Flattening the Learning Curve for OrCAD-Cadence PSpICE

Kenneth J. Soda
Department of Electrical Engineering
U.S. Air Force Academy

Abstract: The circuit simulation program PSpICE, and its companion schematic capture tool Capture, find widespread use in contemporary Electrical Engineering curricula. This software suite, offered by Cadence PCB Systems Division, can be a powerful instructional asset as well as an important tool for student design. PSpICE also finds consistent application in industry, further motivating use in academia. In recent years however, the investment required to train students to apply this software has become ever more burdensome. Most especially in lower division courses, the unavoidable software training often confuses students and diverts attention away from the very principles the simulator is intended to illustrate. In order to “flatten the learning curve”, we have developed and implemented several techniques to simply student application of Capture and PSpICE, particularly in introductory digital, signals and circuits courses. We apply a building block approach across our curriculum, incrementally applying specially prepared software templates and custom instructional materials. Our “just in time” approach minimizes frustration and builds skill while continuing to support our course and institutional objectives. This paper details our instructional approach, illustrates our use of templates, and describes our supplemental materials. Internet links to our important software templates are also provided.

Introduction

The USAF Academy’s Department of Electrical Engineering is fully committed to the integration of contemporary design tools in its course of instruction. Indeed, one of our institutional objectives is to ensure our graduates possess competence in the use of these tools. As in many other undergraduate programs in our discipline, computer aided circuit analysis software has become an all but essential design tool. Not only does this software support instruction and validate student designs, it also realistically prepares our students for the environment in which they will find themselves after graduation. The difficulty with computer based CAD tools is their very finite potential to confuse, frustrate, even bewilder students. Modern CAD tools possess so many features and controls that the novice operator can be easily overwhelmed. Without careful instruction, the very principles we hope to illustrate become buried behind a host of radio buttons, icons and full down menus. After some years of development, our department has developed a carefully tiered process though which Cadence PSpICE\(^1\) circuit simulation is introduced and applied across our curriculum. We find this process significantly “flattens the learning curve” and permits our students to develop

\(^{1}\)OrCAD Family Release 9.2, Cadence Design Systems, Copyright 1986-1999, San Jose, CA.
competence without undue frustration and confusion. This paper outlines this process and details some of our innovative methods of instruction.

The underlying philosophy of circuit simulation instruction has three fundamental premises. First, we introduce only the simulation capabilities necessary to support the instruction at hand. Second, circuit simulation is always preceded by theoretical prediction of circuit behavior. In this way, the general nature of result of any exercise may be anticipated and grossly incorrect simulation may be dismissed. Third, we repeatedly use a the same circuit simulation software package across at least six courses taken during a three year span of instruction. This philosophy places the greatest burden of instruction on those courses appearing earlier in our program. Here careful preparations must be made to guide students through their first simulations. It also requires careful coordination across several courses taught by many different instructors. In some cases, a pedagogical price must be paid since the capabilities of a single circuit simulator may not meet the needs of all courses equally well.

In the sections which follow, the general features of OrCAD PSPICE, nature of software instruction and course specific applications will be described on a semester-by-semester basis. We will also describe innovative instructional materials and software we have ourselves developed to support this instruction.

*Cadence PSPICE – A Brief Primer*

In order to fully understand our pedagogical approach, some background on the nature and basic operation of the software tool is in order. Cadence PSPICE Version 9.2 Lite is used in most of our courses. This full featured “student” version operates identically to the more capable “professional” version. We restrict use of the professional version to advanced courses where circuits with high active part counts are encountered.

The major OrCAD-Cadence PSPICE family circuit simulation programs are described in Figure 1. The components, topology, interconnections and characteristic values of a circuit are specified through “Capture”. (See Figure 1a below) To the user, Capture operates in a fashion similar to a schematic drawing program. A spreadsheet like feature within Capture, called the Property Editor, is used to specify component settings. (Fig 1b) However, Capture is also used to specify the type of simulation to be completed. In our curriculum, we employ DC bias, transient (time dependent), frequency and component value dependent analysis. Simulation Profiles (Fig 1c) are used to define the type of simulation, along with its limits and parameters. Radio windows with tabs, push buttons and blanks are used to select or specify these settings.

