A Preliminary Pedagogical Comparison Study Using the Digital Logic Concept Inventory

Geoffrey L. Herman and Joseph Handzik
University of Illinois at Urbana-Champaign, glherman@illinois.edu, jhandz2@illinois.edu

Abstract – **Digital logic instructors have created a myriad of new, innovative teaching methods. Comparing the effectiveness of these methods is elusive because of differences in topic coverage and assessment content. We have created a standardized assessment tool to provide a means to directly compare student conceptual learning in different digital logic design courses – the Digital Logic Concept Inventory (DLCI). We have used the DLCI along with instructor interviews and analysis of student final exams to compare student learning in two courses at the University of Illinois at Urbana-Champaign. These courses differed in course content, teaching philosophy, and pedagogical technology. We provide evidence that the DLCI can provide useful insight into what techniques and methods improve student learning.**

Index Terms – Digital logic, misconceptions, assessment, teaching methods

INTRODUCTION AND BACKGROUND

The engineering education community continually develops new, innovative teaching methods for engaging students in the classroom. These methods attempt to increase student learning, satisfaction, and retention. While there are accepted means for measuring satisfaction and retention, reliably assessing achievement of student learning can sometimes be elusive and complicated. When using tools such as final exam scores or course grades, rigorous assessment of learning is elusive, because differences in topic coverage, curriculum and course goals, and exam content do not allow for direct comparison of two teaching methods. Because of these difficulties, computing educators have issued a general call for the adoption of assessment tools to critically evaluate new teaching methods [1], [2].

One way to compare the effectiveness of teaching methods is to measure students' overall conceptual knowledge. Increasing conceptual learning is important, because students who can organize facts and ideas within a consistent conceptual framework are able to learn new information quickly and are able to more easily apply what they know to new situations [3]. By accurately assessing conceptual understanding, instructors can match instruction to their students' learning needs. One solution for measuring students' conceptual understandings is a concept inventory (CI). CIs are standard assessment tools that evaluate how much a student's conceptual framework matches the accepted conceptual framework of a discipline.

The power of CIs to motivate the adoption of new teaching techniques has been shown in the physics education community through the success of the first CI – the Force Concept Inventory (FCI) [4]. The FCI provided evidence that interactive engagement pedagogies increased learning compared with traditional lectures at all levels of instruction of introductory physics [5]. Since those pedagogical studies took place, introductory physics education has undergone a revolution in instruction [6].

Digital logic design courses are taught using a rich variety of techniques. These techniques vary along a number of different bases: lab vs. lecture, simulation vs. FPGA vs. breadboard, self-paced vs. instructor-paced, industry processors vs. instructional processors, top-to-bottom vs. bottom-to-top, and others. Each of these approaches has its strengths and weaknesses (cost, time, student enjoyment, etc.), but rigorous comparison of student conceptual learning has rarely been used as a metric of comparison. To enable the comparison of conceptual learning, we have developed the Digital Logic Concept Inventory (DLCI). For those who are interested in how we developed the items and distractors, measured reliability, and assessed validity, we refer to our previous work [7]. We ask that all requests to see the DLCI items by directed to the first author so we can protect the integrity of the DLCI as we finish development. In this paper, we treat the DLCI as a completed instrument where Table IV presents the list of concepts assessed.

Because the DLCI is intended to be a tool that compares student conceptual learning in different courses, we have conducted a "proof of concept" pedagogical comparison study to demonstrate what can be learned from using the DLCI as an assessment tool. We compare and contrast what can be learned from studying two courses final exams as well as results from completing the DLCI. We specifically wanted to learn which method – qualitatively studying final exams or quantitatively examining DLCI results – was better at comparing student conceptual learning in two courses. We also hoped to investigate how these two methods could complement each other to better inform us about which pedagogies encouraged the development of misconceptions. We present what we learned from using the DLCI as part of a mixed-methods study to compare courses with slightly different pedagogical styles, course content, and teaching philosophies. The study uses grounded theory analysis of course examinations [8], interviews and informal conversations with the instructors, and results from four administrations of the DLCI [7].

