AC 2012-3650: PROFESSIONAL DEVELOPMENT OPPORTUNITY FOR ELECTRICAL ENGINEERING TECHNOLOGY EDUCATORS IN VHDL AND FPGA DESIGN

Dr. Nasser Alaraje, Michigan Technological University

Nasser Alaraje is currently the Electrical Engineering Technology program Chair, as well as a faculty member at Michigan Technological University. He taught and developed courses in the computer engineering technology area at the University of Cincinnati and Michigan Technological University. Alaraje's research interests focuses on processor architecture, system-on-chip design methodology, field-programmable logic array (FPGA) architecture and design methodology, engineering technology education, and hardware description language modeling. Alaraje is a Fulbright scholar; he is a member of the American Society for Engineering Education (ASEE), a member of the ASEE Electrical and Computer Engineering Division, a member of the ASEE Engineering Technology Division, a member of the Institute of Electrical & Electronic Engineers (IEEE), and a member of the Electrical and Computer Engineering Technology Department Heads Association (ECETDHA).

Prof. Aleksandr Sergeyev, Michigan Technological University

Aleksandr Sergeyev is currently an Assistant Professor in the Electrical Engineering Technology program in the School of Technology at Michigan Technological University. Sergeyev is earned his bachelor's degree in electrical engineering in Moscow University of Electronics and Automation in 1995. He obtained the master's degree in physics from Michigan Technological University in 2004 and the Ph.D. degree in electrical engineering from Michigan Technological University in 2007. Sergeyev's research interests include high energy lasers propagation through the turbulent atmosphere, developing advanced control algorithms for wavefront sensing and mitigating effects of the turbulent atmosphere, digital in-line holography, digital signal processing, and laser spectroscopy. He is also involved in developing new eye-tracking experimental techniques for extracting 3-D shape of the object from the movement of human eyes. Sergeyev is a member of American Society for Engineering Education (ASEE) and is actively involved in promoting engineering education.

Professional Development Opportunity for Electrical Engineering Technology Educators in VHDL and FPGA design

Abstract

Hardware Description Language and Field Programmable Gate Array (FPGA) have revolutionized the way Digital Logic Design is taught and implemented. Traditional ways of teaching logic design using discrete components (TTL: Transistor-Transistor Logic and CMOS: Complementary Metal Oxide Semiconductors) have been replaced by Programmable Logic Devices (CPLD: Complex Programmable Logic Devices and FPGA). Today, a more standard development process is widely used in industry. The process uses Hardware Description Languages as a design entry to describe the digital systems. The two most widely used Hardware Description Languages in industry are VHDL (Very High Speed Integrated Circuit Hardware Description Language) and Verilog (Verifying Logic). Although most traditional electrical and computer engineering programs have updated their curriculum to include topics in hardware description language and programmable logic design (FPGA/CPLD), two-year and four-year electrical engineering technology programs have fallen behind and are moving slowly in updating their curriculum. This paper will discuss the offering of two-day VHDL and FPGA design workshop for electrical engineering technology faculty as part of National Science Foundation- Advanced Technological Education grant.

