BOOLEAN BLUNDERS:
IDENTIFICATION AND ASSESSMENT OF STUDENT MISCONCEPTIONS IN A DIGITAL LOGIC COURSE

BY

JAMES T. LONGINO

B.S., The University of Texas at Austin, 2004

THESIS

Submitted in partial fulfillment of the requirements
for the degree of Master's of Science in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Urbana-Champaign, 2006

Urbana, Illinois

Put cert. of committee approval here

ABSTRACT

Many engineering and computer science instructors desire to improve their teaching, to meet ABET accreditation requirements, and to address the need for quantitative evaluation of pedagogy. In order to meet these goals, instructors require effective, concise assessment tools such as concept inventories.

Concept inventories are proven tools for classroom assessment. The Force Concept Inventory (FCI) has been used to successfully determine the effectiveness of active engagement techniques in introductory physics courses. Despite the critiques of the FCI, instructors are working to develop concept inventories in other engineering and science fields.

To identify student misconceptions in digital logic, we interviewed 16 students who had previously completed either ECE 290 or CS 231. We then created a multiple-choice problem for each misconception. After tuning the problems, we compiled them into a draft assessment test, which we administered to over 200 ECE 290 and CS 231 students.

Unsurprisingly, students had the most difficulty with concepts involving abstraction and with transferring concepts to different contexts. We could assess many of these misconceptions using multiple-choice problems, and found that students who took the assessment tests showed the same misconceptions as the students we interviewed. We also found that students in CS 231 performed similarly to students in ECE 290, despite pedagogical differences between the two courses.

.

ACKNOWLEDGMENTS

I am grateful to the College of Engineering and the Department of Computer Science at the University of Illinois at Urbana-Champaign for providing financial assistance to support my research.

I would like to thank my graduate advisers, Professors Craig Zilles and Michael C. Loui. More than just two professors interested in education research, you provided an excellent mix of guidance, freedom, motivation, and inspiration. While many students would dread the thought of being co-advised, I found the insight and advice that you provided to be complementary instead of contradictory. My work would not be nearly as successful as it was without both of you. Special thanks to Michael C. Loui for your help in revising this thesis. Your comments and suggestions greatly enhanced the quality of my writing.

I would like to thank Donna Brown and the ECE 290 staff of the 2005 – 2006 academic year. Your class time is valuable, and I appreciate that you let me use some of it for my work.

Thanks to the ECE 290 and CS 231 students who volunteered to be interviewed and who allowed their responses to be used in my work.

Of course, my deepest thanks to my family and friends. None of this would be possible without your love and support.

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

Engineering and computer science instructors have many reasons to assess student learning outcomes. First, the program accreditation standards of the Accreditation Board for Engineering and Technology (ABET) requires “a system of ongoing evaluation that demonstrates achievement of these objectives and uses the results to improve the effectiveness of the program” [1]. Second, instructors who want to improve their teaching effectiveness require an objective, reliable tool to evaluate the effects of different teaching methods. Third, education researchers need a standard tool to compare pedagogies. I believe that classroom assessment can be used to achieve institutional, faculty, and research assessment objectives [2].

In this thesis, I describe initial steps toward a concept inventory to assess student understanding of digital logic. A concept inventory is a short, multiple-choice test that assesses students’ conceptual understanding of a field. Concept inventories have proven valuable in science and engineering. In particular, the Force Concept Inventory (FCI), which probes conceptual understanding of Newtonian mechanics, has revolutionized the teaching of introductory physics courses. Chapter 2 of this thesis describes the history and impact of concept inventories; it also describes one method used to develop concept inventories.

We conducted our research at the University of Illinois at Urbana-Champaign, which offers two introductory logic design courses. ECE 290, Computer Engineering I, is offered by the Department of Electrical and Computer Engineering, and CS 231, Computer Architecture I, is offered by the Department of Computer Science. Both courses are sophomore-level courses that are required for the departments’ undergraduate majors. The two courses cover almost identical material: representation of information, combinational and sequential circuit analysis and design, and computer organization and control. Both courses use the same textbook [3], but they are not jointly offered, primarily due to their large enrollments of about 200 ECE 290 students and 120 CS 231 students per semester. Most of the students are traditional age, residential students. About 90% of ECE 290 and CS 231 students are male, and most are white or Asian. In both courses, students attend three hours of class every week: two hour-long lectures taught by faculty, and one hour-long recitation taught by a graduate teaching assistant. Students in both courses complete bi-weekly schematic-capture and simulation-based laboratory assignments. Students submit written homework every other week for CS 231 and every week for ECE 290. In addition, students complete frequent Web-based homework problems and receive instant feedback. During the semester, students take two mid-term exams and a comprehensive final exam. Further information can be found at http://courses.ece.uiuc.edu/ECE 290 and http://www.cs.uiuc.edu/class/CS 231.

To develop a concept inventory, we needed to identify which concepts students find difficult to understand and to determine why students find those concepts difficult. During the 2005–2006 academic year, we identified student misconceptions in digital logic through a series of 16 interviews with students who had completed either CS 231 or ECE 290. To check whether other students held these misconceptions, we developed and administered two multiple-choice assessment tests; the first test was administered to 28 ECE 290 students at the end of Fall 2005, and the second test was administered to over 200 ECE 290 and CS 231 students at the end of Spring 2006. I describe the method we used to identify student misconceptions in Chapter 3 of this thesis.

Our first research goal was to identify common student misconceptions in digital logic. As we identify these misconceptions, instructors of introductory digital logic will know which concepts might require more explanation. Further, instructors of subsequent courses will know which concepts they might need to reinforce for students.

Our second research goal was to determine whether the different pedagogies used in ECE 290 and CS 231 affect students' conceptual understanding of course material. ECE 290 instructors use rigorous and time-consuming problem sets to "drill" concepts, whereas CS 231 instructors believe that complex, repetitious problem sets are not needed for students to gain a conceptual understanding of digital logic.

Student interviews and the assessment tests revealed a number of misconceptions about digital logic:

- Students are often content to have a level of understanding consistent with the "Remember" level rather than the "Apply" level of the revised Bloom's Taxonomy of Anderson et al. [4].

- Students are uncertain about the output dependencies of different logic devices such as latches, flip-flops, and combinational logic gates.
- Students have difficulty applying previous knowledge to new situations.
- Students have difficulty designing with medium-scale integration (MSI) components such as multiplexers and decoders.
- While students do understand some of the implications of complements and duals of Boolean expressions, they often cannot derive the complement and the dual of a particular expression.
- Many students do not understand the concept of state in sequential design. They do not always know how to describe a state or how to tell when transitions between states occur. Students also have difficulty distinguishing an abstract state machine from specific implementations of that state machine.
- When they analyze a complex circuit, students rarely abstract functionality from subsystems of the circuit.
- Students are sometimes uncertain about when combinational circuits should be used and when sequential circuits should be used.

Despite significant differences between CS 231 students' performance and ECE 290 students' performance on a few problems on the Spring 2006 assessment test, we found that average assessment test scores were not significantly affected by course choice.

In Chapter 4 of this thesis, I discuss student misconceptions, student performance on the Spring 2006 assessment test, and the differences between CS 231 and ECE 290 student performance.

Although we did not develop a finished concept inventory in digital logic, our work represents a promising first step. We identified a number of student misconceptions in digital logic and developed a draft assessment test that can be the basis for future work. Much work remains before we create a valid, reliable concept inventory, however. In Chapter 5 of this thesis, I describe some of the lessons that future concept inventory developers could learn from our research as well as future work needed for a concept inventory in digital logic.

CHAPTER 2 BACKGROUND

In this chapter, I review the literature on the development and use of concept inventories in science and engineering. I will describe the history, critiques, and impact of the Force Concept Inventory (FCI), and I will discuss recent attempts to construct concept inventories in other fields.

2.1 The Force Concept Inventory

Hestenes et al. [5] developed the FCI to demonstrate the extent to which students became “Newtonian thinkers” after formal instruction in Newtonian mechanics. Hestenes et al. identified six different dimensions of the Newtonian force concept. To test each dimension, they created several multiple-choice problems. They used 29 of these problems in the original version of the FCI (the current version has 30 problems). The problems require little or no calculation, and each problem focuses on only one concept. Each problem contains one correct answer choice based on Newtonian understanding and four alternatives based on commonly held misconceptions. Although physics professors who had reviewed the inventory considered the problems too trivial to be useful, initial administrations of the FCI showed that students did not understand these basic concepts as well as their professors had hoped. Early data showed that students who had completed a semester of introductory Newtonian physics had average FCI scores between 63% and 77% correct.

2.1.1 Critiques of the FCI

The earliest critique of the FCI focused on the division of the concept of Newtonian force into six dimensions. Huffman and Heller [6] performed a factor analysis on the student FCI scores reported by Hestenes et al. and found that students did not conceive of force in the six dimensions Hestenes had described. Because the FCI did not measure the concept of force in the way that students conceived of force, Huffman and Heller believed that the FCI was not a good measure of student understanding. Hestenes and Halloun [7] replied that true Newtonian thinkers did conceive of force in those dimensions, and that the correct interpretation of the factor analysis was that students did not think of force in the same way as Newtonian thinkers. They believed that FCI results were valid because the FCI measured the difference between student thought and Newtonian thought. This disagreement was never fully resolved and was left as a difference in interpretation of FCI results. The disagreement highlights how difficult it is to create a concept inventory that assesses conceptual understanding in a manner that is acceptable to all educators.

Another set of critiques focused on the multiple-choice format of the FCI. Although the FCI was designed to reduce the number of false positives (non-Newtonian thinkers selecting Newtonian answer choices) and false negatives (Newtonian thinkers selecting non-Newtonian answer choices), some educators worried that contexts of the problems and the candidate answer choices could affect test results. Mahajan [8] found that students with significant physics experience might select the correct answer choice only after great deliberation and without being fully convinced of their answer. His administration of the FCI to 10 physics students resulted in very high scores with an average of about 90%. However, he found many students revealed non-Newtonian thinking when the problems were recast as circular motion problems. Steinberg and Sabella [9] also found that student performance depended on the context of the problem. In their study, students performed better with open-ended problems testing the same physics concepts as FCI problems. Rebello and Zollman [10] examined the effects of the distracters on students. They created open-ended problems from FCI problems by removing the choices, and they compared student open-ended responses with the choices offered in the original FCI problem. They found that the students correctly answered a similar number of problems on both the multiple-choice format and the open-ended format, but that students' incorrect solutions on the open-ended format did not align to FCI distracters.

Finally, some researchers have found that student background affects FCI results. Mahajan's students, who had more physics background than the average American physics students, answered correctly even when they did not fully understand the concepts behind their answers [8]. McCullough [11] found that gender also affected students' responses. She found that changing some of the stereotypically male contexts of the problems (cannons, rockets, hockey pucks, etc.) to stereotypical female contexts (shopping, family, etc.) affected students' responses to those problems. However, the effects were not predictable. Cultural context also affects students' responses, as Schecker and Gerdes [12] show. Simply changing the context of a problem from golf to soccer can affect student responses.

No assessment tool is appropriate for all students in all cases. While there is still work to be done, few educators familiar with the FCI can doubt its impact on physics education.

2.1.2 Impact of the FCI

The developers of the FCI suggested three uses for the inventory [5]. First, the FCI could be given early in the course, to show professors which concepts were misunderstood by students. Second, it could be used as a placement exam. However, because the FCI is not designed to test students' ability to perform physics calculations, Hestenes et al. suggested that students also take a math test to determine whether they have the necessary background for a more advanced course. Third, the FCI could be used to evaluate instruction. The inventory could be administered to students at the beginning and the end of the semester, and the gain in scores should relate to the effectiveness of instruction. It is with this third use that the FCI has had the most impact on physics education.

From the reports of its initial administrations, the FCI began to point out the shortcomings of traditional, lecture-based physics education [5]. Posttest scores were much lower than many instructors had expected. Some instructors believed that their students had low posttest scores only because their students had little previous physics experience. However, FCI data showed that pretest scores varied little with previous physics experience, so it was more likely that low posttest scores were a result of teaching methods. Many educators were surprised by the small differences between pre- and posttest scores. In some courses, students' posttest scores were only 11% higher than their pretest scores. However, the number of students tested was still small. A larger dataset was needed to give educators a baseline measure for the effectiveness of their own teaching.

Hake ([13-15], summary in [16]) surveyed 62 introductory physics courses enrolling 6542 high school, college, and university students. He examined whether teaching methods based on student-student and student-teacher interaction (interactive engagement methods) resulted in more effective student learning than teaching methods based on passive-student lectures. Hake calculated a student's relative gain by dividing the difference between pre- and posttest scores by the difference between the pre-test score and the highest possible score. He then used the average relative gain for all students in a course to measure the effectiveness of instruction. Hake found that the average relative gain in courses using interactive engagement methods was more than two standard deviations greater than the average relative gain in courses using traditional lecture methods. Further, Hake found that high FCI gains had a strong positive correlation with the Mechanics Baseline Test [17], a test designed to measure problem solving ability rather than conceptual understanding.

Educators continued using the FCI to conduct research into teaching methods at multiple institutions. These institutions include the University of Maryland [18-21], the University of Montana [22], Rennselaer and Tufts [23], North Carolina State University [24], Carnegie Mellon University [25, 26], The Ohio State University [27], and upper secondary schools in Finland [28]. These studies have provided statistical evidence to support interactive engagement methods. Some researchers believe that, without the evidence proved by a simple, effective assessment tool like the FCI, physics instructors would not realize the increase in student learning that results from interactive engagement methods [29] and thus would not have seriously considered interactive engagement methods.

2.2 Other Concept Inventories

In an attempt to replicate the success of the FCI and produce similar pedagogical change in other disciplines, concept inventories are being developed for other science and engineering fields. Many of the teams developing concept inventories are working as part of the Foundation Coalition [30] to share best practices and disseminate information [31]. Concept inventories are being developed in chemistry [32, 33], dynamics [34, 35], electricity and magnetism [36-39], fluid mechanics [40], heat transfer and thermodynamics [41-43], materials [44-47], signals and systems [48-51], statics [52-54], and statistics [55, 56].

We are aware of only one prior attempt to develop a concept inventory for digital logic. This concept inventory, developed at The University of Massachusetts at Dartmouth and The University of Alabama, consists of 13 problems over seven different topics [57]. However, this concept inventory remains incomplete. Further, some problems cover computer organization, a topic not covered in all introductory digital logic courses. Other problems rely too deeply on formalism such as register transfer notation, or on algorithms such as binary to decimal conversion that can be applied without a firm understanding of the underlying concepts.

2.3 Concept Inventory Development

Richardson [58] describes concept inventory development as an iterative process consisting of five steps: determine the concepts to be included in the inventory; study and articulate the student learning process regarding those concepts; construct several multiple choice problems for each concept; administer the inventory and perform statistical analysis to determine validity, reliability, and fairness; and revise the inventory to improve readability, validity, reliability, and fairness.

In order for a concept inventory to be useful in evaluating teaching methods, it must assess the student's understanding of a set of concepts that is widely accepted as foundational knowledge in the tested field. The creation of this set of concepts can be particularly difficult if similar courses at different institutions cover significantly different topics. Including all possible concepts would artificially lower inventory scores and would cause the inventory to take too much time for students to complete. Good concept sets should arise from the agreement of multiple experts. Some professional organizations, such as the Association for Computing Machinery, have developed "model curricula" which could provide the topic set in those fields [59]. For fields without model curricula, concept inventory developers would work with experts both in the university and industry to define a set of topics. One method mentioned in concept inventory literature [60] is the Delphi method. To follow the Delphi method, concept inventory developers first select a panel of experts in the field the concept inventory will test. The developers then send the experts a survey asking them to submit important concepts. These experts anonymously submit their concept lists to the developers, who compile the data and then send an aggregated concept list and relevant statistics back to the experts for comment. This process repeats until the results stabilize.

After concept inventory developers have a list of concepts, they determine how students think about and learn those concepts. One way developers can determine how students think about concepts is through interactions with students. These interactions could include individual interviews, focus groups, and surveys. Developers can synthesize the data obtained by these methods to devise a comprehensive model of student understanding. Knowledge of cognitive psychology and consultation with cognitive psychologists can increase the effectiveness of this step.

Developers who understand the nature and causes of student misconceptions can then construct multiple-choice problems. Ideally, for each concept, developers should construct several problems to reduce the likelihood of false positives because students will be less likely to guess correct answers for all of the problems on that concept. Distracters can be created by examining student responses to open-ended versions of the problems, [5, 31]. The interpretation of student performance is simplified when each problem focuses on only one concept.

Developers should then administer the inventory to as many students as possible and should analyze the data to determine the validity, reliability, and fairness of the inventory [56]. A *valid* concept inventory measures what it claims to measure. A *reliable* inventory would result in similar scores for similar populations. A *fair* inventory would result in comparably valid scores regardless of the student's gender or ethnic group. Because the analysis of validity, reliability, and fairness requires sophisticated statistical methods, concept inventory developers may need the assistance of an expert in psychometric analysis. Statistical analysis of a concept inventory might reveal changes that need to be made in inventory problems or answer choices. The developers then repeat the process of administration and analysis with a revised inventory.

The research described in this thesis corresponds with the first two steps Richardson describes, even though we produced problems that test student misconceptions in digital logic. Although the long-term goal is to develop a concept inventory in digital logic that is statistically sound and can be used nationally, our initial goals are to identify student misconceptions in digital logic through a small number of student interviews and to verify that we could assess the misconceptions we found using multiple-choice problems. The next section describes the method we used to accomplish these goals.

CHAPTER 3 METHODOLOGY

3.1 Concepts

Before we could identify which concepts students have difficulty understanding, we needed a list of concepts that students who completed a digital logic course would be expected to know. We started with the list of ECE 290 course objectives compiled for accreditation with ABET because this document was specifically developed to capture the key concepts of the course. ECE 290 includes some topics on computer organization, but we removed those topics. Not every digital logic course covers these topics, and we wanted our list to be as general as possible. I include the ECE 290 list as Appendix A.1.

We validated our list of concepts using multiple methods. The final list, found in Appendix A.2, corresponds with two textbooks used to teach digital logic design [3, 61]. We asked an experienced CS 231 instructor to provide a list of topics covered in CS 231. The CS 231 list was almost identical to the ECE 290 list, but the similarity was not surprising because both courses use the same textbook. We showed our list of concepts to students whom we interviewed during Spring 2006, and we asked them whether other concepts should be added. No student suggested additions. Finally, we examined course descriptions and syllabi from digital logic courses at institutions that had detailed syllabi on the Web, including the University of Texas [62], the University of Maryland [63], the University of Colorado [64], and the Association for Computing Machinery (ACM) Computer Science Model Computing Curriculum 2001 [59]. Although the ACM Computer Science Model Curriculum does not have the same detail as the course descriptions, the topics on the ECE 290 objectives roughly match the topics in the Architecture and Organization 1 (AR1) – Digital Logic and Digital Systems area of the model curriculum [65].

3.2 Student Interviews

After we created a candidate list of concepts, we needed to determine which concepts students found difficult to understand. Based on suggestions from previous concept inventory developers, we decided to conduct individual student interviews to explore how students understood digital logic concepts [66]. We conducted 16 student interviews during the 2005–2006 academic year; each interview lasted about one hour. I conducted the interviews, and Craig Zilles was present for many of them. Most interviews were videotaped. Each interviewed student received compensation of $20. Because our research involved human subjects and required IRB approval, we created an informed consent form for students to sign. This form is found in Appendix A.3.

We interviewed eight students during Fall 2005. Student volunteers were identified through three rounds of e-mail. In the first round, we contacted students who had taken ECE 290 and CS 231 during Summer 2005, and all three respondents had received “A” grades. To achieve a diversity of achievement levels in our interview subjects, we contacted students who had taken these courses during Spring 2005 and had received course grades of “C and lower” and “B/B-” in the second and third solicitations, respectively. Five additional students volunteered for the interviews. Of the eight students whom we interviewed, three had taken CS 231, and five had taken ECE 290. Two students were women. The ratio of men and women we interviewed is representative of engineering students at the University of Illinois.

We first asked students to reflect on their experiences in either ECE 290 or CS 231 and to share their perceptions of the most important, the easiest, and the most difficult topics in the courses. Because students’ responses were too general to be useful, we provide no data here on their responses. The time between instruction and interview may have contributed to the lack of meaningful responses.

For the remainder of the interview, we asked students to explain digital logic concepts and perform “think alouds,” during which students solved logic design problems — either on the whiteboard or on paper — while verbalizing their thought processes. We asked follow-up questions to probe why students chose particular techniques and how they arrived at their answers.

The interview problems can be found in Appendix A.5.

Initially, many problems were similar to problems students would encounter in a textbook or in a course problem set. The initial set of problems consisted of at least one problem testing each course objective. As the interviews progressed, we removed from the interviews the problems that students correctly answered most often. For example, we removed problems on converting binary numbers to decimal numbers and on drawing timing diagrams for combinational circuits. The remaining problems contained at least one concept that students found difficult. If students found only one difficult concept in a problem, we simplified the remaining parts of the problem to highlight the difficult concept. If students had difficulty with multiple concepts in a problem, we created separate problems so that each problem would focus on only one concept.

During the Fall 2005 interviews, we found that we could not cover material for both combinational and sequential logic during the hour we allocated for the interviews. We found that students performed poorly on problems about latches and flip-flops, but we did not know what misconceptions drove the poor performance. Because we did not have sufficient data about students' understanding of sequential logic concepts, we interviewed eight more students during Spring 2006 to probe their understanding of sequential logic. Six of the students had taken CS 231 in Fall 2005 and two students had taken ECE 290 in Fall 2005.

The format of the Spring 2006 interviews was similar to the format of the Fall 2005 interviews with some minor changes. Instead of asking students to recall difficult, easy, and important concepts, we gave students the list of course objectives created for ABET accreditation for ECE 290. We then asked students to mark the objectives they found easy, difficult, or important. We also asked students to mark the objectives they did not encounter in the course and to add any objectives they felt should be on the list. Student responses exceeded student responses from Fall 2005, but mostly validated information we had already discovered during previous interviews. No student suggested additions to the objectives, and a few students said that some objectives were not covered in the course.

Questions and problems in the Spring 2006 interviews focused primarily on sequential logic concepts, but otherwise the format of the Spring 2006 interviews differed little from the Fall 2005 interviews. We asked questions about the difference between combinatorial and sequential logic, latch and flip-flop operation and design, transforming and analyzing information from next-state tables, state diagrams, timing diagrams, and creating sequential logic designs from word problems.

3.3 Assessment Tests

We developed two assessment tests to verify the presence of misconceptions that we found during the interviews. For each misconception, we developed a multiple-choice problem that tests for a deep, conceptual understanding and that requires little or no calculation. Among the answer choices, we included incorrect answers that we heard during student interviews. We also included nonobvious or nonintuitive correct answer choices for some problems. Problems that had more than one correct answer instructed students to select all correct answers.

We administered the first test to 28 students in two recitation sections of ECE 290 as a review during the final class period of the Fall 2005 semester. One test problem is shown Figure 3.1; the full test with results is in Appendix B of this thesis. The Fall 2005 assessment test validated many of the misconceptions we had found during the Fall 2005 interviews, but we realized that students could have correctly selected or rejected answer choices for reasons that were not related to the concept the problem was based on. For example, students might reject answer choice c in the problem in Figure 3.1 because it does not follow the pattern of the other answer choices.

A sequential state diagram with *n* states and requires at least *m* flip-flops. If a different state diagram has 2**n* states, what is the minimum number of flip-flops?

a.) *m*

b.) *m* + 1

c.) *m* + 2

d.) 2**m*

e.) 2**m* + 1

f.) *m*^2

g.) *m*^2 + 1

h.) None of the above

Figure 3.1. An example Fall 2005 assessment test problem

In the Spring 2006 semester, we created a second assessment test. We used five of the problems from the first test without change and modified four problems to fix errors in the problem statements or change the answer choices based on the results of the first assessment test. We removed problem 1 and problem 10 from the Fall 2005 assessment test because neither problem tested a misconception. We also removed problem 11 from the Fall

2005 assessment test because the problem was too complex. We added four problems (problems 1, 5, and 6 on Form 1 and problem 4 on Form 2) to the Spring 2006 assessment test based on the misconceptions that we had found during the Spring 2006 interviews. We pilot-tested the Spring 2006 test with three students who had completed ECE 290 in Fall 2005 to ensure that the problem statements and answer choices were clear and made sense.

To reduce the time required for students to take the second assessment test, we divided the 13 problems of the second assessment test into two test forms, each with 7 problems; one problem appeared on both test forms. These test forms appear in Appendix 0 of this thesis. We administered each test form to about half of the students in ECE 290 during the final lecture period on May 3, 2006. We gave students 30 minutes to complete the test, but most students finished within 15 minutes. Because students had taken a quiz in every ECE 290 lecture during the Spring 2006 semester, the assessment test served as the quiz for May 3. Lecture quizzes did not affect students' grades, but were used only to take attendance. Similarly, students' performances on the assessment test did not count toward their course grades. On each test form, the top sheet explained our research. Students wrote their names on this sheet, detached the sheet from the test, and submitted the sheet for attendance credit.

CS 231 students took the second assessment test as part of their final exams on May 6, 2006. About half of the students received each of the two test forms. Unfortunately, the CS 231 students received slightly different versions of some of the problems. The test forms for CS 231 appear in Appendix C.3, and the differences between the ECE 290 and CS 231 test versions are discussed in Section 4.1.

CHAPTER 4 RESULTS

We found that students held many misconceptions after they took an introductory digital logic course. We anticipated some student misconceptions. These anticipated misconceptions include students' inability to use MSI components such as multiplexers and decoders to implement arbitrary Boolean functions, students' uncertainty about the output dependencies of elements in sequential logic circuits, and students' inability to derive the complement and dual of a Boolean expression. We were surprised about some student misconceptions that we found. The most surprising misconceptions were students' reluctance to use Karnaugh maps in unfamiliar problems, students' inability to correctly identify logically complete sets, and students' unwillingness to abstract functionality from subsystems of complex circuits. We could assess many misconceptions using multiple-choice questions. I list misconceptions that we confirmed using the assessment test below along with the subsections where I discuss them further.

- Students often have a level of understanding consistent with the "Remember" level rather than the "Apply" level of the revised Bloom's taxonomy of Anderson et al. [4]. We identified some instances specific to digital logic:
    - While most students know that a NAND gate is logically complete, many students cannot determine whether other sets of logic gates are logically complete (Section 4.2.4).
    - Students do not always reason correctly about the effect of constant inputs on logic gates. For example, some students believe that a logic-1 input has no effect on the output of a NOR gate (Section 4.2.3).
    - Some students think that every sequential logic device changes output only on clock edges (Section 4.2.5).
    - Students may recall that a state diagram with n states requires at least $\log_2 n$ flip-flops to implement, but have difficulty determining the number of flip-flops needed to implement a state diagram with $2n$ states (Section 4.3.3).

- Students are uncertain about the output dependencies of different logic devices such as latches, flip-flops, and combinational logic gates (Sections 4.2.5 and 4.2.7).
- Students have difficulty applying previous knowledge to new situations:
    - Students think that Karnaugh maps are solutions for particular types of problems rather than tools to be used in a variety of situations. As a result, students are reluctant to use Karnaugh maps in problems that do not specifically call for them (Section 4.3.6).
    - Some students are unable to implement arbitrary Boolean expressions with MSI components (Section 4.3.7).
    - Students do not understand that a combinational circuit has a unique output for every combination of inputs (Section 4.2.1).
    - Students have difficulty determining when the output changes in circuits that have both sequential and combinational logic, particularly in Mealy machines (Section 4.2.5).
- Students have difficulty designing with MSI components (Section 4.3.7).
- While students do understand some of the implications of complements and duals, they often have difficulty deriving the complement or dual of a particular expression (Section 4.2.2, Section 4.3.1, and Section 4.3.2).

While we could assess many misconceptions using multiple-choice problems, we found that some misconceptions were too abstract to assess using multiple-choice problems. We mention these abstract misconceptions here as a starting point for future work:

- Many students have difficulty with the concept of state in sequential design. They do not always know how to describe a state or how to tell when transitions between states occur. Students also have difficulty distinguishing an abstract state machine from specific implementations of that state machine.
- When students analyze complex circuits, they do not simplify the analysis by abstracting functionality from subsystems of the circuit. In particular, when students are presented with gate-level implementations of larger components like

latches or MSI devices, they are reluctant to replace a gate-level implementation with a box that contains the functionality of the larger device.