SUBJECTS

The study compares two digital logic courses taught at the University of Illinois at Urbana-Champaign: one course taught by the computer science department (CS 231) and the other by the electrical and computer engineering department (ECE 290). Both courses are large lecture courses (~150 students each) taken during the sophomore year of their respective majors, are the first course on digital logic that the students will take, and are attended by students of similar ability and backgrounds. The two courses have different meeting time structures, use different software packages to simulate circuit design and administer homework, have minor variations in their topic coverage, and emphasize different learning goals.

Each course had a different instructor and each instructor taught their course for both semesters of our investigation. For the remainder of the paper, we refer to the courses as C1 and C2 with a semester prefix, for the sake of instructor and student privacy (See Table I).

TABLE I
ANONYMOUS COURSE NAMES

	Course 1	Course 2
Spring Semester	SpC1	SpC2
Fall Semester	FaC1	FaC2

Because students are rarely exposed to digital logic concepts prior to these courses, it is reasonable to assume that any digital logic knowledge that the students possess after taking these courses results from the instruction they have received. Given the similarities of the students, the comparison of any learning achieved during the two courses will largely be a reflection of the different pedagogies and course structures.

QUALITATIVE METHODOLOGY

Throughout the Fall semester, we collected midterm and final exam papers from C1 and C2. We examined these course artifacts using a grounded theory approach. A detailed description of our methodology is available in our previous work [9]. In addition to analyzing these classroom artifacts, we interviewed the instructors concerning their teaching goals and philosophies.

QUALITATIVE RESULTS

Our qualitative research revealed new student misconceptions and indentified what course content was taught, how that content was delivered, and how student understanding was assessed.

I. Teaching Goals and Philosophies

During interviews with the instructors, we learned that each course had different goals and different philosophies to match those goals. C1 students are expected to begin their first intensive design course while learning about digital logic and computer architecture. C2 students are expected to simply learn the basics of digital logic and computer architecture as a stepping stone to future coursework. The differences in philosophies were evidenced in the instructors' approaches toward teaching number representations. The C1 instructor wanted her students to know how the different number representations developed and understand the broader concepts of designing number bases, while the C2 instructor wanted her students to be able to recall the different bases and convert numbers between the different bases.

Our analysis of the exams and grading procedures confirmed these different emphases. C1 exams contained many more design questions, while C2 exams contained more conceptual, multiple-choice, and fill-in-the blank questions. We even observed that the courses tested students using the same questions in different formats. For example, C1 and C2 exams both required students to create a sequential "divide by 3" circuit. The C1 exam simply gave students the design specifications and asked them to make all design decisions of the circuit, while the C2 exam gave students the specifications and some design decisions already made before asking students to complete the design (e.g., fill in a next-state table and draw the circuit). We discovered that most C2 students correctly completed the circuit's design while most C1 students struggled to begin the design process even though they had a very similar problem (a divide by 5 circuit) on a homework assignment.

Both courses' exams contained problems that required simultaneous use of multiple concepts or tasks to solve the problems. These types of problems were used to test a variety of topics and appeared more commonly on C1 exams. For example, both courses tested students' knowledge of ROM implementations. The C1 exam problem required students to determine the size of the ROM and replace all logic in the circuit with the ROM (two tasks). The C2 exam problem gave students the size of the ROM, and required students only to replace all logic in the circuit with the ROM (one task). Students performed better on one task problems and performed worse as the number of simultaneous tasks increased. Even requiring two tasks caused students to leave answers blank at an alarming rate.

Despite these differences in teaching goals and philosophies, the course syllabi of C1 and C2 are fairly similar. Although the two courses have considerable overlap in concept coverage, they vary on a few topics. C1 covers topics such as parity bits and Hamming codes, coincident decoding of RAMs, and Mealy and Moore machines. C2 covers topics such as the universality of gates, active-low vs. active-high, and compilers.

