Effectively Incorporating Hardware Experience into a Digital Electronics Service Course

William C. Dillard
200 Broun Hall ECE Department
Auburn University, AL 36849-5201
Voice (334) 844-1840 Fax (334) 844-1809
dillard@eng.auburn.edu

Abstract

A new approach to incorporating digital hardware in an introductory digital electronics service class has been successfully demonstrated at Auburn University's Electrical and Computer Engineering department. Having no room in the curriculum for a formal laboratory, the central theme is to combine theory, simulation and hardware within the existing classroom/study time allotments. Teams of 5 or 6 students construct experiments "at home" and submit their circuits as homework for grading. PSPICE simulations are used to support both the lecture material and the hardware experience. Additional reading materials and tutorials have been created for better utilization of both in-class lecture and out-of-class study time. Lecture style has been modified to incorporate class time for introduction to hardware and digital simulation with PSPICE without sacrificing course content. This approach has proven to be an effective tool in introducing students to hardware issues and implementation alternative as well as improving student learning and motivation.

Introduction

Traditionally, the Electrical and Computer Engineering department at Auburn University taught an introductory digital electronics course at the junior level that served both electrical and computer engineers. With three hours of lecture each week, the course topic included:

- Binary, decimal and hexadecimal bases
- Logic operations and gates
- LED's, 7-segment displays and switches
- Boolean algebra and Karnaugh maps
- SOP and POS formats
- Minterms, maxterms and lists
- Timing diagrams and gate delay
- Signed number systems
- Adder circuits
- Parity, ASCII, BCD and Gray Code
- Arithmetic Logic Unit
- Decoder, encoder, MUX and DeMUX
- Programmable logic – ROM's and PLD's
- Flip-flops and sequential circuits
- Registers and shift registers
- Counters

An accompanying weekly, three-hour laboratory was also required, providing the student with a significant semester long hardware investigation. After a recent campus-wide curricula change, the original course has been replaced by two sophomore-level courses; one for electrical and computer engineering majors (with a reduced lab schedule), and the other, a service for computer science (CS) and software engineering (SE) students with no laboratory time at all. Three hourly lectures are conducted each week with no change in the course content. While these curricula models fit into the institution's core curriculum requirements, the lack of any laboratory in the service class left the CS and SE majors disadvantaged in that this service course is the only "hardware" class in their curricula. Given industrial emphasis on hardware/software co-design, it is vital that these students have exposure to hardware, the multitude of implementation options that exist, and the role software plays in modern digital circuit design.

Initially the service course was strictly in-class lectures, although the design content of the homework and tests was significant. Surprisingly, the CS and SE students themselves recognized the need for hardware and petitioned for a solution. However, having no formal laboratory infrastructure in the new curriculum significantly restricted the possible solutions. A new pedagogy had to be found that did not compromise the integrity of the course just to include some hands-on wiring. Seven goals became immediately apparent.

1. With no budget for laboratory teaching assistants or facilities, the departmental costs must be kept to a minimum and, if possible, nonrecurring.
2. Having no formal laboratory infrastructure, students must be able to assemble and verify designs at home without access to tradition test and measurement equipment. Also, their costs should be kept as low as feasible.
3. The curriculum constrains the hardware/simulation homeworks to fit into the traditional "out-of-class" time allotment of approximately three hours for each "in-class" hour. Therefore, hardware experiments should require no more time than traditional homeworks.
4. Some interesting experiments are just too large to construct in the allotted time. In those cases, simulations should be used to validate the design. Also, simulations should directly support either a pen-and-paper homework and/or an experiment.
5. The course content should not be altered to allow room for any hardware/simulation introductory lectures.
6. The hardware homeworks must enhance retention of the lecture material by supporting it rather than driving it.
7. To facilitate the introduction to PSPICE and emphasize design, the lectures should be taught in a multimedia classroom with access to PSPICE and the Internet so that manufacturers datasheets can be accessed as needed.

In addressing all of these issues, a new teaching model called, the "Hardware Experience", was adopted for evaluation during spring semester 2001. It was decided that the class would be divided in teams of five to six students with each team purchasing a single hardware kit, thus reducing the cost per student. Having students own their kits brought many advantages. First, institutional costs were significantly reduced. Second, due to pride of ownership, very few parts were destroyed. And third, curious students anxious to implement their own ideas had the equipment in hand to proceed.