- Students are sometimes uncertain about when combinational circuits should be used and when sequential circuits should be used.

During the interviews, CS 231 students performed poorly on sequential logic problems. Because of this performance, we hypothesized that CS 231 students would perform worse on the Spring 2006 assessment test, particularly on the sequential logic problems. Assessment test data suggest no statistically significant differences between the two student populations overall, however. Although there were a few questions on which the two populations had statistically significant response rates, we cannot draw any strong conclusions because CS 231 students and ECE 290 students took slightly different versions of the test under different testing conditions. I discuss these differences in the next section of this chapter, and discuss each assessment test problem in subsequent sections.

4.1 ECE 290 and CS 231 Students and the Spring 2006 Assessment Test

Though both ECE 290 and CS 231 cover a similar set of topics, the two courses have very different philosophies: ECE 290 incorporates a large number of exercises in homework, interactive Web assignments, and simulation labs to teach students digital logic concepts. In contrast, CS 231 has a slightly larger number of shorter, less complex assignments. ECE 290 and CS 231 students also have different backgrounds. Most ECE 290 students have taken ECE 110, a first-year course that covers basic logic gates and introduces digital systems. For most computer science students, CS 231 is their first experience with logic gates and digital systems.

Although we wanted to see which course resulted in a better conceptual understanding of course material, CS 231 students received an older version of the assessment test. The basic idea of each problem was the same on both versions of the test, but the differences between test versions preclude a decisive comparison between the conceptual understanding ECE 290 students and CS 231 students. We discuss some of the insights that we gained, but we do not give a full discussion of the CS 231 version of the test.

The test version given to CS 231 students and the data from our administration of that test are included in Appendix C.3 of this thesis.

Further, we cannot make a full comparison between ECE 290 and CS 231 students because they took the assessment test under different conditions. The ECE 290 assessment test conditions may have artificially lowered ECE 290 students' scores. First, we did not give ECE 290 students advance notice about the assessment test. Second, we administered the assessment test as an attendance quiz on May 3, 2006, during the final ECE 290 lecture period. Because attendance was required for ECE 290 students, they received an attendance quiz every lecture period. ECE 290 teaching assistants looked only at the names on the quizzes for attendance purposes and rarely examined the answers closely. Consequently, many students did not take the attendance quizzes seriously, and their attitudes about the attendance quizzes may have affected their attitudes about the assessment test. Finally, the instructor allotted the final 30 min. of the lecture period for the assessment test; the average attendance quiz was allotted five minutes. Although most students finished the test in less than 15 minutes, students who became bored may have hurried through the final problems in the test.

We administered the assessment test to CS 231 students as part of the final exam on May 6, 2006; each test question was worth one percent of the final exam grade. Because the assessment test was part of the final exam, CS 231 students probably had reviewed the material covered by the assessment test and took the test seriously. In contrast with ECE 290 students, we do not know whether CS 231 students had time to review their solutions to assessment test problems before submitting them.

4.1.1 Overview

I measured performance on the assessment test by the number of decisions students needed to make. A problem with just one correct answer required one decision, and a problem with multiple correct answers required one decision for each answer choice. Consequently, a student's score on a test is the total number of correct decisions. Although this scoring policy places larger emphasis on problems with multiple correct answers, I believe the larger emphasis is justified because students have to make several, often independent, decisions about the answer choices for these problems. Table 4.1 shows statistics for both forms of the test.

Table 4.1. Statistics from the Spring 2006 assessment test

Form 1	Maximum possible score – 28		
	Mean	Standard Deviation	Sample size
ECE 290	22.0	9.7	51
CS 231	22.2	8.5	51
Form 2	Maximum possible score – 26		
	Mean	Standard Deviation	Sample size
ECE 290	19.9	8.9	56
CS 231	19.7	7.1	43

Since all distributions appear to be approximately normal (see Figures C.1 and C.2 in Appendix C for histograms), a *t* test will tell whether there is a significant difference between the ECE 290 scores and the CS 231 scores. This statistical test determines the probability that two normal distributions belong to the same population. I used a two-tailed *t* test to determine whether there is any difference between ECE 290 scores and CS 231 scores; I would use a one-tailed *t* test if I expected CS 231 scores to be higher than ECE 290 scores or vice versa. The *t* test showed that there was no statistically significant difference between ECE 290 scores and CS 231 scores on either Form 1, $t(100) = -0.23$, $p = 0.82$, or Form 2, $t(95) = .42$, $p = 0.67$.

Based on the interviews, we found that CS 231 students usually had more difficulties when asked about gate-level details of sequential design such as flip-flop implementation, the difference between flip-flops and latches, and signal transitions. Thus, I hypothesized that CS 231 students and ECE 290 students would perform similarly on the combinational portion of the exam, but ECE 290 students would perform better on the sequential portion of the exam. Except for problem 3 on Form 2 of the assessment test (relating number of states to the number of flip-flops needed to minimally implement a sequential design), all sequential problems focus on gate-level details.

Students' performance during this administration of the assessment test did not confirm our hypothesis. I measured students' performance on combinational problems only (problems 1-4 on Form 1 and problems 1-3 and 6-7 on Form 2) and sequential problems only (problems 5-7 on Form 1 and problems 4 and 5 on Form 2) in the same way that I measured performance for the whole test. The statistics for the combinational logic questions and sequential logic questions are shown in Table 4.2. Two-tailed *t* tests showed no significant difference between ECE 290 scores and CS 231 scores for the combinational logic problems on Form 1 ($t(100) = -1.52$, $p = 0.13$), the combinational logic problems Form 2 ($t(90) = -0.28$, $p = 0.78$), or the sequential logic problems on Form 2 ($t(94) = 1.33$, $p = 0.19$). ECE 290 scores and CS 231 scores were significantly different ($t(100) = 2.18$, $p = 0.03$) for the sequential logic problems on Form 1, but this difference is likely due to differences in the problem statement of the latch timing problem on Form 1, discussed in the next section.

Table 4.2. Statistics for combinational and sequential problems on the Spring 2006 assessment test

Form 1 Combinational	Maximum possible score – 21		
	Mean	Standard Deviation	Sample size
ECE 290	16.6	6.4	51
CS 231	17.4	5.8	51
Form 1 Sequential	Maximum possible score – 7		
	Mean	Standard Deviation	Sample size
ECE 290	5.4	1.9	51
CS 231	4.8	2.1	51
Form 2 Combinational	Maximum possible score – 19		
	Mean	Standard Deviation	Sample size
ECE 290	14.6	4.3	56
CS 231	14.7	4.4	43
Form 2 Sequential	Maximum possible score – 7		
	Mean	Standard Deviation	Sample size
ECE 290	5.3	1.9	56
CS 231	4.9	1.7	43

4.1.2 Differences between problems

I used Pearson's chi-square test to test for any correlation between the course a student took and the likelihood that the student would select only the correct answers for a test problem. A test problem that has a strong correlation could show a correlation between pedagogy and conceptual understanding. In this section, I show how I used Pearson's chi-square test for the first problem of Form 2 of the Spring 2006 assessment test. Table 4.3 shows the data for students of each course for this problem.

Table 4.3. Data for problem 1 on Form 2

	ECE 290 students	CS 231 students	Total
Solved correctly	13	11	24
Solved incorrectly	43	32	75
Total	56	43	99

I calculated the expected value for the number of ECE 290 students who should solve this problem correctly by multiplying the proportion of the total number of students who correctly solved the problem (24/99) by the number of ECE 290 students who took the test (56) to get 14. The other expected values are shown in Table 4.4.

Table 4.4. Expected values for problem 1 on Form 2

	ECE 290 students	CS 231 students
Expected to solve correctly	14	10
Expected to solve incorrectly	42	33

I squared the difference between the expected values and the observed values and divided that sum by the expected value to get four numbers that are a measure of the difference for each cell, shown in Table 4.5.

Table 4.5. Measure of difference between expected values and actual values for problem 1 on Form 2

	ECE 290 students	CS 231 students
Solved correctly	0.02	0.03
Solved incorrectly	0.01	0.01

I then summed all of the differences from Table 4.5 to get the chi-square statistic (0.07). Finally, I found the probability that the chi-square sampling distribution is equal to or greater than the chi-square statistic. The chi-squared distribution assumes there is no association between the course taken and the likelihood of solving the problem correctly. Using the chi-square distribution with one degree of freedom, the probability that the distribution is equal to or greater than the calculated chi-square statistic is 0.79. Because the probability value is greater than 0.05, I cannot statistically reject the null hypothesis that there is no association between the course taken and the likelihood of solving the problem.

Because Pearson’s chi-square test can be used only when all options are mutually exclusive, I tested only whether students’ answers were fully correct (they selected all of the correct answers and only the correct answers) or not fully correct for most problems. Testing only whether students' answers were fully correct may mask some results from the data, such as whether CS 231 students preferred one distracter for a problem but ECE 290 students preferred a different distracter. Although we are interested in whether ECE 290 students and CS 231 students approached the problems differently, Pearson's chi-square test cannot analyze answer distributions for problems with multiple correct answers, and I lack the statistical background to perform a more thorough analysis. Future work towards a concept inventory would benefit from a more thorough statistical analysis of student data.

Table 4.6 compares the CS 231 performance with the ECE 290 performance for each test problem; the problems are listed in the order they appear in the ECE 290 version of the test. An asterisk denotes a statistically significant difference. I discuss problems that have statistically significant differences between ECE 290 students and CS 231 students after the table.

Table 4.6. Chi-square statistics for Spring 2006 assessment test problems

Problem	ECE 290 % fully correct	CS 231 % fully correct	Chi-square statistic	*p*
Form 1				
1. Combinational Extrapolation	65%	61%	0.2	0.69
2. Complement and DeMorgan's Law	29%	53%	5.8	0.02*
3. Nontrivial Output	57%	53%	0.2	0.69
4. Logical Completeness	16%	18%	0.1	0.79
5. Sequential Timing – FF Output	53%	49%	0.2	0.69
6. D-Latch Identification	65%	76%	1.7	0.19
7. Latch Timing – Form 1 Version	57%	12%	23.	< 0.005*
Form 2				
1. Implementing a Complement	23%	26%	0.1	0.79
2. Implications of Equality	46%	23%	5.6	0.02*
3. Minimal Number of States	63%	65%	0.1	0.79
4. Sequential Timing – Z Output	45%	53%	0.8	0.38
5. Latch Timing – Form 2 Version	57%	49%	0.7	0.41
6. Minimal Gate Implementation	43%	79%	13.2	< 0.005*
7. MSI Implementation	11%	30%	6.0	0.01*

Implications of equality (Section 4.3.2): ECE 290 students (46% fully correct) outperformed CS 231 students (23% fully correct) on this problem about the implications of functions with equivalent truth tables. The only difference between two test versions is that the ECE 290 version calls the functions "Boolean functions," whereas the CS 231 version calls the functions "functions." The answer choice stating that the duals of two functions with the same truth table are equal accounts for most of the student differences; ECE 290 students were much more likely to select this answer choice than CS 231 students ($\chi^2(1) = 9.6$, $p < 0.005$). If this answer choice were removed, then CS 231 students would perform about the same (56% fully correct versus 54% fully correct for ECE 290 students). We are not sure whether CS 231 students forgot that the duals of two equivalent expressions are also equivalent or whether they were not taught that the duals of two equivalent expressions are also equivalent.

Complementation and DeMorgan's law (Section 4.2.2): CS 231 students (53% fully correct) outperformed ECE 290 students (29% fully correct) on this problem, which asked students to use DeMorgan's law to determine whether expressions were equivalent. There is no difference between the problems on different versions. Because this problem was located at the end of the ECE 290 version of the test, ECE 290 students may have approached the problem less carefully than problems at the beginning of the test.

Karnaugh maps (Section 4.3.6): CS 231 students (79% fully correct) outperformed ECE 290 students (43% fully correct) on this problem, which asked students to determine the number gates needed to minimally implement an expression. The problem statement was different on the two versions of the test. The ECE 290 version includes this text:

> The truth table below defines a Boolean function. Suppose you plan to implement this function as a combinational circuit specified by a sum-of-products expression. Assuming only AND and OR gates are available, determine the minimum required number of gates. Assume you have inputs {*a*, *b*, *c*} and their complements {*a*', *b*', *c*'} available as inputs to the circuit.

The CS 231 version is shorter and less precise. While the general idea remains the same, students may have required more time to understand the CS 231 version of the problem. The CS 231 version includes this text:

> If you implement the following truth table as a sum of products, what is the minimum required number gates, assuming only AND and OR gates are available? Assume you have inputs and their complements available as inputs to the circuit.

More ECE 290 students (21%) than CS 231 students (7%) selected answer choice f, a significant difference ($\chi^2(1) = 5.6$, $p = 0.02$). We believe students who selected answer choice f tried to implement the product-of-sums form of the expression. Even with answer choice f removed, the difference between ECE 290 performance and CS 231 performance is still significant ($\chi^2(1) = 9.1$, $p < 0.005$).

MSI implementation (Section 4.3.7): CS 231 students (30% fully correct) outperformed ECE 290 students (11% fully correct) on the problem, which asked students to select which circuits using MSI components were implemented correctly. The problem statements are similar except the ECE 290 version referred to the function in the problem statement as a "Boolean function" while the CS 231 version just says "function." The difference between students' performance on answer choices corresponding to decoder implementations was not significant ($\chi^2(1) = 3.65$, $p = 0.06$), but the difference between students' performance on the answer choices corresponding to multiplexer implementations was significant ($\chi^2(1) = 8.43$, $p < 0.005$).

Latch timing (Section 4.2.7): ECE 290 students outperformed CS 231 students on a problem present in both forms asking students to find the output of a small latch circuit at different times, but the difference was significant only on Form 1 (see Table 4.6). The problem statement was significantly different between the ECE 290 version and the CS 231 version. The ECE 290 version explicitly stated that the latch was not a positive-edge triggered flip-flop, whereas the CS 231 version lacked this statement. As a result, CS 231 students might have mistaken the latch for a flip-flop. Consequently, no meaningful insight can be gained from the difference between students' performance.

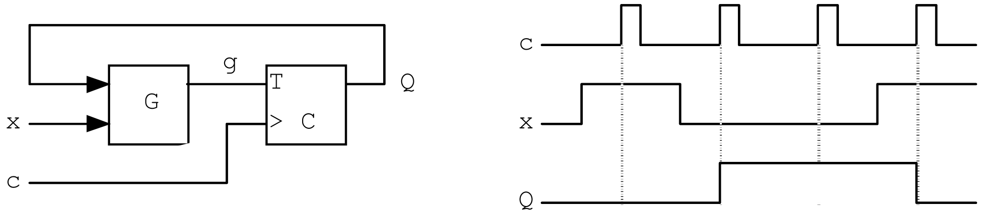
4.2 ECE 290 Spring 2006 Assessment Test Form 1 Problems

The sections below describe the problems in Form 1 of the ECE 290 Spring 2006 assessment test in the order they appear on the test. Form 1 of the ECE 290 Spring 2006 assessment test is in Appendix C.2 of this thesis.

4.2.1 Combinational extrapolation

Students often create timing diagrams from circuit designs, but they rarely derive circuit designs from timing diagrams. We hoped that asking students to derive a circuit design from a timing diagram would reveal students' understanding of sequential circuits and sequential design. Thus, during the interviews, we asked students the problem in Figure 4.1.

The sequential circuit below has a positive-edge-triggered T flip-flop and a combinational circuit G whose output is the function *g*(*x*,*Q*). Complete the timing diagram by showing the waveform for *g*, assuming that the delay in G is negligible. Also specify *g* with a truth table. Explain your reasoning.



Solution:

According to the specification of the T flip-flop, if *Q* changes at a positive clock edge (when *c* rises from 0 to 1), then *g* must be 1 immediately before that clock edge. If *Q* does not change at a positive clock edge, then *g* must be 0 immediately before that clock edge. Because every possible combination of *x* and *Q* is present immediately before some clock edge, *g* can be determined at all places in the timing diagram. When $x = Q = 0$ or $x = Q = 1$, then *g* is 1. When $x \neq Q$, then *g* is 0.

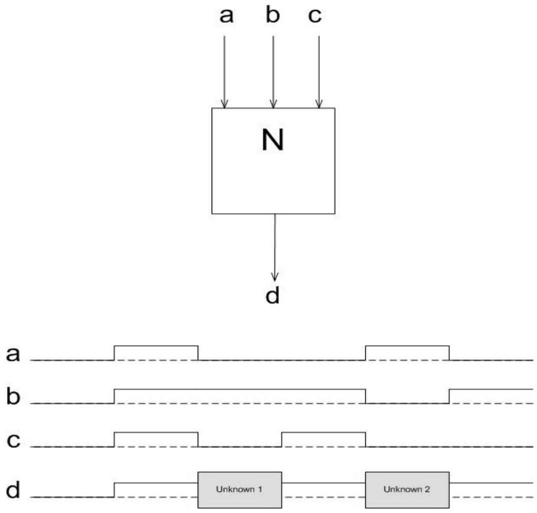
Figure 4.1. A problem on the derivation of a circuit design from a timing diagram

Students' solutions to this problem revealed two misconceptions. Although most students realized that *g* was 1 immediately before positive clock edges (when *c* rises from 0 to 1) where *Q* changed values and *g* was 0 immediately before positive clock edges where *Q* did not change values, many students would assign that value to *g* for the entire clock period. We believe students did not realize that *g* could change when either *Q* or *x* changed. This misconception about combinational circuits is discussed in Section 4.2.5. Even after students determined the value for *g* for all values of *x* and *Q*, many students believed the values of *g* could not be determined before the first change of *x* or after the last positive clock edge.

We created the problem in Figure 4.2 to focus on the concept that each set of input values in a combinational circuit always produces the same set of output values.

ECE 290 Spring 2006 Assessment Test Form 1, Problem 1

The block *N* below represents some combinational circuit that implements an unknown Boolean function. It has one output {*d*} and three inputs {*a*, *b*, *c*}.



What are the values of d in the boxes?

Unknown 1	Unknown 2
a. 0	a. 0
b. 1	b. 1
c. Cannot be determined	c. Cannot be determined

Solution:

Unknown 1 can be determined because the input {$a = 0$, $b = 1$, $c = 0$} is the last set of inputs in the timing diagram with a 1 output. Unknown 2 cannot be determined because the input {$a = 1$, $b = 0$, $c = 0$} is not present anywhere else in the timing diagram.

Figure 4.2. A problem on finding unknown outputs in a combinational circuit

Table 4.7 shows students' results for this problem. The bottom rows indicate the number of students who selected a particular combination for the unknowns. For example, the row for a / b indicates the students who selected answer choice a for Unknown 1 and answer choice b for Unknown 2.

Table 4.7. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.2

Form 1	Problem 1	49 of 51 students answered this problem	
Unknown 1	Number of Students	Percentage of Students	Correct answer
a	1	2%	
b	41	80%	*
c	7	14%	
Unknown 2			
a	7	14%	
b	2	4%	
c	40	78%	*
Combinations			
b / c	33	65%	*
c / c	7	14%	
b / a	5	10%	
b / b	3	6%	

The combinations in Table 4.7 reveal the most about student understanding. The students who selected c for both unknowns likely believed that there was not enough information to solve the problem. Some students selected the correct answer for Unknown 1 but did not select the correct answer for Unknown 2. These students may have reasoned that Unknown 1 must be 1 because the input combination associated with Unknown 1 is seen later in the timing diagram, but were uncomfortable with the "cannot be determined" answer choice for Unknown 2. These students may have guessed both unknowns.

A fully correct answer to this problem reveals a deep level of student understanding since the student must know both when an unknown can be determined and how to determine an unknown value when it can be determined. Although students could guess both correct answers, we believe the correct combination of answers would seem inconsistent to students who do not know how to solve the problem. Rather, we believe they would surmise that either both unknowns can be determined or neither unknown can be determined.

4.2.2 Complement and DeMorgan's law

During the Fall 2005 interviews, some students could not derive the complement of the Boolean expression in Figure 4.3. Students were asked to fully propagate the complement until only literals were complemented: for example, students should answer $(x' + y')$ instead of $(xy)'$.

What is the complement of $wx(y'z + yz') + w'x'$?

Solution:

$$(w' + x' + (y + z')(y' + z))(w + x)$$

Figure 4.3. An interview problem on complementation

Students incorrectly solved this problem for multiple reasons. Some students had forgotten DeMorgan’s law and could not begin the problem. Some students did not remember how to apply DeMorgan’s law while taking operation precedence into account and they omitted the parentheses around $w' + x' + (y + z')(y' + z)$. Other students did not propagate the complement and gave answers like $(w + x + (y'z + yz'))(w'x')$. We constructed the problem in Figure 4.4 based on how students derived the complement of an expression. Table 4.8 shows ECE 290 student performance on the Spring 2006 assessment test.

ECE 290 Spring Assessment Test Form 1, Problem 2

Which of the following expressions are equivalent to $(a + b')(bc + a'c')$?

Select all correct answers.

a.) $(a + b')((b' + c')(a' + c'))'$

b.) $((a' + b)(b'c' + ac))'$

c.) $(a'b)+(bc + a'c')'$

d.) $((a'b) + (b' + c')(a + c))'$

e.) $(a'b)'((b' + c')' + a'c')$

f.) $(ab') + (b + c)(a' + c')$

Solution:

Answer choice a is incorrect because the right half of the expression is equivalent to $(bc + ac)$. Answer choice b is incorrect because only the literals are complemented. Answer choice c is incorrect because students cannot transform $(a'b)$ to $(a + b')$. Answer choice d is correct; the expression inside the outermost parentheses is equivalent to the complement of the expression in the problem statement. Answer choice e is also correct. Answer choice f is incorrect because it is the dual of the expression in the problem statement.

Figure 4.4. A problem on complementation and DeMorgan's Law

Table 4.8. ECE 290 Spring 2006 assessment test results for the problem in

Form 1	Problem 2	48 of 51 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	4	8%	
b	3	6%	
c	4	8%	
d	31	61%	*
e	25	49%	*
f	5	10%	
d, e	15	29%	*

Most distracters were ineffective and were selected less than 10% of the time. Answer choice f was the most effective distracter. Unfortunately, this distracter could be selected for multiple reasons. Students might recognize this expression as the dual of the expression in the problem statement and think that the dual expressions are equivalent. They might also view the expression in answer choice f as the complement of the expression in the problem statement and believe that an expression and its compliment are equivalent. More students selected answer d than answer e, even though the expression in answer e requires fewer algebraic steps to match the expression in the problem statement.

It is difficult to measure student understanding of complementation. The expression must capture some of the finer points of DeMorgan's law by having students complement expressions that OR product terms, such as $(ab' + a'b)'$, but complex expressions require more time to derive. This problem does not check students' understanding of complementing the constants 0 and 1.

4.2.3 Nontrivial output

Students sometimes make mistakes when handling extra input lines to logic gates, particularly NAND and NOR gates. In both real designs and classroom problems, devices with the required number of input lines are not always available. Some input value must go into these lines, but students do not always know what input is needed. We first saw this uncertainty when interviewed students tried to solve the problem in Figure 4.5 with a limited set of devices. Function F uses only three of the four NOR gate inputs. One student reasoned that, since an OR gate requires a 0 to result in nontrivial output (not a constant 1 or 0), a NOR gate would require a 1 to result in nontrivial output.

$F(x, y, z) = \text{AND}(\text{M1}, \text{M4}, \text{M6})$

$G(x, y, z) = \text{OR}(\text{m1}, \text{m3}, \text{m5})$

$H(x, y, z) = x \text{ XOR } z$

Draw a circuit which implements the three functions *F*, *G*, and *H*. You may use only one 3-to-8 decoder, two 4-input OR gates, and one 4-input NOR gate.

Solution:

The OR gate for *G* and NOR gate below could also have the constant 0 going into the bottom input.

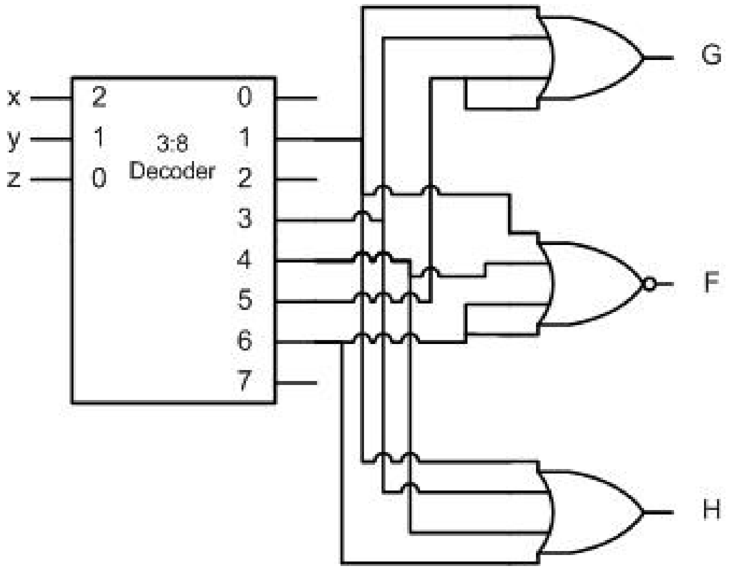


Figure 4.5. A problem on implementing functions with a decoder

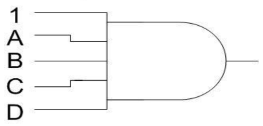
We constructed a multiple-choice problem to examine students' knowledge about when the output of common logic gates is nontrivial. Our first attempt was problem 7 of the Fall 2005 assessment test in Appendix B.2. Because this problem used two-input logic gates, students could base their answers on a small truth table rather than a conceptual understanding about NAND and NOR gates. Because we wanted to discourage the use of truth tables, we added three inputs to each gate for the Spring 2006 assessment test, resulting in the problem in Figure 4.6. Student performance on this problem is tabulated in Table 4.9.

ECE 290 Spring Assessment Test Form 1, Problem 3

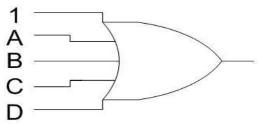
Which of the following will result in nontrivial output (not always 0 or 1)?

Select all correct answers.

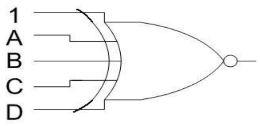
a)

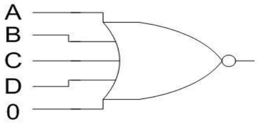


b)

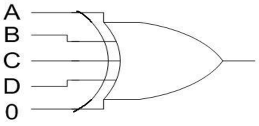


c)

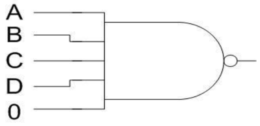




d)



e)



f)

g.) None of the above

Solution:

Answer choice a should be selected and answer choice f should not be selected because AND and NAND gates will have trivial output with a constant 0 input. Answer choice b should not be selected and answer choice d should be selected because OR and NOR gates will have trivial output with a constant 1 input. Answer choices c and e should be selected because XOR and XNOR gates do not have trivial output.

Figure 4.6. A problem on nontrivial input

Table 4.9. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.6

Form 1	Problem 3	50 of 51 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	42	82%	*
b	2	4%	
c	37	73%	*
d	40	78%	*
e	43	84%	*
f	6	12%	
g	3	6%	
a, c, d, and e	29	57%	*

As expected, students performed better on AND, OR, and XOR gates (answer choices a, b, and e) than on NAND, NOR, and XNOR gates (answer choices c, d, and f). Some answer patterns raise interesting questions. The three students who selected answer choices a and f probably believed that the constant input value should be complemented when the gate is a NAND gate. We would expect these students to either select both answer choices b and d or reject both answer choices b and d because the gates in answer choices b and d have a complemented input value when the gate is a NOR gate. One of the three students did choose similarly for answer choices b and d, but the other two students selected only answer choice d. It is not clear whether the students who selected answer choices a, d, and f were using different reasoning for the AND/NAND pair than the OR/NOR pair or whether the students hurried through the problem.

Some students may have misinterpreted the problem statement. Two students selected answer choices b and f, which were the answer choices corresponding to the gates that did have trivial output. The students who selected answer choices b and f may have thought the question was actually asking them to select gates that had trivial output. The three students who selected answer choice g may have selected different answer choices if the phrase “nontrivial output” was more clearly defined. We expected all students to correctly select answer choice a, but some students did not. These students may have also misinterpreted the problem statement. Interviews with students about this problem should point out any ambiguity in the problem statement.