Once a circuit and its simulation profile are established, the simulation engine may then be activated. The output of the Berkeley SPICE2 based simulator is presented in one of two places. The results of DC bias analysis may be displayed directly upon the same Capture schematic that was first used to describe the circuit. Such an annotated schematic is shown in Figure 2a below. Time, frequency or component value dependent analysis may be presented graphically in the PSPICE output window as shown in Figure 2b. The presentation of these data in this window may be optimized through scale modifications, introduction of new axes or plots.
Figure 1. OrCAD-Cadence Capture Screens. A) Capture window with completed circuit schematic. B) Property Editor window through which component values may be set in spreadsheet fashion. C) Simulation profile radio window for selecting type and duration or span of simulation.

Figure 2. Methods for presenting results of OrCAD-Cadence PSPICE simulation. A) Capture schematic with simulated DC bias currents and voltages marked directly on the schematic. B) PSPICE window showing results of a transient simulation. Axes have been modified to accent key results.
Building Fundamental Skills

Cadets enrolled in the Electrical Engineering and Computer Engineering majors take their first course in circuits and signals, El Engr 231, in their third or fourth academic semester. It is in this course where they first encounter PSPICE. Two exercises are used here to support the concepts of source transformation and first order RC circuits. PSPICE is used to evaluate the DC bias condition of a two-source resistor circuit and the transient response of a pulse source driven RC circuit. Parametric analysis of the RC circuit is desired in a nested simulation.

The essential skills which must be developed for these exercises are the creation of a blank PSPICE project, placement of independent voltage and current sources as well as resistors and capacitors; setting component values and source parameters; creating a simple simulation profile; executing the simulation and printing results. These simple skills are communicated via in-class demonstrations and written exercise guides. These guides consist of step-by-step instructions prepared in Microsoft Power Point briefing slide format. The slides illustrate the sequential process through which the exercise simulation may be created and executed. We use screen captures of the simulator windows integrated with text to illustrate essential exercise processes and settings. A sample of one such guide is illustrated in Figure 3. These guides may be created and modified with relative ease, are easily distributed to our students via our local area network and are easily understood and applied by our students. These guides are based upon the known condition of the simulator’s default “blank project” profile. Since all students download the simulation software from the same network drive, the initial condition of the PSPICE project is known in advance. These instructions are therefore carefully scripted against this known condition.

Figure 3. Example slides from the EE 231 Computer Exercise 2 instruction slides. Screen captures of OrCAD PSPICE windows are combined with text to illustrate key icons, screen faces and process in chronological order.
In these introductory PSPICE exercises, no effort is made to explain the mathematical operation of the simulator, to enumerate the number and types of possible simulations, to manage the project files, or to change the simulator's default settings. Most of the Capture and PSPICE icons are ignored in the instructions. By keeping the explanations very simple and the breadth of the exercises narrow, we communicate two basic PSPICE analysis modes, reinforce instruction and minimize the chances for student induced errors. Typical Capture and PSPICE input and output screen faces seen in the EE 231 exercises are illustrated in Figures 4 and 5 below.

**Initial Simulation Skills - I**

*DC Bias Analysis*

**DC Circuit Schematic with Finished Bias Analysis**

![Simulation Profile](image)

**Figure 4.** First application of the PSPICE simulator in EE 231 is a simple resistive circuit with two DC sources. Simulation profiles require one radio button setting. Results are displayed directly on the Capture circuit schematic.

It is increasingly common to find textbooks accompanied by simulation software guidebooks. In the case of El Engr 231, Nilsson and Ridel’s “Introduction to PSPICE” comes as ready source material with the course textbook “Electric Circuits”. While these guides are helpful, the continuous improvement of the software often renders these guidebooks obsolete in just a few semesters. This trend has motivated us to rely upon our own instructional materials which are reviewed and updated after each software upgrade. We find this practice better assures consistency between classroom instruction and simulator behavior.

**Digital Circuit Simulation**

The careful restriction of PSPICE projects practiced in El Engr 231 must be relaxed as our students enter their first course in digital circuits. El Engr 281, Introductory Digital Systems, is normally taken in the fourth academic semester and is therefore the second course in which...
Figure 5. Second EE 231 application of the PSPICE simulator is an RC circuit driven by a pulsed voltage source. Parametric variation of the capacitor’s value is set through the simulation profile. A single voltage marker selects the capacitor’s voltage for display in this transient simulation.

PSPICE finds application. Here the standard PSPICE 7400 series logic gate models serve as a basis for demonstrating the operation of combinational and sequential logic gates and the design of MSI level digital systems. It is our observation that a number of complications arise which make PSPICE digital simulation more troublesome to students. In order to deal with these complications, special instructional materials have been developed exclusively for EE 281.

