

PROFESSIONAL DEVELOPMENT OPPORTUNITIES FOR ELECTRICAL ENGINEERING TECHNOLOGY EDUCATORS IN VHDL AND FPGA DESIGN

Nasser Alaraje, Aleksandr Sergejev, William Kennedy and Carl Seidel, Michigan Technological University

Abstract

Hardware Description Languages (HDLs) and Field Programmable Gate Arrays (FPGAs) 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using discrete components (TTL: Transistor-Transistor Logic and CMOS: Complementary Metal Oxide Semiconductors) have been replaced by Programmable Logic Devices (PLDs), which include Complex Programmable Logic Devices (CPLDs and FPGAs). Today, a more standard development process is widely used in industry. The process uses HDLs as a design entry tool to describe digital systems. The two most widely used HDLs 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 curricula to include topics in HDL 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 curricula. This paper presents information on two-day VHDL and FPGA workshops designed to educate faculty in electrical engineering technology as part of a National Science Foundation Advanced Technological Education grant.

Introduction

Programmable Logic Devices in general and FPGA-based re-programmable logic designs became more attractive as a design medium during the last decade and, as a result, industrial use of FPGAs in digital logic design is increasing rapidly. Due to this 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 industry's need for FPGA design skills, universities are updating their curricula with courses in HDL and programmable logic design. Although most traditional electrical and computer engineering programs have updated their curricula to include these topics, only 19.5% of 4-year and 16.5% of 2-year electrical and computer engineering technology programs at U.S. academic insti-

tutions currently have a curriculum component in HDL and programmable logic design [2-5]. Additionally, experience with traditional high-level programming languages such as C will be helpful to students wanting to learn HDLs and work with PLDs. To effectively meet the workforce needs of the next generation, electrical and computer engineering technology curricula must be current, relevant, and teach technology that is widely used in industry. To meet this goal, the authors pursued the development of a digital logic design curriculum for 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 employment. As a result, while academia needs to be fully aware of the current state-of-the-art of knowledge requirements, industry must be driving the curriculum development. Therefore, in this curriculum development process, a strong link between academia and industry must be established. This partnership is a "two-way street" with advantages for both parties. The Electrical Engineering Technology (EET) program is collaborating with Altera's [6] University Program and those faculty members leading the project attended a set of training workshops developed by Altera. 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, gain the knowledge and expertise needed to teach both VHDL digital Design and Programmable Logic (FPGA) design courses. Exposure to industry-taught courses can help 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 program has resulted in digital logic design curriculum development in the electrical engineering technology programs. This curriculum revision led to the addition of two new courses beyond the current course in Digital Electronics. As a result, the EET program introduced two new three-credit-hour courses: Digital Design Using VHDL and Topics in Programmable Logic. The new curriculum will provide students with a hands-on educational experience well respected by industry. The principal investigators of the project shared 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.

Research Background

Historically, EET programs have included a traditional logic design course that covers topics in combinational logic and sequential logic circuits. The course is based on discrete components such as TTL and CMOS. And although these topics represent fundamental concepts in logic design and optimization theory, they are far from the most current industry practice in applied logic design. Topics that have been traditionally taught in logic design courses are outdated and less important to current employers. The time spent teaching Boolean algebra and how to minimize Boolean expressions using Boolean algebra or Karnaugh Maps (K-Maps) could be better spent teaching current and industry-relevant topics in logic design. This line of thought applies to both the design of combinational circuits and the design of sequential circuits [2]. For the curriculum to adequately meet the current needs of industry, EET programs must teach digital logic using VHDL and FPGAs [7]. Consequently, students will be equipped with design skills that are current, relevant, and widely used in industry.

Recent research supports the idea that the proposed curricular shift is industry-relevant. In one study, Furtner and Widmer [2] conducted an employer survey to rank currently taught logic design concepts at Purdue University. The survey included questions about many topics that are heavily explained in logic design courses such as Boolean algebra, design simplifications using K-Maps or Quine McClusky, and design implementation using discrete gates. Each was given a low priority from the employer perspective. On the other hand, topics that cover designing with a hardware description language such as VHDL or Verifying Logic (Verilog) received high-priority rankings from employers [2]. There is definitely a great need for community colleges and universities to continually update their programs and resources, and provide ongoing faculty development to include the latest information about digital logic design. Additional studies [8] identify faculty development as a major concern for academic institutions and suggest faculty development through industry consulting and training. Similarly, in another study [7], researchers built an industry-academic collaborative partnership to train almost 200 faculty members with innovative teaching methodologies, which had a great impact on student learning. This industry-academia partnership helped faculty members become more active partners, mentors, and facilitators of the student learning process. Universities [9] and Community Colleges [10] need to develop such pro-