I. Introduction

Programmable Logic Devices in general and FPGA-based re-programmable logic design became more attractive as a design medium during the last decade, and as a result, industrial use of FPGA in digital logic design is increasing rapidly. As would be expected following technology change in industry, the need for highly qualified logic designers with FPGA expertise is increasing at a fast rate. According to the United States Department of Labor, the job outlook is on the rise and will continue to expand for at least the short- to medium-term future [1]. To respond to the industry needs for FPGA design skills, universities are updating their curriculum with courses in hardware description languages and programmable logic design. Although most traditional electrical and computer engineering programs have updated their curriculum to include topics in hardware description language and programmable logic design (FPGA/CPLD), only 19.5 % of 4-year and 16.5 % of 2-year electrical and computer engineering technology programs at US academic institutions currently have a curriculum component in hardware description language and programmable logic design [3]. To effectively meet the next generation's workforce needs, the electrical and computer engineering technology curriculum must be current, relevant, and teach technology that is widely used in industry. To meet this goal, we propose a digital logic design curriculum development in the EET program in the School of Technology at Michigan Tech University. Faculty involved in developing and teaching the new curriculum must be well-informed of advances in technology currently used in industry. Likewise, industry wants to have qualified and well-educated employees coming out of academia who are ready to implement their knowledge on day one of their employment. As a result, while academia needs to be fully aware of the current state-of-the-art knowledge requirements, industry must be driving the curriculum development. Therefore, in this curriculum development, a strong link between academia and industry must be established. This partnership is a “two-way street” and advantageous for both parties. The Electrical Engineering Technology (EET) program is collaborating with Altera University program. Those faculty members leading the project attended a set of training workshops developed by Altera’s university program. These workshops are targeted toward professional individuals and college faculty seeking knowledge and expertise in programmable logic design. Faculty members having the opportunity to attend these workshops will gain the knowledge and expertise to teach both VHDL digital Design and Programmable Logic (FPGA) design courses. The exposure to industry-taught courses will help the faculty members to impact the learning experience of undergraduate students by providing them with skills that are highly marketable and appreciated by industry. This industry-led faculty training have resulted in digital logic design curriculum development in the electrical engineering technology programs. This curriculum revision incorporates the addition of two new courses that added to the current course (Digital Electronics). As a result, the EET program introduced two new courses (Digital Design Using VHDL and Topics in Programmable Logic). Each of these courses is three credit hours (2 class, 3 lab). The new curriculum development will provide students with a hands-on educational experience well-respected by industry. The project Principal Investigators will share their knowledge and expertise in digital logic curriculum development by offering this professional development opportunity to interested faculty members at similar institutions as part of the dissemination plan.

II. Faculty Workshop

The goal of this workshop is to combine technical information from the vendor training with practical curriculum planning and strategies for developing courses like those developed at Michigan Tech University under this project. The participating faculty members learn introductory material on the impact of teaching engineering technology students relevant skills in hardware modeling and FPGA design. In subsequent sessions, the faculty members learn fundamental concepts of VHDL and gain knowledge on FPGA design environment using Altera Quatrus development software. Participants gain hands-on lab experience in modeling basic building blocks of digital systems and learn the FPGA design flow from HDL design entry and circuit simulation to verifying the correctness of the design. Participating faculty members tour the Re-configurable Computing Lab and learn the hardware and software necessary to establish a re-configurable lab at their respective institutions. Michigan Tech University faculty members assist participating faculty members in further development of curriculum through a post-workshop follow-up. Curriculum material developed at Michigan Tech is made available for use by participating faculty members both during and after the workshop.

The first summer faculty workshop was offered in September 2011, the project PIs conducted an intensive, two-day workshop on VHDL and FPGA design. There was an overwhelming positive response to the opportunity announced on the Electronics Theses and Dissertations (ETD) listserv, which forced The PI to close the registration after only two hours following the announcement. Representatives from seven institutions in six states (Indiana, Illinois, Kentucky, Pennsylvania, Virginia, and Georgia) engaged in the hands-on learning experience, working with both the software and the hardware. The workshop provided faculty members of community colleges and four-year electrical engineering technology programs with the opportunity to expand their expertise in VHDL and FPGA design. The participants will utilize these skills to develop new courses in digital logic design, using VHDL and FPGA, at their respective institutions. The workshop participants will:

- Be able to identify the importance of teaching engineering technology students relevant skills in hardware modeling and FPGA design
- Demonstrate the understanding of the fundamental concepts of hardware description languages and gain knowledge on programmable logic devices (PLD)
- Gain hands-on expertise on the hardware and software necessary to establish a re-configurable lab at their respective institutions
- Gain hands-on lab experience by practicing modeling basic building blocks of digital systems and learn the FPGA design flow
- Develop potential curricular resources to be used at their respective institutions

III. Faculty Workshop Curriculum Modules

Hands-on learning is infused into a sequence of instructional modules. The first module focuses on Quartus Software Development, the second module focuses on introduction to VHLD, and the third module focuses on advanced topics in VHDL. Each module has an associated laboratory exercise to enforce the learning experience of participants. The following is a description of each module, relevant topics that are covered and expected learning outcomes. The breakdown of the workshop into three modules allows participants to pick and choose components to match his/or her learning needs. All of the laboratory exercises are conducted using The Altera® Development and Education (DE2) board which provides an ideal vehicle for learning about digital logic, computer organization, and FPGAs. Featuring an Altera Cyclone® II FPGA, the DE2 board offers state-of-the-art technology suitable for our laboratory use [2]. Altera also provides the Quartus® II development software free to universities. Both DE2 FPGA evaluation boards and Quartus Development software have been received as a donation from Altera Corporation.