4.2.4 Logically complete sets

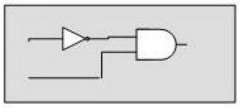
Although logical completeness is not a major concept in digital logic, we believe that students who understand logical completeness will also thoroughly understand Boolean logic. One interview question asks students to explain when a set of logic functions can be used to express all Boolean functions. During the interviews, we found that few students could answer this question precisely. As a result, we began to ask students if particular sets of logic gates were logically complete. The problem in Figure 4.7 contains the most interesting sets of logic gates, and it was included in the assessment tests.

ECE 290 Spring Assessment Test Form 1, Problem 4

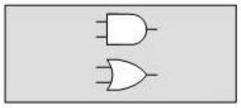
Which of the following are complete logic families (i.e., all possible Boolean functions can be implemented using just these gates and the constants 0 and 1).

Select all correct answers.

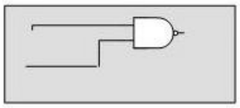
a)



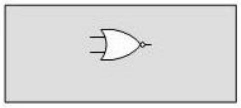
b)



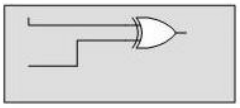
c)



d)



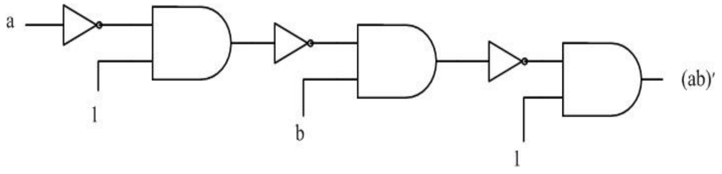
e)



f.) None of the above

Solution:

The circuit in answer choice a is complete because three gates can implement a NAND gate, which is complete.



The set of circuits in answer choice b is not complete because there is no inverter. The NAND gate in answer choice c and the NOR gate in answer choice d are both complete. The XOR gate in answer choice d is not complete.

Figure 4.7. A problem on logical completeness

ECE 290 student Spring 2006 assessment test results for the problem in Figure 4.7 are tabulated in Table 4.10.

Table 4.10. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.7

Form 1	Problem 4	51 of 51 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	13	25%	*
b	22	43%	
c	44	86%	*
d	37	73%	*
e	3	6%	
f	2	4%	
a, c, and d	8	16%	*

Students' answers reveal the depth to which they understand logical completeness. Students who selected answer choice c most likely recalled that a NAND gate is complete; the NAND gate is a common example of logical completeness in introductory digital logic courses. Students who selected the NOR gate either recalled that a NOR gate is also complete, realized that NOR gates can be used to construct NAND gates, or guessed due to the symmetry between the NAND answer choice and the NOR answer choice. One student selected the NOR gate as logically complete but not the NAND gate.

Students who did not select answer choice b with just the AND and OR gates likely recalled that a logically complete set must be able to implement AND, OR, and NOT.

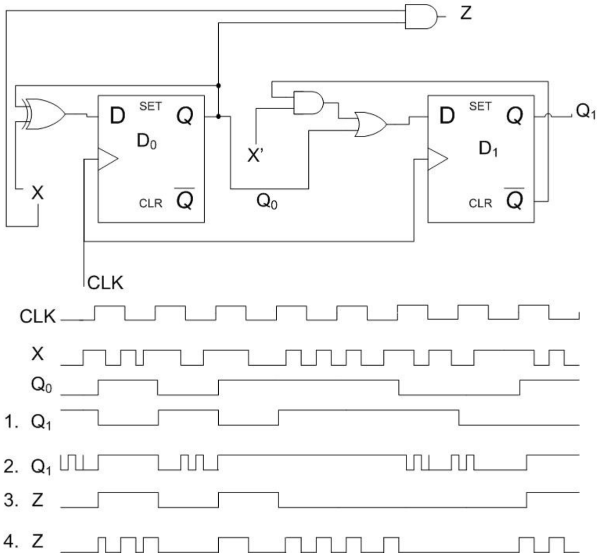
Students who selected answer choice a were able to implement a NAND gate or AND, OR, and NOT gates out of the gate in the answer choice. These students revealed the deepest level of understanding of logical completeness tested by this problem.

4.2.5 Flip-flop and combinational circuit timing

ECE 290 and CS 231 focus on sequential circuits that implement Mealy machines. The output of a Mealy machine depends on the current state and the current input. We found that students do not understand when and under what conditions the state and output change. When the flip-flops are edge triggered, the state, based on flip-flop outputs, can change only on clock edges, but the output of the whole circuit may change on clock edges or on input changes. Some students do not understand when flip-flop outputs change, and other students do not know which inputs cause the circuit output to change. We used the problem in Figure 4.8 on the Fall 2005 assessment test and during Spring 2006 interviews to explore how students understand when the state and the circuit output change.

ECE 290 Fall 2005 Assessment Test Problem 11

Select which combination of Z and Q_1 are correct. Assume gate delays and flip-flop propagation delays are negligible. All flip-flops are positive edge triggered D flip-flops.



a.) 1 and 3   b.) 1 and 4   c.) 2 and 3   d.) 2 and 4

Solution:

Waveform 2 has Q_1 state transitions between positive clock edges so it cannot be the waveform of a flip-flop output. Waveform 3 changes only on positive clock edges and ignores the effect of the input X. Waveforms 1 and 4 are correct, resulting in answer b.

Figure 4.8. A problem on output dependencies in sequential circuits

Although 62% of ECE 290 students correctly selected answer choice b on the Fall 2005 assessment test, we were not convinced that students approached the problem in Figure 4.8 as we had intended. For example, we wanted students to reject waveform 2 because it had transitions when there was no positive clock edge, but Spring 2006 interviews showed that many students based their answers on the combinational logic generating the input to the D_1 flip-flop. None of the six students who solved this problem during Spring 2006 interviews based their rejection of waveforms 2 and 4 on when the outputs should change.

As a result, we split the problem in Figure 4.8 into two new problems to separate the flip-flop output dependencies from combinational output dependencies. We also replaced the

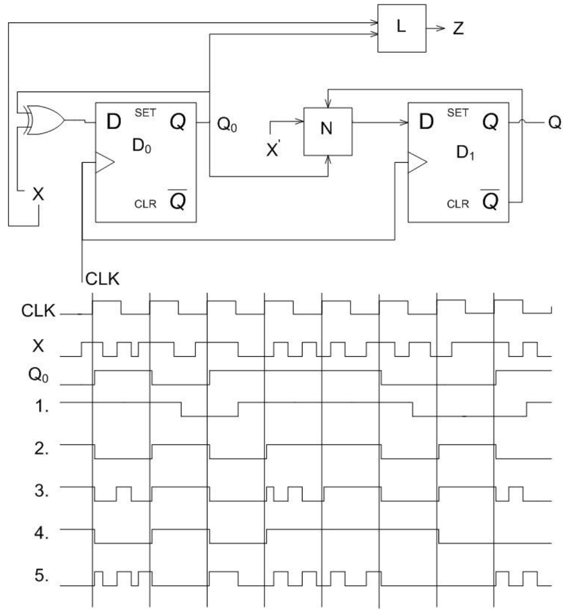
AND gate that generated the *Z* output and the logic gates that generated the D_1 flip-flop input with combinational circuit boxes to hide implementation details. We adjusted the waveforms in response to student interviews until students based decisions on answer choices on a conceptual understanding of output dependencies.

The problem focusing on flip-flop output dependencies is shown in Figure 4.9.

ECE 290 Spring Assessment Test Form 1, Problem 5

The sequential circuit below has two positive-edge triggered D flip flops D_0 and D_1 and two unknown combinational circuits *N* and *L*. Which of the following waveforms could be waveforms for Q_1?

Select all correct answers.



a.) 1 b.) 2 c.) 3 d.) 4 e.) 5

Solution:

Waveform 1 is for a negative-edge triggered flip-flop. Waveform 2 is a correct choice (N implements the expression $X' + Q_1'$). Waveform 4 is a correct choice (N implemented the expression $X'Q_1' + Q_0$). Waveforms 3 and 5 have transitions when there is no positive clock edge.

Figure 4.9. A problem on flip-flop output dependencies

ECE 290 Spring 2006 assessment test results for the problem in Figure 4.9 are tabulated in Table 4.11.

Four of the six students who selected answer a also selected answers b and d. These students might have known that the output of a flip-flop changed on a clock edge but did not notice that the problem statement specifies positive-edge flip-flops. The students who selected answer choices c or e likely only examined the waveforms at the clock edges.

Table 4.11. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.9

Form 1	Problem 5	49 of 51 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	6	12%	
b	36	71%	*
c	5	10%	
d	39	77%	*
e	4	8%	
b and d	27	53%	*

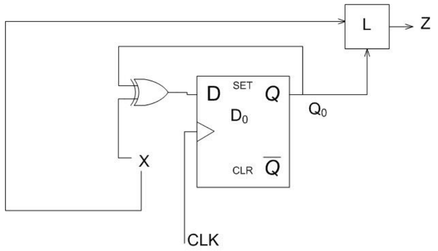
We constructed the problem in Figure 4.10 to focus on the output *Z* of a combinational logic block that has an input from the D_0 flip-flop. This problem is the same as the problem in Figure 4.8 with the D_1 flip-flop and the *N* combinational block removed because those elements added unnecessary complexity. ECE 290 student Spring 2006 assessment test results are tabulated in Table 4.12.

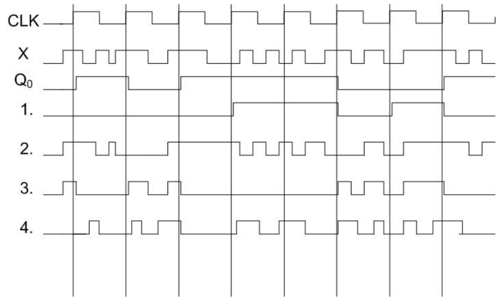
Students selected the distracters from the problem in Figure 4.10 more frequently than they selected the distracters from the problem in Figure 4.9. This result could indicate that students are less certain about when outputs based on both sequential and combinational logic change than they are certain about when outputs based on sequential logic alone change. This may not be the best interpretation, however, because the waveforms for the problem in Figure 4.10 are more complicated than the waveforms in the problem in Figure 4.9. While the waveforms in the problem in Figure 4.9 could be selected or rejected based on the sole criteria of transitions that were not around a clock edge, the waveforms in the problem in Figure 4.10 required closer examination to select or reject them. Waveform 2 in the problem in Figure 4.10 is particularly complicated since it has no apparent relationship to the clock edges.

ECE 290 Spring Assessment Test Form 2, Problem 4

The sequential circuit below has a positive-edge triggered D flip flop D_0 and a combinational circuit *L*. Which of the following waveforms could be waveforms for *Z*?

Select all correct answers.





a.) 1 b.) 2 c.) 3 d.) 4 e.) None of the above

Solution:

Waveform 1 is the Boolean expression *X′* evaluated only at positive clock edges. Waveform 2 is the Boolean expression $Q_0'X + Q_0X'$ evaluated only when *X* changes. Waveform 3 is the Boolean expression $Q_0'X$ evaluated when either Q_0 or *X* change. Waveform 4 is a waveform that has changes without any *X* or Q_0 change.

Figure 4.10. A problem on the output dependencies of combinational logic in sequential circuits

Table 4.12. ECE 290 Spring 2006 assessment results for the problem in Figure 4.10

Form 2	Problem 4	54 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	10	19%	
b	17	32%	
c	40	74%	*
d	5	9%	
e	4	7%	

The problem in Figure 4.10 might be unfamiliar to some students who have completed a digital logic course whose topics differ significantly from ECE 290 and CS 231. Courses that teach digital logic but focus on computer organization or on discrete mathematics may not cover sequential design in enough detail to prepare students for this problem. Students who take courses that focus on Moore machines, in which the output is based only on the state, might not have seen circuits where the input directly affected the output. Although we hope that students who have focused on Moore machines would still be able to solve this problem, students who have significant experience with Mealy machines would have an advantage. Students who studied both Moore and Mealy machines should perform equally well on the flip-flop output problem and will have an advantage over students from courses that do not cover sequential design thoroughly.

The problems in Figure 4.9 and Figure 4.10 can take a lot of time for students to solve. Students who understand when the outputs change could quickly reject waveforms 3 and 5 from the flip-flop output problem and waveform 4 from the problem in Figure 4.10. Students who do not understand when outputs change could spend a lot of time looking at the waveforms to make sure every part of those waveforms are correct. We could reduce the time needed to solve this problem by including only one right answer per problem or eliminate the "none of the above" answer choice in the problem in Figure 4.10, but removing these options increases the chance that students could simply guess the correct answer. If we reduced the number of clock cycles in the waveform from eight to three or four, we could still retain the fundamental nature of the distracters.

4.2.6 Latch identification

During the interviews, we asked students to examine the circuit element shown in Figure 4.11, to identify the element, and to explain how the element works. The element is a positively-clocked D-latch: *E* is the data input, *F* is the clock, and *G* and *H* are the outputs *Q* and *Q*′, respectively. None of the students identified the element as a clocked D-latch even though many students tried multiple approaches. Further, although most students recognized that the cross-coupled NAND gates were some kind of latch, no student specifically mentioned an SR-latch. Students' approaches revealed a significant deficiency in students' problem-solving ability and student misconceptions about latches.

Identify the circuit element below:

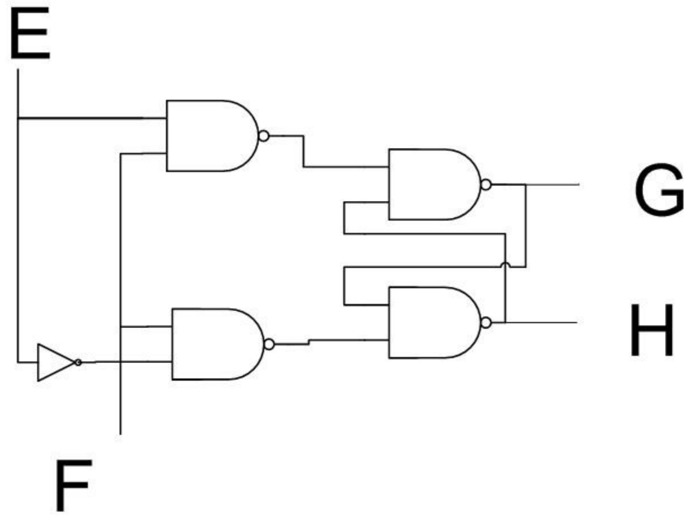


Figure 4.11. A problem on identification and analysis of a clocked D-Latch

Even though abstraction is a fundamental process to solve problems, we found students rarely used abstraction to solve problems. In particular, no student used abstraction in the problem in Figure 4.11 to replace the cross-coupled NAND gates with an SR-latch. While students may have considered this circuit too simple to warrant abstracting out a functional unit, we were surprised that no student mentioned any kind of abstraction when they had failed in other attempts to determine how circuit element worked.

In place of abstraction, students used other approaches that revealed misconceptions about sequential circuit elements, particularly latches. Most students tried to derive Boolean expressions for *G* and *H*, or they set values for *E*, *F*, *G*, and *H* and traced through the circuit. Students using these methods rarely viewed *G* and *H* as both inputs and outputs. Students who used Boolean algebra often visualized the circuit as seen in Figure 4.12. These students did not always consider the output *G* of one SR-latch NAND gate and the corresponding input *M* of the opposite SR-latch NAND gate to be connected. These students derived expressions for *G* in terms of *L* and expressions for *H* in terms of *M*, but these expressions did not help students understand the circuit.

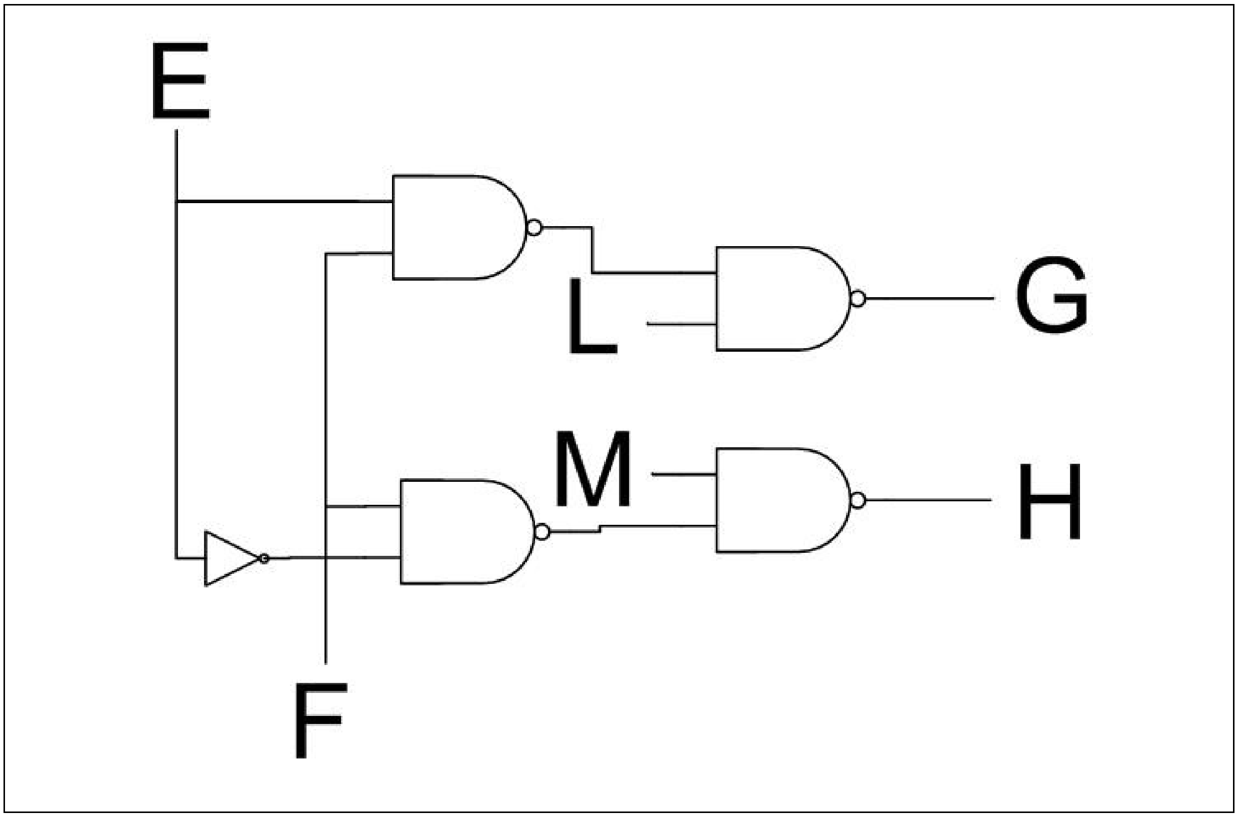


Figure 4.12. Students' conception of the D-Latch from Figure 4.11

Students who traced through the circuit varied *E* and *F* but kept *G* and *H* constant. They created non-symmetric excitation tables like in Table 4.13.

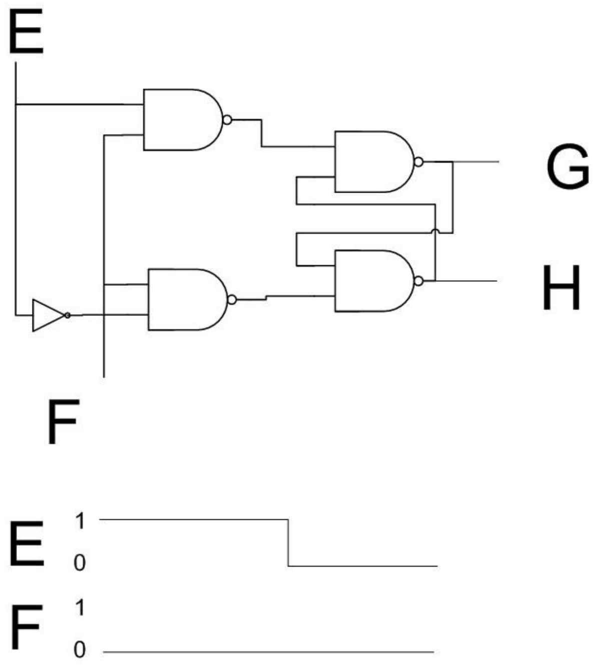
Table 4.13. An example excitation table for the circuit element in Figure 4.11with G = 1 and H = 0

E	*F*	*G* Result	*H* Result
0	1	Set to 0	Set to 1
0	0	Remain the same	Remain the same
1	0	Remain the same	Remain the same
1	1	Remain the same	Remain the same

Although we think that abstraction is important, we do not think it is easy to assess using a multiple-choice problem. We constructed the problem in Figure 4.13 to assess student understanding of G and H as both inputs and outputs.

ECE 290 Spring 2006 Assessment Test Form 1, Problem 6

The inputs *E* and *F* are applied as shown to the circuit below when *G* and *H* are in a steady state. What are the final values of *G* and *H* after these inputs are applied?



a.) $G = 0, H = 0$

b.) $G = 0, H = 1$

c.) $G = 1, H = 0$

d.) $G = 1, H = 1$

e.) Cannot be determined

Solution:

Since *F* is never 1, *G* and *H* will not change based on *E* but will retain their original values. Since these values are never provided, they cannot be determined.

Figure 4.13. A problem on D-latch response to input

Table 4.14 shows the ECE 290 Spring 2006 assessment test results for this problem.

Table 4.14. ECE 290 Spring 2006 assessment results for the problem in Figure 4.13

Form 1	Problem 6	51 of 51 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	4	8%	
b	3	6%	
c	7	14%	
d	4	8%	
e	33	65%	*

Students who selected any answer choice other than answer choice e probably assigned values to *G* and *H*. These students likely believed that *G* and *H* were only outputs, and so they did not examine all possible values of *G* and *H*. Students who selected either answer choice a or answer choice d did not recognize the right half of the circuit as an SR latch, or they did not remember that an SR latch is never in a "steady state" when outputs are both 0 or 1. Students could have selected answer choice e without understanding the operation of a latch, however.

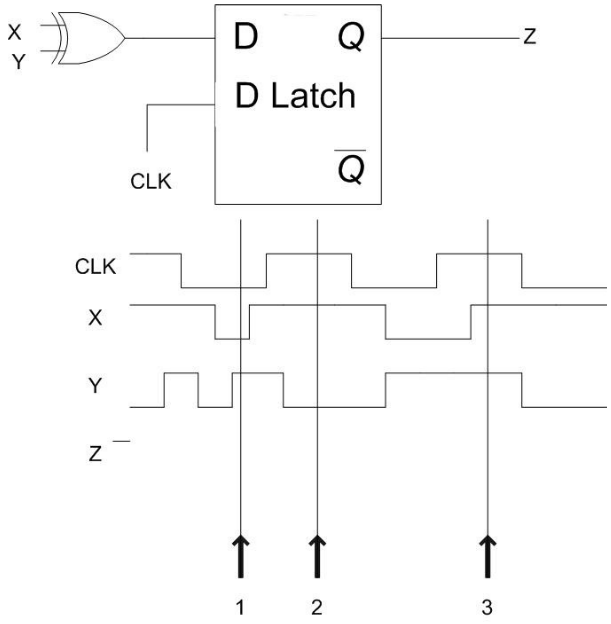
4.2.7 Latch timing

Students commonly misunderstand the difference between flip-flops and latches. One interview question asked students whether there was a difference between latches and flip-flops, and few students answered correctly. Most answers were very general, such as "a flip-flop is more complicated," or, "I know they're different, but I don't remember how." On the Fall 2005 assessment test, 40% of students correctly solved the problem in Figure 4.14. Almost as many students (30%) selected answer choice b, which is consistent with positive-edge triggered flip-flop operation.

We gave this problem to students during Spring 2006 interviews and found that most students did not understand the operation of the clocked latch. We first asked students to describe the timing of a clocked latch; most students gave the definition of a positive-edge triggered flip-flop. When the operation of a positively clocked latch was defined, most students correctly solved this problem.

ECE 290 Spring 2006 Assessment Test Form 1, Problem 7 and Form 2, Problem 5

What are the values of Z at times 1, 2, and 3 if the logic element below is a *positively clocked latch (not a flip-flop)*? Assume that propagation times for the XOR gate and the latch are negligible and that Z is initially one.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

Solution:

Before time 1, while CLK is high, D input is first 1 (when $X = 1$, $Y = 0$), then 0 (when $X = Y = 1$). When CLK first falls from 1 to 0, the value 0 is stored as the state of the latch. Thus, the output Z is 0 at time 1. Since CLK is high during times 2 and 3, the output Z at those times is the same as the D input. At time 2, the output Z is 1 because $X = 1$ and $Y = 0$. At time 3, the output Z is 0 because $X = Y = 1$.

Figure 4.14. A problem on completing a latch timing diagram

We included this problem on both forms of the Spring 2006 assessment test. On both forms, we added the words "(not a positive-edge triggered flip-flop)" to the problem statement. On Form 2, we added the sentence "A positively clocked D latch will change values only when the clock input is high" to the end of the problem statement. The results from both forms are in the Table 4.15. The number before the slash represents data from the problem on Form 1, and the number after the slash represents data from the problem on Form 2.

Table 4.15. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.14

Form 1 / 2	Problem 7 / 5	49 / 55 of 51 / 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	1 / 0	2% / 0%	
b	1 / 5	2% / 9%	
c	29 / 32	57% / 57%	*
d	3 / 2	6% / 4%	
e	2 / 0	4% / 0%	
f	1 / 1	2% / 8%	
g	8 / 12	16% / 22%	
h	1 / 3	2% / 5%	
i	3 / 0	6% / 0%	

As seen above, the extra sentence in Form 2 had little impact on students' performance. A chi-square test also indicates that the sentence was not related to the probability that a student would correctly answer the problem ($\chi^2(8) = 5.23$, $p = 0.73$), but the chi-square test is weak since expected values are low for some answer choices (see Section 4.1.2 for more on the chi-square test).

Students who viewed the latch as a positively-edged flip-flop would select answer choice f. Because Form 2 had an incorrect diagram (the *Z* waveform was initially 0 on the diagram instead of the stated 1), students who received Form 2 and viewed the latch as a positive-edged flip-flop could select answer choice b if they looked primarily at the diagram.

It was not clear initially why answer choice g was selected so often, but the waveforms for *Z* that students drew on the tests revealed two potential reasons. Students may have mistaken the long vertical lines crossing all the waveforms for the clock edges. If these students believed that the latch was actually a flip-flop, they would have looked at the values of *X* and *Y* immediately before the times and obtained *Z* values consistent with answer choice

g. Students also may have viewed the latch as an unclocked D-latch. These students also would have obtained *Z* values consistent with answer choice g from the values of *X* and *Y* immediately before times 1, 2, and 3.

Because some distracters could be chosen for multiple reasons, we cannot determine students' misconceptions from their selection of distracters. Our understanding of students' misconceptions is also confounded because the XOR gate adds unnecessary complexity. If we removed the XOR gate, we could be more certain that incorrect answers were a result of misconceptions about latch timing.

In both CS 231 and ECE 290, latches are covered only briefly, usually on the way to a discussion of flip-flops. Students might see latches on only one homework assignment. Consequently, students may believe latches are less important than flip-flops, so they forget the differences. CS 231 students in particular may consider latches less important because many computer science students do not focus on hardware; CS 231 students' explanations during interviews about the differences between flip-flops and latches tended to be less clear than ECE 290 students' explanations. Even though latches are not well understood by either ECE 290 or CS 231 students, we are not certain that a deep understanding of latch timing is important long term.

4.3 ECE 290 Spring 2006 Assessment Test Form 2 Problems

The sections below describe the problems in Form 2 of the ECE 290 Spring Assessment Test in the order they appear on the test. Form 2 of the ECE 290 Spring Assessment Test is in Appendix C.2.2.

4.3.1 Alternate implementations of expressions

Although there are many ways to implement Boolean expressions as combinational circuits, students often consider only a few of these ways. For example, implementing the complement of an expression as a sum-of-products and then changing the OR gate to a NOR gate is sometimes simpler than implementing the original expression directly. During the Fall 2005 interviews, we found that students were reluctant to use this method when they solved the problem in Figure 4.15.

$F(x, y, z) = \text{AND}(M1,M4,M6)$

$G(x, y, z) = \text{OR}(m1,m3,m5)$

$H(x, y, z) = x \text{ XOR } z$

Draw a circuit which implements the three functions *F*, *G*, and *H*. You may use *only* one 3-to-8 decoder, two 4-input OR gates, and one 4-input NOR gate.