grams allowing their faculty to advance professionally. The approach described here for faculty development targets faculty members who acknowledge the need to update the digital logic design curricula at their institutions, but who do not have the time to pursue it on their own. The project's approach is to combine digital logic design best practice in industry with practical curricular planning. Participating faculty members are provided with the latest digital logic expertise using a series of educational resources and professional development sessions aimed at "educating the educator".

A recent survey was sent to the employers who hire electrical engineering technology graduates of both two-year and four-year institutions in order to assess how well this educational initiative aligns with their current and future human resource needs. This survey was designed to judge the necessity of providing the type of training identified as important. Forty organizations responded to this survey. Overwhelmingly, the survey respondents identified the ability of technicians and technologists to be able to work with FPGAs as a critical skill. Results from the assessment survey further showed that almost 80% of the respondents view knowledge of VHDL and FPGAs as a critical skill for making a technician more employable and marketable. As a result, there is a great need for training community college and university technology professors in VHDL and FPGA design in order to provide them with the latest digital logic expertise. Through the training program described here, the participating faculty members are not only getting trained on VHDL and FPGA design, but are also receiving access to curricular components that have already been developed at Michigan Tech University, learning best practices for using innovative strategies, and learning about those tools most effective for teaching digital systems design using VHDL and FPGA technologies.

Faculty Workshop Objectives

The goal of this workshop is to combine technical information from the vendor with training on practical curricular 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 environments using Altera's Quartus development software. Participants gain hands-on lab experience in modeling basic building blocks of digital systems and learning FPGA design flow from HDL design entry and circuit simulation to verifying the correctness of the design. Participat-

ing 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 faculty members assist participating faculty members in further development of their own curricula through a post-workshop follow-up. Curricular materials developed at Michigan Tech are 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 Engineering Technology Division (ETD) listserv, which forced the PI to close the registration after only two hours following the announcement. All ten seats were taken and a waiting list of 15 more participants was created. Representatives from seven institutions from 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 their new skills gained through the workshop to develop new courses in digital logic design, using VHDL and FPGAs, at their respective institutions. The workshop participants learned how to:

- identify the importance of teaching engineering technology students relevant skills in hardware modeling and FPGA design;
- demonstrate an understanding of the fundamental concepts of hardware description languages and gain knowledge on programmable logic devices (PLDs);
- gain hands-on expertise with 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 FPGA design flow; and,
- develop potential curricular resources to be used at their respective institutions.

Faculty Workshop Curriculum Modules

Hands-on learning was infused into a sequence of instructional modules, as shown in Figure [1]. The first module focused on Quartus software development; the second module focused on an introduction to VHDL; and the third module focused on advanced topics in VHDL. Each module had 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 allowed participants to pick and choose components to match his or her learning needs. All of the laboratory exercises were 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 laboratory use [6]. Altera also provides the Quartus® II development software free to universities [6]. Both DE2 FPGA evaluation boards and Quartus Development software were received as a donation from Altera Corporation.

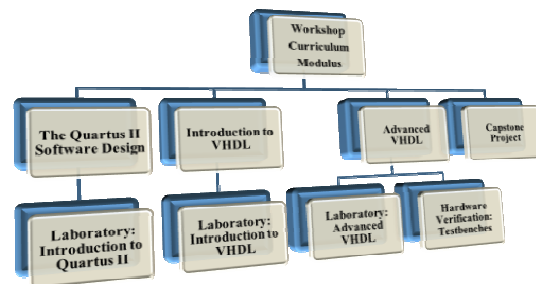


Figure 1. Workshop Curriculum Modules

The Quartus II Software Design Series

This module of the workshop provided extensive training on how to use Quartus® II development software to develop an FPGA or CPLD. Faculty members were able to create a new project, enter in new or existing design files, and compile their designs. Faculty learned how to plan and manage I/O assignments and apply timing analyses of their designs in order to achieve design goals using Quartus® II development software [6]. Additionally, faculty members learned how to constrain and analyze a design for timing using the TimeQuest timing analyzer in the Quartus® II software. This included 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 component objectives were to have class participants be able to:

- make pre-project decisions to prepare for Quartus II design;
- create, manage and compile Quartus II projects;
- use Quartus II tools to view the results of compilation;
- plan and 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;
- use Quartus II software RTL Viewer to verify correct synthesis results;
- incorporate Altera structural blocks in VHDL designs;
- write simple testbenches for verification; and,
- create parameterized designs.

