AC 2007-1752: ACTIVE-HDL, MULTISIM, CADENCE... THERE HAS GOT TO BE A BETTER WAY TO TEACH CAD/E TOOLS

Jeff Gribschaw, D/EECS, USMA

MAJ Jeff Gribschaw is an instructor in the Electrical Engineering Program at the US Military Academy. He has a Master of Science Degree in Electrical and Computer Engineering from the Georgia Institute of Technology and is a member of IEEE.

Paul Patterson, D/EECS, USMA

MAJ Paul Patterson is an instructor in the Electrical Engineering Program at the US Military Academy. He has a Master of Science Degree in Engineering Management from the University of Misouri – Rolla and a Master of Science Degree in Electrical Engineering from George Washington University. He is a licensed Professional Engineer in the state of Missouri and a member of IEEE.

Bryan Goda, USMA

COL Bryan Goda is a permanent Academy Professor and director of the Information Technology Program at the US Military Academy. He has a Ph.D. in Computer Engineering from RPI and is a senior member of IEEE.

Active-HDL, Multisim, Cadence…
There has got to be a better way to teach CAD/E Tools

Abstract

Numerous computer aided design (CAD) and engineering (CAE) software products exist to automate the design process, but how does an instructor efficiently incorporate those tools into the classroom to facilitate learning? A typical electrical engineering major at this school may use up to twenty different software products over the course of two and a half years. CAD/E tools play an important role in enabling students to take concepts learned in the classroom and apply them to real world problems and significantly enhances student learning. Many text books come with support for a specific CAD/E tool with many examples, but gloss over the use of the CAD/E application and expect that the student already knows the software or will learn it on their own. Many courses do the same, based on the amount of material they are required to cover, and there is not enough time to also instruct students on the operation of a specific CAD/E tool. In our digital logic and computer architecture courses, we currently spend approximately two hours over two semesters teaching students to use Active-HDL; we require students to use the program in multiple labs and homework assignments to reinforce key concepts in Very High Speed Integrated Circuit Hardware Description Language (VHDL). Due to the fact that our instruction only touches the surface of the program's capabilities, students expend a significant amount of additional time and effort learning to use Active-HDL at the expense of learning the key concepts we would like to emphasize with VHDL. Too often the students spend countless hours attempting to learn the software and fail to grasp the actual concepts that the software was supposed to reinforce. One course of action to eliminate this problem is to stream line the software tools used throughout the curriculum. This would require a conscious, program-wide effort to redesign the course curriculum using only one CAD/E product and is not a feasible solution to our problem. Instead, we propose a coordinated software effort within a program and an assessment program geared specifically towards the CAD/E tools used in each course to help instructors enhance classroom instruction and out-of-class independent student learning of new CAD/E Software. This paper will focus on the software application Active-HDL, the programming language VHDL, and an assessment tool we use to improve student proficiency with these CAD/E tools.

Background

The original objective of our current research was to develop a set of tools we could apply to help students learn the software program Active-HDL and the programming language VHDL. Our intent was to develop these tools within the digital logic thread of our Electrical Engineering program, and then apply these tools to other software applications and programming languages within other threads of our EE program. We believed that we could develop innovative techniques to facilitate improved learning.

In our Master Teacher Program, instructors learn about student learning techniques, course analysis and course design.[1] One of the requirements is to complete a classroom assessment research project. As we looked at course assessment in our digital logic and computer architecture courses, we identified issues with students spending an inordinate amount

of time learning the software and not using the CAD/E tools to reinforce the principles or purpose of the assignments. We have used and are continuing to use multiple assessment tools to evaluate the student's knowledge and performance and will address some of the techniques within this paper.

We can apply the lessons learned from various assessment techniques to improve both formal and informal instruction. We refer to formal instruction as dedicated class time in which the instructor has set aside time and planned the lesson to provide instruction specifically on the CAD/E tool. We refer to informal instruction as those occasions at the start, middle or end of class where an instructor spends a couple of minutes providing simple direction or guidance to assist students in learning or using the CAD/E tool. This may occur as a result of a survey response, in response to a question a student had during additional instruction/office hours, or as a result of a question before or during class. An example of informal instruction that could result from the survey would be to show the Active-HDL help index at the beginning of the next class period. We learned in our Master Teacher Program that addressing the survey (or other assessment technique used) with students and answering their concerns greatly helps build a better learning environment.