Solution:

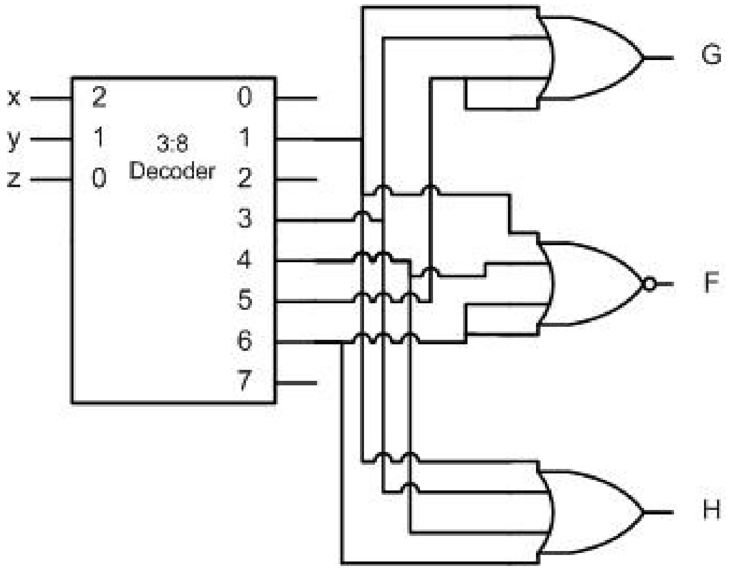


Figure 4.15. A problem on implementing functions with a decoder

During the interviews, many students could implement function *G* easily, but they could not use the NOR gate to implement function *F*.

We constructed the problem in Figure 4.16 based on the interview results and included it on the Spring 2006 assessment test. This problem required students to understand the operation of a decoder and to perceive that *G* could be implemented with either a NOR gate or an OR gate.

Students' performance on this problem is summarized in Table 4.16. While the decoder may introduce unnecessary complexity, many students showed their understanding of decoder operation by correctly selecting answer choice b. Further, nine students who did not select answer choice b did select answer choice e based on the minterm form of *G*. This combination of answer choices indicates understanding of decoder operation.

ECE 290 Spring 2006 Assessment Test Form 2, Problem 1

$G(a, b, c)$= AND(M1,M3,M5) = $(a + b + c')(a + b' + c')(a' + b + c')$

Which of the following could be used to implement Boolean function G? Assume that the constants 0 and 1 are *not* available and that all gates must have unique inputs.
Select all correct answers.

a.) A 3:8 decoder and a 3-input OR gate b.) A 3:8 decoder and a 3-input NOR gate
c.) A 3:8 decoder and a 4-input OR gate d.) A 3:8 decoder and a 4-input NOR gate
e.) A 3:8 decoder and a 5-input OR gate f.) A 3:8 decoder and a 5-input NOR gate

Solution:

The Boolean function *G* has five input combinations that result in an output of 1, and it has three input combinations that result in an output of 0. We can implement *G* by ORing any decoder output that could be 1 when *G* = 1. We can also implement *G* by NORing any decoder output that could be 1 when *G* = 0. If the order of decoder inputs from most to least significant bit is {*a*, *b*, *c*}, then we can OR decoder outputs 0, 2, 4, 6, and 7, or we can NOR decoder outputs 1, 3, and 5. Thus, the correct answer choices are e and b.

Figure 4.16. A problem on implementing a function using a decoder

Table 4.16. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.16

Form 2	Item 1	52 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	6	11%	
b	35	63%	*
c	5	9%	
d	4	7%	
e	24	43%	*
f	7	13%	
b and e	13	23%	*

Students who selected either answer choice b or answer choice e, but not both, most likely did not realize that they could implement expressions with decoders using both OR and NOR gates. The student who selected only answer choices a and f and the student who selected only answer choices c and d tried to implement the expression with both an OR and a NOR gate, but they did not determine the correct number of gate inputs.

While students should understand both that decoders are minterm generators and that an expression can be implemented by implementing the sum-of-products form of the complement and replacing the OR gate with a NOR gate, these two concepts would ideally be explored in separate problems to remove unnecessary complexity. Also, using the maxterm form for G may lead students to recall that an expression can be implemented with a decoder and a NOR gate. If we used the minterm form of G in the problem statement, then students could forget that the expression could be implemented by NORing the decoder outputs that would be a 1 for the combinations of input values that cause the function to be 0.

4.3.2 Duality and implications of equality

During the interviews, very few students correctly derived the dual of the expression in Figure 4.17 on their first attempt. The types of errors varied widely. Some errors were typographical, such as removing the complement from the first occurrence of y while copying from one line to another. Other students confused taking the dual with complementing the expression; their answer was the same as the solution with every literal complemented. The expression requires an understanding of operation precedence; students must remember to separate $(y' + z)$ from $(y + z')$. We found that most students understood operation precedence. The expression does not contain a constant 1 or 0 to see if students remember to complement constants.

What is the dual of $wx(y'z + yz') + w'x'(y' + x)(y + z')$?

Solution:

$$(w + x + (y' + z)(y + z'))(w' + x' + y'x + yz')$$

Figure 4.17. An interview problem on duality

It is unclear exactly why students had so much difficulty with this problem. Students may have forgotten the duality concept because they thought that duality is unimportant. Also, duality is covered very quickly near the beginning of the term and usually is not seen again. However, we believe that some students might have some conceptual understanding of duality without having a procedural understanding of how to derive the dual of an expression. Conversely, some students could perform the derivation without having a conceptual understanding of duality. While we want students to have both a procedural and conceptual understanding of duality, we decided to test conceptual understanding. The problem in Figure 4.18 assesses students' understanding of duality in the context of asking about the implications of equality.

ECE 290 Spring 2006 assessment test results for the problem in Figure 4.18 are tabulated in Table 4.17.

ECE 290 Spring Assessment Test Form 2, Problem 3

Let F, G, and H be Boolean functions of x, y, and z. The truth tables of F and G are the same. Which of the following must be true?

Select all correct answers.

a.) $F = G$

b.) $F' = G'$

c.) Dual(F) * G = 0 (Dual of F AND G)

d.) $F'G$=0

e.) $FH = HG$

f.) Dual(F) = Dual(G)

Solution:

If the truth tables of two functions are the same, then those functions are equal, so answer choice a is true. Answer choice b should be selected because if two functions are equal, then their complements are equal. Answer choice c should not be selected: for example, if $F = G = x$, then Dual(F) = x and Dual(F) * $G = x$. Answer choice d should be selected because a function AND its complement is 0. Answer choice e should be selected because of the commutative property of AND. Answer choice f should be selected because if two functions are equal, then their duals are equal.

Figure 4.18. An assessment test problem on the implications of equality

Table 4.17. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.18

Form 2	Problem 3	56 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	53	95%	*
b	50	89%	*
c	11	20%	
d	49	88%	*
e	50	89%	*
f	48	86%	*
a, b, d, e, f	26	46%	*

Specifically, answer choice f tested whether students understood that, if two functions are equal, then the duals of those functions are also equal. Most ECE 290 students selected answer choice f. While this apparent understanding of a main idea of duality is encouraging, it is possible that students selected answer choice f because the same operation is performed on both sides of the equation and not because they understood duality. Also, the students who selected answer choice c may have thought that the truth table of the dual was the same as the truth table of the complement. All of the students who selected answer choice c selected all of the answer choices.

The five students who selected answer choice f on item 2 of form 1 (see Section 4.2.2) may also have recognized the expression as the dual of the expression in the problem statement, but incorrectly believed dual expressions to be equivalent.

More investigation is needed to understand how students understand duality on a conceptual level. A problem that focuses more specifically on the implications of duality, such as the problem in Figure 4.19, would expose students' conceptual understanding of duality (answer choices would be determined through student interviews).

If E, F, G, and H are Boolean expressions and $EF = G + H$, then what is Dual(E) + Dual(F)?

Figure 4.19. A problem on duality

4.3.3 Minimum number of flip-flops needed in a minimal state diagram

In most introductory digital logic courses, students learn how to design a sequential circuit from a state diagram specification. Transforming a state diagram into a sequential circuit design is an algorithmic task that students can execute without a firm understanding of why or how the transformation works. We designed the problem in Figure 4.20 to capture a part of students' conceptions of this process.

ECE 290 Spring Assessment Test Form 2, Problem 3

A state diagram with n states requires at least m flip-flops to implement a sequential circuit. If a different state diagram has $2n$ states, what is the minimum number of flip-flops needed for an implementation?

a.) m

b.) $m + 1$

c.) $2m$

d.) $2m + 1$

e.) m^2

f.) $m^2 + 1$

g.) None of the above

Solution:

A state diagram with n states requires at least ceiling($\log_2(n)$) = m flip-flops to implement. If the number of states is doubled to $2n$, then at least ceiling($\log_2(2n)$) = $m + 1$ flip-flops are needed to implement the state diagram.

Figure 4.20. A problem on the number of flip-flops needed to implement a state diagram

This problem probes students' understanding of the relationship between states and flip-flops. Many students remember that the minimum number of flip-flops needed for n states is $\log_2(n)$, but this problem cannot be solved by merely recalling that fact.

The version of this problem on the Fall 2005 assessment test did not include the correct answer of $m + 1$ as an option. We believed students would find answer choices that contained various combinations of the number "2" more compelling distracters and that $m + 1$ would be out of place as an answer choice. However, we could not tell whether students selected "None of the above" because they knew the answer was $m + 1$ or for some other reason. Few students selected answer such as $m + 2$ or $2m + 2$ during the Fall 2005 test, so we decided to include the correct answer as an answer choice for the Spring 2006 test and modify the remaining answer choices so that *m* + 1 did not appear different from the other answer choices. The results from the Spring 2006 assessment test are in Table 4.18.

Table 4.18. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.20

Form 2	Problem 3	54 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	1	2%	
b	35	65%	*
c	9	17%	
d	1	2%	
e	5	10%	
f	0	0%	
g	3	6%	

As expected, answer choices c and e were the most effective distracters. The other distracters were not very effective, but are necessary to maintain the symmetry of the answer choices. Since few students chose “None of the above,” it could be removed.

The process implementing a state diagram may not always be taught with the goal of minimum flip-flop design. For example, courses that cover sequential design quickly may teach students one-hot design, which allocates one flip-flop for every state. Courses that cover digital logic as only part of the course may not cover sequential design techniques at all. While many of the basic concepts are the same regardless of the goal, students who did not learn minimal flip-flop design techniques would have a disadvantage on this problem.

4.3.4 Flip-flop and combinational circuit timing

See Section 4.2.5.

4.3.5 Latch timing

See Section 4.2.7.

4.3.6 Karnaugh maps

Even though Karnaugh maps are a major topic in both ECE 290 and CS 231, students are generally reluctant to use them in all but the most obvious of circumstances. During the interviews, we asked students to solve two problems that required the use of Karnaugh maps to complete quickly. We first presented the open-ended problem in Figure 4.21.

Find a two-level multiple-output AND-OR gate network to realize the following expressions using 6 gates.

a'c + *a'd'* + *b'c* and *c'd'* + *ab'* + *ac'*

Solution:

Below are Karnaugh maps for both expressions.

a'c + *a'd'* + *b'c*

cd

		00	01	11	10
	00	1		1	1
ab	01	1		1	1
	11				
	10			1	1

c'd' + *ab'* + *ac'*

cd

		00	01	11	10
	00	1			
ab	01	1			
	11	1	1		
	10	1	1	1	1

We can implement the gate network with four AND gates and two OR gates. The AND gates implement *a'c'd'*, *ab'c*, *a'c*, and *ac'*. We implement the first expression by ORing together *a'c'd'*, *ab'c*, and *a'c*, and we implement the second expression by ORing together *a'c'd'*, *ab'c*, and *ac'*.

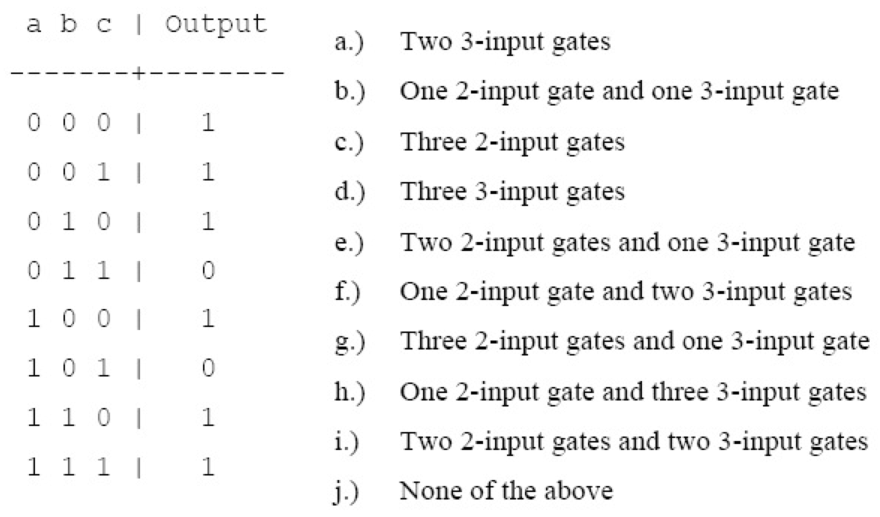
Figure 4.21. A problem on implementing two expressions using a limited number of logic gates

To solve this problem, students must find the common product terms *a'c'd'* and *ab'c*. Although students could find these terms using Boolean algebra, it is easier to construct Karnaugh maps for both expressions. Students usually tried Boolean algebra or truth tables, and many of these students either gave up or gave an incorrect solution. One student said he was looking for common product terms but decided that a truth table was a better tool than a Karnaugh map. Only the top students instinctively used a Karnaugh map for this problem; other students would use a Karnaugh map only after we asked follow-up questions that suggested a Karnaugh map would be useful.

We constructed the multiple-choice question in Figure 4.22 to test whether students would use a Karnaugh map to solve a minimization problem. The problem indicates that a Karnaugh map might be needed because it asks for the “minimum number of gates,” phrasing that both ECE 290 and CS 231 students would associate with Karnaugh maps. Results from the problem are summarized in Table 4.19

ECE 290 Spring 2006 Assessment Test Form 2, Problem 6

The truth table below defines a Boolean function. Suppose you plan to implement this function as a combinational circuit specified by a sum-of-products expression. Assuming only AND and OR gates are available, determine the minimum required number of gates. Assume you have inputs {*a*, *b*, *c*} and their complements {*a*', *b*', *c*'} available as inputs to the circuit.



```
 a b c | Output
-------+--------
 0 0 0 |   1
 0 0 1 |   1
 0 1 0 |   1
 0 1 1 |   0
 1 0 0 |   1
 1 0 1 |   0
 1 1 0 |   1
 1 1 1 |   1
```

a.) Two 3-input gates
b.) One 2-input gate and one 3-input gate
c.) Three 2-input gates
d.) Three 3-input gates
e.) Two 2-input gates and one 3-input gate
f.) One 2-input gate and two 3-input gates
g.) Three 2-input gates and one 3-input gate
h.) One 2-input gate and three 3-input gates
i.) Two 2-input gates and two 3-input gates
j.) None of the above

Solution:

The minimal sum-of-products expression for this function is $a'b' + ab + c'$. This expression can be implemented with two 2-input AND gates and one 3-input OR gate.

Figure 4.22. A problem on the minimum number of gates needed to implement a truth table

Most students who correctly selected answer choice e had drawn correct Karnaugh maps on their test papers. A few students had correct Karnaugh maps but did not derive the minimal expression. Students who selected answer choice f most likely implemented the product-of-sums expression for the function because they did not read the problem statement carefully. Future revisions of the problem statement could put more emphasis on "sum-of-products expression."

Table 4.19. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.22

Form 2	Problem 6	54 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	2	4%	
b	1	2%	
c	0	0%	
d	1	2%	
e	24	43%	*
f	12	21%	
g	4	7%	
h	5	9%	
i	4	7%	
j	1	2%	

We want students to learn to use Karnaugh maps correctly and at the appropriate times. Introductory digital logic courses require students to use Karnaugh maps in limited contexts, such as problems that specifically require students to use Karnaugh maps, and problems that ask students to "find the minimal number of gates" for an implementation of an expression. As a result, students may not realize they can use Karnaugh maps when problems are presented in different contexts, such as the assessment test problem asking students to analyze MSI implementations in Section 4.3.7.

4.3.7 MSI devices

Interviews revealed that students have difficulty understanding and using decoders and multiplexers. These devices are usually introduced to students as the devices are used in real designs; for example, a multiplexer would be introduced as a device that selects one of many inputs to appear on the output. Many course problems ask students to use the devices to implement arbitrary Boolean expressions, however. These problems can confuse students because students cannot use the devices in the ways that the devices were introduced. Rather, students must develop a deeper understanding of how MSI devices work. On the Spring 2006 assessment test, we asked students the problem in Figure 4.23 to test their understanding of MSI components. Spring 2006 assessment test results for this problem are in Table 4.20.

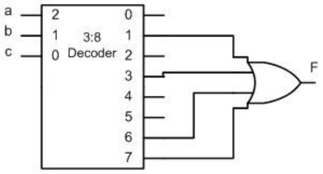
ECE 290 Spring 2006 Assessment Test Form 2, Problem 7

Which of the following is an implementation of the following Boolean function:

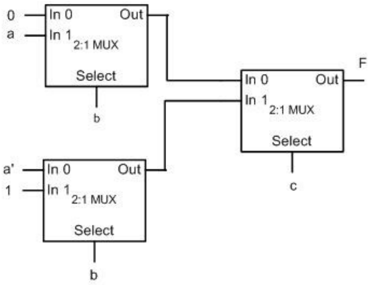
$F(a, b, c) = \mathrm{OR}(m1, m3, m6, m7) = a'b'c + a'bc + abc' + abc$

Select all correct answers.

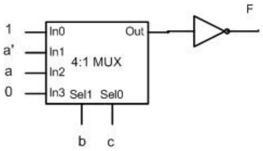
a.)



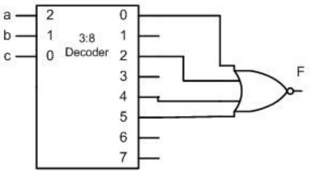
b.)



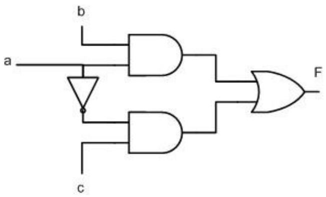
c.)



d.)



e.)



f.) None of the above

Solution:

Answer choices a and d are correct implementations; answer choice a ORs together the decoder outputs that would have value 1 for the input combinations at which *F* outputs 1 and answer choice d NORs together the decoder outputs that would have value 1 for the input combinations at which *F* outputs 0. Answer choice b is a correct two-level multiplexer implementation. Answer choice c would be correct if the In1 and In2 inputs are swapped. Answer choice e is a correct AND-OR implementation.

Figure 4.23. A problem on implementing an expression with MSI devices

Table 4.20. ECE 290 Spring 2006 assessment test results for the problem in Figure 4.23

Form 2	Problem 7	55 of 56 students answered this problem	
Answer	Number of Students	Percentage of Students	Correct answer
a	53	95%	*
b	23	41%	*
c	15	27%	
d	38	68%	*
e	28	50%	*
f	0	0%	
a, b, d, and e	6	11%	*
a and d	38	68%	
b and not c	14	25%	

Except for answer choices a and d, it is difficult to tell why students would choose some and not others. We found that students understand decoders better than they understand multiplexers. Students selected both answer choice a and answer choice d, the correct decisions for answer choices based on decoders, 68% of the time. In contrast, students selected answer choice b and did not select answer choice c, the correct decisions for answer choices based on multiplexers, only 25% of the time. However, we do not understand what conceptual misunderstandings drive this difference.

Finally, we do not understand the low performance on answer choice e. Although students could have easily determined whether answer choice e is correct by using a Karnaugh map, few students drew Karnaugh maps on their tests (see Section 4.3.6 for more information on students and Karnaugh maps). Because most students correctly selected answer choice a, it is unlikely that students misunderstood the problem statement. As a result, low performance on other answer choices more likely results from student misconceptions.

Students' conceptual understanding of MSI components should be explored further so that misunderstandings of individual components are more clearly identified. The problem in Figure 4.23 could be broken up into a problem focusing on decoders and another problem focusing on multiplexers. Each of these new problems could then have a more distracters to focus on student misconceptions.

CHAPTER 5 CONCLUSIONS AND FUTURE WORK

5.1 Student Preconceptions

While the Force Concept Inventory (FCI) has been used to successfully determine the effectiveness of active engagement techniques in introductory physics courses, not all concept inventories will be successful in the same way. Because Newtonian mechanics describes common, physical phenomena, most students hold well-defined preconceptions that can be accurately measured by the FCI. Introductory physics instructors who use the FCI as a pretest can then address student preconceptions as they teach. In contrast, digital logic instructors will not need to overcome student preconceptions because most introductory digital logic students lack the experience to form preconceptions. Consequently, students' performance on a pretest concept inventory in digital logic would likely be similar to their performance if they randomly guessed all answers.

Because students have few preconceptions about digital logic, we assumed all concepts were equally likely to be misunderstood. Based on this assumption, we used a "shotgun" approach and selected different kinds of problems about different topics for our initial interview problem set. Our approach limited the effects of our biases about which concepts should be "easy" or "hard." Nevertheless, I believe we could have better focused our problem set using cognitive development theory [67]. An expert on cognitive development theory may have identified classes of concepts that students would find difficult, such as concepts that required abstraction and knowledge transfer. Then we could have ensured that our interview problem set covered the identified difficult concepts. While we would still want to cover a wider range of concepts than the identified difficult concepts, we could be more confident that we had covered the concepts that students would be most likely to misunderstand.

5.2 Future Work

Students' performance on the Spring 2006 assessment test provided evidence that we had identified student misconceptions. Future research is needed to show why students find these concepts difficult. It is apparent that students tend to find some classes of concepts more difficult than others. These more difficult concepts require abstraction and knowledge transfer. Because we lack a good model of digital logic concepts, it is difficult to determine whether the concepts themselves are difficult or whether the types of problems we used to assess the concepts were difficult. Two kinds of models could be helpful: the first model would describe how students approach and learn digital logic concepts, and the second model would describe digital logic concepts similar to Hestenes et al.'s six dimensions of force. Both models could be useful in developing and analyzing a digital logic concept inventory. I believe that an overarching cognitive model would facilitate analysis of digital logic as a whole, rather than just analysis of individual problems. This deeper analysis could enable instructors to present digital logic concepts to best facilitate student learning.

Although we interviewed students only individually, future developers might understand student misconceptions better if they interviewed small groups of students. While students were often reluctant to explain how they arrived at their solutions to the problems, they might reveal this information as they defended their solutions to other students. I am not sure whether the benefits of interviewing small groups would be worth the increased complexity in logistics and interview protocol.

We found that students in CS 231 performed similarly to students in ECE 290 on the Spring 2006 assessment test, despite pedagogical differences between the two courses. Because students received two different versions of the test under significantly different conditions, we are unable to attribute differences in performance to differences in pedagogy. Future administrations of the assessment test could provide more information on the relationship between pedagogy and conceptual understanding.

I recommend a number of improvements for the assessment test. First, many of the problems should be adjusted to minimize confounding factors from other concepts or from clues in the answer choices. Some adjustments are mentioned in this thesis. Second, the assessment test should undergo rigorous statistical analysis of the data and the test problems by psychometrics experts. This analysis would reveal the best adjustments to make to the problem statements and answer choices. Finally, instructors at other institutions should have the opportunity to administer and review the revised assessment test to gather data from different populations of students. Data from other institutions will ensure that the test problems are not biased toward digital logic courses at this institution.

5.3 Conclusion

The work described in this thesis makes progress toward a concept inventory in digital logic. Through student interviews, we identified common student misconceptions in digital logic. We confirmed these misconceptions when we administered a multiple-choice assessment test that we developed. With revision and review from other disciplines and institutions, I believe that a digital logic concept inventory could become a catalyst to advance student learning.

APPENDIX A FALL 2005 AND SPRING 2006 INTERVIEWS

A.1 ECE 290 Objectives

Representation of information

- Convert between decimal, binary, octal, and hexadecimal representations of integers
- Determine the number of errors that a code can detect or correct
- Understand two's complement representation of integers and determine whether overflow occurs in arithmetic operations
- Distinguish between a variety of decimal and alphanumeric codes

Design and analysis of combinational networks

- Understand the operation of discrete logic gates
- Analyze a combinational network using Boolean expressions
- Convert a verbal specification into a Boolean expression
- Understand basic properties of Boolean algebra: duality, complements, standard forms
- Apply Boolean algebra to prove identities and simplify expressions
- Use Karnaugh maps to find minimal sum-of-products and products-of-sums expressions
- Design combinational networks that use NAND, NOR, and XOR gates
- Design with MSI components such as encoders, decoders, multiplexers, adders, arithmetic-logic units, ROMs, and programmable logic arrays
- Calculate delays in ripple carry adders and combinational arrays

Design and analysis of sequential networks

- Understand the operation of latches; clocked, master-slave, and edge-triggered flip-flops; shift registers; and counters
- Plot and interpret timing diagrams

- Determine the functionality of sequential circuits from state diagrams and timing diagrams
- Translate sequential circuit specifications into state diagrams
- Design sequential circuit components (latches, flip-flops, registers, synchronous counters) using logic gates
- Synthesize general sequential circuits
- Understand trade-offs in register and counter design

Computer organization

- Understand the operation of tri-state buffers and their uses in multiplexing outputs and enabling bidirectional signaling
- Understand the operation of random access memories
- Synthesize a large memory from smaller memories and decoders
- Design datapath components, including register files, buses, and functional units
- Design a hardwired control unit to implement an instruction set
- Design a microprogrammed control unit to implement an instruction set
- Understand trade-offs between hardwired and microprogrammed control
- Understand instruction formats and addressing modes
- Understand the operation of stack instructions, control flow, and interrupts
- Specify new instructions and addressing modes in register transfer language
- Translate register transfer language statements into microcode
- Analyze the effects of individual instructions and machine-level programs
- Write short machine-level programs

Laboratory skills

- Use professional CAD software on engineering workstations for schematic capture and simulation of small digital systems

A.2 Final Objectives List

Representation of information

- Convert between decimal, binary, octal, and hexadecimal representations of integers
- Determine the number of errors that a code can detect or correct
- Understand two's complement representation of integers and determine whether overflow occurs in arithmetic operations
- Distinguish between a variety of decimal and alphanumeric codes

Design and analysis of combinational networks

- Understand the operation of discrete logic gates
- Analyze a combinational network using Boolean expressions
- Convert a verbal specification into a Boolean expression
- Understand basic properties of Boolean algebra: duality, complements, standard forms
- Apply Boolean algebra to prove identities and simplify expressions
- Use Karnaugh maps to find minimal sum-of-products and products-of-sums expressions
- Design combinational networks that use NAND, NOR, and XOR gates
- Design with MSI components such as encoders, decoders, multiplexers, adders, arithmetic-logic units, ROMs, and programmable logic arrays
- Calculate delays in ripple carry adders and combinational arrays

Design and analysis of sequential networks

- Understand the operation of latches; clocked, master-slave, and edge-triggered flip-flops; shift registers; and counters
- Plot and interpret timing diagrams
- Determine the functionality of sequential circuits from state diagrams and timing diagrams
- Translate sequential circuit specifications into state diagrams

- Design sequential circuit components (latches, flip-flops, registers, synchronous counters) using logic gates
- Synthesize general sequential circuits

A.3 Institutional Review Board (IRB) Consent Form

ECE 290/CS 231 Computer Architecture Concepts Consent Form

Purpose and Procedures:
The purpose of this research is to determine how well students are learning the computer architecture concepts outlined in the course objectives of ECE 290 and CS 231. This research is being undertaken by James Longino, Prof. Craig Zilles, and Prof. Michael Loui of the University of Illinois at Urbana-Champaign.

Students will participate in videotaped interviews with the researchers. During these interviews, students will be asked questions about their experiences in ECE 290 and CS 231 and may be asked to solve problems aloud.

Requirements:
All participants must be at least 18 years old.

Participation is Voluntary:
Participation in this research is voluntary. Students may refuse to participate or may discontinue participation at any time. The decision to participate, decline, or withdraw from participation will have no effect on the subjects grades at, status at, or future relations with the University of Illinois. Compensation will be prorated for the period of participation.

Benefits and Risks:
As a result of participating in these interviews, students may have a better idea as to what teachers of computer architecture consider to be foundational concepts. This may aid the student in any future portion of this course and later courses which will build on this material. Students may experience limited, temporary discomfort, such as stress, related to solving problems out loud. Since these interviews do not have any effect on the student's grade for the course, this discomfort should be very limited.