The root cause of student difficulties in digital simulation is the high level of realism which PSPICE seeks to achieve. Through the Cadence PSPICE evaluation library which is...
supplied with Version 9.2 Lite, logic gate function can be easily simulated. Capture schematics of these gates even include correct part pin numbers. However, the simulator also models gate signal delays, set-up/hold times, and open collector gate behaviors. Advanced courses consider these digital behaviors in detail. However, El Engr 281’s course goals are more readily met if these effects are simply masked. It is also important that students have digital clock and binary counters signals readily available to exercise their digital designs. Each one of these course specific requirements brings with it a unique series of component characteristics, simulator settings and schematic naming conventions and symbols. Herein lies the great potential to confuse the first time students of digital simulation.

Just as in El Engr 231, we have found detailed process descriptions an important educational asset. We have developed a set of step-by-step instructions demonstrating advanced simulation skills based upon those developed earlier in the curriculum. For El Engr 281, we shift the focus to file management skills, model library assignment, repeated part placement, data bus naming, connection to data buses, marking and assessing digital signals, re-formatting simulated data, and identification of high impedance or indeterminate output signals. Using this information, our students are easily able to save, retrieve and modify their schematics as well as better display and interpret their simulation results.

El Engr 281’s unique simulation requirements have spurred the development of perhaps our most unique teaching innovation, the project “template”. These templates serve as the starting point for student simulations with either static (bias point) or dynamic (transient) behavior. The templates are partially completed PSPICE projects from which new student schematics may be created. These templates possess static logic signals, a clock source and binary counters of two to four bits width. These signals have been carefully adjusted relative to one another to avoid set up and hold time violations. The templates also store key default settings. For example, the template’s simulation profile is preset for a fixed duration which is longer than one cycle of any anticipated course exercise. The template’s profile is also prepared to initialize all flip-flops to logic zero. By eliminating analog components of any kind, the templates allow students to see only logic signals (as opposed to analog voltages) in their simulation outputs. This practice simplifies simulation and maximizes the number of logic gates which may be studied with the student version of the software. The process descriptions have been carefully coordinated with the simulation templates, but have been written in a general enough fashion that they become course, rather than exercise specific.

It is worth mentioning that the static simulation template is actually based upon a brief transient simulation. We have discovered that in so doing, the results of the bias point simulation will be displayed on the circuit schematic, much as they were in El Engr 231. This practice minimizes student frustration in early static logic simulations.

An example student project, based upon our transient digital circuit template, is illustrated in Fig 6. In the figure, elements of the schematic which belong to the original template schematic are differentiated from those belonging to the student’s circuit.

Figure 6. Digital circuit simulation based upon the author’s “Trans.opj” simulation template. Fixed logic signals, digital clocks and binary counters are included in the template. Student’s replicate the templates and then add to them to create and simulate their exercise circuits. Templates also carry information on digital signal frequency, simulation duration and initial settings of flip flops.

Solidifying Simulation Skills

During semesters 5 and 6, cadets become more skilled in the operation of PSPICE as a consequence of their study of electronics. Instruction on electron device physics and their application in fundamental circuits is delivered in El Engr 321 and 322. PSPICE supports prediction of circuit behavior in five of the eight hardware laboratory exercises delivered in these courses.

While a fundamental understanding of Capture and PSPICE is assumed, cadets review file management, circuit wiring and basic simulation settings during the first laboratory exercise. However, the use of the property editor to set component values is stressed to a much greater extent. Students are also expected to properly select the type and duration or limits of their bias point and transient analyses, link component libraries and manipulate PSPICE output. As such, we do not provide circuit templates. However, written process guides similar to those of El Engr 281 have been developed as reminders of these basic skills.
El Engr 321 and 322 naturally demand two new simulation types. The non-linear behavior of diodes and transistors can be better illustrated using both primary and secondary DC sweeps. Thus the traditional IC vs VCE with varying IB curves of the BJT may thus be generated. The frequency dependent behavior of transistor circuits, including circuits with feedback, naturally require AC sweeps. We find very little instruction is required to support these new analysis types as modifications to the Simulation Profiles are so trivial.

It is important that the device modeling parameters we discuss in class be applied in simulation. To that end, we must demonstrate the method by which simple PSPICE active device models may be created. This skill is appropriately integrated into the junction laboratory exercise via a single page process description. This process is applied again in the exercise on field effect transistors.