The Quartus II Software Design Series: This module of the workshop provides extensive training on how to use Quartus® II development software to develop an FPGA or CPLD. Participants are able to create a new project, enter in new or existing design files, and compile their design. Faculty learn how to plan and manage I/O assignments and apply timing analysis of design to achieve design goals using Quartus® II development software [2]. Additionally, Participants learn how to constrain & analyze a design for timing using the TimeQuest timing analyzer in the Quartus® II software. This includes understanding FPGA timing parameters, writing Synopsys Design Constraint (SDC) files, generating various timing reports in the TimeQuest timing analyzer, and applying this knowledge to an FPGA design. The workshop part objectives are to have class participants are able to:

- Make pre-project decisions to prepare for Quartus II design
- Create, manage & compile Quartus II projects
- Use Quartus II tools to view the results of compilation
- Plan & manage device I/O assignments using Pin Planner
- Use the basics of the TimeQuest timing tool
- Review compilation results in various Quartus II software reports and graphical viewers
- Understand the TimeQuest timing analyzer timing analysis design flow
- Apply basic and complex timing constraints to an FPGA design

- Analyze an FPGA design for timing using the TimeQuest timing analyzer
- Write and manipulate SDC files for analysis and controlling the Quartus II compilation

Introduction to VHDL: This module of the workshop provides introduction to the VHDL language and its use in programmable logic design. The emphasis is on the synthesis constructs of VHDL. Faculty members gain a basic understanding of VHDL. The course is laboratory intensive and includes a hands-on experiment to design, test, and simulate and synthesize a basic logic circuit as part of Quartus® II development software [2]. The workshop component objectives are to have class participants are able to:

- Understand simulation versus synthesis environments
- Build basic VHDL models using the VHDL design units (entity, architecture, configuration, package)
- Use behavioral modeling constructs and techniques to describe logic functionality
- Use structural modeling constructs and techniques to create hierarchical designs

Advanced VHDL: In this module of the workshop, faculty members learn how to write efficient coding techniques for VHDL synthesis, particularly for Altera® devices. The faculty member gain experience writing behavioral and structural code and learn how to effectively code common logic functions including registered, memory, and arithmetic functions. As part of the course topics, Participants learn how to write testbenches to verify the functionality of the design [2]. The workshop component objectives are to have class participants are able to:

- Develop coding styles for efficient synthesis when:
- Targeting device features
- Inferring logic functions
- Using arithmetic operators
- Writing state machines
- Use Quartus II software RTL Viewer to verify correct synthesis results
- Incorporate Altera structural blocks in VHDL designs
- Write simple testbenches for verification
- Create parameterized designs

Hands-on Laboratories Exercises:
A set of five laboratory exercises were developed. These labs consisted of an introduction to the Altera Quartus II software used to code, compile, and program the Altera DE2 FPGA Development Boards, an introduction to the basics of the VHDL language, an advanced VHDL primer, a lab focusing on testbenching a design, and finally a complete project of creating a reaction timer using VHDL and the FPGA development board. Following is a description of each laboratory exercise, all labs are available on the project website: http://www.tech.mtu.edu/NSFATE/.

Lab 1: Introduction to Quartus II: This lab is designed to familiarize the participant with using many of the common aspects of the Quartus II software through a complete design phase. Participants will learn to create a new project, create a new VHDL file, use the MegaWizard Plug-In Manager, compile the design, plan and manage I/O assignments, apply timing analysis using the TimeQuest Timing Analyzer, write Synopsys Design Contraint (SDC) files, and program a design onto the Altera DE2 FPGA Development Board. Specifically, in this laboratory exercise the participant creates a new project, names it, and learns all of the appropriate project settings for using the Altera DE2 FPGA Development Board. Then they create a new VHDL file and paste the VHDL code provided to them to create the top level design entity for this circuit. Next, the MegaWizard Plug-In Manager is used to create a four-bit three by one multiplexer component. The appropriate pins are then assigned to the inputs and outputs of the design. Basic use of the TimeQuest Timing Analyzer is then shown including: creating a timing netlist, setting timing constraints, adding SDC files, editing SDC files, and running the timing analyzer. Finally, the circuit created is programmed to the Altera DE2 FPGA Development Board and the participant can work the switches and see the multiplexer in action.