Compensation:
By participating in an interview of approximately one hour, students will be given $20. If the participant elects to end the interview early, compensation will be prorated at $20/hour.

Confidentiality:
Video recordings and written notes resulting from this interview will be analyzed by only James Longino, Prof. Craig Zilles, and Prof. Michael Loui. In the event of publication of this research, no personally identifying information will be disclosed. Under no circumstances will the video recordings be made public.

Whom to Contact with Questions:
Questions about this research should be directed to James Longino (longino2@uiuc.edu). Questions about your rights as a research participant should be directed to the UIUC Institutional Review Board at (217)-333-2670.

I certify that I have read this form and volunteer to participate in this research study.

Printed name: ______________________________________
I give my permission to have my research session videotaped ______ (check here)

Signature: ________________________________ Date: _______________

A.4 Interview Questions Usage

		Subj. 1	Subj. 2	Subj. 3	Subj. 4	Subj. 5	Subj. 6	Subj. 7	Subj. 8	Subj. 9	Subj. 10	Subj. 11	Subj. 12	Subj. 13	Subj. 14	Subj 15.	Subj. 16
		290	290	231	231	290	290	290	231	231	231	231	231	231	231	290	290
Problem	Description	9/27/2005	9/30/2005	10/7/2005	10/14/2005	10/18/2005	10/21/2005	11/8/2005	11/17/2005	2/13/2006	2/17/2006	2/22/2006	2/23/2006	2/24/2006	2/27/2996	3/13/2006	3/13/2006
1.1	Convert binary	Yes	Yes	Yes		Yes											
1.2	Comb. Gates	Yes		Yes	Yes	Yes		Yes	Yes								
1.3	Dual	Yes		Yes		Yes		Yes									
1.4	Add zeros						Yes										
1.5	MIMO red.	Yes		Yes		Yes	Yes										
1.6	MUX synth.	Yes		Yes	Yes		Yes	Yes									
1.7	Diagram red.																
1.8	Complement		Yes		Yes												
1.9	completeness		Yes		Yes	Yes											
1.10	MUX add																
1.11	Timing		Yes		Yes												
1.12	MSI		Part a		Part a	Yes	Yes										
1.13	Logical Equiv.								Yes								
1.14	# of gates								Yes								
1.15	Log. Compl. Ex.																
1.16	Pinned Values																
1.17	MSI options								Yes								
1.18	Comb. Extra															Yes	Yes
2.1	Latch and comb.																
2.2	FF types	Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes		Yes		Yes	Yes	Yes	Yes	Yes
2.2.1	MS vs. Edge									Yes	Yes		Yes				
2.2.2	D Latch Ident									Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2.2.3	SR latch design									Yes	Yes	Yes	Yes			Yes	Yes
2.3	Rev. Count.	Yes					Yes					Yes	Yes				
2.4	Seq. Det.			Yes		Yes				Yes					Yes		
2.5	Input Delay		Yes				Yes	Yes			Yes		Yes			Yes	
2.6	Diag. Synth.					Yes			Yes	Yes	Yes	Yes	Yes			Yes	Yes
2.7	FF Timing																
2.8	Seq. Timing																
2.9	Min # of states									Yes	Yes	Yes		Yes	Yes		
2.10	Pseudo FF																
2.11	Diagram -> func.																
2.12	Synth. Process									Yes	Yes	Yes	Yes	Yes	Yes	Yes	Yes
2.13	Timing -> Func										Yes	Yes		Yes	Yes	Yes	Yes
2.14	Timing -> State										Yes	Yes				Yes	
2.15	FF/Comb outputs								Yes	Yes	Yes	Yes	Yes	Yes			Yes
2.16	Latch Timing									Yes		Yes	Yes	Yes		Yes	
2.17	FF explaination												Yes	Yes	Yes	Yes	Yes
2.18	FF Options (FF)														Yes	Yes	Yes
2.19	FF Options (Z)														Yes	Yes	Yes
2.20	Comb or Seq															Yes	Yes

A.5 Interview Questions

1.1 Convert to decimal

11001.1011

1.2 Describe AND, OR, NOT, XOR, etc. in terms of everyday situations.

1.3 What is the dual of $wx(y'z + yz') + w'x' (y' + x)(y + z')$

1.4 A switching network has three inputs (x, y, and z) and two outputs (a and b). The output variables a and b represent the most and least significant bits, respectively, of a binary number N. N equals the number of inputs which are 0. For example, if $x = 1$, $y = 0$, $z = 1$, than $a = 0$, $b = 1$.

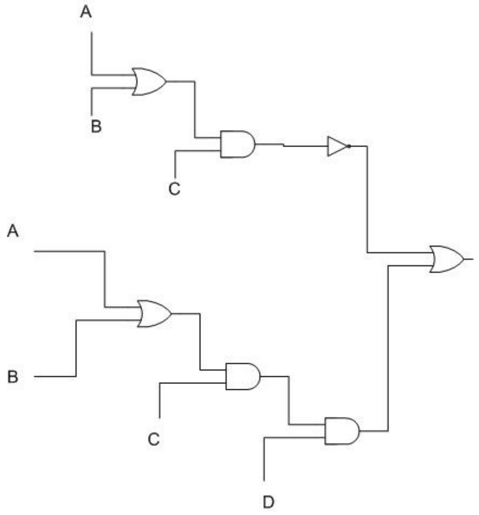
Find the minterm and maxterm expressions for a and for b.

1.5 Find a two-level multiple-output AND-OR gate network to realize the following functions using six gates.

$a'c + a'd' + b'c$ and $c'd' + ab' + ac'$

1.6 Show how two 2-to-1 MUXes can be connected to form a 3-to-1 MUX with $AB = 00$ select I_0, $AB = 01$ select I_1, and $AB = 1X$ select I_2

1.7 For the following network, find the output and design a similar network having the same output.

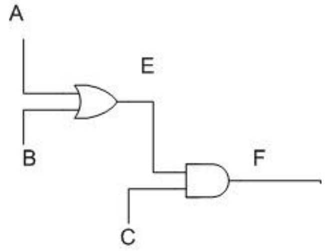


1.8 What is the complement of $wx(y'z + yz') + w'x'$?

1.9 Why can we design any combinational system using only NAND gates?

1.10 Design a network that will add either 2 or 3 to a 3-bit binary number X. Let the inputs X_2, X_1, X_0 represent X, and C be a control signal. Let S_2, S_1, S_0 represent the output S. When $C = 0$, let $S = 2 + X$. When $C = 1$, let $S = 3 + X$. Assume that S will never be greater than 111. Design the whole circuit using only NAND and XOR gates.

1.11 Complete a timing diagram for the following network. Assume all AND gates have a propagation delay of 3 ns and that all OR gates have a propagation delay of 2 ns.



At time 0, $A = B = C = 0$

At time 4, $A = C = 1$

At time 10, $A = 0$

1.12

$$f(x, y, z) = \text{AND}(\text{M1},\text{M4},\text{M6})$$

$$g(x, y, z) = \text{OR}(\text{m1},\text{m3},\text{m5})$$

$$h(x, y, z) = x \text{ XOR } z$$

a.) Draw a circuit which implements the three functions f, g, and h. You may use *only* one 3-to-8 decoder, two 4-input OR gates, and one 4-input NOR gate.

b.) Draw a circuit which implements f, g, and h. You may use only three 4-to-1 MUX's and, if necessary, one inverter.

c.) Implement f, g, and h using *only* a ROM. Specify the ROM size (the number of words and the number of bits per word). Give the ROM truth table and draw the corresponding ROM programming.

1.13 Which of the following are equivalent to $(A + B')(BC + A'C')$?

a) $(A + B')((B' + C')(A' + C'))'$ b) $((A' + B)(B'C' + AC))'$

c) $(A'B)+(BC + A'C')'$ d) $((A'B) + (B' + C')(A + C))'$

e) $(A'B)'((B' + C')' + A'C')$ f) $(AB') + (B + C)(A' + C')$

There may be more than one right answer.

1.14 If you implement the following truth table as a sum of products, what are the minimum required number of AND and OR gates. Assume you have inputs and their complements available as inputs to the circuit.

A	B	C	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

a) one 2-input gate

b) one 3-input gate

c) two 2-input gates

d) two 3-input gates

e) one 2-input gate and one 3-input gate

f) three 2-input gates

h) two 2-input gates and one 3-input gate

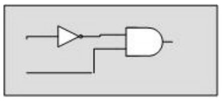
i) one 2-input gate and two 3-input gates

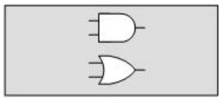
j) three 2-input gates and one 3-input gate

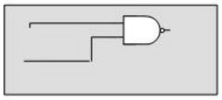
k) one 2-input gate and three 3-input gates

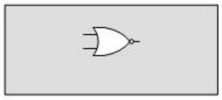
l) two 2-input gates and two 3-input gates

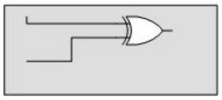
1.15 Which of the following are complete logic families (i.e., all possible combinational logic circuits can be implemented using just these gates and the constants 0 and 1).





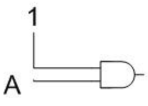


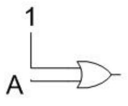


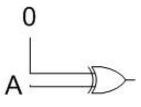


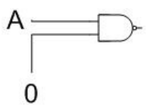
There may be more than one right answer.

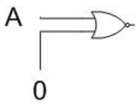
1.16 Which of the following will result in the output being the input *A*?

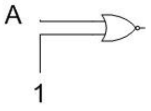










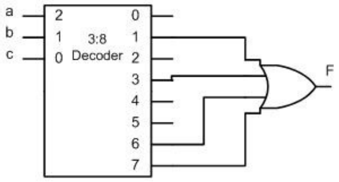


There may be more than one right answer.

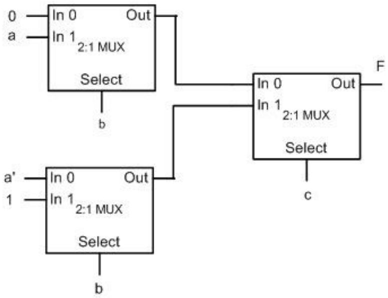
1.17 Which of the following is an implementation of the following function:

$F(a, b, c) = \mathrm{OR}(m1, m3, m6, m7) = a'b'c + a'bc + abc' + abc$

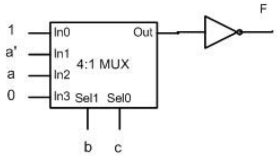
a.)



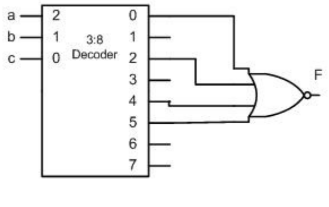
b.)



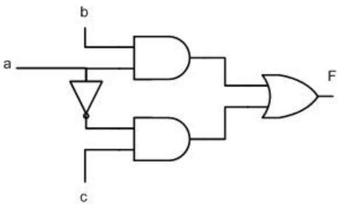
c.)



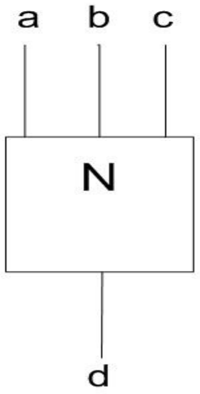
d.)

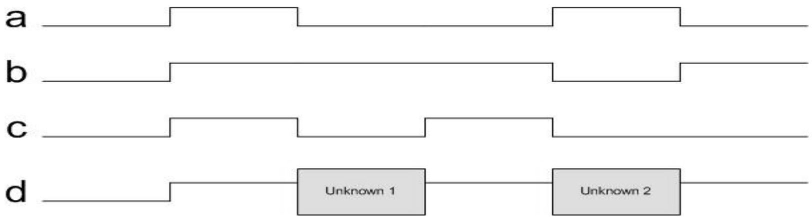


e.)



1.18 The block *N*, below, is a combinatorial function such that $d = N(a, b, c)$. It has one output, $\{d\}$, and three inputs, $\{a, b, c\}$.





What are the values of d in the boxes?

The answer of 0,1 corresponds to Unknown 1 = 0, Unknown 2 = 1.

a. 0, 0

b. 0, 1

c. 0, can't be determined

d. 1, 0

e. 1, 1

f. 1, can't be determined

g. can't be determined, 0

h. can't be determined, 1

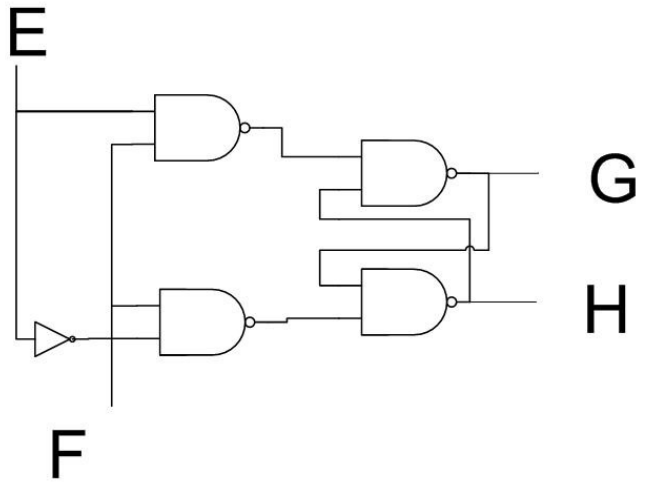
i. can't be determined, can't be determined

2.1 Why don't we put a latch with combinational elements? How do we solve this problem?

2.2 What kinds of flip-flops are there? How are they different? List as many as you can remember.

2.2.1 What kinds of flip-flops will act differently if implemented as master-slave instead of edge triggered?

2.2.2 Identify this circuit element:



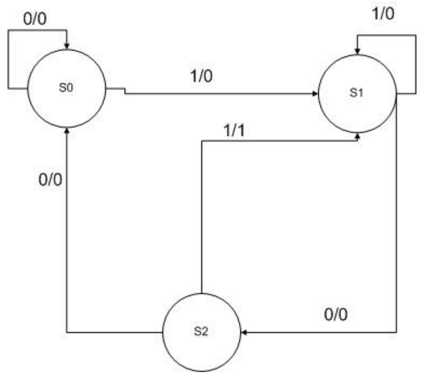
2.2.3 Design an SR latch using 2-input logic gates.

2.3 Design a counter that counts backwards from 1111 to 0000 using any kind of flip-flops and gates.

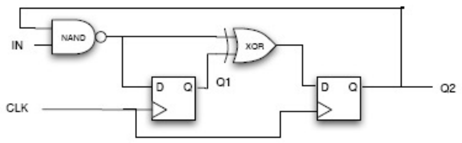
2.4 A network has one input, {X}, and two outputs, {$Y_{1,}$ Y_2}. An output of Y_1 = 1 occurs every time the input sequence 001 is completed provided that the sequence 111 has never occurred. An output Y_2 = 1 occurs every time a sequence 111 occurs. Construct a state graph and state table.

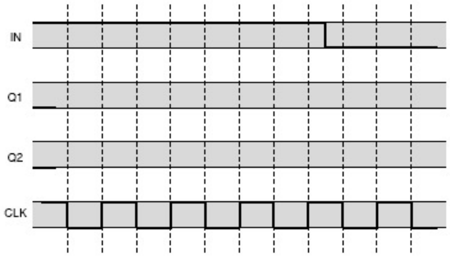
2.5 A network has one input, {X}, and one output, {Y}. The output is the same as the input two cycles previously. The first two values of the output are zero. Construct a state graph and state table for this network.

2.6 Synthesize the following state diagram using any type of flip-flop. The labels on edges are input/output.

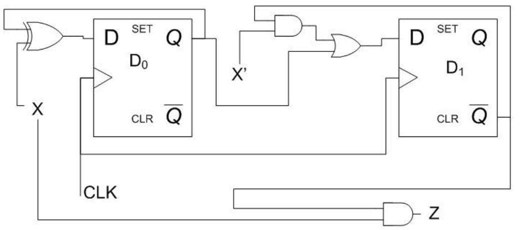


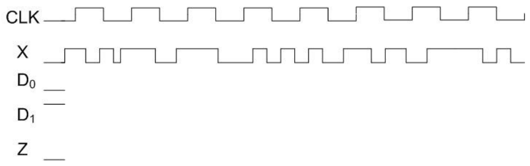
2.7 For the circuit below, compute the state (0 or 1) for points *Q1* and *Q2* as a function of time (the initial state is 0 for both points). Both D-type flip-flops are triggered on the rising edge.





2.8 Complete the timing diagram for flip-flop states *D0* and *D1* and output *Z.* All flip-flops are positive-edge triggered.





2.9 A sequential state diagram with *n* states and requires at least *m* flip-flops. If a different state diagram has 2**n* states, what is the minimum number of flip-flops?

a.) *m*	b.) *m* + 2
c.) 2**m*	d.) *m*^2
e.) 2*(*m*^2)	f.) None of the above

2.10 A clocked circuit has two inputs {*c*, *d*} and one output {*Q*}. The output changes only on a positive clock edge.

If *c* = *d* = 0, then the circuit output *Q* = 0 on the next positive clock edge.

If *c* = 0 and *d* = 1, then the circuit output doesn't change.

For example, if the previous output was *Q* = 1, and the current output is *Q* = 0, then the next output would be *Q* = 0.

If *c* = 1 and *d* = 0, then the circuit output becomes the output immediately before the last clock edge.

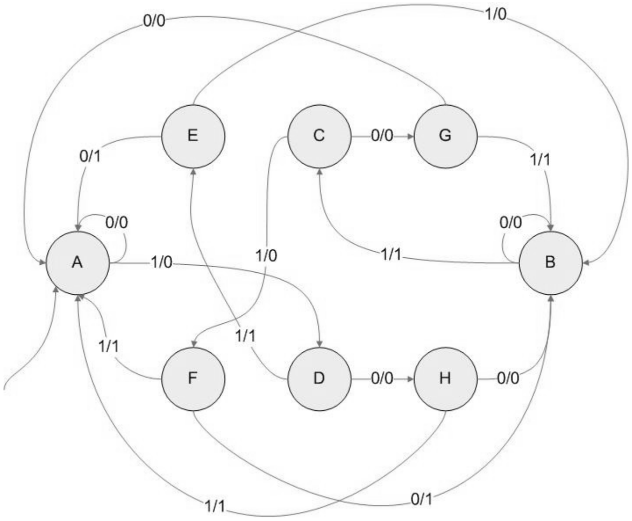
For example, if the previous output was *Q* = 1, and the current output is *Q* = 0, then the next output would be *Q* = 1.

If *c* = *d* = 1, then the circuit outputs *Q* = 1 on the next positive clock edge.

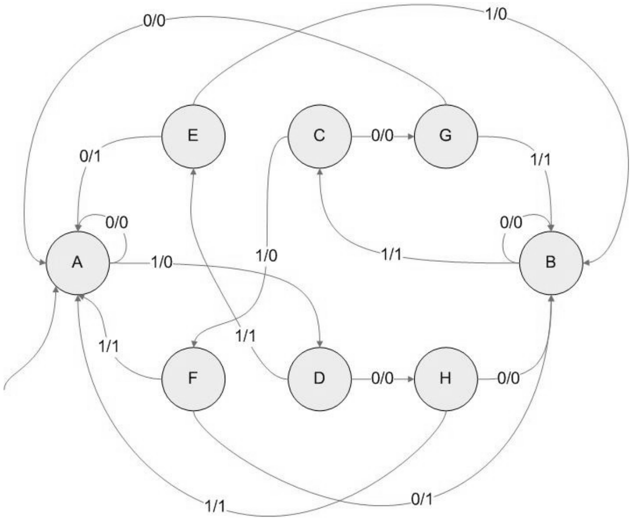
Implement this circuit using any digital logic elements you'd like.

2.11 What is the function of this state diagram?

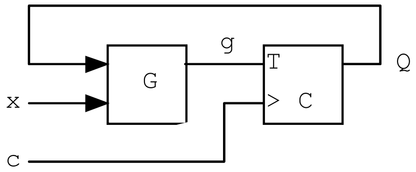
Note: edges are labeled input/output. The unlabeled edge points to the start state.

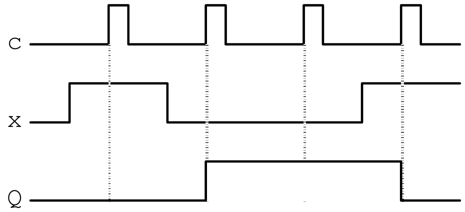


2.12 Given the following state diagram, describe the process you would use to create an implementation of the state diagram. Explain each step and why you would use that step.

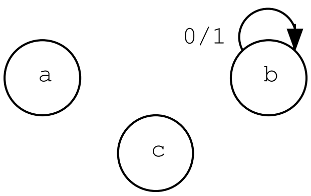


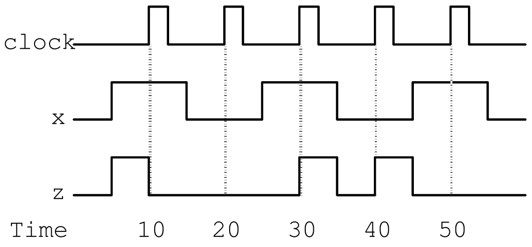
2.13 The sequential circuit below has a positive-edge-triggered T flip-flop and a combinational circuit G whose output is the function *g*(*x*,*Q*). Complete the timing diagram by showing the waveform for *g*, assuming that the delay in G is negligible. Also specify *g* with a truth table. Explain your reasoning.





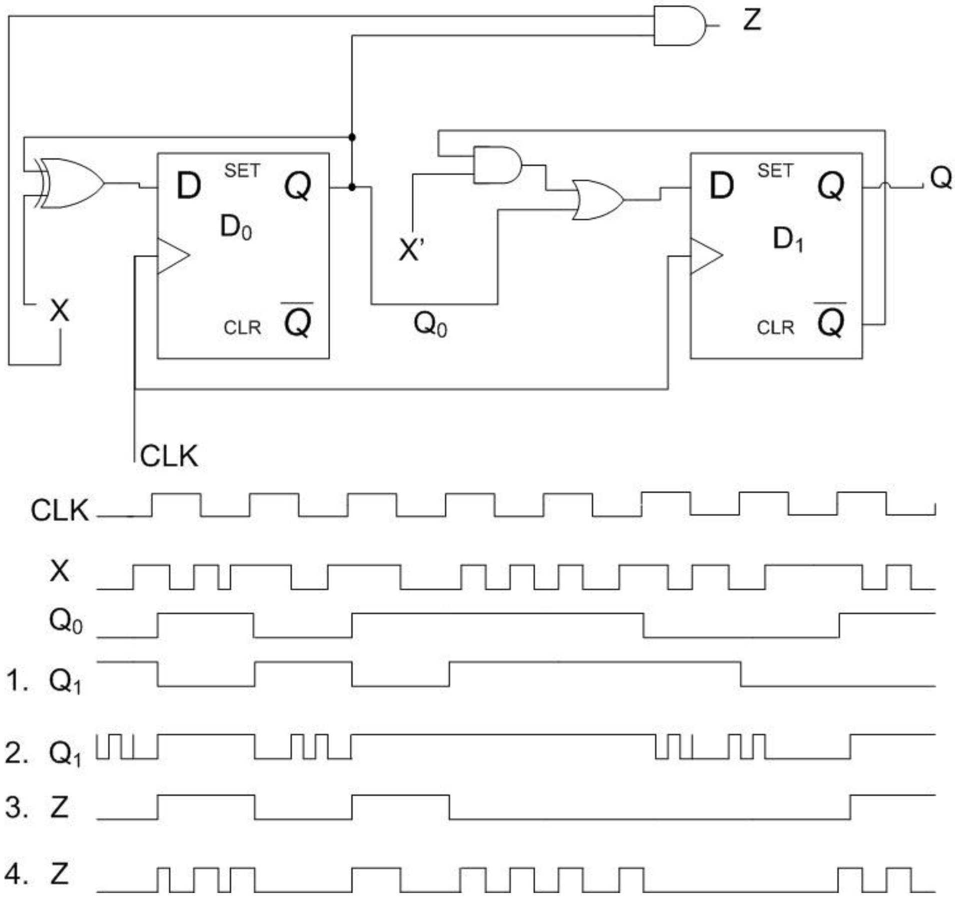
2.14 You have know that a sequential circuit has three states, {*a*, *b*, *c*}, with one input variable, {*x*} and one output variable, {*z*}. A portion of the state diagram is shown below; arcs are labeled with *x*/*z*. Through careful testing, you have obtained the timing diagram shown below, where the initial state is *a*. The delays in the combinational part of the circuit are negligible.





Determine the complete state diagram.

2.15 Select which combination of *Z* and Q_1 are correct. Assume gate delays and flip-flop propagation delays are negligible. All flip-flops are positive edge triggered D flip-flops.



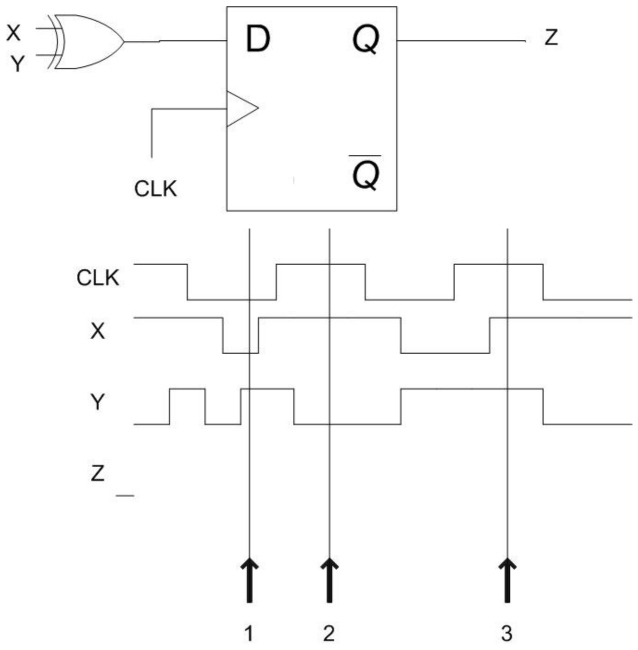
a.) 1 and 3

b.) 1 and 4

c.) 2 and 3

d.) 2 and 4

2.16 What are the values of *Z* at times 1, 2, and 3 if the logic element below is a *positively clocked D latch*? Assume that propagation times for the XOR gate and the latch are negligible and that *Z* is initially zero.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

2.17 Explain a flip-flop in a way that an intelligent high school senior would understand.

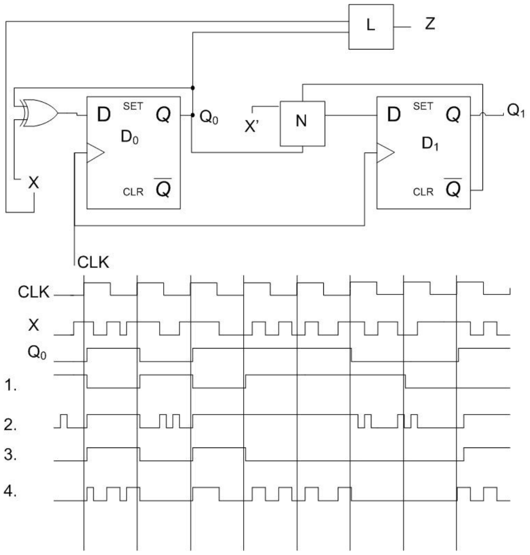
a.) Why do we use flip-flops?

b.) How does the input relate to the output?

c.) What is the deal with the clock? Why is it important?

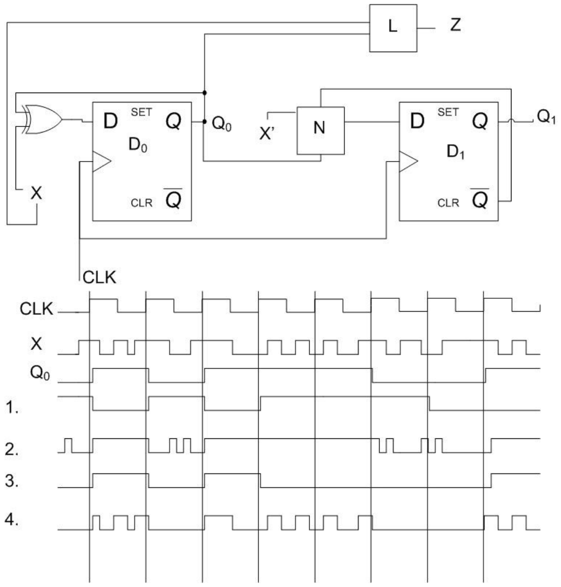
d.) What, if anything, differentiates a flip-flop from a latch?