The only remaining new simulation skills taught during this academic year are advanced output plotting and scaling techniques. The creation of multi-vertical scale, multi-plot graphs with mathematically manipulated traces is demonstrated in class. Since El Engr 321 and 322 students have the opportunity to perform simulation during class time, the limited number of pull downs and key strokes required to create these plots are easily communicated and reinforced. An example of such a plot is shown in Figure 7.

Advanced Simulation Skills and Printed Circuit Board Layout

During the 7th academic semester, one new Capture related skill is taught. El Engr 463, Design Project Techniques was created to better arm our students for the hardware realization of their capstone design projects3. A key element of this course is instruction in computer aided printed circuit board (PCB) design. Here we take advantage of the schematic wiring skills our cadets have developed over the two preceding academic years. Capture schematics can be modified to carry component footprint as well as interconnection information to the companion Cadence printed circuit design tool “Layout”. Layout merges the interconnections and component package types defined through Capture with stored component dimensions and printed circuit board attributes. The resulting “rats nest” representations of a PCB can be either manually or automatically translated into a finished printed circuit board designs. Using Layout produced photo-masks, we can create finished double-sided PCBs in our simple in-house process. An example PCB design is illustrated in Figure 8. Although it is not demanded in all projects, cadets are highly encouraged to apply this skill in their individual design projects completed under the auspices of El Engr 464, taken in their eighth and final semester.

As has been so successfully applied throughout our curriculum, El Engr 463 makes use of illustrated process descriptions and in-class practice to teach the PCB CAD process. The process description actually extends beyond the operation of the computer tools to include the operation of the in-house printed circuit board fabrication facility.

The Capture and PSPICE instruction described in this paper has thus far been limited to “core” courses required of all electrical and computer engineers. The most advanced applications

---

Figure 7. Example of 5th and 6th semester simulation skills. Frequency dependent (AC Sweep) simulation is demonstrated. Multiple output traces are plotted against different axes which may be mathematically manipulated.

of the Cadence tools are made in an elective course, El Engr 473, Introduction to CMOS VLSI Circuit Design. In this course, Capture and PSPICE are used to model the behavior of microchip-scale MOS transistor circuits. Since these circuits tend to be highly repetitive, we make use of Capture’s capabilities in hierarchical design. For example, an SRAM cell need be designed only once. Identical “instances” of this cell may then be applied many times across a larger circuit. By this point in their education, our cadets are highly familiar with the student version of the OrCAD Tool suite. Since the “Professional” version of the software is identical in appearance and function, cadets enrolled in El Engr 473 make a seamless transition to this more capable tool.
Figure 8. Example application of Cadence Capture and Layout creating a printed circuit board design. This skill is taught during the 7th academic semester as part of El Engr 463, Project Design Techniques.

Perhaps the most unique application of Cadence PSPICE in our curriculum is the simulation of integrated circuits designed in VLSI polygon representation. We make use of a non-Cadence VLSI layout design tool. The VLSI layout tool converts polygon representations of integrated circuits into standard PSPICE netlists. These netlists are universally understood by PSPICE simulators, including the Cadence tool to which our students have grown so accustomed. Again though a written process description, our students modify the netlists to include source, control and output statements into a text file. This file is then read and executed by Cadence PSPICE. The performance of the circuit can be evaluated using the graphing methods which have been so well reinforced in other courses. A sample of just such a simulation is presented in Figure 9.

Measures of Success

As mentioned at the beginning of this paper, our department believes all of its graduates should possess a minimum competence in the operation of modern design tools, including electronic circuit simulators. As such, it is important to examine the measures by which we assess our success in reaching achieving this objective. We can thereby judge the adequacy of our instruction in electronic circuit simulation.
Proceedings of the 2002 American Society of Engineering Education Annual Conference &Exposition  
Copyright © 2002, American Society of Engineering Education

Figure 9. Although not taught to all students in the electrical engineering major, students enrolled in El Engr 473 learn to extend their simulation skills to SPICE netlists created by a non-Cadence VLSI editor. The very same simulation skills developed in the preceding four semesters are applied in this course. New simulation skills learned in this course include hierarchical circuit design using OrCAD Capture.

We annually assess first classmen (seniors) enrolled in the electrical engineering major on their ability to interpret the results of OrCAD Cadence PSPICE simulation. This assessment is performed as part of a “Skills Review” conducted at the beginning of their final semester. Each cadet is presented with a hardware circuit which may or may not contain a fault. Each circuit is accompanied by selected PSPICE simulation results which illustrate the desired behavior of the hardware circuit. In order to successfully pass this assessment, a satisfactory analysis of the hardware circuit’s condition is required. A correct interpretation of the simulation output is a major element of this assessment. While this assessment does not focus exclusively upon PSPICE simulation, the very high (>98%) passing rate on this evaluation tends to indicate our students have developed a good understanding of these data.