Lab 2: Introduction to VHDL: This laboratory exercise is used as an introduction to the VHDL language including entity declaration, process statements, behavioral coding, structural coding, port mapping, component declaration, and signals, among others. There is a lot to learning any programming language, but the authors found that it is best to have good basic examples to fall back on that are clear and concise. The goal of creating this laboratory exercise is that the participant should be exposed to a wide variety of programming styles and techniques in order to form a solid foundation to build upon later. If the participant were simply asked to write their own code in the exercise based upon what they learned from the lecture portion of the workshop they would have no way of knowing if the code they wrote was well written or even standard practice. Instead, this laboratory exercise manual acts as a compendium of basic VHDL programming styles and techniques that can be referenced at a later date in order to employ the correct usage and syntax. For this laboratory exercise the participant creates a ripple-carry four-bit full adder/subtractor. This circuit is made up of a number of smaller design elements including: exclusive OR logic gates, two-input AND gates, and three-input OR gates. These are arranged to create four one-bit full adders with the ability to add the two's compliment of one of the numbers (subtraction) if desired. The participant is also instructed on how to simulate inputs and outputs to the circuit using a Vector Waveform File (vwf) including using both Functional and Timing simulator modes. In the end, the project is programmed to the Altera DE2 FPGA Development Board and the participant can interact with their ripple-carry four-bit full adder/subtractor physically.

Lab 3: Advanced VHDL: This laboratory exercise was created after reviewing the Altera laboratory manual titled Advanced VHDL Design Techniques. Using this material as a reference the topics were selected to be suited to an advanced VHDL laboratory exercise. These topics include operator balancing, resource sharing, preventing unwanted latches, pipelining, and state machine encoding schemes. These advanced VHDL techniques are used to improve the speed and efficiency of the code and its implementation on the hardware. In this laboratory exercise the participant creates two separate designs and uses them to demonstrate the varying advanced VHDL techniques. The first design is a multiplier that demonstrates operator balancing and resource sharing. First, the code is compiled and analyzed as-is. The maximum specified clock speed is recorded for the design using the Timing Analyzer Summary, and use of the Register Transfer Level (RTL) viewer in introduced to observe the number of multipliers used in the design. Resource sharing is introduced by using parenthesis to group mathematical operations. Once recompiled, the participant can see the effect that this technique has on increasing the maximum available clock speed and by reducing the number of components necessary. Next, the code is modified to make use of pipelining, by using temporary registers, which increases the maximum available clock speed again. The second design used in this laboratory exercise is then created and compiled. This code first demonstrates creating unintentional latches by not properly setting up the state machine. The State Machine Viewer is then introduced as a tool used to visually identify state machine operation or problems. Then, different state machine encoding schemes are used to illustrate how they affect the maximum clock speed of the design. The various encoding schemes used for demonstration are One-Hot Encoding, Minimum Bits, Gray Encoding, Johnson Encoding, and Sequential Encoding.

Lab 4: Testbenching: This laboratory exercise uses Mentor Graphics ModelSim software [4] integrated with Altera Quartus Software. In this exercise a circuit design is loaded, a testbench code is written, and signal waveform graphs are generated. The circuit design used in this exercise is the full adder/subtractor circuit from Lab 2: Introduction to VHDL. The participant then copy and pastes the testbench code provided into the file they created from the beginning steps of this exercise. The circuits, and its inputs, are then simulated and the participant is instructed on how to view the resulting waveforms efficiently. The testbench file is then edited to test other input conditions and the results are viewed after simulation.

Lab 5: Capstone Project: This lab acts as a capstone to the entire VHDL and FPGA Design Workshop. This integrating experience develops participant competencies in applying VHDL and FPGA technical skills in solving a design problem. It covers various topics previously discussed and adds some even more advanced techniques and algorithms, it gives a good real-world application of what can be accomplished with FPGAs.

IV. Assessment

Assessment is a vital part of any curriculum reform project and helps provide useful information for workshop enhancements and determining if the workshop has met its objectives. Formative evaluation occurred during the workshop delivery and will be used to inform adjustments for subsequent workshop offerings. Embedded assessment is used to measure each workshop objective and determine whether goals are met. Assessment of the effectiveness of the faculty workshops training sessions offered is conducted anonymously using pre- and post-surveys. Assessment data collected and analyzed from the workshop will result in continuous improvement actions to be implemented in year two faculty workshop. We use a pre-test/post-test design and pre-survey/post-survey employing both direct and indirect measures of student learning. The indirect assessment instrument also included questions regarding participants' satisfaction while direct assessment instrument include a set small design problems and multiple choices problems.