2.18 The sequential circuit below has two positive-edge D flip flops D_0 and D_1 and two combinatorial circuits *N* and *L*. The output of *N* is *N*(*X*', Q_0), and the output of *L* is *L*(*X*, Q_0) = *Z*. Which of the following waveforms could be waveforms for Q_1? There may be more than one right answer.



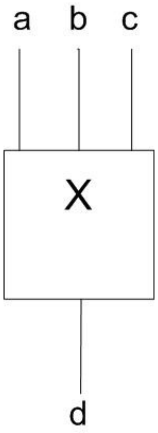
a.) 1 b.) 2 c.) 3 d.) 4 e.) None of the above

2.19 The sequential circuit below has two positive-edge D flip flops D_0 and D_1 and two combinatorial circuits *N* and *L*. The output of *N* is *N*(*X*', Q_0), and the output of *L* is *L*(*X*, Q_0, Q_1) = *Z*. Which of the following waveforms could be waveforms for *Z*?



a.) 1 b.) 2 c.) 3 d.) 4 e.) None of the above

2.20 The circuit below has 3 inputs {a, b, c} and one output {d}. If you were given a timing diagram, what would you look for to determine if the logic block X was combinatorial or sequential logic?



APPENDIX B FALL 2005 ASSESSMENT TEST

B.1 ECE 290 Fall 2005 Assessment Test Results

We administered the Fall 2005 assessment test to 28 students. The number in the second column underneath the problem number indicates the number of students who skipped the problem. Percentages are calculated out of number of attempts.

Problem	1													
Answer		a	b	c	d	e								
	0	0	0	7	2	20								
		0%	0%	25%	7%	64%								
Problem	2													
Answer		a	b	c	d	e	f							
	0	26	12	5	20	18	0							
		93%	43%	18%	71%	64%	0%							
Question	3													
Answer		a	b	c	d	e	f	g						
	2	26	26	3	24	23	24	21						
		100%	100%	12%	92%	88%	92%	81%						
Problem	4													
Answer		a	b	c	d	e	f							
	2	5	15	3	2	17	8							
		19%	58%	12%	8%	65%	31%							
Problem	5													
Answer		a	b	c	d	e	f							
	5	0	1	3	19	15	2							
		0%	4%	13%	83%	65%	9%							
Problem	6													
Answer		a	b	c	d	e	f	g	h	i	j	k	l	m
	6	0	0	0	1	0	0	1	14	0	1	0	2	3
		0%	0%	0%	5%	0%	0%	5%	64%	0%	5%	0%	9%	14%
Problem	7													
Answer		a	b	c	d	e	f	g						
	2	25	0	25	2	22	2	0						
		96%	0%	96%	8%	85%	8%	0%						
`	8													
Answer		a	b	c	d	e	f	g						
	4	0	2	7	1	2	0	12						
		0%	8%	29%	4%	8%	0%	50%						
Problem	9													
Answer		a	b	c	d	e	f							
	2	8	10	24	24	2	0							
		31%	38%	92%	92%	8%	0%							
Problem	11													
Answer		a	b	c	d									
	4	1	15	1	7									
		4%	63%	4%	29%									
Problem	12													
Answer		a	b	c	d	e	f	g	h	i	j			
	6	2	7	9	0	1	0	1	1	0	0			
		9%	32%	41%	0%	5%	0%	5%	5%	0%	0%			

B.2 ECE 290 Fall 2005 Assessment Test Questions

1.) What is the decimal equivalent of 1001100?

a.) -12

b.) -51

c.) 76

d.) None of the above

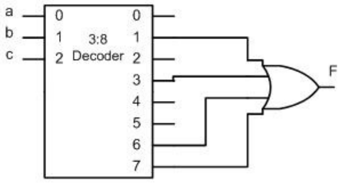
e.) Insufficient information

2.) Which of the following is an implementation of the following function:

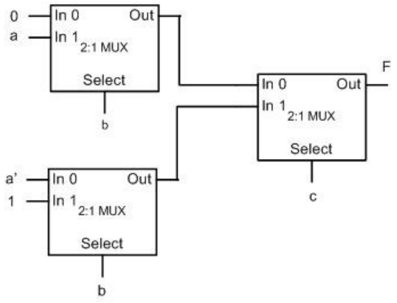
$F(a, b, c) = \mathrm{OR}(m1, m3, m6, m7) = a'b'c + a'bc + abc' + abc$

There may be more than one answer

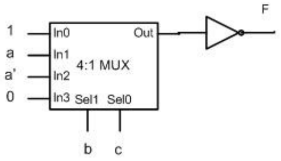
a.)



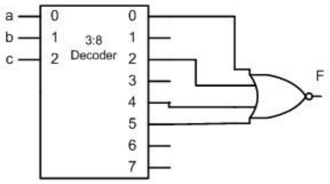
b.)



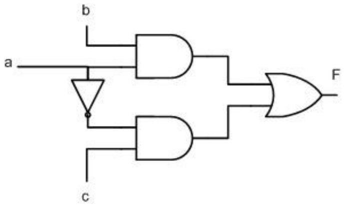
c.)



d.)



e.)



f.) None of the above

3.) Let F, G, and H be functions of x, y, and z. The truth tables of F and G are the same. Which of the following must be true? There may be more than one answer

a.) $F = G$

b.) $F' = G'$

c.) Dual(F) * G = 0 (Dual of F ANDed with G)

d.) $F'G = 0$

e.) $FH = HG$

f.) $FH = GH$

g.) Dual(F) = Dual(G)

4.) $G(a,b,c)$= AND(M1,M3,M5) = $(a + b + c')(a + b' + c')(a' + b + c')$

Which of the following could be used to implement function G? Assume that the constants 0 and 1 are not available. There may be more than one right answer.

a.) A 3:8 decoder and a 3-input OR gate

b.) A 3:8 decoder and a 3-input NOR gate

c.) A 3:8 decoder and a 4-input OR gate

d.) A 3:8 decoder and a 4-input NOR gate

e.) A 3:8 decoder and a 5-input OR gate

f.) A 3:8 decoder and a 5-input NOR gate

5.) Which of the following are equivalent to $(a + b')(bc + a'c')$?

There may be more than one right answer.

a.) $(a + b')((b' + c')(a' + c'))'$

b.) $((a' + b)(b'c' + ac))'$

c.) $(a'b)+(bc + a'c')'$

d.) $((a'b) + (b' + c')(a + c))'$

e.) $(a'b)'((b' + c')' + a'c')$

f.) $(ab') + (b + c)(a' + c')$

6.) If you implement the following truth table as a sum of products, what are the minimum required number of AND and OR gates. Assume you have inputs and their complements available as inputs to the circuit.

A	B	C	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

a.) One 2-input gate

b.) One 3-input gate

c.) Two 2-input gates

d.) Two 3-input gates

e.) One 2-input gate and one 3-input gate

f.) Three 2-input gates

g.) Three 3-input gates

h.) Two 2-input gates and one 3-input gate

i.) One 2-input gate and two 3-input gates

j.) Three 2-input gates and one 3-input gate

k.) One 2-input gate and three 3-input gates

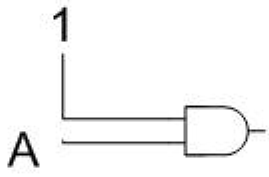
l.) Two 2-input gates and two 3-input gates

m.) None of the above

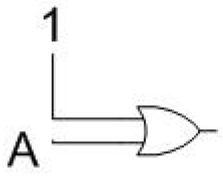
7.) Which of the following will result in nontrivial output (not always 0 or 1)?

There may be more than one right answer.

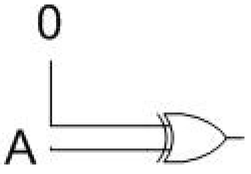
a.)

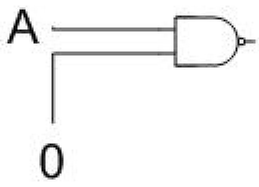


b.)

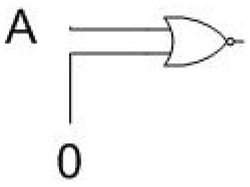


c.)

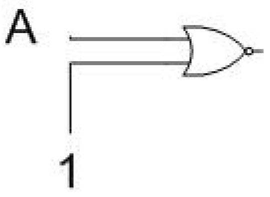




d.)



e.)



f.)

h.) None of the above

8.) A sequential state diagram with *n* states and requires at least *m* flip-flops. If a different state diagram has 2**n* states, what is the minimum number of flip-flops?

a.) m

b.) $m + 2$

c.) $2*m$

d.) $2*m + 1$

e.) m^2

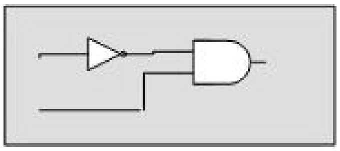
f.) $2*m^2$

g.) None of the above

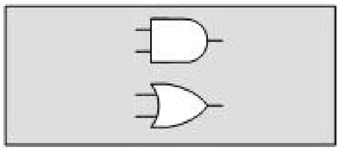
9.) Which of the following are complete logic families (i.e., all possible combinational logic circuits can be implemented using just these gates and the constants 0 and 1).

There may be more than one right answer.

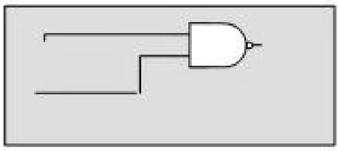
a.)



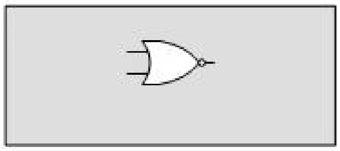
b.)



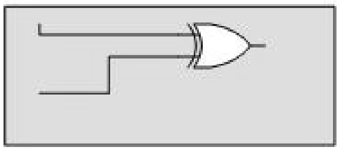
c.)



d.)



e.)



f.) None of the above

10.) For each of the following situations, would you want to use combinatorial logic (C), sequential logic (S), or neither (N)? Circle one.

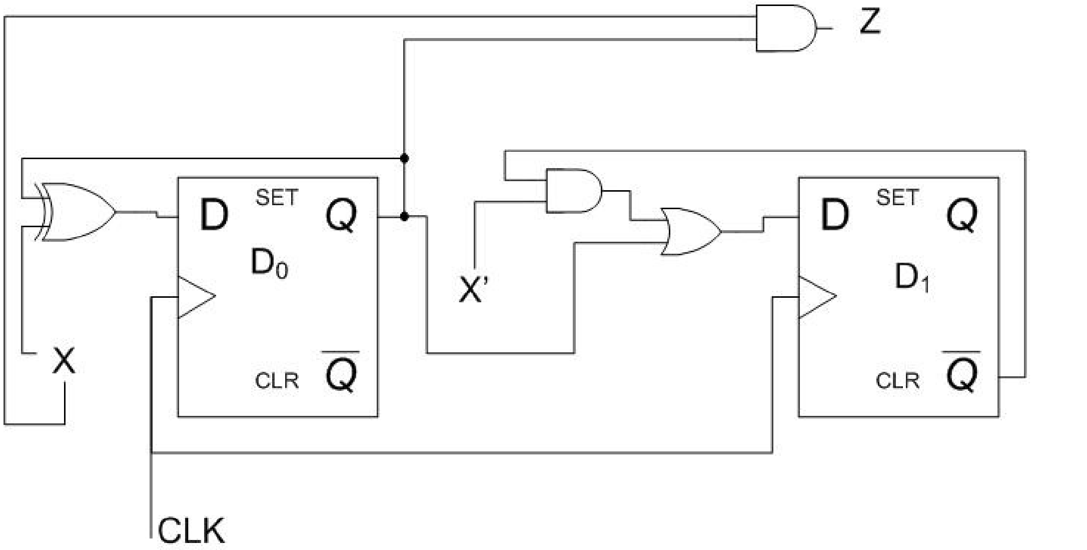
a.) You want to go on a trip by plane. To save money, you look for a plane ticket that has an overnight Saturday stay, has at least one stop, and takes off early in the morning. However, you have a lot of frequent flier miles on a particular airline, so the above conditions apply only for flights on different airlines. C S N

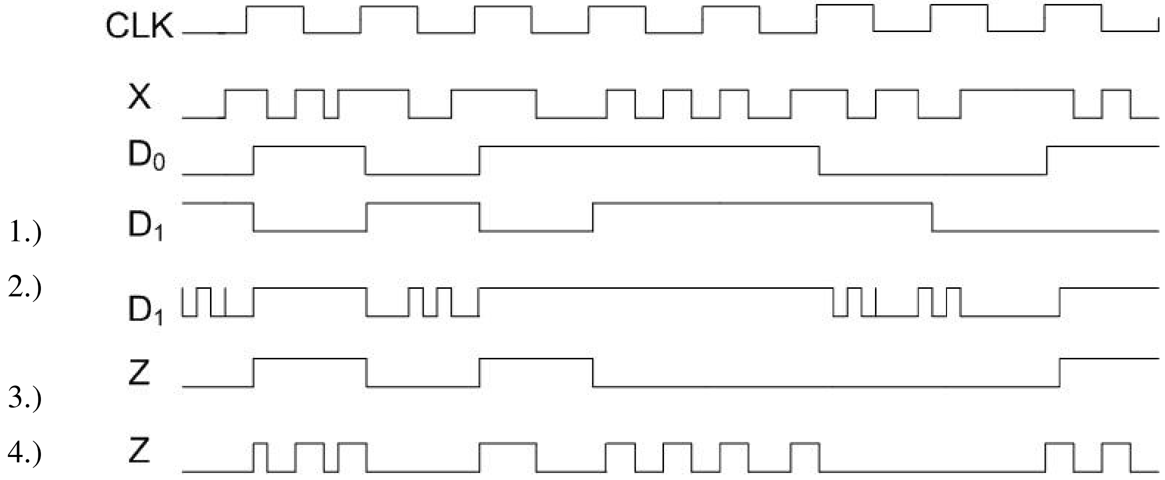
b.) You run quality assurance for a potato chip company. If you see four broken potato chips or make less than twenty potato chips a second, then you know the assembly line has problems and you must stop it to repair it. C S N

c.) You want to design a full adder with a minimum number of logic gates. C S N

d.) You want to find words of any length that are palindromes. A palindrome is a word that is the same spelled both forwards and backwards. For example, “mom” or “solos.” C S N

11.) Select which combination of *Z* and D_1 are correct. Assume gate delays and flip-flop propagation delays are negligible. All flip-flops are positive edge triggered D flip-flops.





a.) 1 and 3

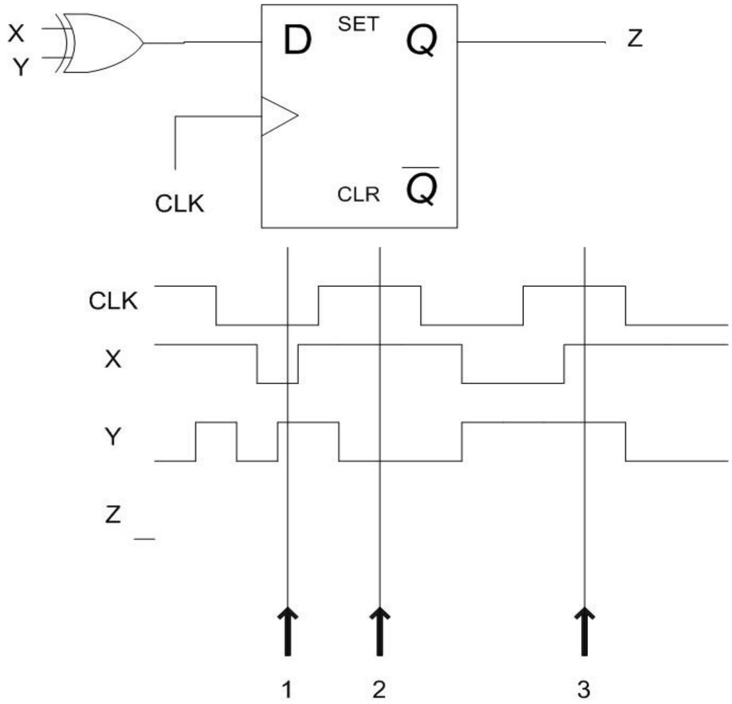
b.) 1 and 4

c.) 2 and 3

d.) 2 and 4

e.) None of the above

12.) What are the values of *Z* at times 1, 2, and 3 if the logic element below is a positively clocked latch? Assume that propagation times for the XOR gate and the latch are negligible and that *Z* is initially zero.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 0

e.) 0, 1, 1

f.) 1, 0, 0

g.) 1, 0, 1

h.) 1, 1, 0

i.) 1, 1, 1

j.) Not enough information

APPENDIX C SPRING 2006 ASSESSMENT TEST

C.1 Test Results

Figure C.1 shows student performance on Form 1 of the Spring 2006 assessment test. Figure C.2 shows student performance on Form 2 of the Spring 2006 assessment test.

Problem		Form 1		
	ECE students	Percentage	CS students	Percentage
	51		51	
Problem 1 both right	33	64.71%	31	60.78%
Problem 2 all right	15	29.41%	27	52.94%
Problem 3 all right	29	56.86%	27	52.94%
Problem 4 all right	8	15.69%	9	17.65%
Problem 5 all right	27	52.94%	25	49.02%
Problem 6 right	33	64.71%	39	76.47%
Problem 7 right	29	56.86%	6	11.76%
Average score (of 28)	22.03921569	0.787114846	22.17647059	0.792016807

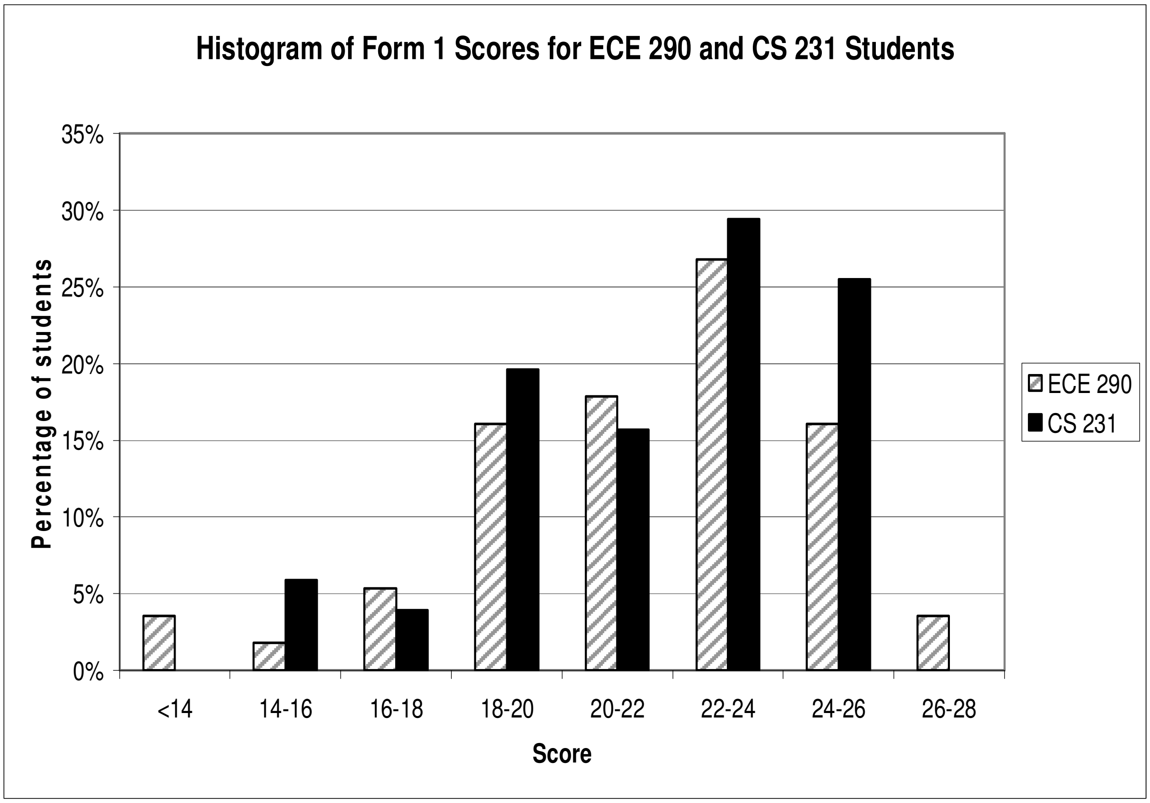


Figure C.1. Histogram for ECE 290 and CS 231 Form 1 scores

Problem		Form 2		
	ECE students	Percentage	CS students	Percentage
	56		43	
Problem 1 both right	13	23.21%	11	25.58%
Problem 2 all right	26	46.43%	10	23.26%
Problem 3 all right	35	62.50%	28	65.12%
Problem 4 all right	25	44.64%	23	53.49%
Problem 5 all right	32	57.14%	21	48.84%
Problem 6 right	24	42.86%	34	79.07%
Problem 7 right	6	10.71%	13	30.23%
Average score (of 26)	19.89285714	0.76510989	19.65116279	0.755813953

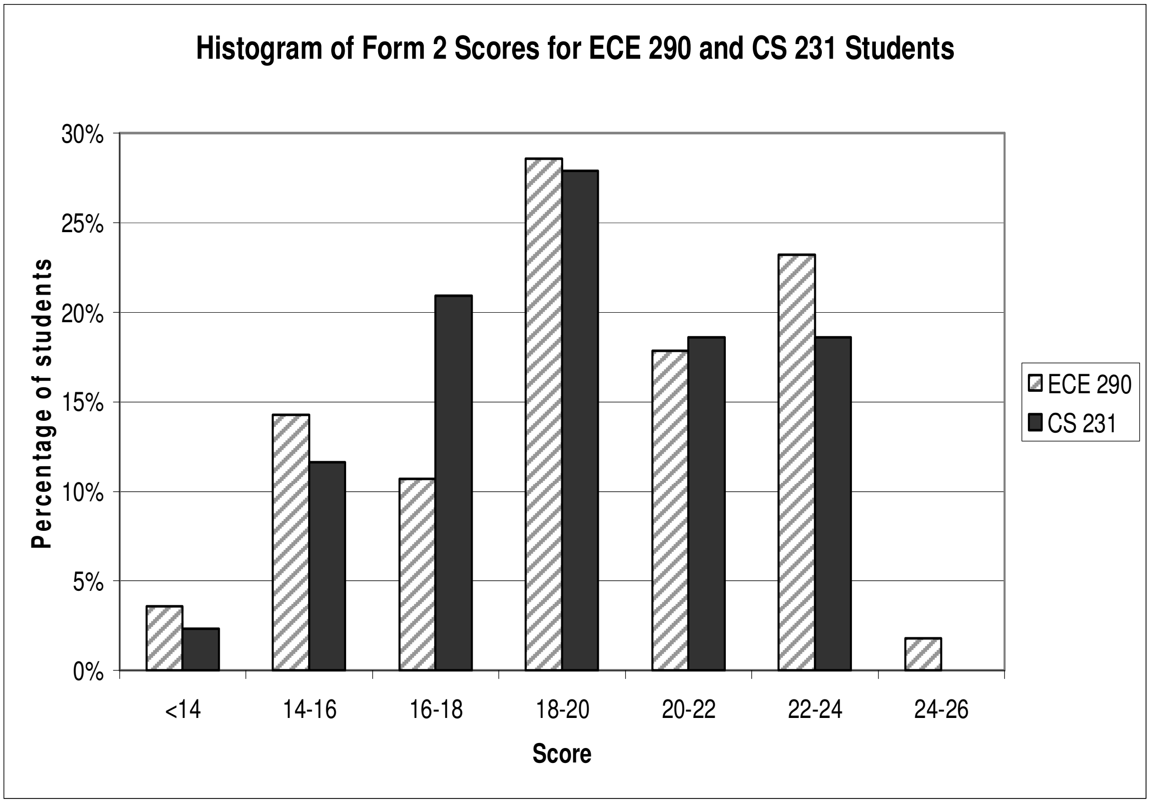


Figure C.2. Histogram for ECE 290 and CS 231 Form 2 scores

C.2 ECE 290 Spring 2006 Assessment Test

C.2.1 Form 1

May 3, 2006

Your Name: ___________________________________

Section (number or time): ___________

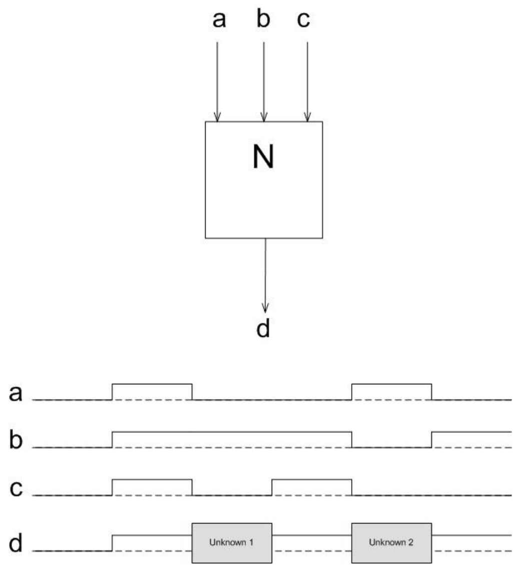
Teaching Assistant:____________________________________

Today's attendance quiz consists of seven problems covering topics from combinational and sequential logic design. These problems can help you find areas you might want to study for the final exam.

In addition, your answers to these problems may be used in research being conducted by James Longino, Professor Craig Zilles of the C.S. department, and Professor Loui on improving education in digital logic. Your performance on this quiz (like other attendance quizzes) will have no impact on your course grade. To ensure your confidentiality, please remove this top page and turn it in separately for attendance credit. In the event of publication of this research, no personally identifying information will be disclosed.

The questions and their answers will be posted on Mallard after class today so you can review them later.

1.) The block N below represents some combinational circuit that implements an unknown Boolean function. It has one output $\{d\}$ and three inputs $\{a, b, c\}$.



What are the values of d in the boxes?

Unknown 1	Unknown 2
a. 0	a. 0
b. 1	b. 1
c. Cannot be determined	c. Cannot be determined

2.) Which of the following expressions are equivalent to $(a + b')(bc + a'c')$?

Select all correct answers.

a.) $(a + b')((b' + c')(a' + c'))'$

b.) $((a' + b)(b'c' + ac))'$

c.) $(a'b)+(bc + a'c')'$

d.) $((a'b) + (b' + c')(a + c))'$

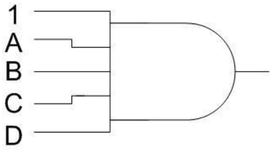
e.) $(a'b)'((b' + c')' + a'c')$

f.) $(ab') + (b + c)(a' + c')$

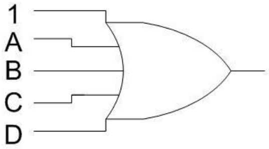
3.) Which of the following will result in nontrivial output (not always 0 or 1)?

Select all correct answers.

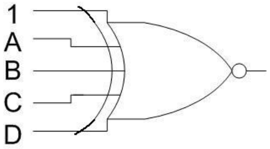
a)

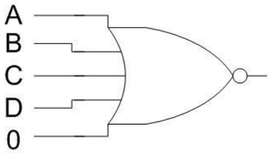


b)

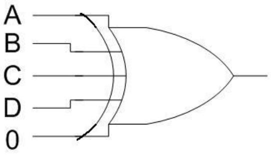


c)

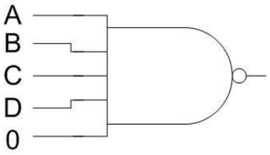




d)



e)



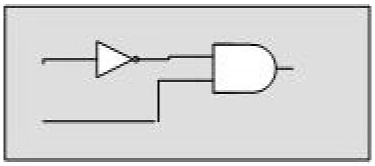
f)

g.) None of the above

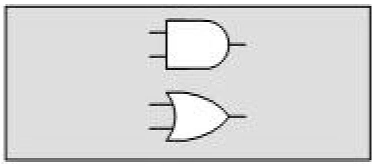
4.) Which of the following are complete logic families (i.e., all possible Boolean functions can be implemented using just these gates and the constants 0 and 1).

Select all correct answers.

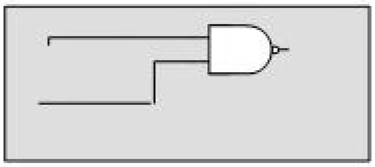
a.)