The potential exists for students to be overwhelmed with the number of CAD/E tools they are required to learn over the course of earning an undergraduate electrical engineering degree. Table 1 lists the twenty-three software programs used in the United States Military Academy (USMA) Electrical Engineering program and all of the courses that use each software. A quick scan of the table reveals that we expect students to learn a tremendous number of applications over the course of their final two and a half years. Not reflected in the diagram is that we spend less than twenty hours of formal instruction on how to use the programs, relying heavily on self-study in the form of tutorials, homework assignments and lab exercises. The focus of this paper is to understand the requirements of a modern EE program's use of CAD/E tools and identify methods to enhance student learning of these tools in a time constrained environment. We propose a coordinated effort to streamline software use where possible. We highlight the importance of a good learning environment for the formal instruction conducted on CAD/E tools. We also emphasize the importance of in course assessments of student proficiency with CAD/E tools to better focus formal and informal instruction (both during the current semester and for future semesters). Our approach is novel in that it will lend itself to improving student ability to learn CAD/E tools across the entire EE curriculum and in that it incorporates an assessment mechanism that has the flexibility to support multiple CAD/E tools. Our findings emphasize the importance of developing a teaching focused learning environment to maximize student potential.

The related work section will address some of the tools available for helping students to learn CAD/E tools. In the methodology section, we will address several potential approaches to dealing with the large number of programs required within an undergraduate EE program, and we will lay the foundation for the assessment mechanism we have employed to a limited extent. The results section will provide the survey data we have collected. We will analyze that data in the analysis section and will then provide our conclusions.

Software	Course	Use
Agilent LogicWave	Microprocessors	Logic analyzer, capture waveforms
Active HDL	Digital Logic	Digital Logic Simulation, FPGA prog
	Computer Arch	Register File, ALU, organization
	Adv Comp Arch	Pipelining, branch predict, other speedups
Cadence	Electronic Design	Op Amps, Active Filters
	Photonics, Power	PSpice
	Electronics, Solid State	PSpice
Chipmaster	Digital Logic	Chip programming
Circuit Maker	Design	Circuit Boards
EXCEED	Microprocessors	GCC Compiler
GNU Radio	Telecommunications	Signal Simulation
	Digital Networks	FM Receiver
	Wireless Systems	FM Transceiver
IC-CAP	Photonics, Electronics	Semiconductor modeling
	Electronics	Diode, BJT modeling
Imagecraft ICC	Microprocessors	C Compiler
Labvolt	Wireless Systems	Modulation Simulation
Lattice IspLever	Digital Logic	Program CPLDs
Tanner Tools Pro	Solid State	L-Edit Layout editor
Logical Devices AllPro	Digital Logic	Chip Programmer
MARASM Assembler	Computer Architecture	Assembly Language programming
MATLAB	Signals and Systems	Block diagram simulation
	EM Fields	Field effects
	Controls	Block diagram simulation
	Telecommunications	Filter Design, Signal Processing
	Digital, Wireless Net.	Filter Design, FFT
Microload	Microprocessor	Downloader for 68HC11
Microsoft Office	Senior Design	Report creation, data collection
Microsoft Project	Senior Design	Project administration and scheduling
Multisim	Basic EE	Circuit Simulation
OPNET/NETWARS	Telecommunications	Communications Model Library
Pbasic	Basic EE	Robot programming
Simulink	Communications	System Modeling
THRSim11	Microprocessors	68HC11 simulator

Table 1 – CAD/E Tools/Applications used in USMA EE Program

Related Work

A plethora of tools and papers exist to help teach specific CAD/E tools, but there is very little research from a macroscopic perspective that one can generally apply to an EE program or to a general CAD/E tool. Many publications describe the problems associated with learning the syntax (or what button to push) and not having enough time to develop skills in program design. Often a new tool is developed to reduce the time learning the syntax of a programming

language.[2] Advances in software design now allow a student to use a top down design tool to describe a system's behavior in VHDL, simulate its performance, download the circuit into a programmable chip, and then measure actual performance.[3] This approach has been successfully used in several of our computer architecture courses.