II. Misconceptions

By analyzing the midterm and final exam papers, we found several new misconceptions. We present the two misconceptions that became the basis for new DLCI items.

We found the first misconception when examining student responses to a circuit analysis problem. The problem required students to complete a next-state table. Students typically found the correct solution to the problem, except that they would fill in one column incorrectly. The problematic column corresponded to properly analyzing the output of a three-input XOR gate. The two most common errors are shown in Figure 1.

FIGURE 1

$x\,y\,z$	$x \oplus y \oplus z$	$x \oplus y \oplus z$	$x \oplus y \oplus z$
0 0 0	0	0	0
0 0 1	1	1	1
0 1 0	1	1	1
0 1 1	0	0	1
1 0 0	1	1	1
1 0 1	0	0	1
1 1 0	0	0	1
1 1 1	1	0	0

CONCEPTIONS OF THREE-INPUT XOR GATES. THE FIRST COLUMN IS THE CORRECT CONCEPTION. THE OTHER COLUMNS ARE MISCONCEPTIONS

Because students struggled to fill-in only this column of the next-state table we are confident that the students knew how to complete a next-state table and had misconceptions about only three-input XOR gates. The XOR operator can be defined using a two-part definition “(a) one input variable is 1, (b) but not both.” Student misconceptions corresponded to two over-generalizations of this intuitive definition. The third column of Figure 1 corresponds to a misconception that the three-input XOR is 1 when only one input variable is1 (over-emphasis on (a)). The last column corresponds to the misconception that the three-input XOR is 1 when one or two input variables is 1 but not all three (over-emphasis on (b)). Students manifested these misconceptions even though they were given the property that $p \oplus s \oplus o = (p \oplus s) \oplus o$. C1 students did not manifest the XOR misconceptions, but many C2 students manifested these misconceptions.

We found a second misconception when students struggled to interpret bitwise manipulations of two’s complement arithmetic such as $a_2\, a_1\, a_0 + 1\,1\,1$ and $a_2\, a_1\, a_0 + \overline{b_2}\,\overline{b_1}\,\overline{b_0}$. These manipulations should be interpreted as A-1 and A+B-1 respectively (when A=$a_2\, a_1\, a_0$ and B=$b_2\, b_1\, b_0$). Several students interpreted the first manipulation to be A+1 or –A and the second manipulation to be A-B or A-B+1. The A-B+1 mistake may be a computational mistake, but there are no computational mistakes that can yield the other three mistakes. We cannot offer a full explanation of these misconceptions without interviewing students to elicit their reasoning.

C1 students were not tested on bitwise concepts. We could not compare the two courses misconceptions.

QUANTITATIVE METHODOLOGY

We have administered the DLCI to C1 and C2 for two semesters. SpC1 and SpC2 both offered credit for completing the DLCI. SpC1 made the DLCI required and SpC2 made the DLCI an extra credit assignment. During the second semester, FaC1 still made the DLCI a required part of the course, while FaC2 did not offer credit for completing the DLCI. Because the different types of credit could affect the level of student effort, we decided to test if these varying credit options created bias between the administrations. We also tested for bias on the instrument as a whole, so we could more accurately compare student comprehension of individual topics. If one course performed significantly better on the DLCI than the other course then we must be more cautious in determining how pedagogies affect learning.

Between the first semester and the second semester, we added and removed items from the DLCI. Because we changed the DLCI, we constructed a subtest of 12 items that were present on all versions to compare the performance between the fall and spring courses. We hypothesized that if the performance between courses and semesters was the same then we should not see a significant difference between the scores. We tested for statistical significance using 2-sided *t*-tests and considered *p*-values less than 0.05 to be significant.

To test for significant differences in performance on individual questions, we used *t*-tests. The *p*-values used for significant differences for these tests would depend on the presence of bias between the administrations. If bias was present in the overall administration, then we would use smaller *p*-values to test for significance.

