fall semesters have a moderate positive correlation with original final course scores, where it has a weak positive correlation in the spring semester.

When instruction was fully in-person, participation rate among the groups were at similar levels. As we switched to half-online and then fully-online instruction, participation rate dropped in Group 3 while increased in Group 1. Since the optional extended learning modules provided opportunities to earn extra-credit in the course, we also examined final score increase and letter grade improvement of participating students. Due to calculation of the extra-credit, not every student who participated will see a score increase. Overall, 80.60% students in Fall 2019 saw a score increase, 95.54% in Spring 2020 and 86.49% in Fall 2020. Out of those who had a score increase, 16 students (7.96%) received a higher letter grade in Fall 2019, 55 (35.03%) in Spring 2020 and 52 (23.42%) in Fall 2020. Letter grade improvement is observed in different grade levels, not just in the A and B ranges.

Based on wide participation in ECE 120, optional learning modules are also introduced in the subsequent courses. In ECE 220, a module is created on LC-3 multi-tasking and it consists of a set of slides and a video recording. Students will need to complete a timed assessment after they went through the materials. A graded programming assignment is also available for those who are interested to explore the topic further. In ECE 385, two modules are created: one examines the performance of students' implementation of Carry Look-ahead adder and Carry-select adders on top of a standard ripple adder; another one focuses on performance analysis of a simplified LC-3 (SLC-3) CPU. These modules were deployed in ECE 220 and ECE 385 starting in Fall 2020.

For future work, we plan to use short answer surveys to collect feedback to better understand students' motivation in completing these optional extended learning modules and whether their

participation has led to an increased interest in PDC topics. Surveys will be specifically designed for the introductory sequence of courses and the specialized parallel computing course.

References

[1] Mohamed Zahran and Marsha J. Berger. Parallel computing at the undergraduate level: Lessons learned and insights. In *Proceedings of the Workshop on Computer Architecture Education*, WCAE'19, New York, NY, USA, 2019. Association for Computing Machinery. ISBN 9781450368421. doi: 10.1145/3338698.3338889. URL https://doi.org/10.1145/3338698.3338889.

[2] C. M. Brown, Y. Lu, and S. Midkiff. Introducing parallel programming in undergraduate curriculum. In *2013 IEEE International Symposium on Parallel Distributed Processing, Workshops and Phd Forum*, pages 1269–1274, 2013. doi: 10.1109/IPDPSW.2013.270.

[3] M. Allen, B. Wilkinson, and J. Alley. Parallel programming for the millennium : Integration throughout the undergraduate curriculum. 1997.

[4] Suzanne J. Matthews, Joel C. Adams, Richard Brown, and Elizabeth Shoop. Incorporating parallel computing in the undergraduate computer science curriculum. In *Proceedings of the 51st ACM Technical Symposium on Computer Science Education*, SIGCSE '20, page 1399, New York, NY, USA, 2020. Association for Computing Machinery. ISBN 9781450367936. doi: 10.1145/3328778.3372511. URL https://doi.org/10.1145/3328778.3372511.

[5] Donald Johnson, David Kotz, and Fillia Makedon. Teaching parallel computing to freshmen. In Chris Nevison, editor, *Proceedings of the Conference on Parallel Computing for Undergraduates*. Colgate University, Colgate University, June 1994. URL https://www.cs.dartmouth.edu/ kotz/research/johnson-freshmen/index.html.

[6] Charles H. Roth, Larry L. Kinney, and Eugene B. John. *Fundamentals of logic design*. Cengage Learning Aisa Pte Ltd, 2021.

[7] Sushil K. Prasad, Ramachandran Vaidyanathan, Jerry L. Trahan, and Suresh Rai. *Chapter 5 Introducing Parallel and Distributed Computing Concepts in Digital Logic*, page 360. Elsevier, Morgan Kaufmann, 1st edition, 2015.

[8] Zeran Zhu. Carry lookahead adder slides, 2019. URL https://uofi.box.com/s/vsb4cuyzi8d1248ez6kpglyh1vy7ldly.

[9] Zeran Zhu. *Carry Lookahead Adder Video*. 2019. URL https://uofi.box.com/s/oznypil2etry1z3hj2tgahc02r97v0n5.

[10] Zeran Zhu. Counters slides, 2019. URL https://uofi.box.com/s/ye6r3xtx5fr454y9kqgjr4u9f4yh45ti.

[11] Zeran Zhu. *Counters Video*. 2019. URL https://uofi.box.com/s/bhw3i83gya58y73ywkn703lotvgea7j0.

[12] IBM Corp. IBM SPSS statistics for macintosh. URL https://www.ibm.com/products/spss-statistics.

[13] Bruce Ratner. The correlation coefficient: Its values range between+ 1/- 1, or do they? *Journal of targeting, measurement and analysis for marketing*, 17(2):139–142, 2009.