Introduction to VHDL

This module of the workshop provided an introduction to VHDL language and its use in programmable logic design. The emphasis was on the synthesis constructs of VHDL. Faculty members gained a basic understanding of VHDL. The course was laboratory intensive and included a hands-on experiment to design, test, and simulate and synthesize a basic logic circuit as part of Quartus® II development software [4]. The workshop component objectives were to have class participants be 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; and,
- use structural modeling constructs and techniques to create hierarchical designs.

Advanced VHDL

In this module of the workshop, faculty members learned how to write efficient coding techniques for VHDL synthesis, particularly for Altera® devices. The faculty member gained experience writing behavioral and structural code and learned how to effectively code common logic functions including registered, memory, and arithmetic functions. As part of the course topics, faculty members learned how to write testbenches to verify the functionality of the design [6]. The workshop component objectives were to have class participants be able to develop coding styles for efficient synthesis when:

- targeting device features;
- inferring logic functions;
- using arithmetic operators; or,
- writing state machines.

Hands-on Laboratories Exercises

A set of five laboratory exercises was 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 to create a reaction timer using VHDL and the FPGA development board. Following is a description of each laboratory exercise.

Lab 1: Introduction to Quartus II

This lab was designed to familiarize the participants with the use of many of the common aspects of the Quartus II software through a complete design phase. Participants 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 a timing analysis using the TimeQuest Timing Analyzer, write Synopsys Design Constraint (SDC) files, and program a design onto the Altera DE2 FPGA Development Board. Specifically, in this laboratory exercise the participants create a new project, name it, and learn 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 in order 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 participants 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 the best teaching approach for VHDL is through

clear and concise examples. The goal of creating this laboratory exercise was 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 participants 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 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 complement 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 participants can physically interact with their ripple-carry four-bit full adder/subtractor.

Lab 3: Advanced VHDL

This laboratory exercise was created after reviewing the Altera laboratory manual titled *Advanced VHDL Design Techniques*. Using Altera's materials as a reference, the topics were selected to be suited to an advanced VHDL laboratory exercise. The 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 participants create two separate designs and use 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 is 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 de-

sign 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 [11] 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 participants then copy and paste the testbench code provided into the file they created from the beginning steps of this exercise. The circuit and its inputs are then simulated and the participants are 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 even more advanced techniques and algorithms. It gives a good real-world application of what can be accomplished with FPGAs.

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 workshop delivery and was used to make adjustments for subsequent workshop offerings. Embedded assessment was used to measure each workshop objective and determine whether goals were met. Assessment of the effectiveness of the faculty workshop training sessions offered was conducted anonymously using pre- and post-surveys. Assessment data collected and analyzed from the workshop will be used for continuous improvement actions to be implemented in year-

two faculty workshops. The authors used 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 about participants' satisfaction, while the direct assessment instrument included a set of small design problems and multiple-choice 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. It is 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 pre-test portion (Mastery of Course Outcomes) yielded the following scores on a five-point scale with 5 indicating "Complete Mastery" and 1 indicating "No Mastery".

Table 1. Quality of Instruction Participants' Feedback Assessment Results

Quality of Instruction (5= Strongly Agree, 1=Strongly Disagree) Measurable Outcomes	Post-Test Overall Rate
The instruction was clearly presented	4.86
Any questions I asked were	4.71
The materials provided helped me to learn	4.86
The pace of the course was appropriate for	4.57

Table 2. Introduction to VHDL Participants' Feedback Assessment Results

Introduction to VHDL (5= Strongly Agree, 1=Strongly Disagree) Measurable Outcomes	Pre-Test Overall Rate	Post-Test Overall Rate
Ability to implement basic	2.57	4.71
Ability to implement	1.86	4.57
Ability to use software tools to	2.57	4.57

It is 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 is also clear that these participants valued the quality of the instruction.