QUANTITATIVE RESULTS

Table II presents the *p*-values from the t-tests that compared the performance on the DLCI subtest. We found no evidence of bias between any of the DLCI administrations that offered credit for completing the DLCI. We did find evidence of bias between the administration that offered no credit and all administrations that offered credit.

TABLE II
P-VALUES FROM *T*-TESTS COMPARING PERFORMANCE ON THE DLCI SUBTEST (*DENOTES SIGNIFICANT DIFFERENCES WHEN $P<0.05$).
(** DENOTES WHICH COURSE OFFERED CREDIT FOR THE DLCI)

	SpC1**	SpC2**	FaC1**	FaC2
SpC1	1.000			
SpC2	0.628	1.000		
FaC1	0.326	0.155	1.000	
FaC2	0.011*	0.036*	0.001*	1.000

These results suggest that students who did not receive credit for completing the DLCI performed worse because of the lack of credit. We believe that the lack of credit translated into a lack of motivation. This lack of motivation was also evidenced by the difference in DLCI completion rates for the Fall courses: 99% for FaC1 students versus less than 50% for FaC2 students.

Because no credit reduced motivation and scores for students in FaC2, we tested for significant differences in performance on individual items using small *p*-values. We decided that *p*-values less than 0.01 indicate marginal significance and *p*-values less than 0.005 indicate significance. If a *p*-value revealed a significant difference in learning between the two courses, we tried to determine what caused the change. If several questions that covered similar concepts exhibited marginal significance, we would investigate what caused this change in learning. Our follow-up investigations needed to determine if these significant differences in performance resulted from differences in motivation or differences in instruction.

Table III presents the descriptive statistics of FaC1's and FaC2's overall performance on the DLCI. Table IV lists the concepts tested by each DLCI item and highlights which concepts presented a significant difference in performance on individual items. FaC1 performed significantly better on all highlighted items. The overall administration bias made it difficult to determine on which items FaC2 performed better relative to their overall performance.

TABLE III
PERFORMANCE METRICS ON THE DLCI FOR FAC1 AND FAC2.

FaC1	Metric	FaC2
108	N	60
14.4	Mean	11.1
3.7	STD DEV	4.1
14	Median	10
4	Minimum score	4
22	Maximum score	21
Normal	Distribution	Normal

TABLE IV
LIST OF DLCI ITEMS AND THE CONCEPT TESTED IN EACH ITEM.
HIGHLIGHTED CONCEPTS SHOWED A SIGNIFICANT DIFFERENCE IN PERFORMANCE BETWEEN FAC1 AND FAC2
(* DENOTES MARGINAL SIGNIFICANCE, ** DENOTES SIGNIFICANCE)

Item #	Concept
1	Boolean operators (negated variables)
2	Relationship between states and flip-flops**
3	Number bases
4	Decoders and multiplexers
5	Boolean operators (3-variable XOR)**
6	Relationship between states and flip-flops
7	Time dependence of state
8	Boolean operators (negated variables)
9	Boolean operators (NAND and NOR)
10	Two's complement representation*
11	Decoders and multiplexers (select bits)**
12	RAM inputs and outputs*
13	RAM inputs and outputs**
14	Two's complement representation
15	Time dependence of state
16	Number bases
17	Relationship between states and flip-flops**
18	Relationship between states and flip-flops
19	Boolean operators (NAND and NOR)
20	Boolean operators (3-variable XOR)
21	Underspecified Boolean functions
22	Two's complement representation
23	Two's complement representation**

DISCUSSION

We found that the mixed method nature of this study provided a clear framework to interpret the results of our work. The DLCI results highlighted which concepts were understood differently by the two courses. These highlighted concepts helped us find the pedagogies that might be responsible for the differences in understanding.

Before we describe how we think that C1 pedagogies improved performance on some items, we want to emphasize that there was no significant difference in overall performance between the courses and instructors during the Spring semester. Some of the items that show a significant difference in performance during the Fall administration did not show significant differences during the Spring administration (e.g., items 2, 10, and 12). The difference in course credit offerings may have exaggerated the differences in conceptual understanding of individual topics.