Teaming

While many institutions have chosen to closely monitor and control team dynamics [1], our approach was more relaxed. Students were allowed to form themselves in teams of five to six members. No formal peer evaluation process was implemented. Instead, team problems were identified by informal conversations with the teams throughout the semester. It is our opinion that resolving conflicts is an inherent part of realistic teaming. It may in fact be the most important aspect in maintaining an effective work environment. Therefore, teams were encouraged to manage themselves courteously and professionally. Irresolvable conflicts were handled expeditiously in the instructor's office face-to-face. No attempt was made to determine what percentage of the work each team member might have performed as we consider equitable load sharing to be an essential part of team self-management. Therefore, every team member received the same grade on team-based assignments.

The Hardware Kit

The core of the Hardware Experience is the Hardware Kit. Components, listed in Table 1, were chosen to expose the student to a variety of 74xx series SSI, MSI and LSI chips. To lower costs (below $8.00 per student) teams were formed of 5 to 6 students. A 9-V battery and 5-V regulator provided a TTL compatible supply voltage. Binary inputs were implemented using DIP switches and momentary pushbuttons while outputs were monitored by LED's. Much of the success of the hardware kit should be credited to the Scientific Supply Store, part of Chemistry Department at Auburn University, which acted as retailer, obtaining stock in bulk and reselling to the students. Given the low cost of each component and on-campus access, failed devices were simply replaced at the student's expense. The ECE department provided a mounting board that accommodated both the breadboard and battery and a carry case that holds the entire kit. These components are shown in Figure 1. Initial institutional cost per kit was less than $3.00 per kit with a recurring cost of about 25¢ for wire. At the end of the semester, the mounting boards and cases are collected for reuse the following term.

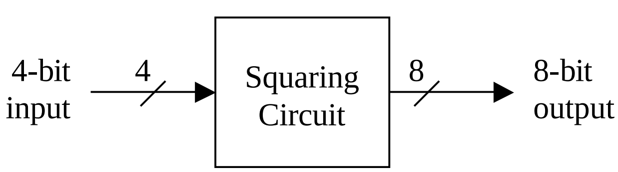
Table 1. Hardware Parts List

Part	**No.**	**Part**	**No.**	**Part**	**No.**	**Part**	**No.**
Protoboard	1	'20 NAND	2	Bargraph LED	1	'153 Dual MUX	1
Wire Strips	1	'08 AND	2	Cap. – 10uF	1	28C16 EEPROM	1
Wire Kit	1	'32 OR	1	LM555C Timer	1	DIP Switch	1
'04 NOT	2	'86 XOR	2	50 kΩ & 10 kΩ	2	Storage Box	1
'00 NAND	2	'175 D-FF	1	5-V Regulator	1	Pushbuttons	2
'10 NAND	2	'161 counter	1	9-V Battery	1	Resistors 330 Ω	20

A conscience decision was made to wire all circuits by hand. We feel that at the sophomore level, experiments performed on pre-fabricated PC boards, while convenient, do not empower students and should be avoided if possible. Three considerations motivated our choice. First, no circuitry is transparent to the student. Second, students are forced to reference datasheets and consider circuit layout issues – two important engineering skills. Third, and most importantly, we wanted students to understand that as sophomores they have all the technical competence necessary to design and build functional circuits not only for class, but also for their own projects. Working with such a simple arrangement of parts, wire and breadboard, students are empowered to explore digital electronics.

Figure 1. The ELEC 2200 Hardware Kit. The protoboard base and box are provided by the ECE department. All other components are purchased by student teams.

One chip in the Hardware Kit of particular interest is the 28C16A 2Kx8 EEPROM. With 12 inputs and 8 outputs, it allows students to easily design and construct a variety of circuitry too complicated for gate level construction. For example, consider a 4-bit squaring circuit, that is two 4-bit inputs and an 8-bit output, as seen in Figure 2a. Beginning with the appropriate ROM table, the student simply enters the necessary data into a simple assembly language file, listed in Figure 2b. A rudimentary freeware assembler called UNIASM produces the corresponding HEX file. A ROM burner is then used to program the ROM and the circuit can be constructed and tested. For more complicated ROM implementations with more than 4 inputs, students often use C++, Java or EXCEL to automatically create the assembly text file. Given their backgrounds and interest in programming, this is often one of their easier assignments.