There has been a movement towards a Microsoft Office for EE programs, a total integrated software package to support a wide variety of EE courses. The closest version of Microsoft Office for EE programs is probably the Cadence Design Systems.[4] While this product is attractively priced and is widely used in industry, there is a steep learning curve in utilizing this tool. For the foreseeable future we will continue to have a large mixture of tools such as those listed in the CAD table.

Methodology

The obvious approach to minimize the large number of CAD/E tools used in an Electrical Engineering program is to pick a single program that does everything you need. This solution is not feasible because there is not a single solution software that meets the broad requirements of an ABET accredited electrical engineering program; such an application would have to address everything from solid state design layout, to analog circuit analysis, to high level digital design simulation. Even if there were such a program, unless industry exclusively used the program, EE programs would need to develop graduates who can apply their current knowledge to new applications (or new CAD/E tools). One of our institute's Academic Program goals is to graduate students that, "in response to a technological problem, learn new concepts in engineering and learn about new technologies without the aid of formal instruction." [5] We want graduates who have the ability to learn a new CAD/E tool on their own to solve a real world problem. An important requirement for a modern EE program becomes determining the appropriate balance within the curriculum between formal teaching and self-learning of CAD/E tools.

One method to accomplish this objective is to have a coordinated effort within the program to minimize the number of different CAD/E tools used. For example, it does not make sense for a digital logic course to use Active-HDL to teach VHDL programming, and then for a follow-on computer architecture course to use Multisim to complete a VHDL assignment. One potential challenge here is that individual instructors may resist moving to a different CAD/E tool; the goal remains to find an appropriate balance—a program will need some variance to meet the institute and industry requirement for graduates with the ability to learn and apply new technologies. A second method is to base the amount of formal instruction on where the specific course falls within the curriculum. An introductory level course (for sophomores or juniors in our program) would involve more formal teaching, and more advanced courses would involve less or no formal teaching. The specific CAD/E tool should also considered when determining this balance; for example, Ledit is a complex tool for which the instructor spends three class hours providing formal instruction in a senior level course.

As the program and individual instructors determine where to include formal instruction on CAD/E tools, the classroom or computer lab for that instruction should provide an enhanced

learning environment which maximizes student learning potential. While observing a computer architecture course taught last semester, our EE program director observed a poor learning environment for software demonstration. (See Figure 1A.) As noted in the survey results below, the digital logic lab also does not provide a good environment for demonstrating CAD/E tools. (See Figure 1B.) In both cases, many of the computer workstations face away from the front of the room, making it difficult for students to follow along with the instruction. To address this issue, the EE program converted a standard classroom into a computer lab focused on maximizing the learning environment within the digital logic thread. (See Figure 1C.) This new teaching lab provides a workstation facing the front of the room for all students in the course, allowing them to easily follow along during formal CAD/E instruction. The room also supports conducting digital logic labs at student workstations, so that each student gets the hands on experience, instead of having to work in groups of two or more.