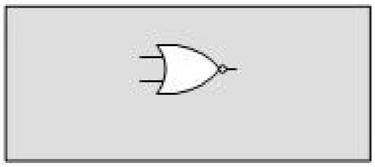
b.)



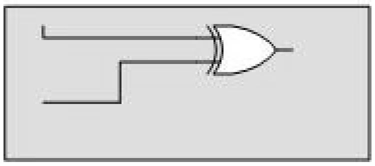
c.)



d.)



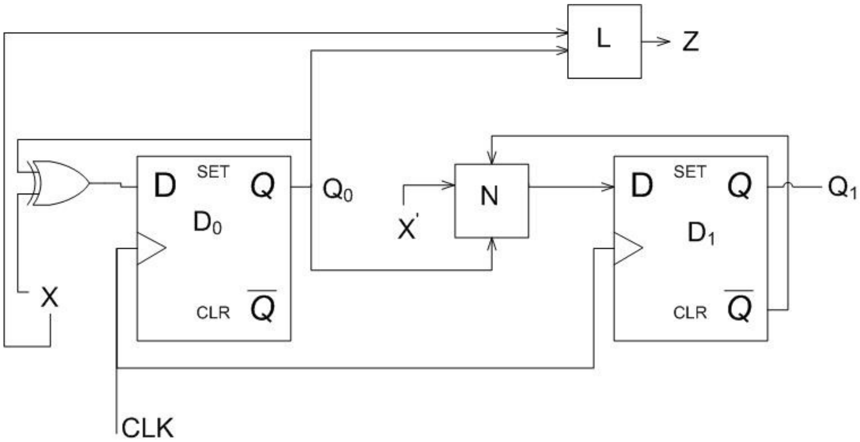
e.)

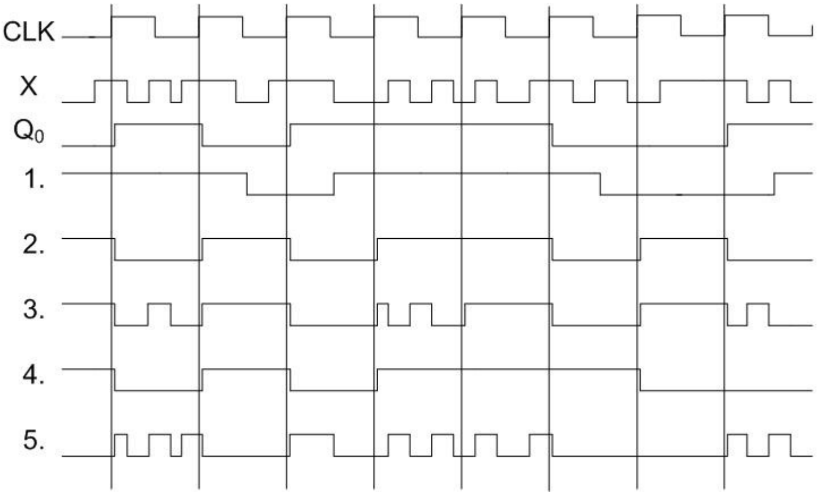


f.) None of the above

5.) The sequential circuit below has two positive-edge triggered D flip flops D_0 and D_1 and two unknown combinational circuits *N* and *L* which implement unknown Boolean functions. Which of the following waveforms could be waveforms for Q_1?

Select all correct answers.





a.) 1

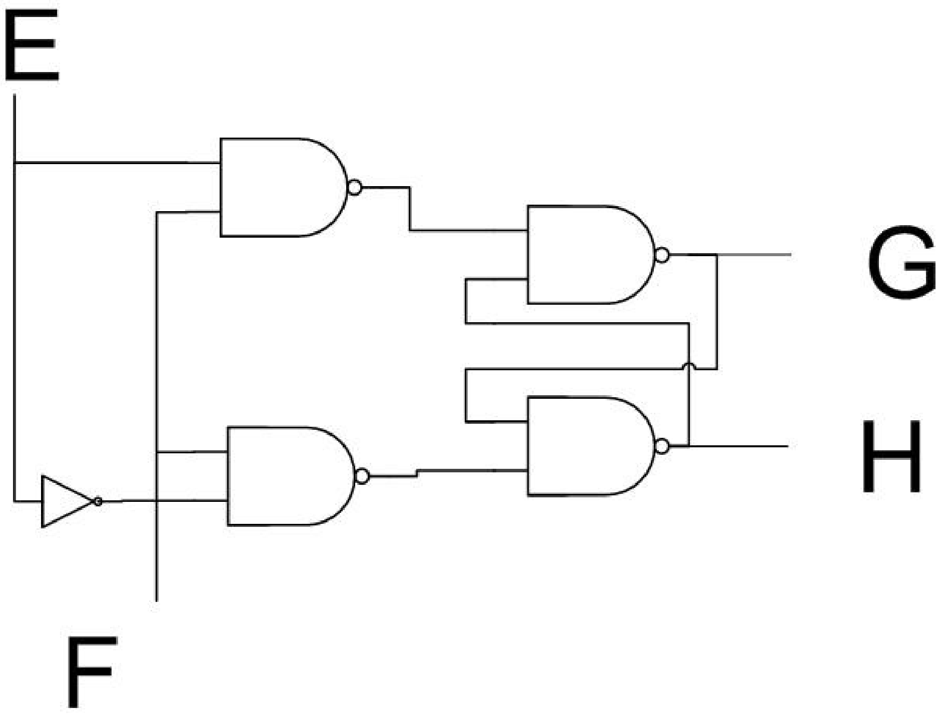
b.) 2

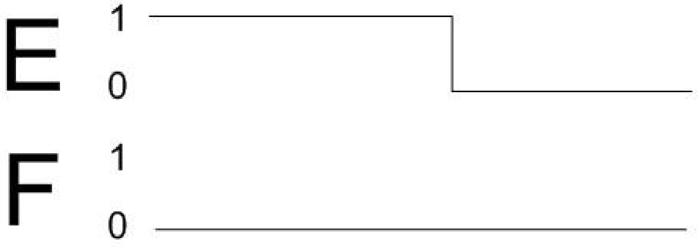
c.) 3

d.) 4

e.) 5

6.) The inputs *E* and *F* are applied as shown to the circuit below when *G* and *H* are in a steady state. What are the final values of *G* and *H* after these inputs are applied?





a.) $G = 0, H = 0$

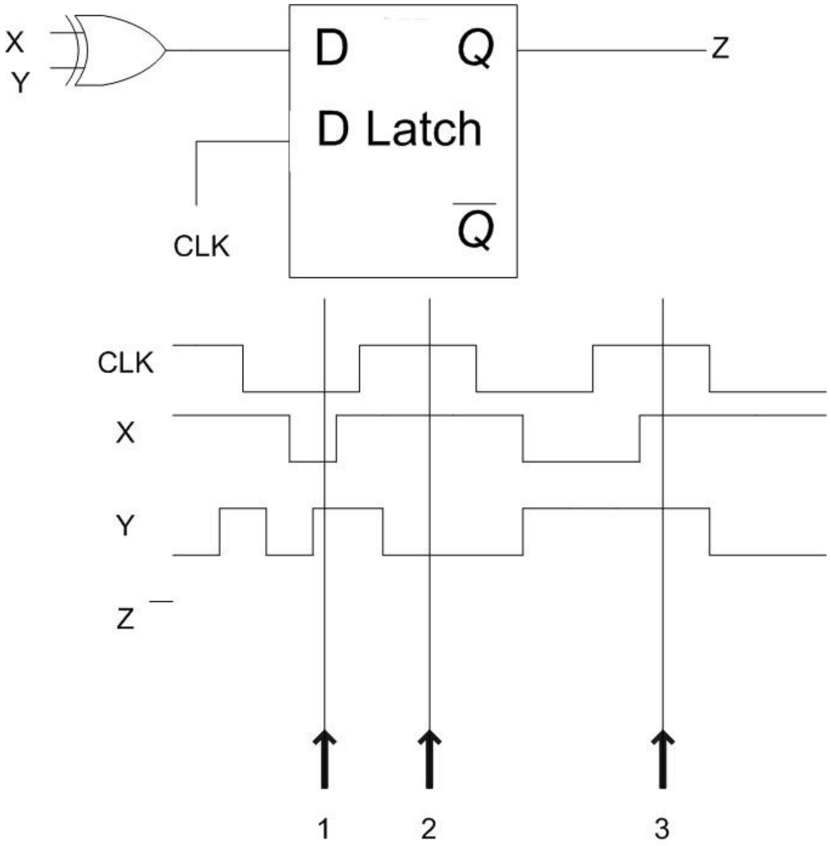
b.) $G = 0, H = 1$

c.) $G = 1, H = 0$

d.) $G = 1, H = 1$

e.) Cannot be determined

7.) What are the values of *Z* at times 1, 2, and 3 if the logic element below is a *positively clocked D latch (not a positive-edge triggered D flip-flop)*? Assume that propagation times for the XOR gate and the latch are negligible and that the initial value of *Z* is 1.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

C.2.2 Form 2

May 3, 2006

Your Name: ____________________________________

Section (number or time): ___________

Teaching Assistant:_____________________________________

Today's attendance quiz consists of seven problems covering topics from combinational and sequential logic design. These problems can help you find areas you might want to study for the final exam.

In addition, your answers to these problems may be used in research being conducted by James Longino, Professor Craig Zilles of the C.S. department, and Professor Loui on improving education in digital logic. Your performance on this quiz (like other attendance quizzes) will have no impact on your course grade. To ensure your confidentiality, please remove this top page and turn it in separately for attendance credit. In the event of publication of this research, no personally identifying information will be disclosed.

The questions and their answers will be posted on Mallard after class today so you can review them later.

1.) $G(a,b,c)$= AND(M1,M3,M5) = $(a + b + c')(a + b' + c')(a' + b + c')$

Which of the following could be used to implement Boolean function G? Assume that the constants 0 and 1 are **not** available and that all gates must have unique inputs.

Select all correct answers.

a.) A 3:8 decoder and a 3-input OR gate

b.) A 3:8 decoder and a 3-input NOR gate

c.) A 3:8 decoder and a 4-input OR gate

d.) A 3:8 decoder and a 4-input NOR gate

e.) A 3:8 decoder and a 5-input OR gate

f.) A 3:8 decoder and a 5-input NOR gate

2.) Let F, G, and H be Boolean functions of x, y, and z. The truth tables of F and G are the same. Which of the following must be true?

Select all correct answers.

a.) $F = G$

b.) $F' = G'$

c.) Dual(F) * G = 0 (Dual of F ANDed with G)

d.) $F'G = 0$

e.) $FH = HG$

f.) Dual(F) = Dual(G)

3.) A state diagram with n states requires at least m flip-flops to implement a sequential circuit. If a different state diagram has $2n$ states, what is the minimum number of flip-flops needed for an implementation?

a.) m

b.) $m + 1$

c.) $2m$

d.) $2m + 1$

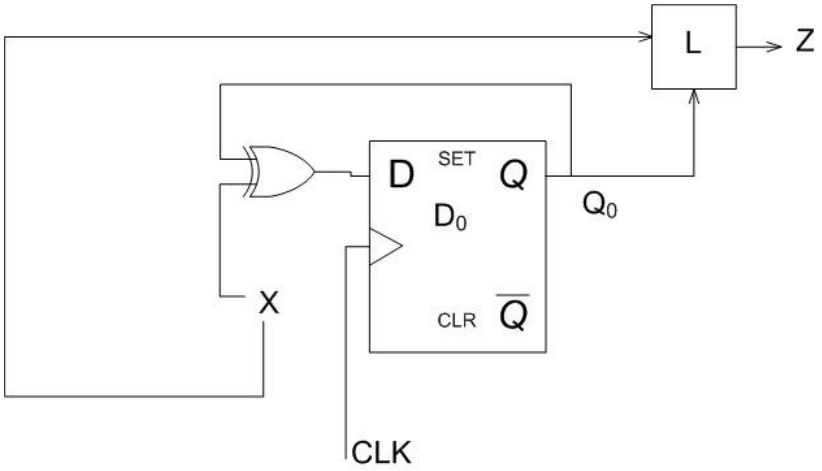
e.) m^2

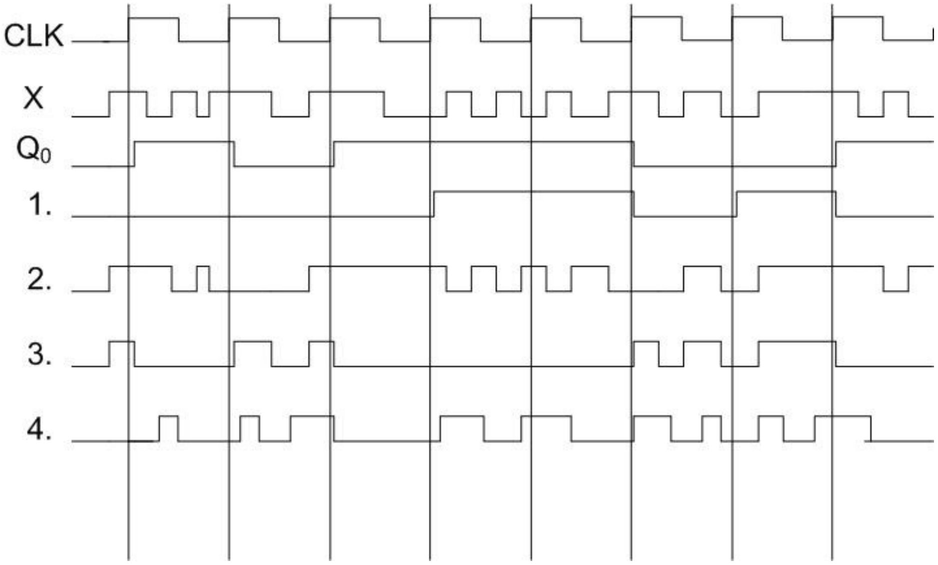
f.) $m^2 + 1$

g.) None of the above

4.) The sequential circuit below has a positive-edge triggered D flip flop D_0 and a combinational circuit *L* which implements some unknown Boolean function. Which of the following waveforms could be waveforms for *Z*?

Select all correct answers.





a.) 1

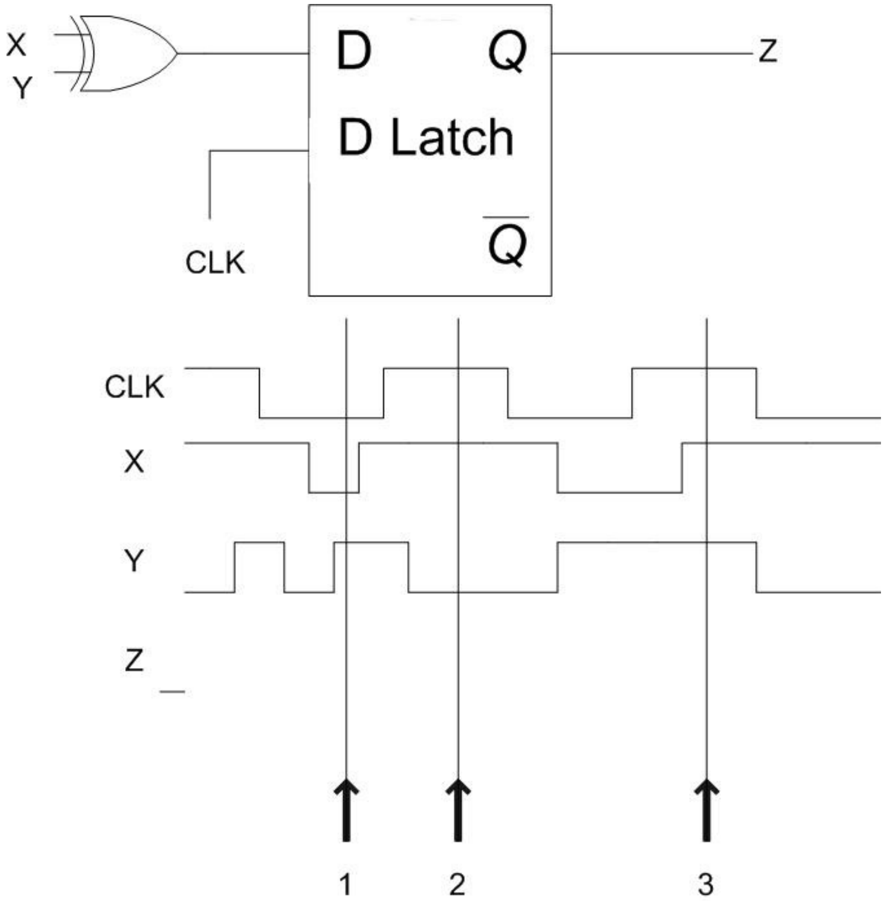
b.) 2

c.) 3

d.) 4

e.) None of the above

5.) What are the values of *Z* at times 1, 2, and 3 if the logic element below is a *positively clocked D latch (not a positive-edge triggered flip-flop)*? Assume that propagation times for the XOR gate and the latch are negligible and that *Z* is initially one. A positively clocked D latch will change values only when the clock input is high.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

6.) The truth table below defines a Boolean function. Suppose you plan to implement this function as a combinational circuit specified by a sum-of-products expression. Assuming only AND and OR gates are available, determine the minimum required number of gates. Assume you have inputs {*a*, *b*, *c*} and their complements {*a*', *b*', *c*'} available as inputs to the circuit.

```
 a b c | Output
-------+--------
 0 0 0 |   1
 0 0 1 |   1
 0 1 0 |   1
 0 1 1 |   0
 1 0 0 |   1
 1 0 1 |   0
 1 1 0 |   1
 1 1 1 |   1
```

a.) Two 3-input gates

b.) One 2-input gate and one 3-input gate

c.) Three 2-input gates

d.) Three 3-input gates

e.) Two 2-input gates and one 3-input gate

f.) One 2-input gate and two 3-input gates

g.) Three 2-input gates and one 3-input gate

h.) One 2-input gate and three 3-input gates

i.) Two 2-input gates and two 3-input gates

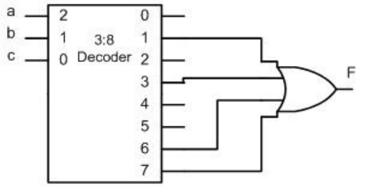
j.) None of the above

7.) Which of the following is an implementation of the following Boolean function:

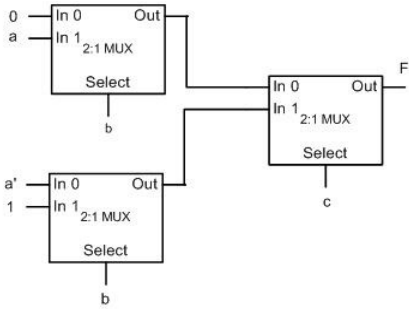
$F(a, b, c) = \mathrm{OR}(\mathrm{m1}, \mathrm{m3}, \mathrm{m6}, \mathrm{m7}) = a'b'c + a'bc + abc' + abc$

Select all correct answers.

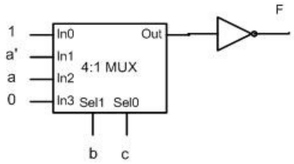
a.)



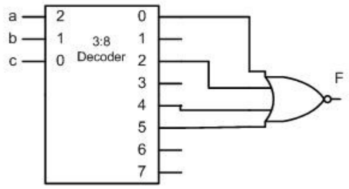
b.)



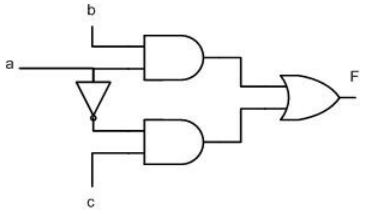
c.)



d.)



e.)

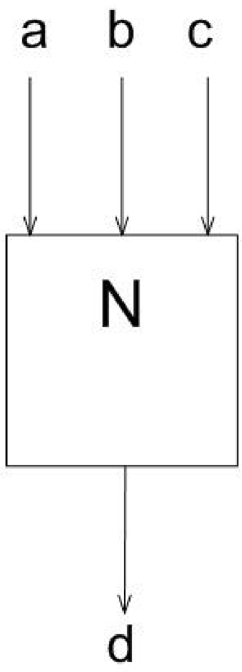


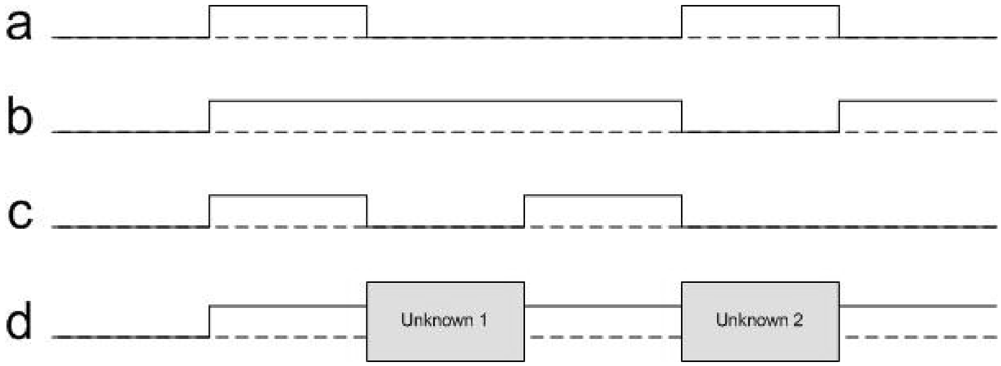
f.) None of the above

C.3 CS 231 Spring 2006 Assessment Test

Problems for this test reordered to match the ECE 290 version, with the original problem position indicated.

C.3.1 Form 1





What are the values of *d* in the boxes?

Unknown 1	Unknown 2
a. 0	a. 0
b. 1	b. 1
c. Cannot be determined	c. Cannot be determined

2.) (Originally problem 7)

Which of the following expressions are equivalent to $(a + b')(bc + a'c')$?

Select all correct answers.

a.) $(a + b')((b' + c')(a' + c'))'$

b.) $((a' + b)(b'c' + ac))'$

c.) $(a'b)+(bc + a'c')'$

d.) $((a'b) + (b' + c')(a + c))'$

e.) $(a'b)'((b' + c')' + a'c')$

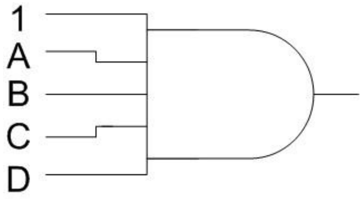
f.) $(ab') + (b + c)(a' + c')$

3.) (Originally problem 2)

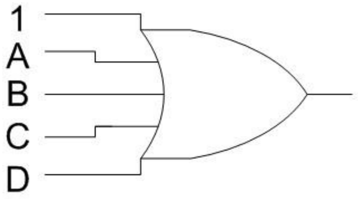
Which of the following will result in nontrivial output (not always 0 or 1)?

Select all correct answers.

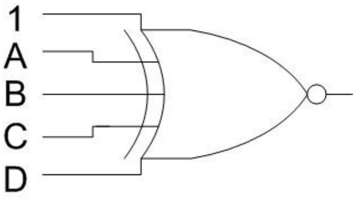
a)

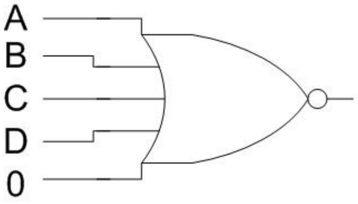


b)

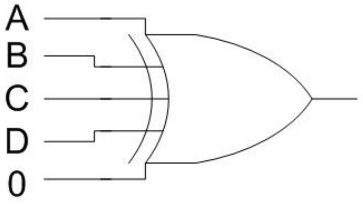


c)

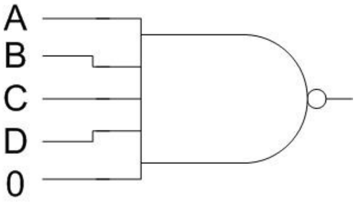




d)



e)



f)

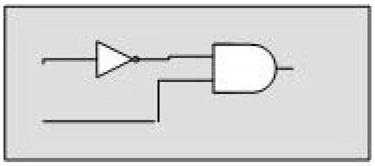
g.) None of the above

4.) (Originally problem 3)

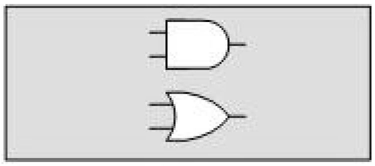
Which of the following are complete logic families (i.e., all possible combinational logic circuits can be implemented using just these gates and the constants 0 and 1).

Select all correct answers.

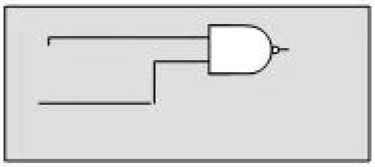
a.)



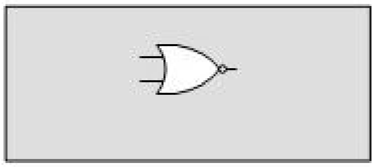
b.)



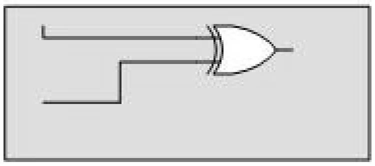
c.)



d.)



e.)

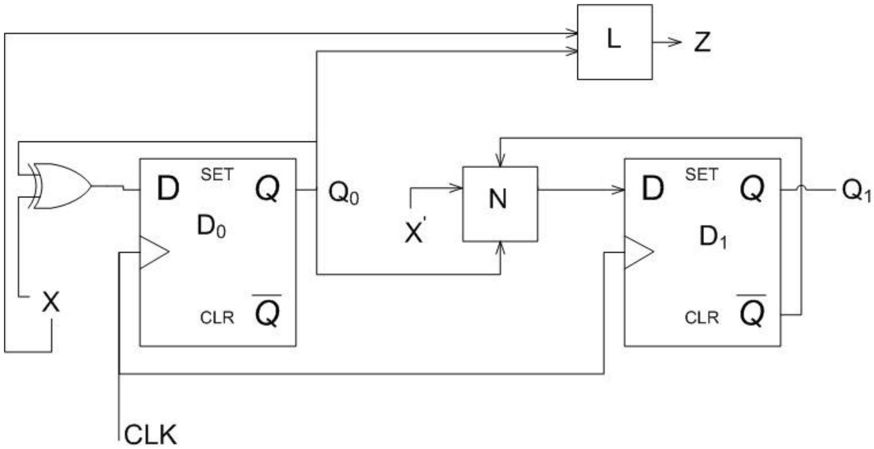


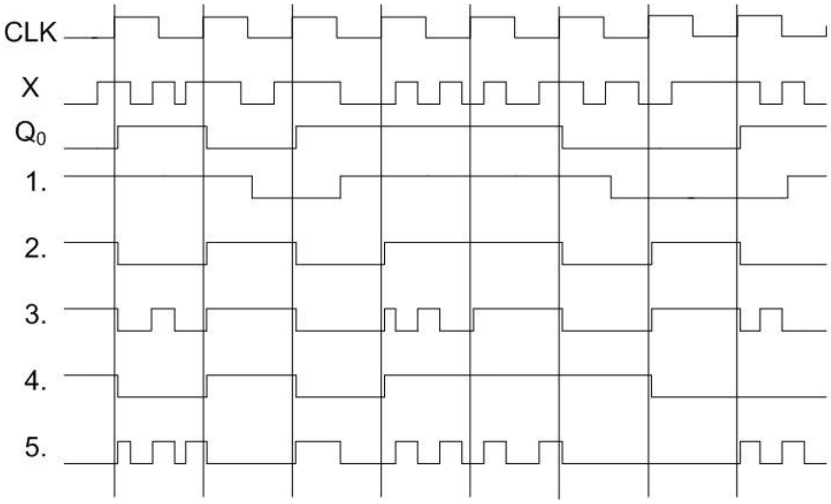
f.) None of the above

5.) (Originally problem 4)

The sequential circuit below has two positive-edge D flip flops D_0 and D_1 and two unknown circuits *N* and *L* which implement unknown Boolean functions. Which of the following waveforms could be waveforms for Q_1?