(a)

```
       ORG 0
EX_1:  DB 0
       DB 1
       DB 4
       DB 9
       DB 16
       DB 25
       DB 36
       DB 49
       DB 64
       DB 81
       DB 100
       DB 121
       DB 144
       DB 169
       DB 196
       DB 225
       END
```

(b)

Figure 2. A ROM-base squaring circuit (a) and the corresponding assembly language file (b).

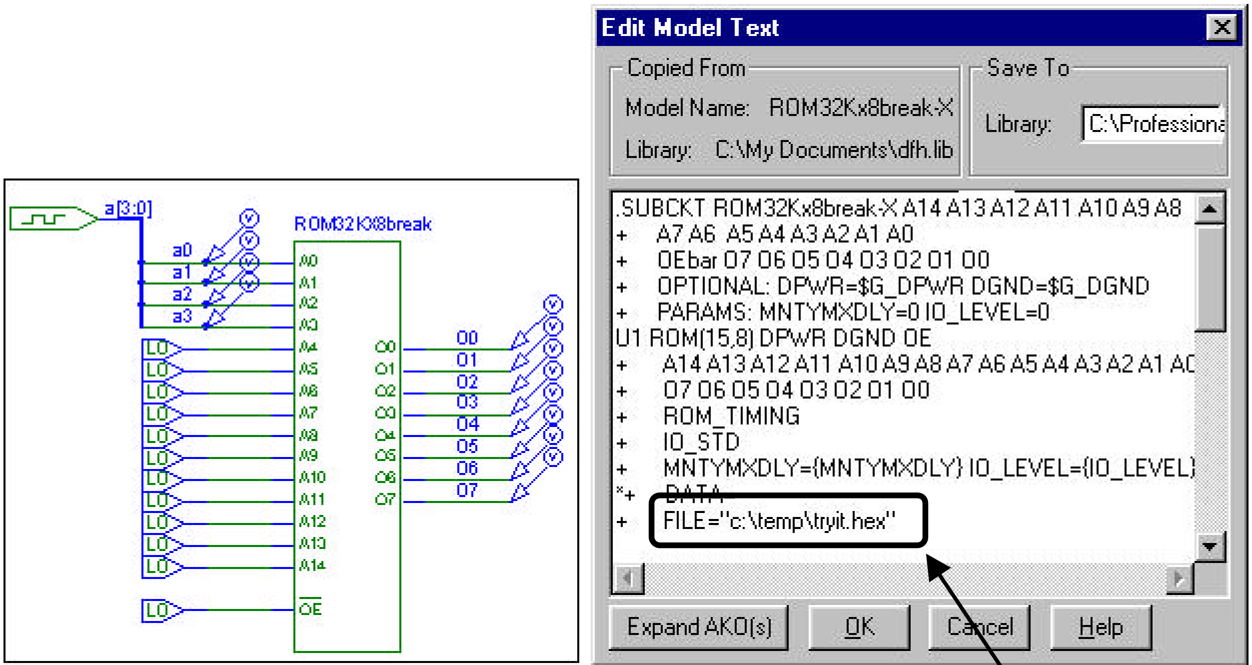
PSPICE for Simulation

The effectiveness of simulation tools versus hardware experiments in teaching electrical engineering has been explored by several investigators. In 1989, Gokhale[2] compared an unspecified simulation program with traditional laboratories. He found no significant differences in posttest scoring between the two groups. Garren[3] came to the same conclusion in 1990 when comparing software versus hardware approaches in digital electronics instruction. However, significant progress has been made in circuit simulators over the last decade, particularly in the human interface, or schematic entry. Also, today's entering freshmen are much more computer savvy than freshmen ten years ago. A more recent evaluation (1999) by Hall[4] compared the use of Electronics Workbench, a SPICE-based simulator, against hardware. He also found no quantitative posttest difference between the two approaches. However, in qualitative testing, he found that students had strong opinions for each technique. Junior level students felt better served by the hardware, even though their quantitative scores indicated that hardware yielded no advantage. On the other hand, freshmen regarded each method as equally effective.

Hall recommended that both hardware and software be employed in teaching the theoretical topics. This is a reasonable concept given the variation in learning styles; audio, visual, hands-on, etc. that one encounters in any collection of students. In our work there is another motivation for including both simulation and hardware. The cost limitation on the hardware kit and the time constraints caused by the curriculum, there simply is not enough time to construct, debug and verify large digital circuits. Thus, in the Hardware Experience, when the circuit gets large, simulation is used exclusively.