Direct Measures of Student Learning:
Participants were given the same instruments for the pre-test and for the post-test. The average on the pretest was 40% correct answers. On the post-test, following the two days of instruction, the average on the test rose to 72% correct answers. Since only seven students were involved in this summer workshop, generalizing these results is somewhat problematic, but it's clear that these participants made substantial progress towards mastering course concepts during the two-day workshop.

Indirect Measures of Student Learning:
Participants were given the indirect measure instrument prior to the beginning of instruction. The only relevant pretest portion (Mastery of Course Outcomes) yielded the following scores on a five point scale with five indicating "Complete Mastery" and one indicating "No Mastery". It's clear from these results that participants made substantial progress towards achieving course outcomes as a result of the instruction. This conclusion is supported by both direct and indirect measures. It's also clear that these participants valued the quality of the instruction.

Quality of Instruction (5= Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree) **Measurable Outcomes**	**Post-Test Overall Rate**
The instruction was clearly presented	4.86
Any questions I asked were properly answered	4.71
The materials provided helped me to learn	4.86
The pace of the course was appropriate for the amount of material to be learned	4.57

Table 1: Quality of Instruction Participants' feedback assessment results

Introduction to VHDL (5= Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree) **Measurable Outcomes**	**Pre-Test Overall Rate**	**Post-Test Overall Rate**
Ability to implement basic constructs of VHDL	2.57	4.71
Ability to implement modeling structures of VHDL	1.86	4.57
Ability to use software tools to check the code for correctness and to correct errors	2.57	4.57

Table 2: Introduction to VHDL Participants' feedback assessment results

Advanced VHDL Design Technique Learning Objectives (5= Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree **Measurable Outcomes**	**Post-Test Overall Rate**
Write synthesizable VHDL	4.43
Control state machine implementation	4.57
Optimize a system design, using operator balancing, resource sharing, and pipelining	4.29
Create a test bench and run a simulation	4.71

Table 3: Advanced VHDL Participants' feedback assessment results

Quartus II Software Design Series: Foundation Learning Objectives (5= Strongly Agree, 4=Agree, 3=Neutral, 2=Disagree, 1=Strongly Disagree **Measurable Outcomes**	**Post-Test Overall Rate**
Create a new Quartus II project	4.86
Create design components using MegaWizard manager	4.71
Compile a design and view results	4.86
Use settings and assignments to control results	4.71
Make pin assignments and evaluate	4.71
Use the TimeQuest timing analyzer	4.71

Table 4: Quartus II Software Design Participants' feedback assessment results

V. Conclusion

This paper discussed the offering of two-day VHDL and FPGA design workshop for electrical engineering technology faculty as part of National Science Foundation- Advanced Technological Education grant. Curriculum resources and workshop materials are also be made available to faculty in other electrical and computer engineering technology programs. The educational materials are shared directly with participating faculty who attend the workshops and made available electronically through a project web site. This Professional Development activity provides both two-year and four-year electrical engineering technology faculty with the pedagogical and subject matter knowledge, digital teaching tools, and teaching strategies that will attract and effectively prepare students for STEM careers in reconfigurable electronics and other advanced electronics fields. For the United States to remain competitive in electronics technology, universities and community colleges need to continually update programs and facility resources, and provide ongoing faculty development to include the latest information about reconfigurable systems. There was an overwhelming positive response to the opportunity announced on the Engineering Technology Division (ETD) listserv, which forced The Principal Investigator to close the registration after only two hours following the announcement. Assessment results showed that students made substantial progress towards achieving course outcomes as a result of the instruction using both direct and indirect measures. Additionally, Workshop participants valued the quality of the instruction grading the quality of instruction at 4.86 on a 5 points scale.

Bibliography

[1] http://www.bls.gov/oco/home.htm.

[2] http://www.altera.com

[3] R. Furtner and N. Widmer, “Technology Education and the new frontier of digital electronics,” *ASEE Annual Conference & Exposition (ASEE 2006)*, June 2006

[4] http://www.model.com

Acknowledgments

The authors gratefully acknowledge the support for this project under the National Science Foundation – Advanced Technological Education Award No. DUE-1003389.