I. Three Input XOR

We believe that the FaC1 students understand the three-input XOR better than the FaC2 because of the way that the XOR concept is introduced. The XOR concept is introduced in C2 during Boolean translation instruction while the XOR concept is introduced in C1 during parity bit checking instruction (a topic not taught in C2). During parity bit checking instruction, students are taught to think of the XOR as an odd/even detection circuit – XOR is 1 when it receives an odd number of 1 inputs. During Boolean translation instruction, students are taught to think of the XOR as "one or the other input variables is 1, but not both." C1 students are also taught this translation definition later in the course.

The C1 students are taught a universally applicable definition of XOR that can easily scale to any number of inputs without additional cognitive effort. The C2 students are taught a logical definition of XOR that is accurate, but is difficult to scale to any number of inputs without requiring the student to complete two tasks (XOR two input variables and then XOR the result of the first XOR operation with the third input variable).

The additional number of tasks needed to correctly interpret the three-input XOR may discourage students from being systematic and encourage incorrectly generalizing the "one or the other is 1, but not both" definition. Naively over-generalizing that definition could yield either the "only one of the input variables is 1" or "one or two of the input variables is 1, but not all three" misconceptions. By teaching XOR as an odd/even detection circuit, C1 students never needed to make a naive over-generalization to reduce their cognitive load.

We are confident in this assertion because three times as many C2 students chose the wrong interpretation of XOR. We also observed that three times as many C2 students chose the three-input XOR distractor on Item 20 which asks students to translate a statement that says "[a sandwich must] have bacon, lettuce, or tomatoes, but not all three" into a Boolean expression. This distractor matches with the over-generalization that we described earlier and it was more attractive to C2 students. We also observed that more C2 students answered Item 5 wrong *and* chose the three-input XOR distractor on Item 20. Finally, our qualitative observations of exams corroborate our hypothesis.

II. Two's Complement

We observed a significant difference in understanding on two items pertaining to two's complement representation (marginal significance on item 10 and significance on item 23). Item 10 asks students to choose the best explanation for *why* computers use two's complement representation of numbers. Item 23 asks students to complete the $a_2\,a_1\,a_0 + \overline{b_2}\,\overline{b_1}\,\overline{b_0}$ two's complement bit-wise manipulation.

We were surprised by these significant differences because both courses introduce two's complement number representation in a similar fashion. Both courses teach signed magnitude binary, then one's complement representation, and then two's complement representation. Both teach two's complement as an introduction to subtraction circuits.

We do not think that the difference in performance is due to instruction, but rather due to the difference in credit offered. First, the difference in performance on Item 10 was marginally significant and therefore not enough to warrant definitive conclusions. Second, C2 students performed better (non-significantly) on the other two's complement item – Item 14 – and comparably on the other bit-wise manipulation item – Item 22. Third, the final three items were the most frequently unanswered items from the FaC2 administration (note: our statistical analysis included only those responses with at most one unanswered item). The lower completion rate of these items is probably due to the lower motivation. Since Item 23 was the last item, we suspect that the lack of motivation resulted in more guessing or hurried analysis as students ran out of time. This increased guessing may have caused the significant difference in performance.

III. Information Encoding

The remaining topics that showed significant differences in performance are the "Relationship between states and flip-flops," "Multiplexer (MUX) select bit assignment," "RAM inputs and outputs," (Items 2, 11, 12, 13 and 17) These topics seem to cover three distinct concepts (state and flip-flops, MUXes, and RAM). Deeper inspection reveals another grouping for these distinct concepts – information encoding.

To understand the relationship between states and flip-flops, a student must understand that the circuit's state is encoded in the individual bits of the flip-flops. To understand how to assign select bits in a MUX, a student must understand that the binary value of the selected input of the MUX is encoded in the binary ordering of the select bits. To understand the difference between RAM address lines and RAM data inputs/outputs, a student must understand what type of information can be encoded in bits and what type of information cannot be encoded in bits. All three concepts require the student to understand how and why information is encoded into bits. These five items all focus on how to encode information in their distinct contexts, and they are the only five items on the DLCI that assess information encoding.