Due to its ease of use, low cost (free of charge), simple user interface and large parts library, the evaluation version of PSPICE 9.1, free from Cadence Inc., was selected as the simulation tool. Students could access the program in on-campus computer labs or install it at home. The 134 digital components available in the evaluation version are more than sufficient for an introductory course. Various digital sources, clock signals and bus options are also available. To lower the learning curve, an introductory tutorial was written and distributed on-line.

As an example of how software and hardware can be used together, a PSPICE file can be created for the 4-bit squaring circuit in Figure 2a. In the PSPICE schematic shown in Figure 3a, the ROM32Kx8breakout part in PSPICE emulates the 28C16A 2Kx8 ROM found in the hardware kit. The corresponding HEX file is simply referenced by path and name within the ROM32Kx8 model in Figure 3b. When investigating timing concerns such a delay, glitches, hazards and critical paths, the hardware approach, with its LED output indicators is useless. For these issues PSPICE is the superior tool. Digital inputs can be edited within PSPICE or a large text file of test vectors can be used. Gate delays from any input pin to any output pin can be specified for minimum, maximum and typical values. For flip-flops, setup and hold times can also be edited. Another strength in PSPICE, particularly over text entry simulation such as HDL, is the similarity between schematic entry, where wiring is simply point and click, and wiring hardware by hand. Our experience is that these two activities, schematic entry and wiring by hand, makes the other that much easier to master.



(a) (b)

Figure 3. Emulating the 28C16A EEPROM using the ROM32KX8break part in PSPICE (a) requires the corresponding hex file to be referenced in the ROM32KX8break model (b).

Hardware Homeworks and Simulations

The hardware homeworks, briefly listed below, were used in the Spring 2001 class. (All homeworks are available in their entirety as WORD documents at the author’s website: http://www.eng.auburn.edu/users/dillard/.) They ranged in complexity from simple gate-level circuits to EEPROM implementations to sequential machines and were selected to support the lecture material rather than for “high-tech” impact. Since the kits are portable and battery powered, hardware homeworks can be evaluated during the lecture hour. This is particularly useful when circuits can be cascaded. For example, full adders can be cascaded to produce a ripple carry adder that can be tested in class for functionality. Bringing a portable logic analyzer or oscilloscope to class, the effect of carry rippling on timing delays can be measured.

1. Simple AND/OR logic gate with switches and LED’s
2. NAND-NAND implementation of a fire alarm
3. 8-bit parity checker (both construction and PSPICE simulation)
4. 7-segment displays and the 74154 decoder (both construction and PSPICE simulation)
5. Multiplexers and full adders (both construction and PSPICE simulation)
6. Floating point numbers
7. Rotary Gray Code encoders
8. A sequence detector for a serial data transfer protocol

The simulations listed below were homework assignments during the Spring 2001 semester. Some were combined with a paper homework while others were part of a paper/simulate/build assignment.

1-3. See Hardware homework number 3, 4 and 5
4. Comparing ripple carry versus carry look-ahead adders
5. Asynchronous sequential

In-class Time Restrictions and Solutions

Incorporating the Hardware Experience into an already crowded syllabus required a new approach to teaching the course. Brief introductions to PSPICE and the hardware kit components consumed three lectures – one week of classes – that had to be recovered. Additionally, to treat hardware and PSPICE as quickly as possible, topics such as gates, Boolean algebra and K-maps had to be introduced very early in the semester. Our solution was to restructure the first three weeks of the course. A comparison between the old and new approaches is listed in Table 2.