A more direct assessment of student competence with PSPICE is accomplished as part of El Engr 321 and 322 during the sixth and seventh semester. A Pass / Fail grade is awarded for demonstrated skill in circuit simulation as part of the course laboratory report assessment process. In order to receive a passing mark, cadets must demonstrate competence in creating...
schematics, running simulations appropriate to the exercise and interpreting the results. To date, this single measure indicates nearly all our students (> 90%) demonstrate a passing minimum competence in simulation.

While these two measures give encouraging indications on overall competence in simulation, they do no offer much guidance into student perceived adequacy of instruction. A better indicator of our instructional effectiveness may be garnered from surveys and exercise performance data. The academic course chosen for survey should be early in the curriculum, where the simulator is used extensively and where the level of student experience is still relatively low. We have chosen El Engr 281 for this purpose.

The table below summarizes the results of the most recent semester survey completed after the second laboratory exercise. Those questions related to the operation of PSPICE indicate a general agreement that our simulation instructional materials and methods are correct and useful to students. Given the total number of times the simulator was used prior to this exercise, it is not surprising to find a slightly negative average response regarding simulator ease of use. Since the introduction of course specific PSPICE instructional materials, the level of student frustration, as perceived by instructors, has dropped “significantly”, while the level of student performance in simulation has risen.

<table>
<thead>
<tr>
<th>Student Survey Statements</th>
<th>Average Score, n = 73</th>
<th>Std Deviation</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(1 – Strongly Agree</td>
<td></td>
</tr>
<tr>
<td></td>
<td>3 – Neutral, 5 –</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Strongly Disagree)</td>
<td></td>
</tr>
<tr>
<td>The illustrated operating instructions for PSPICE were an important and useful guide.</td>
<td>2.30</td>
<td>1.24</td>
</tr>
<tr>
<td>The illustrated operating instructions for PSPICE were incorrect.</td>
<td>3.77</td>
<td>1.17</td>
</tr>
<tr>
<td>The circuit template files were completely worthless to my design effort.</td>
<td>3.79</td>
<td>1.44</td>
</tr>
<tr>
<td>Overall, I found the PSPICE simulator easy to operate.</td>
<td>3.20</td>
<td>1.33</td>
</tr>
<tr>
<td>I found the simulation output graphs easy to interpret.</td>
<td>2.19</td>
<td>1.18</td>
</tr>
</tbody>
</table>

| Table 1. Results of El Engr 281 Lab 2 Student Survey. The average response of 73 students is listed on the 1(Strongly Agree) to 5 (Strongly Disagree) scale. |

While they are difficult to quantify, there is much anecdotal evidence that PSPICE becomes more easily applied as the level of student experience grows. I have often heard cadets in their final semesters comment to younger students how “user friendly” PSPICE is. I have observed students create extremely complex multi-level VLSI simulations with only minimal instruction. I have seen the ease with which students successfully apply our 26 page PCB design process guide with only one brief in-class instruction session. There was even a case this past semester where a cadet attempted to create a PCB design without the benefit of classroom instruction. Even without formal training, the student was able to create a fully functional 6"X10" PCB. These informal measures are, in my opinion, additional indications that our instructional processes are sound and effective.
Conclusion

During six of their eight semesters, cadets enrolled in the USAF Academy electrical engineering curriculum receive an incremental education in the operation of a single circuit simulation tool suite, Cadence Capture, PSPICE and Layout. We have found them to be important tools which support and reinforce classroom instruction at the same time as they educate our cadets about the capabilities and limitations of CAD. Our philosophy is to teach only those simulator features which are needed to immediately support the instruction at hand. We have also carefully developed our own customized instructional materials which focus upon a limited number of features in each exercise. The consistent reinforcement of simulation entry and control in the same software increases learning effectiveness and minimizes student frustration. We thus create a “flattened” learning curve where the tool’s instructional burden is distributed among many complementary courses while supporting the corresponding course objectives.

KENNETH J. SODA. Dr. Soda is the first permanent civilian faculty member of the USAF Academy’s Department of Electrical Engineering. He holds an advanced degree from University of Illinois, Urbana-Champaign. Dr. Soda is the 1997 Recipient of the Tau Beta Pi Teacher of the Year Award (Colorado Zeta Chapter) and the 1988 Recipient of the USAF Academy Outstanding Educator Award.