Both C1 and C2 cover the "Relationship between states and flip-flops," "Multiplexer (MUX) select bit assignment," "RAM inputs and outputs" with similar techniques: similar assignments, similar lectures and examples, and introduced in similar contexts. These similarities of technique meant that in order to find what may have caused the difference in performance, we would have to look elsewhere in the course for the cause. A survey of the two syllabi revealed one plausible explanation for the difference in performance.

At the beginning of the semester, C1 spends a week emphasizing how and why information is encoded into bits. The course presents not only standard numerical binary encodings (e.g., two's complement and unsigned binary), but weighted binary codes, 2 of 5 binary postal codes, and Hamming codes as well. C2 students simply begin with binary numbers and how to convert decimal numbers to binary. The additional emphasis on the idea that unsigned binary is one means of encoding numerical information rather than simply a new number base may help students to think about flip-flops, select bits, and address lines as information encoding bits.

If C1 students learn to think about unsigned binary as one method for encoding information and C2 students do not learn to think this way, C1 students would be able to more easily learn information encoding for states and flip-flops, MUX select bits, and RAM addressing. While C1 students need to learn how to apply the idea of information encoding to three new contexts, C2 students may not perceive any connection between the different contexts. Learning to apply a general concept into three new contexts is much easier than learning three, new unrelated concepts.

In our misconceptions research to develop the DLCI [9], [10], we have found that students try to apply information encoding schemes in different ways in different contexts. When encoding circuit state information in flip-flops, many students wrongfully believe that adding one flip-flop to a circuit allows the circuit to enter one more state (linear relationship) rather than doubling the number of states that the circuit can enter (exponential relationship). Students have yet to show that they believe that there is a linear relationship between the number of select bit inputs and the number of data inputs on a MUX. When working with MUXes, Students struggle to conceptualize how changing the value of one select bit of a MUX will change which data input is selected. When working with state, they do not struggle to understand how changing one flip-flop value can change the state of the circuit.

CONCLUSIONS AND FUTURE WORK

The DLCI is a better tool for comparing student conceptual learning in different courses than using normal classroom artifacts such as exam papers, and we believe that the DLCI can be used as a means to rigorously compare student learning in different courses. While the DLCI did not reveal a difference in overall learning in the two courses, it revealed differences in student learning on individual topics. We believe that the two courses had similar DLCI scores, because the pedagogies were sufficiently similar (i.e., lecture-based courses). DLCI data combined with qualitative analysis of teaching methods provides a new tool to assess whether certain pedagogies can improve student learning or remedy student misconceptions.

Our study of student exam papers, though, was not conducive to comparing student learning in the two courses. Studying exam papers was time consuming and did not allow direct comparison of learning in most cases. Students were either tested on different content or were tested on the same content in very different styles. Study of exam papers was useful for discovering new misconceptions and discovering what made certain styles of questions difficult.

We discovered two new misconceptions that we believe are commonly held. We found that students wrongfully generalize the two input XOR to the three input case and we found that students have difficulty interpreting bit-wise manipulations of numbers. We encourage instructors to teach the XOR concept as an even/odd detection circuit in addition to teaching it is as “one or the other input variables is 1, but not both.” We will investigate the bit-wise manipulation misconceptions in future work.

Our analysis of exam papers revealed that when the number of separate cognitive tasks required to solve a problem increases, students perform worse. Students perform worse even when the additional tasks would be deemed very easy by any expert. We plan to investigate this phenomenon in future research as well.