Table 2. Comparing the First Three Weeks under the Old and New Course Schedules

<table>
<tr><th>Old</th><th>Topic</th><th>Chap.</th><th>New</th><th>Topic</th><th>Chap.</th></tr>
<tr><td rowspan="7">Week 1</td><td>History of Computing</td><td rowspan="5">Zero</td><td rowspan="7">Week 1</td><td>Relating 0 and 1 to 0V and 5V</td><td>Manual</td></tr>
<tr><td rowspan="2">Digital Systems</td><td>Decimal, Binary & Hex</td><td rowspan="2">One</td></tr>
<tr><td>Base Conversions</td></tr>
<tr><td rowspan="2">Computer Organization</td><td>Binary Math</td><td rowspan="2">One</td></tr>
<tr><td>Truth Tables</td></tr>
<tr><td rowspan="2">Number Systems</td><td rowspan="9">One</td><td>Logic Operations, Gates</td><td rowspan="2">One
Two</td></tr>
<tr><td>Boolean Algebra</td></tr>
<tr><td rowspan="3">Week 2</td><td>Binary Math</td><td rowspan="3">Week 2</td><td>The Hardware Kit</td><td>Manual</td></tr>
<tr><td>Base Conversions</td><td rowspan="2">The Karnaugh Map</td><td rowspan="2">Three</td></tr>
<tr><td>Signed Numbers</td></tr>
<tr><td rowspan="5">Week 3</td><td rowspan="2">BCD and Gray Code</td><td rowspan="5">Week 3</td><td>Design Procedures</td><td rowspan="4">One</td></tr>
<tr><td>Designs for BCD</td></tr>
<tr><td rowspan="2">Parity</td><td>Designs for Gray Code</td></tr>
<tr><td>Designs for Parity</td></tr>
<tr><td>Boolean Algebra</td><td>Two</td><td>PSPICE</td><td>Manual</td></tr>
</table>

Note that the old schedule follows the course textbook[5] sequentially. Under the new structure getting to K-maps and hardware so quickly requires skipping about in the first three chapters. Furthermore, most of the hardware issues relate directly to the hardware kit and are not part of the text. To ease this ascent, an 80-page course manual was created.

The major teaching change actually occurs after K-maps in Table 3, when Chapter 1 is revisited to discuss some of the drier topics such as BCD, signed number representation and parity. Now, possessing the ability to design combinational circuit quickly, we introduce the Chapter 1 material as circuit designs in class. This approach not only treats the new theory, it also makes the topics more interesting while honing the student's design skills.

Evaluation Instruments

Two instruments are used to assess the success of the proposed methodology. A pre-test was used to gauge student's course expectations, interests and preparedness for hardware construction. Results show that course expectations were skew heavily toward theory and away from circuit design/construction. This is reasonable given that less that 10% of the students had ever constructed even simple circuitry. A posttest, partially listed in Table 3, was used to evaluate the effectiveness of the hardware experience in meeting the course expectations. Questions are answered on a scale from 0 to 5, with 5 being better evaluation. All questions directly related to the effectiveness of the hardware homeworks (questions 4, 5, 10, 12 and 14) scored 4.0 or better. The response to Question 5, relating the hardware to retention, is particularly exciting – a 4.25 average and a minimum response of 3.0! Considering that ELEC 2200 is an electrical engineering service class for non-EE majors, these are encouraging statistics. The student's had a lower opinion of the effectiveness of PSPICE (questions 3, 15 and 16). This is probably a reflection of the number of PSPICE assignments (5) versus hardware (8). As a result, students did not acquire the proficiency required to use PSPICE effectively.

A very important posttest revelation was the time spent on hardware assignments. Students spent an average of 2.9 hours on each hardware homework compared to 2.8 hours on paper-and-pen work, validating that the hardware did not add a significant time burden. All respondents felt that the hardware cost is justified with 83% strongly recommending continued use of the Hardware Experience.

Table 3. The Posttest and the Response Statistics

	Question	**Avg.**	**Min**	**Max**
1	How useful was the course manual?	3.58	3	5
2	Given the price of the manual, rate your opinion of its utility?	3.75	3	5
3	How helpful was the PSPICE tutorial?	2.60	1	5
4	Did the hardware work help you understand digital electronics?	4.13	3	5
5	Will the hardware work help you retain what you've learned?	4.25	3	5
6	Did the hardware work help you understand datasheets?	3.25	1	5
7	Was the content of the hardware homeworks was suitable?	3.92	2	5
8	Did you and your teammates share the hardware load fairly?	3.25	1	5
9	Rate your ability to analyze an existing digital circuit	3.58	3	4
10	To what extent did the hardware contribute to the rating above?	4.25	3	5
11	Rate your ability to design a digital circuit of your own.	3.42	2	4
12	To what extent did the hardware contribute to the rating above?	4.08	3	5
13	Rate your ability to construct and troubleshoot a digital circuit.	3.50	2	5
14	To what extent did the hardware contribute to the rating above?	4.00	3	5
15	Did the PSPICE work help you understand digital electronics?	3.33	2	5
16	Will the PSPICE work help you retain what you've learned?	3.00	1	5
17	Rate your ability to use manufacturer's datasheets.	3.25	2	5