Select all correct answers.





a.) 1

b.) 2

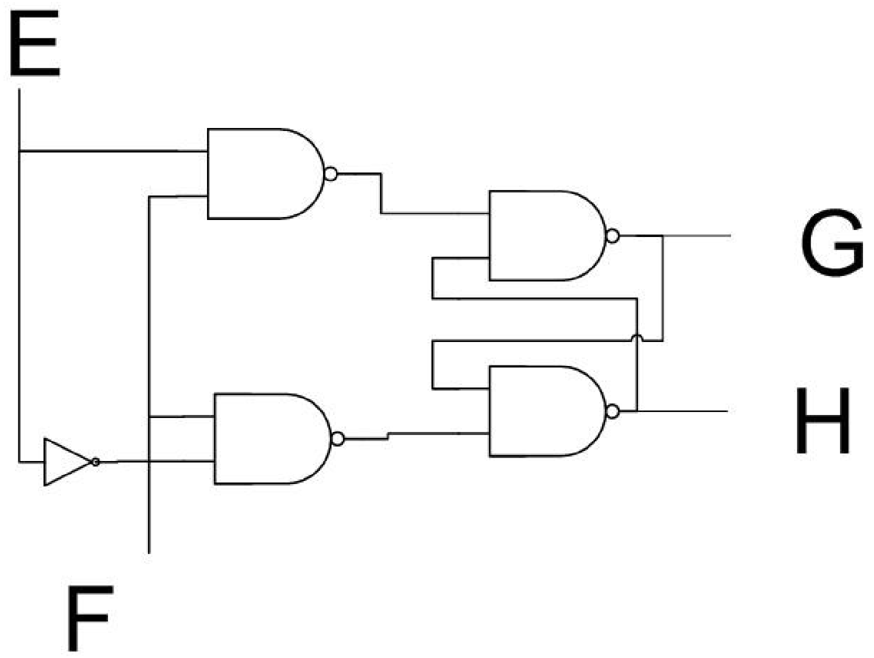
c.) 3

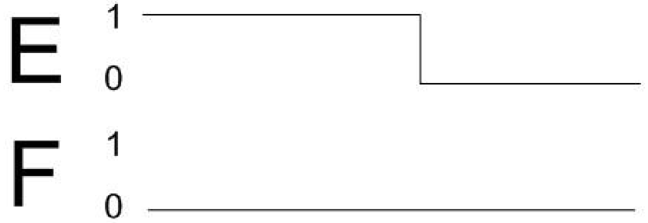
d.) 4

e.) 5

6.) (Originally problem 5)

The inputs *E* and *F* are applied as shown to the circuit below when *G* and *H* are in a steady state. What are the final values of *G* and *H* after these inputs are applied?





f.) *G* = 0, *H* = 0

g.) *G* = 0, *H* = 1

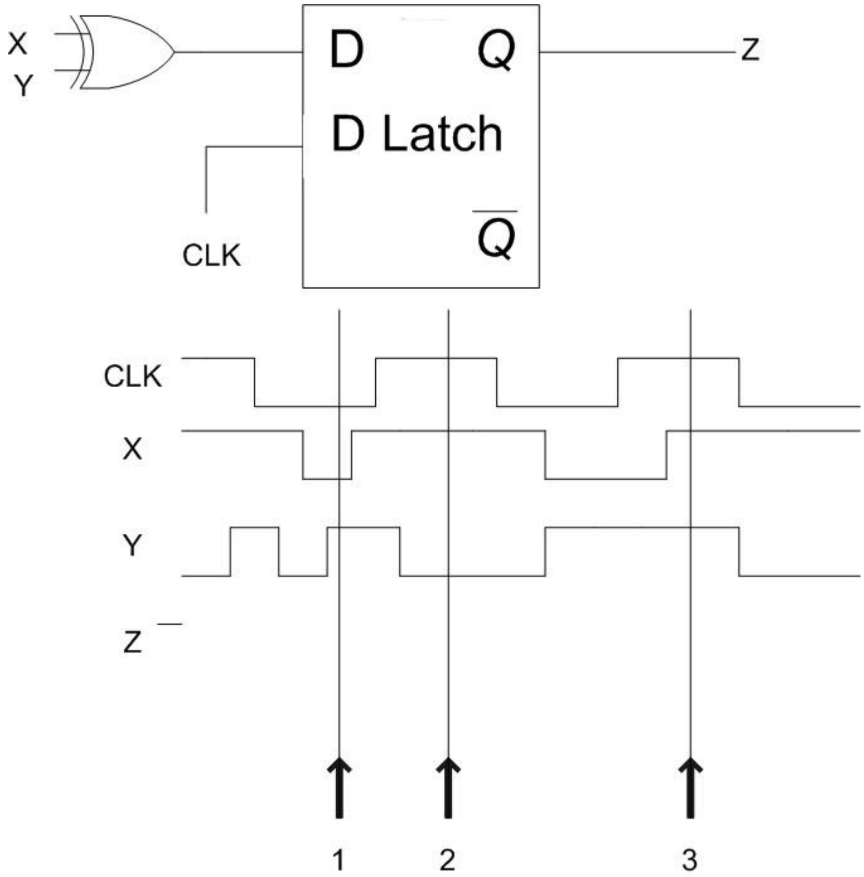
h.) *G* = 1, *H* = 0

i.) *G* = 1, *H* = 1

j.) Cannot be determined

7.) (Originally problem 6)

What are the values of *Z* at times 1, 2, and 3 if the logic element below is a *positively clocked D latch*? Assume that propagation times for the XOR gate and the latch are negligible and that *Z* is initially one.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

C.3.2 Form 2

1.) $G(a, b, c)$= AND(M1,M3,M5) = $(a + b + c')(a + b' + c')(a' + b + c')$

Which of the following could be used to implement function G? Assume that the constants 0 and 1 are **not** available and that all gates must have unique inputs.

Select all correct answers.

a.) A 3:8 decoder and a 3-input OR gate

b.) A 3:8 decoder and a 3-input NOR gate

c.) A 3:8 decoder and a 4-input OR gate

d.) A 3:8 decoder and a 4-input NOR gate

e.) A 3:8 decoder and a 5-input OR gate

f.) A 3:8 decoder and a 5-input NOR gate

2.) Let F, G, and H be functions of x, y, and z. The truth tables of F and G are the same. Which of the following must be true?

Select all correct answers.

a.) $F = G$

b.) $F' = G'$

c.) Dual(F) * $G = 0$ (Dual of F ANDed with G)

d.) $F'G = 0$

e.) $FH = HG$

f.) Dual(F) = Dual(G)

3.) A state diagram with n states requires at least m flip-flops to implement a sequential circuit. If a different state diagram has $2n$ states, what is the minimum number of flip-flops needed for an implementation?

a.) m b.) $m + 2$

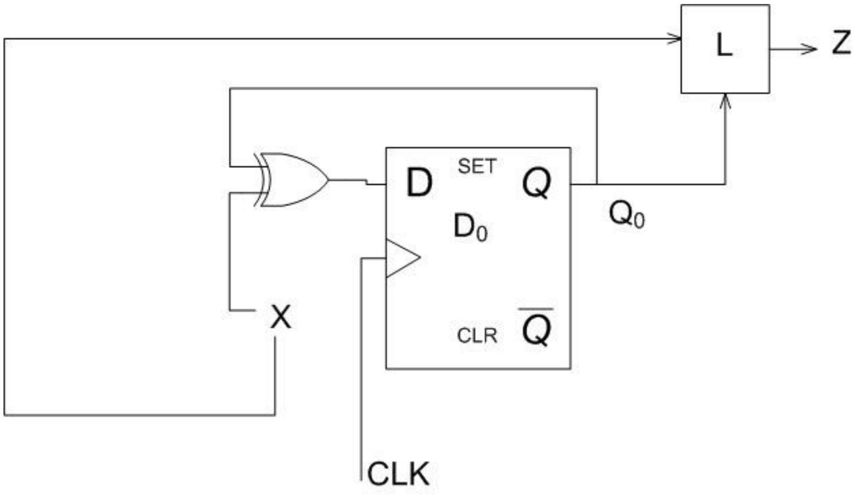
c.) $2m$ d.) $2m + 1$

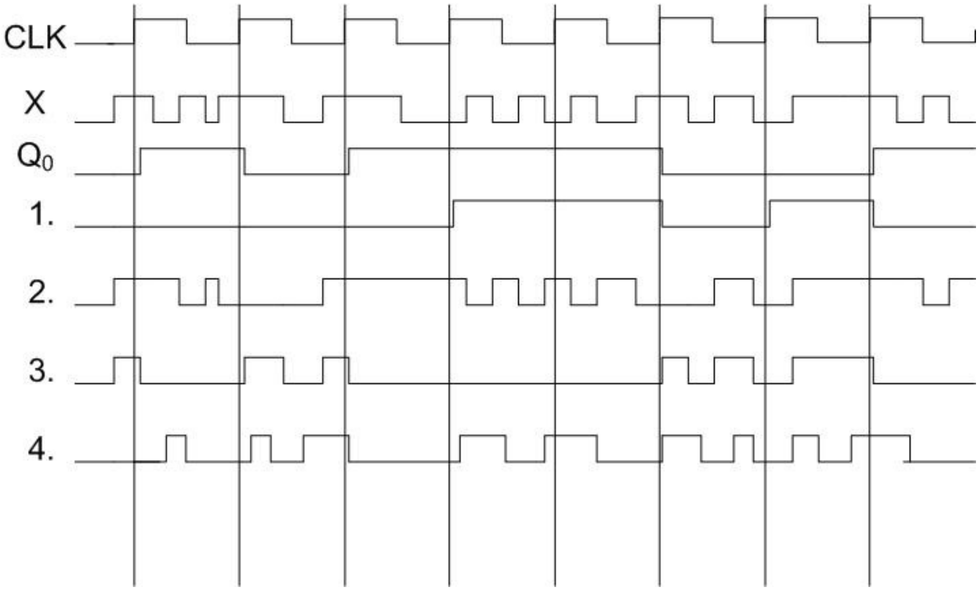
e.) m^2 f.) $2m^2$

g.) None of the above

4.) The sequential circuit below has a positive-edge D flip flop D_0 and a combinational circuit *L* which implements some unknown Boolean function. Which of the following waveforms could be waveforms for *Z*?

Select all correct answers.





a.) 1

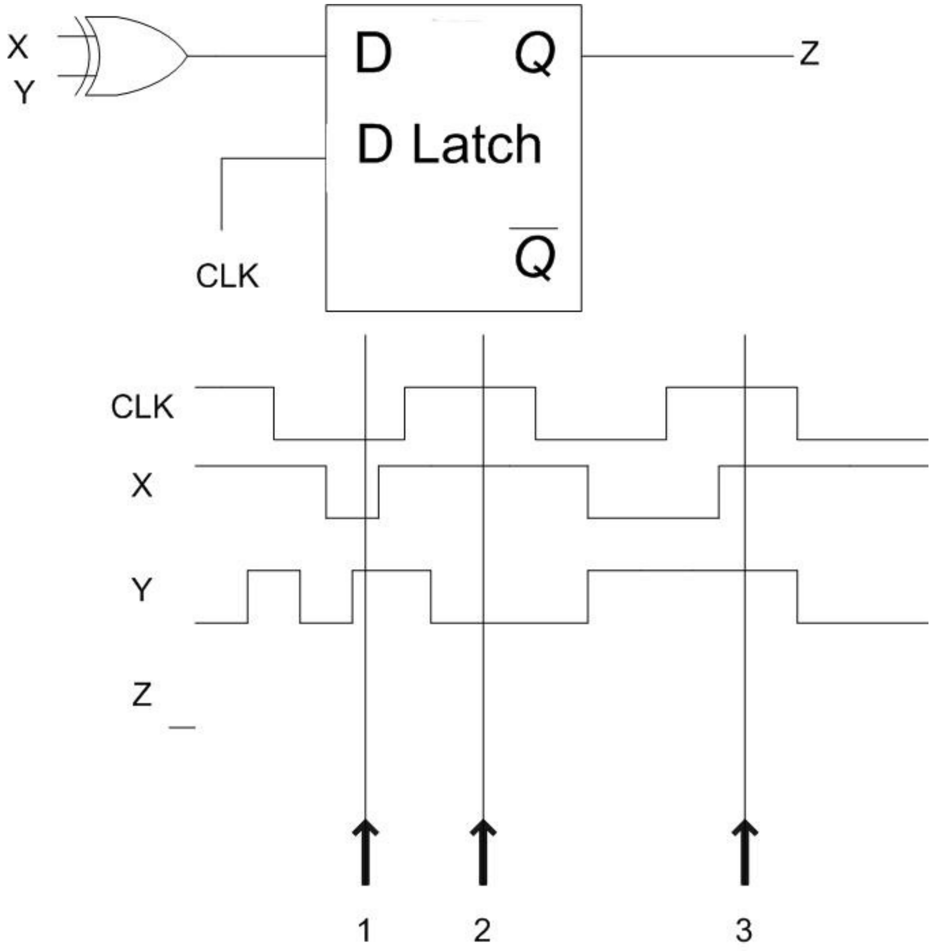
b.) 2

c.) 3

d.) 4

e.) None of the above

5.) What are the values of *Z* at times 1, 2, and 3 if the logic element below is a *positively clocked D latch (not a positive-edged flip-flop)*? Assume that propagation times for the XOR gate and the latch are negligible and that *Z* is initially one. A positively clocked D latch will change values only when the clock input is high.



a.) 0, 0, 0

b.) 0, 0, 1

c.) 0, 1, 0

d.) 0, 1, 1

e.) 1, 0, 0

f.) 1, 0, 1

g.) 1, 1, 0

h.) 1, 1, 1

i.) Not enough information

6.) If you implement the following truth table as a sum of products, what is the minimum required number gates, assuming only AND and OR gates are available? Assume you have inputs and their complements available as inputs to the circuit.

a	b	c	Output
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	1

a.) Two 3-input gates

b.) One 2-input gate and one 3-input gate

c.) Three 2-input gates

d.) Three 3-input gates

e.) Two 2-input gates and one 3-input gate

f.) One 2-input gate and two 3-input gates

g.) Three 2-input gates and one 3-input gate

h.) One 2-input gate and three 3-input gates

i.) Two 2-input gates and two 3-input gates

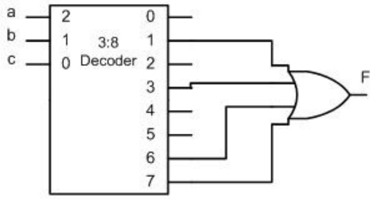
j.) None of the above

7.) Which of the following is an implementation of the following function:

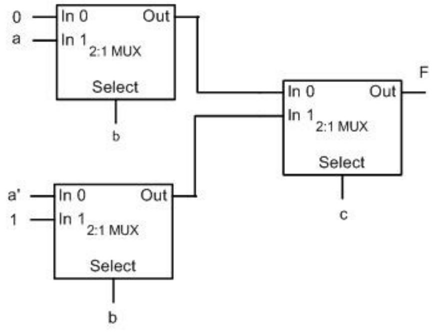
F(*a*, *b*, *c*) = OR(m1, m3, m6, m7) = *a*'*b*'*c* + *a*'*bc* + *abc*' + *abc*

Select all correct answers.

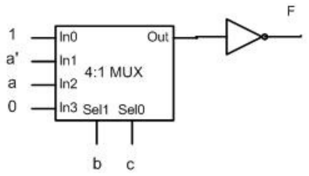
a.)



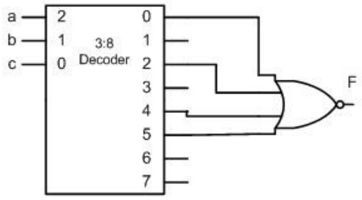
b.)



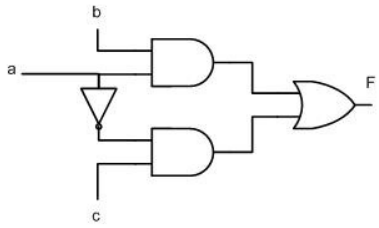
c.)



d.)



e.)



f.) None of the above

REFERENCES

[1] ABET, “Engineering criteria 2000: Criteria for accrediting programs in Engineering in the United States,” *ASEE Prism,* vol. 6, pp. 41-42, 1997.

[2] T. A. Angelo and K. P. Cross, *Classroom Assessment Techniques.* San Francisco: Jossey-Bass, 1993,

[3] M. M. Mano and C. R. Kime, *Logic and Computer Design Fundamentals*, 3rd ed. Upper Saddle River, NJ: Pearson Prentice Hall, 2003,

[4] “Revised Bloom's taxonomy,” June 2006, http://rite.ed.qut.edu.au/oz-teachernet/index.php?module=ContentExpress&func=display&ceid=29.

[5] D. Hestenes, M. Wells, and G. Swackhamer, “Force Concept Inventory,” *The Physics Teacher,* vol. 30, pp. 141-158, 1992.

[6] D. Huffman and P. Heller, “What does the Force Concept Inventory actually measure?” *The Physics Teacher,* vol. 33, pp. 138-143, 1995.

[7] D. Hestenes and I. Halloun, “Interpreting the Force Concept Inventory,” *The Physics Teacher,* vol. 33, pp. 502-506, 1995.

[8] S. Mahajan, “Observations on teaching first-year physics,” *Arxiv Preprint*, December 2005, http://arxiv.org/pdf/physics/0512158.

[9] R. N. Steinberg and M. S. Sabella, “Performance on multiple-choice diagnostics and complementary exam problems,” *The Physics Teacher,* vol. 35, pp. 150-155, 1997.

[10] N. S. Rebello and D. A. Zollman, “The effect of distracters on student performance on the force concept inventory,” *American Journal of Physics,* vol. 72, pp. 116-125, 2003.

[11] L. McCullough, “Gender, context, and physics assessment,” *Journal of International Women’s Studies,* vol. 5, pp. 20-30, 2004.

[12] H. Schecker and J. Gerdes, “Messung von Konzeptualisierungsfähigkeit in der Mechanik: Zur Aussagekraft des Force Concept Inventory,” *Zeitschrift für Didaktik der Naturwissenschaften,* vol. 5, pp. 75-89, 1999.

[13] R. R. Hake, “Interactive-engagement versus traditional methods: A six-thousand-student survey of mechanics test data for introductory physics courses,” *American Journal of Physics,* vol. 66, pp. 64-74, 1998.

[14] R. R. Hake, “Interactive-engagement methods in introductory mechanics courses,” *Physics Education Research*, 1998, http://www.physics.indiana.edu/~sdi/.

[15] R. R. Hake, “Interactive-engagement vs. traditional methods in mechanics instruction,” *APS Forum on Education Newsletter*, 1998, http://www.physics.indiana.edu/~sdi/.

[16] R. Hake, “Lessons from the physics education reform effort,” *Conservation Ecology,* vol. 5, no. 2, 2002, *http://www.ecologyandsociety.org/vol5/iss2/art28/.*

[17] D. Hestenes and M. Wells, “A mechanics baseline test,” *The Physics Teacher*, vol. 30, no. 3, pp. 159-165, 1992.

[18] E. F. Redish, J. M. Saul, and R. N. Steinberg, “On the effectiveness of active-engagement microcomputer-based laboratories,” *American Journal of Physics,* vol. 65, pp. 45-54, 1997.

[19] J. M. Saul, “Beyond problem solving: evaluating introductory physics courses through the hidden curriculum,” Ph.D. dissertation, University of Maryland, College Park, MD, 1998.

[20] E. F. Redish and R. N. Steinberg, “Teaching physics: Figuring out what works,” *Physics Today*, vol 52, no. 1, pp. 24-30, 1999.

[21] E. F. Redish, “Millikan lecture 1998: Building a science of teaching physics,” *American Journal of Physics*, vol. 67, no. 7, pp. 562-573, 1999.

[22] G. E. Francis, J. P. Adams, and E. J. Noonan, “Do they stay fixed?” *The Physics Teacher,* vol. 36, pp. 488-491, 1998.

[23] K. Cummings, J. Marx, R. Thornton, and D. Kuhl, “Evaluating innovations in studio physics. Physics Educational Research,” *Physics Educational Research Supplement to the American Journal of Physics,* vol. 67, pp. S38-S44, 1999.

[24] R. Beichner, L. Bernold, E. Burniston, P. Dail, R. Felder, J. Gastineau, M. Gjertsen, and J. Risley, “Case study of the physics component of an integrated curriculum,” *Physics Education Research Supplement to American Journal of Physics,* vol. 67, pp. S16-S24, 1999.

[25] M. Johnson, “Facilitating high quality student practice in introductory physics,” *Physics Education Research Supplement to American Journal of Physics,* vol. 69, pp. S2-S11, 2001.

[26] P. S. Steif, “Comparison between performance on a concept inventory and solving of multifaceted problems,” in *Frontiers in Education,* 2003, pp. T3D-17-T3D-22.

[27] D. J. Van Domelen and A. Van Heuvelen, “The effects of a concept-construction lab course on FCI performance,” *American Journal of Physics,* vol. 70, pp. 779-780, 2002.

[28] A. Savinainen and P. Scott, “Using the Force Concept Inventory to monitor student learning and to plan teaching,” *Journal of Physics Education,* vol. 37, pp. 53-58, 2002.

[29] J. P. Mestre, “Facts and myths about pedagogies of engagement in science learning,” *Peer Review,* vol. 7, pp. 24-27, 2005.

[30] Foundation Coalition, “Concept Inventory Assessment Instruments,” July 2006, http://foundationcoalition.org/home/keycomponents/concept/index.html.

[31] D. L. Evans, G. L. Gray, S. Krause, J. Martin, C. Midkiff, B. M. Notaros, M. Pavelich, D. Rancour, T. Reed-Rhoads, and P. Steif, “Progress on concept inventory assessment tools,” in *Frontiers in Education,* 2003, pp. T4G- 1-8.

[32] S. Krause, J. Birk, R. Bauer, B. Jenkins, and M. J. Pavelich, “Development, testing, and application of a chemistry concept inventory,” in *Frontiers in Education,* 2004, pp. T1G-1 – T1G-5.

[33] M. Pavelich, B. Jenkins, J. Birk, R. Bauer, and S. Krause, “Development of a chemistry concept inventory for use in chemistry, materials and other engineering courses,” in *American Society of Engineering Education, Annual Conference,* 2004, pp. 3445-3452.

[34] G. Gray, D. Evans, P. Cornwell, F. Costanzo, and B. Self, “Toward a nationwide dynamics concept inventory assessment test,” in *American Society for Engineering Education, Annual Conference,* 2003, pp. 9187-9198.

[35] G. L. Gray, F. Costanzo, D. Evans, P. Cornwell, B. Self, and J. L. Lane, “The dynamics concept inventory assessment test: A progress report and some results,” in *American Society of Engineering Education, Annual Conference,* 2005, pp. 4819-4833.

[36] R. J. Roedel, S. El-Ghazaly, T. R. Rhoads, and E. El-Sharawy, “The wave concepts inventory-an assessment tool for courses in electromagnetic engineering,” in *Frontiers in Education,* 1998, pp. 647-653.

[37] T. R. Thoads and R. J. Roedel, “The wave concept inventory-a cognitive instrument based on Bloom's taxonomy,” in *Frontiers in Education,* 1999, pp. 13C1-14 – 13C1-18.

[38] D. P. Maloney, T. L. O’Kuma, C. J. Hieggelke, and A. Van Heuvelen, “Surveying students’ conceptual knowledge of electricity and magnetism,” *American Journal of Physics,* vol. 69, pp. S12-S23, 2001.

[39] B. M. Notaros, “Concept inventory assessment instruments for electromagnetics education,” in *IEEE Antennas and Propagation Society International Symposium,* 2002, pp. 684-687.

[40] J. Martin, J. Mitchell, and T. Newell, “Development of a concept inventory for fluid mechanics,” in *Frontiers in Education,* 2003, pp. T3D-23 – T3D-28.

[41] K. C. Midkiff, T. A. Litzinger, and D. L. Evans, “Development of engineering thermodynamics concept inventory instruments,” in *Frontiers in Education,* 2001, pp. F2A-F23.

[42] A. Jacobi, J. Martin, J. Mitchell, and T. Newell, “A concept inventory for heat transfer,” in *Frontiers in Education,* 2003, pp. T3D-12 – T3D-16.

[43] A. Jacobi, J. Martin, J. Mitchell, and T. Newell, “Work in progress: A concept inventory for heat transfer,” in *Fronteirs in Education,* 2004, pp. T3F-3 – T3F-4.

[44] J. Richardson, J. Morgan, and D. Evans, “Development of an engineering strength of material concept inventory assessment instrument,” in *Frontiers in Education,* 2001, pp. F2A-F24.

[45] J. Richardson, P. Steif, J. Morgan, and J. Dantzler, “Development of a concept inventory for strength of materials,” in *Frontiers in Education,* 2003, pp. T3D-29 – T3D-33.

[46] S. Krause, J. C. Decker, and R. Griffin, “Using a materials concept inventory to assess conceptual gain in introductory materials engineering courses,” in *Frontiers in Education,* 2003, pp. T3D-7-T3D-11.

[47] W. Jordan, H. Cardenas, and C. B. O'Neal, “Using a materials concept inventory to assess an introductory materials class: Potential and problems,” in *American Society for Engineering Education, Annual Conference,* 2005, pp. 15095-15101.

[48] K. E. Wage and J. R. Buck, “Development of the signals and systems concept inventory (SSCI) assessment instrument,” in *Frontiers in Education,* 2001, pp. F2A-F22.

[49] K. E. Wage, J. R. Buck, T. B. Welch, and C. H. G. Wright, “Testing and validation of the signals and systems concept inventory,” in *IEEE 10th Digital Signal Processing Workshop and 2nd Signal Processing Education Workshop,* 2002, pp. 151-156.

[50] K. E. Wage, J. R. Buck, C. H. G. Wright, and T. B. Welch, “The Signals and Systems Concept Inventory,” *IEEE Transactions on Education,* vol. 48, pp. 448-461, 2005.

[51] K. E. Wage, J. R. Buck, T. B. Welch, and C. H. G. Wright, “The continuous-time signals and systems concept inventory,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing,* 2002, pp. 4112-4115.

[52] P. S. Steif, “Initial data from a statics concept inventory,” in *American Society for Engineering Education, Annual Conference,* 2004, pp. 7291-7300.

[53] P. S. Steif and J. A. Dantzler, “A statics concept inventory: Development and psychometric analysis,” *Journal of Engineering Education,* vol. 94, no. 4, pp. 363-371, Oct. 2005.

[54] P. S. Steif, A. Dollár, and J. A. Dantzler, “Results from a statics concept inventory and their relationship to other measures of performance in statics,” in *Frontiers in Education,* 2005, pp. T3C-5 – T3C-10.

[55] A. Stone, K. Allen, T. R. Rhoads, T. J. Murphy, R. L. Shehab, and C. Saha, “The statistics concept inventory: A pilot study,” in *Frontiers in Education,* 2003, pp. T3D-1 – T3D-6.

[56] K. Allen, A. Stone, T. R. Rhoads, and T. J. Murphy, "The statistics concepts inventory: Developing a valid and reliable instrument," in *American Society for Engineering Education, Annual Conference,* 2004, pp. 12869-12883.

[57] Foundation Coalition, "Computer Engineering Concept Inventory," July 2006, http://www.foundationcoalition.org/home/keycomponents/concept/computer.html.

[58] J. Richardson, "Concept inventories: Tools for uncovering STEM students' misconceptions," in *Invention and Impact: Building Excellence in Undergraduate Science, Technology, Engineering and Mathematics (STEM) Education*, 2004, pp. 19-26, http://www.aaas.org/publications/books_reports/CCLI/PDFs /02_AER_Richardson.pdf.

[59] ACM Special Interest Group on Computer Science Education (SIGCSE), "Computing Curricula 2001," 2001, http://www.sigcse.org/cc2001.

[60] R. A. Streveler, B. M. Olds, R. L. Miller, and M. A. Nelson, "Using a delphi study to identify the most difficult concepts for students to master in thermal and transport science," in *American Society for Engineering Education, Annual Conference,* 2003, p. 4447-4454.

[61] C. H. Roth, *Fundamentals of Logic Design*, 4th ed. Boston: PWS Publishing Co., 2003.

[62] "Course descriptor for ECE 290," April 2006, http://www.ece.utexas.edu/projects/ee316/course_descriptor_sp06.html.

[63] "ENEE 214 course description," April 2006, http://www.ee.umd.edu/Academic/Under/ucourses1.htm#ENEE%20244.

[64] "ECEN 3100 – Digital Logic," April 2006, http://ece-www.colorado.edu/academics/courses/ECEN_3100.html.

[65] SIGCSE, "Architecture and organization," April 2006, http://www.sigcse.org/cc2001/AR.html#AR-DigitalLogic.

[66] T. Newell (private communication), April 2005.

[67] N. J. Evans, D. S. Forney, and F. Guido-DiBrito, *Student Development in College: Theory, Research, and Practice.* San Francisco: Jossey-Bass, Inc., 1998.