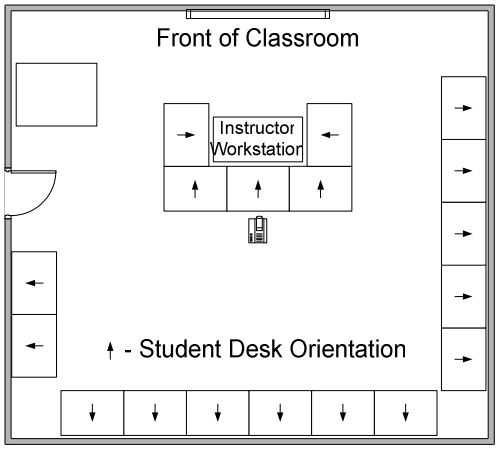

Figure 1A – Poor Teaching Configuration

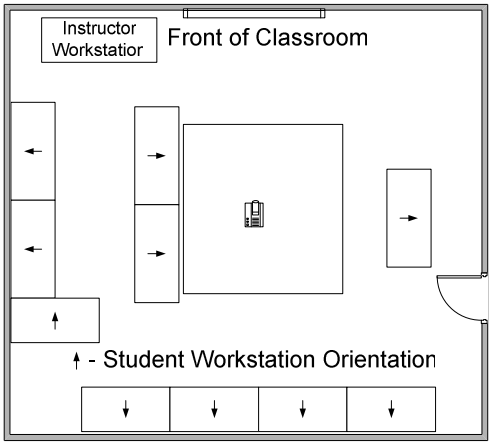

Figure 1B – Poor Lab Configuration

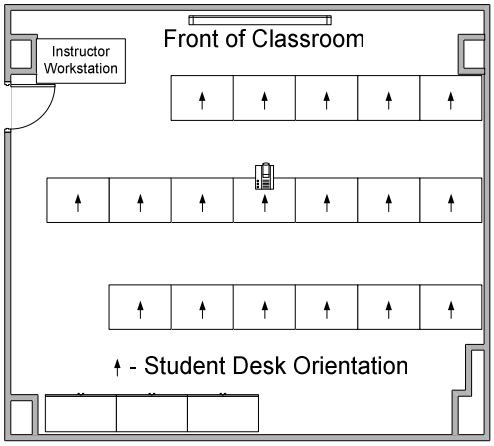

Figure 1C – Enhanced Teaching Configuration

Regardless of the environment and level of coordination within the program (or for any individual course), conducting assessments to gain feedback from current students possesses the potential to greatly improve the ability of instructors to find the right balance between in class instruction and self-study.

At this phase of our classroom research project, we are focused on the CAD/E tools used in our digital logic course and our introduction to computer architecture course. We have developed and employed assessment surveys focused on Active-HDL and VHDL; the surveys purposely separate the Active-HDL application from the VHDL programming language because the goal across the digital logic thread is that students should be able to apply VHDL to simulate hardware systems; Active-HDL is simply one of many CAD/E tools that allows us to accomplish this objective.

We observed some increased motivation to learn Active-HDL and VHDL simply by creating an environment where feedback was requested and subsequently addressed. We mentioned the surveys early in the semester and then issued them between six and eight weeks into the sixteen week semester. This helped ensure feedback was closer to the experience of learning and using the CAD/E tools and prevented the surveys from being hastily completed along with all of the other end-of-course surveys while students are also preparing for term end exams.

The surveys are provided in Table 2 below. Other than the first question of each survey, which we designed to give a numerical snapshot, the questions were specifically designed to be open-ended to solicit a wide range of comments with which we could improve the respective courses. We used the same two surveys in both the digital logic and computer architecture courses.

Active-HDL Survey	VHDL Survey
This is a survey concerning the software program Active-HDL. Please respond constructively to the questions below. Your comments should be directed towards the software program Active-HDL and not the programming language VHDL. Thank you for the time to take this survey. 1. Rate your level of proficiency in the software program Active-HDL on a scale of 1 to 5 (as far as this course is concerned): 1: No proficiency 2: Some proficiency 3. Neutral 4: Mostly proficient 5: Totally proficient 2. Explain your answer to question 1. (What do you think contributes to this level of proficiency?)	This is a survey concerning the programming language VHDL. Please respond constructively to the questions below. Your comments should be directed towards the programming language VHDL and not the software program Active-HDL. Thank you for the time to take this survey. 1. Rate your level of proficiency in VHDL (the language) on a scale of 1 to 5 (as far as this course is concerned): 1: No proficiency 2: Some proficiency 3. Neutral 4: Mostly proficient 5: Totally proficient 2. Explain your answer to question 1. (What do you think contributes to this level of proficiency?)

3. What do you like about the software program Active-HDL? 4. What do you dislike about the software program Active-HDL? 5. What changes would you recommend be implemented next semester to help students learn the software program Active-HDL? (What would help you better learn the software program Active-HDL?) 6. Do you have any other comments about Active-HDL. (Responses are Optional.)	3. What do you like about the hardware descriptive language (VHDL)? 4. What do you dislike about using VHDL (the language – not the software package Active-HDL)? 5. What changes would you recommend be implemented next semester to help students learn VHDL? (What would help you better learn VHDL?) 6. Do you have any other comments about VHDL? (Responses are Optional.)

Table 2 Active-HDL and VHDL Assessment Surveys

Results

Figure 2 provides the results of question 1 from all four surveys. Table 3 includes some selected responses from the other five questions; there are too many responses from the four surveys to include each of them in this paper.