Cognitive research has shown that experts in a field are able to apply their knowledge in a domain more effectively because they have “chunked” distinct situations and problems into meaningful groups. Once a concept or process becomes chunked, experts often fail to realize that they are using the chunked concepts and processes and find it difficult to explain how to use the chunked information to a novice. Based on this previous research and our findings, we tentatively encourage instructors to carefully count the number of tasks a problem requires to gauge how difficult the problem will be for students and to be mindful of when they are manipulating multiple concepts at a time.

The cognitive chunking process may also cause other difficulties for digital logic students. First, instructors may not be aware of how they have chunked certain information together. Digital logic instructors likely see and use the underlying concept of information encoding all the time when manipulating flip-flops, RAM, and MUXes, but may not be aware that they are using the concept. We encourage instructors to explicitly discuss information encoding with students and help them to apply this single concept to the many contexts that require it. This additional emphasis probably does not require instruction on additional coding schemes such as Hamming codes or binary weighted codes.

Second, information encoding is inherently a method for chunking small pieces of information into meaningful groups. The ability to see individual bits in meaningful chunks takes time and repetition, but it can be guided. Students should receive considerable practice learning to see bits as chunks of information rather than seeing bits as relatively meaningless numbers.

The DLCI provides an easy-to-use powerful tool that can quickly encode information about student conceptual knowledge into manageable numerical data. This numerical data revealed the importance of the information encoding concept in digital logic. The DLCI showed that students who learn information encoding better, can better learn a number of seemingly distinct concepts.

The DLCI alone is not sufficient to fully assess student learning in digital logic. Other instruments need to be developed to assess students’ digital logic design skills and problem solving abilities. The DLCI tests only a core subset of digital logic concepts and cannot be used as a final exam.

ACKNOWLEDGMENTS

This work was supported by the National Science Foundation under DUE-0618589. The opinions, findings, and conclusions do not necessarily reflect the views of the National Science Foundation or the University of Illinois.

We thank the instructors of ECE 290 and CS 231 for generously allowing us to test and interview their students.

REFERENCES

[1] M. Ben-Ari. The Concorde doesn’t fly anymore. In *the Technical Symposium on Computer Science Education (SIGCSE), keynote address*, Feb 2005.

[2] Lister, Raymond, "After the Gold Rush: Toward Sustainable Scholarship in Computing." in *the Proceedings of the Tenth Australasian Computing Education Conference (ACE2008)*, pp. 3-17, Wollongong, Australia, January 2008.

[3] J. D. Bransford, A. L. Brown, and R. R. Cocking. *How People Learn: Brain, Mind, Experience, and School*. National Academy Press, Washington, DC, 1999.

[4] D. Hestenes, M. Wells, and G. Swackhamer. “Force Concept Inventory,” *The Physics Teacher*, 30:141–158, 1992.

[5] R. Hake, “Interactive-engagement vs. traditional methods: A Six-thousand-student Survey of Mechanics Test Data for Introductory Physics Courses,” *Am. J. Physics, 66:64–74*, 1998.

[6] J. P. Mestre, “Facts and myths about pedagogies of engagement in science learning,” *Peer Review*, vol. 7, no. 2, pp. 24–27, Winter 2005.

[7] G. L. Herman, M.C. Loui, and C. Zilles, “Creating the Digital Logic Concept Inventory,” in *Proceedings of the 41st Annual ACM Technical Symposium on Computer Science Education*. ACM SIGCSE, March 2010, (to appear).

[8] A. Strauss and J. Corbin, *Basics of Qualitative Research*. Thousand Oaks, CA: Sage, 1998.

[9] G. L. Herman, L. Kaczmarczyk, M. C. Loui, and C. Zilles, “Proof by incomplete enumeration and other logical misconceptions,” in *Proceedings of the Fourth International Computing Education Research Workshop (ICER 2008)*, Sydney, Australia, September 6-7, pp.59-70, 2008.

[10] G. L. Herman, C. Zilles, and M. C. Loui, “Work in Progress: Students’ Misconceptions About State in Digital Systems,” in *Proceedings of the Frontiers in Education Conference 2009*, pp. T4D-1-2, San Antonio, TX, October 18-21, 2009.