Results and Future Work

In gauging the success of the Hardware Experience, we considered four metrics: how much material has covered (depth and breadth), did the hardware/software homeworks stay within the time allotments and budgetary constraints, were the students better motivated and have the students retained the information better or longer? Compared to previous semesters, no course content sacrificed. Perhaps depth in some topics was traded for exposure to PSPICE and hardware issues such as ROM programming, but no lecture topics were omitted. There is however another issue – seeing the big picture. Based on personal conversations, we believe that the combination of software, hardware and theory helped students begin an understanding of the underlying principles upon which the field of digital electronics is based.

From the posttest, we found that the hardware and software portions of the homeworks took essentially the same time outside of class as the pin-and-paper solutions. This validated the Hardware Experience as being consistent with the existing curriculum requirements. Judging student motivation is more difficult, especially over such a short period of time. Certainly the overwhelming recommendation to continue using the Hardware Experience suggests that our teaching method was not demoralizing. The complementary argument however is hard to prove.

Since spring semester 2001, the details of methodology have evolved somewhat. The hardware kits have been streamlined to reduce costs. Both the manual and PSPICE tutorial have been expanded. In Spring semester 2002, we reevaluated the course content, looking for ways to complement the existing CS/SE curriculum. A decision was made to augment the introduction to computing systems by adding design content for a simple ALU and CPU. Over the course of the semester, student teams will create a unique assemble-level instruction set, write the necessary assembler and design a functional CPU. PSPICE will be used to validate the work.

Conclusions

A new approach for incorporating software/hardware into digital electronics service classes at Auburn University, called the Hardware Experience, has been investigated. The method is highly flexible and requires no curriculum allotment for laboratory time or facilities. Student teams using hardware kits, design, construct and verify simple digital circuits as homework. Institutional costs are very low and nonrecurring and student costs are reasonable – roughly $8.00 per student. PSPICE is the simulation tool due to its affordability and ease of use. A PSPICE tutorial and a companion manual have been authored to better structure out-of-class assignments and fully utilize lecture time.

Pre- and posttests show that students find the Hardware Experience has improved their learning and retention of digital circuitry. Although less than 10% of them had ever constructed even the simplest circuitry, by semester's end a full 83% of respondents strongly suggest that the method be adopted.

Acknowledgments

The author must recognize the contributions of the students in ELEC 2200, spring semester 2001, at Auburn University. Their spirited efforts and kind participation in the survey instruments have made this publication possible. Also, Bill Capps of Auburn's Scientific Supply Store has been instrumental in acquiring and stocking the necessary hardware.

Bibliography

1. Parrish, A., Borie, R., Cordes D., Dixon, B., Jackson, J., Pimmel, R., 1999, "An integrated introductory course for computer science and engineering," *29th ASEE/IEEE Frontiers in Education Conference*, pp. 11a3-12 to 11a3-17, 1999
2. Gokhale, A.A., 1989, "To investigate the effectiveness of computer simulation versus laboratory experience, and the sequencing of instruction, in teaching logic circuits," Ph.D. diss., Iowa State University, 1989, Abstract in *Dissertation Abstracts International* 51: 96-A.
3. Garren, W.S. 1990, "An experimental evaluation of using computer-aided design simulations in teaching basic electronics at the college level," Ed.D. diss., North Carolina State University, 1990, Abstract in *Dissertation Abstracts International* 51: 1141-A.
4. Hall, T.M., 2000, "Using simulation software for electronics engineering technology laboratory instruction," *2000 ASEE Annual Conference Proceedings*, St. Louis: American Society Engineering Education.
5. Nelson, V., Nagle, H., Carroll, C., Irwin, J., *Digital Logic Circuit Analysis and Design*, Prentice Hall, Englewood Cliffs, NJ, 1995. (ISBN 0-13-463894-8)

William C. Dillard
William Dillard is a Ph.D. candidate in the Electrical and Computer Engineering Department at Auburn University. He holds M.S. and B.S.E.E degrees from Auburn University. His current research interests are teaching strategies that promote professionalism and career development in students as well as increase long-term retention of course material. Technical interests are in the area of power electronic systems and devices.