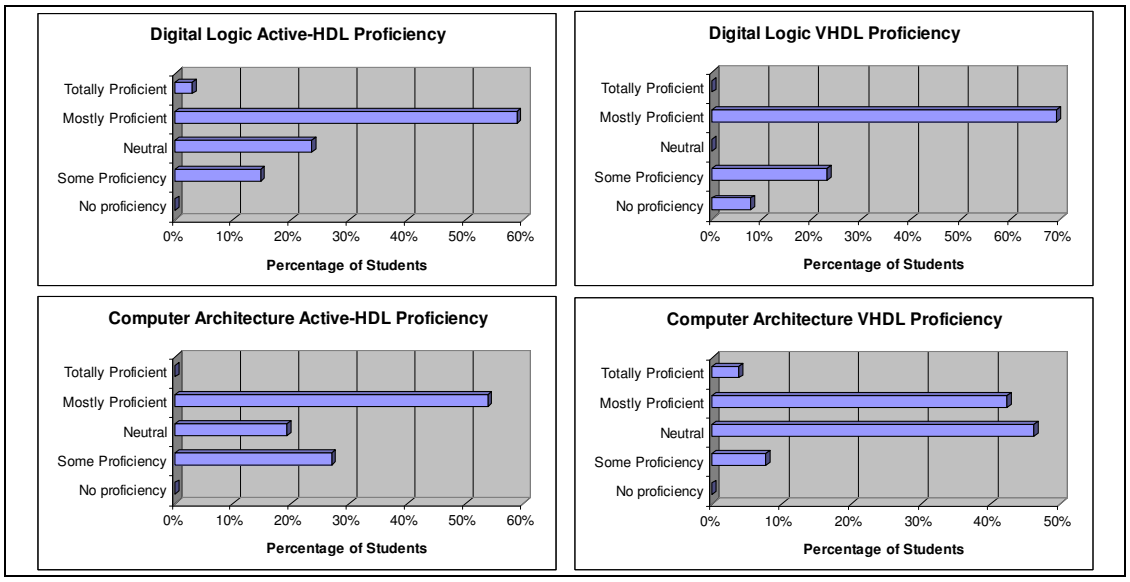

Figure 2 – Student Proficiency Self-Evaluations

Digital Logic -- Active-HDL	Digital Logic – VHDL
• I need more practice to increase my proficiency • There isn't a help index to help with troubleshooting • We should spend more time on HDL in class, including in class exercises • Make the first or second lab focused on an Active-HDL tutorial • Forcing students to use it a lot helps them learn it • Progression from small assignments to larger ones is great • Not sure how to re-open stuff in active-HDL	• Provide a list of commands available • It was troubling to have code compile and simulate in Active-HDL and then not work on the chip—add more instruction on VHDL • Add more classes focused on teaching VHDL • Need a comprehensive reference
Computer Architecture -- Active-HDL	Computer Architecture – VHDL
• I catch on when I work thru problems on my own • I find it hard to link the VHDL code to what happens physically • I find it hard to understand the workspace—I am constantly closing the program to open new workspaces • Troubleshooting problems within the software is my biggest problem • A quick reference on common errors as well as an Active-HDL reference card would be helpful • It would be helpful to add several small coding assignments before lab 1 is due • Add one or two lessons on Active-HDL to digital logic • I think doing problems as a class, where the teacher can explain the code would be helpful; when we did this in digital logic, not all students had a computer and not all computers faced the screen. • Encourage students to start early	• Doing the labs helps understand VHDL • Trial and error (and eventually getting it to work) builds proficiency • Digital logic should have explained the differences between concurrent and sequential statements • More practice—in class exercises • Add book references to the VHDL reference sheet • Highlight the differences between VHDL and other popular programming languages • Add smaller graded events with simple VHDL problems

Table 3 – Selected Survey Results

Analysis

An analysis of Figure 2 reveals that in all four surveys, the majority of students rated their level of proficiency as neutral or higher; in three out of four of the surveys the rating of

mostly proficient received the largest number of responses. Even through a couple of students did, it is unlikely that students in introductory level EE Courses would rate themselves as totally proficient with a software program or a programming language as powerful as Active-HDL and VHDL, respectively. In general, students have the knowledge they need to complete the assignments, some with classmate or instructor assistance. The charts reveal that there are a number of students whose proficiency has room for improvement; this is where the remainder of the survey questions can help the instructor focus formal instruction or highlight potential areas for informal instruction to best help the students improve their learning of the CAD/E tool.