Table 3. Advanced VHDL Participants' Feedback Assessment Results

Advanced VHDL Design Technique Learning Objectives (5= Strongly Agree, 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	4.29
Create a test bench and run a simulation	4.71

Table 4. Quartus II Software Design Participants' Feedback Assessment Results

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

Conclusion

In this paper, the authors present the findings related to the offering of a two-day VHDL and FPGA design workshop for electrical engineering technology faculty as part of National Science Foundation Advanced Technological Education grant. Curricular resources and workshop materials were made available to faculty in other electrical and computer engineering technology programs. The educational materials were shared directly with participating faculty who attended the workshops and made available electronically through a project website. This professional development activity provided both two-year and four-year electrical engineering technology faculty with the pedagogical and subject-matter knowledge, digital teaching tools, and teach-

ing strategies to 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 study's principal investigator to close the registration after only two hours following the announcement, since all seats were taken. Assessment results showed that participants 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-point scale.

Acknowledgments

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

References

- [1] Occupational Outlook Handbook. (2010-2011). Retrieved January 20, 2012, from <http://www.bls.gov/oco/home.htm>
- [2] Furtner, R. & Widmer, N. (2006). Technology Education and the new frontier of digital electronics. *ASEE Annual Conference & Exposition*.
- [3] Nie, H., & Pecan, R. (2007). The Innovative Effects of HDL and FPGA on Digital Hardware Design Education in EET Programs. *ASEE Annual Conference & Exposition*.
- [4] Frenzel, L. E. (2006). Are We Teaching the Right Subjects in AAS Degree Electronics Technology Programs? *The Technology Interface Journal*, 6(1).
- [5] Nie, H., & Pecan, R. (2007). The Impacts of Hardware Description Languages and Large-Scale Programmable Logic Devices on Digital Hardware Design Education in Engineering Technology Programs. *The Technology Interface Journal*, 8(1).
- [6] Altera University Program. (n.d.). Retrieved January 10, 2012, from <http://www.altera.com>
- [7] Narayanan, S., Muniratanam, A., & Creese, R. (2011). Faculty Development in The New Globalized Era Through Industry-Academia Partnership. *ASEE Annual Conference & Exposition*.
- [8] Ocon, R. (2009). Promoting Faculty Development Using Industry Consulting Activities. *ASEE Annual Conference & Exposition*.
- [9] Schulz, N., & Schulz, K. (2004) Faculty Development—The Future of Engineering Education. *ASEE Annual Conference & Exposition*.
- [10] Brewer, E. V., & Dalessio, A. P. (2011). Survey of Professional Development Opportunities for Two-Year Institution and Community College Faculty. *The Technology Interface International Journal*, 11(2).
- [11] ModelSim. (n.d.). Retrieved January 10, 2012, from <http://www.model.com>

Biographies

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 Computer Engineering technology area at University of Cincinnati, and Michigan Technological University. Dr. 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. Dr. Alaraje is a Fulbright scholar; he is a member of American Society for Engineering Education (ASEE), a member of ASEE Electrical and Computer Engineering Division, a member of ASEE Engineering Technology Division, a member of Institute of Electrical & Electronic Engineers (IEEE), and a member of Electrical and Computer Engineering Technology Department Heads Association (ECETDHA). Dr. Alaraje may be reached at alaraje@mtu.edu

ALEKSANDR SERGEYEV obtained the Master degree in Physics from Michigan Technological University in 2004 and the PhD degree in Electrical Engineering from Michigan Technological University in 2007. Dr. Sergeyev is currently an Assistant Professor in the Electrical Engineering Technology program in the School of Technology at Michigan Technological University. Dr. Sergeyev 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 inline 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. Dr. Sergeyev is a member of American Society for Engineering Education (ASEE). Dr. Sergeyev may be reached at avsergue@mtu.edu

WILLIAM A. KENNEDY is the Director of the Center for Teaching, Learning, and Faculty Development at Michigan Technological University. He teaches and does research in the areas of college pedagogy, learning assessment, and the neurobiology of learning and memory. Dr. Kennedy may be reached at wkennedy@mtu.edu

CARL SEIDEL is currently finishing up his last semester for a Bachelors of Science degree in Electrical Engineering Technology at Michigan Tech University. He received an Associate's Degree in Automotive Service Technology from Northern Michigan University in 2005 and worked for 4 years as an automotive technician before coming to MTU. Throughout his undergraduate studies at MTU he has done research in a number of areas including: teaching digital logic courses with FPGA, human machine interface for an EMG sensor controlled pneumatic robotic exoskeleton, and high speed data acquisition for determining the amount of torque applied to a rotating shaft. Mr. Seidel may be reached at csseidel@mtu.edu