One survey comment to highlight is the request that instructors highlight the differences between VHDL and other languages. One of the things highlighted by Dr Mark Evans in our Master Teacher Program is the need for instructors to help students link the concepts they learn throughout a course into the big picture of the course. This can be extended to CAD/E tools, and specifically VHDL, by helping students link what they are already familiar with to the new tool; this helps link their knew knowledge to their current knowledge base and can make it easier for them to learn the new material.

A common theme from the survey, with several responses included in Figure 4, was the need to have ample opportunities to practice in order to better learn the CAD/E tool. One of the specific recommendations was to have a series of small exercises build up to the first major assignment to build confidence with the Active-HDL and VHDL. As instructors try to balance the appropriate number of assignments and corresponding amount of time spent grading, providing optional smaller problems can be beneficial to students. In response to students requesting this type of help (using less formal assessments prior to these surveys), we provided some example problems (with a solution available); many students worked through these additional problems on their own and reported that the exercises were extremely helpful. A corresponding theme was the desire to do more in class exercises where students could work through problems on their own, with the ability to ask the instructor questions if they got stuck so they did not spend hours trying to work around a simple problem with the software or the language. Based on the amount of time available for formal instruction, this may or may not be feasible in a particular course. Providing the examples with solutions mentioned above, may be as close to this request as possible, but time permitting, adding in class exercises to the formal instruction can be beneficial.

Many of the survey comments directly lead to minor improvements in formal and/or informal instruction that can significantly enhance student learning. One student reported in the survey that they did not know how to re-open a workspace within Active-HDL; this can be difficult if you use the "Workspace/Design Explorer." A simple demonstration of using File..Open and browsing to find the *.aws file for the desired workspace could take 30 seconds at the beginning of a class period; this could also be incorporated into the formal instruction (with appropriate emphasis added knowing that this has caused students to struggle in the past). While this may seem minor, a student who wastes fifteen minutes (or much more time) is not spending that time on learning and building confidence with the CAD/E tool. As instructors, we want to set the conditions for student success. The use of a survey can help instructors focus on those simple tasks associated with the CAD/E tools that cause students to struggle.

Conclusion

There are three themes associated with our efforts to find a better way to teach CAD/E tools. First, when possible, streamline the number of CAD/E tools taught over the course of an undergraduate EE program. Second, ensure that when formal instruction is conducted, it is conducted in an environment that will enhance student learning of the CAD/E tools. Finally, the incorporation of a mid-course survey as an integral part of an assessment process can significantly improve the student's ability to learn the CAD/E tools by focusing the time constrained instruction on those areas in which students need the most help.

Future work in this area would be to take the current results and apply them beyond the digital logic thread of an electrical engineering program.

Acknowledgements

We would like to thank Dr. Mark Evans, Director, USMA Center for Teaching Excellence, for his focus on the use of assessment tools in the Master Teacher Program which guided the initial direction of our research. We thank Dr. Greg Conti for guiding the primary author in the right direction for developing the format of the paper and for providing the perspective on how to develop research that adds to the current academic base of knowledge. Finally, we would like to thank Dr. Robert Sadowski for identifying the poor teaching environment in the introduction to computer architecture course which lead to the development of the enhanced EE teaching computer classroom concurrently with this research project; the enhanced classroom is having a positive impact on student education.

The views expressed are those of the authors and do not reflect the official policy or position of the US Military Academy, the US Department of the Army, the US Department of Defense, or the United States government.

References

[1] United States Military Academy Center for Teaching Excellence Master Teacher Certificate Program, (at http://www.dean.usma.edu/centers/cte/Master.cfm)

[2] Al-Imanmy, Samer, "On the Development of a Programming Teaching Tool: The Effect of Teaching by Templates on the Learning Process", Journal *of Information Technology Education*, Volume 5, 2006.

[3] Wu, Angus, "Interactive Learning Toolbox for Logic Synthesis with VHDL", IEEE, 1997.

[4] Cadence Design System, (at http://www.cadence.com/products/kits/index.aspx?lid=cadence_kits)

[5] United States Military Academy. *Educating Future Army Officers for a Changing World*, 2003. (at http://www.dean.usma.edu/support/aad/efaocw.pdf)