Laboratory for the Introductory Digital Course

Otto Fucik, Penn State University at Erie
Bodgan Wilamowski, University of Wyoming
Michael McKenna, University of Wyoming

Abstract

This lab entails practical studying and designing of simple logic gates, combinational logic circuits (adders, multiplexers, decoders, etc.), memory elements (flip-flops), and sequential logic circuits (state machines). The student will design, build, and use his/her own logic circuits. Logic designs will be done using computer-aided design (CAD) tools and implemented using complex programmable logic devices (CPLD). In this laboratory, digital logic circuits will be designed and implemented using the Foundation Series Software and the XC95108PC84 CPLD from Xilinx, Inc. In the later part of the course, students are introduced to VHDL hardware descriptive language.

I. Introduction

The digital course is a required course in all electrical and computer engineering curricula [1][2]. Often this course is also required for the computer science majors. Laboratory is always an essential part of this course [3][4]. The content of this course is constantly being modified. Several years ago the concept of the PLD was introduced [3][5]. HDL is now being introduced into the digital design courses [6][7]. The digital design course at the university of Wyoming is offered in the second semester of the sophomore year, for both electrical engineering and computer science majors. Electrical engineers have already completed the circuits course, while computer scientists have no hardware background, but they are already introduced to the concept of binary systems. In the past, several different design tools were used such as B2L, PLDesign, and VeriBest. This software is executed on the Microsoft Windows platform. The Unix base design tools were also considered but the cost was too high and these tools were too complex for the introductory digital course.

The primary goal of this course is to excite students about digital design. This course starts basically without any prerequisite (the formal prerequisite is the Calculus class, but no information form calculus is required). At the end of the class students are capable of designing relatively large digital circuits with a design complexity greater than 1000 gates and they are able to implement their own design into a silicon chip.

This is possible because of extensive usage of various CAD programs, which are part of the XILINX Foundation Series package. Students leave the course with the satisfaction that they have learned something.

II. Software and Laboratory Equipments

Choice of Software

The laboratory is now developed using Xilinx Foundation Series software and a specially developed printed circuit board to program and operate the XC95108PC84 CPLD chip. The Xilinx Foundation Series software has almost all features needed for this introductory digital laboratory, such as:

1. Schematic capture.
2. Logic simulation providing timing diagrams.
3. Possible entry with Hardware Descriptive Languages such as Verilog and VHDL.
4. Ability to implement digital design into CPLD and FPGA chips.

Two specially developed lab boards consist of a socket for the CPLD chip, two 7-segment displays, and one 10-bit LED bar on the first, and 12 DIP switches on the second board. With these boards, students can easily visualize a digital output from their design. The board can be plugged into a breadboard to allow the addition of any necessary input (e.g. clock generator) and output (e.g. stepper motor driver) circuitry.

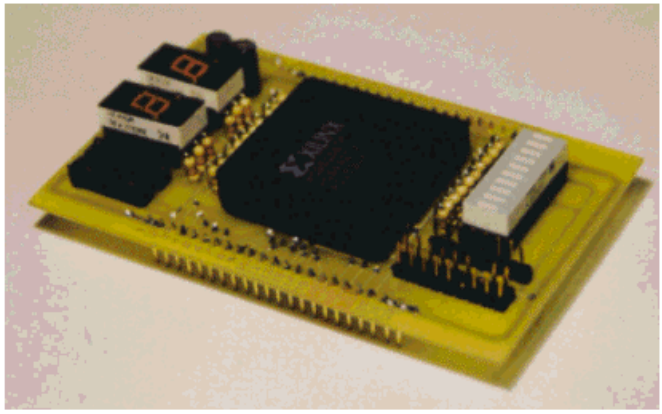


Fig. 1a Photo of the EVB2390 PC Board

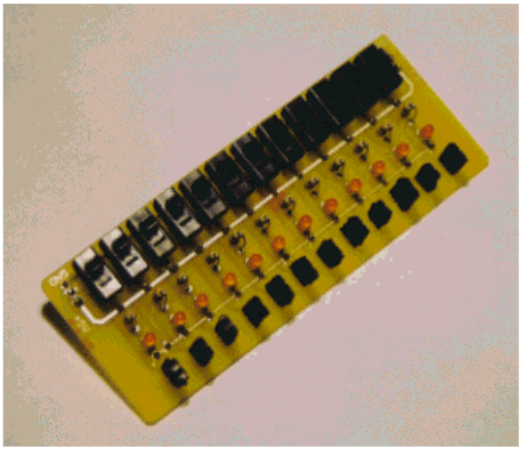


Fig. 1b Photo of the Switch PC Board

Dedicated PC Boards

Two special PC boards were designed for this lab. The photo of the first board is shown in Fig. 1a. The EVB2390 is en evaluation board dedicated for use in an introductory digital design course. This board uses a Complex Programmable Logic Device (CPLD), two 7-segment LED displays, one bar graph with 10 LED diodes, and a crystal chip oscillator running at a frequency 1MHz. The components are mounted on a single board. All of these components are available for the user to customize during design. There are also available the CPLD pins, which can be used as additional inputs/outputs. These pins are accessible by two 22-pin connectors. The EVB2390 can be placed into a breadboard system, which should provide power supply, input switches, output signal lights, clock sources, and any other circuitry necessary for the design. If the breadboard system does not include switches, the switch board can be used. The photo for this board is shown in Fig. 1b. The switch board can be placed into the power and ground strip in a breadboard to provide power to the switches. This board has 12 switches and there are two access points to place wires for each switch.

A template file, which defines CPLD pins, is available to the user. Thus, one can add a circuit into this template (see Fig. 2) to start with a design. The template provides all the available pins. Fig. 3 shows PC Board layout for both top and bottom layers of the EVB2390 board. Fig. 4 shows the PC Board layout for both top and bottom layers of the switch board.

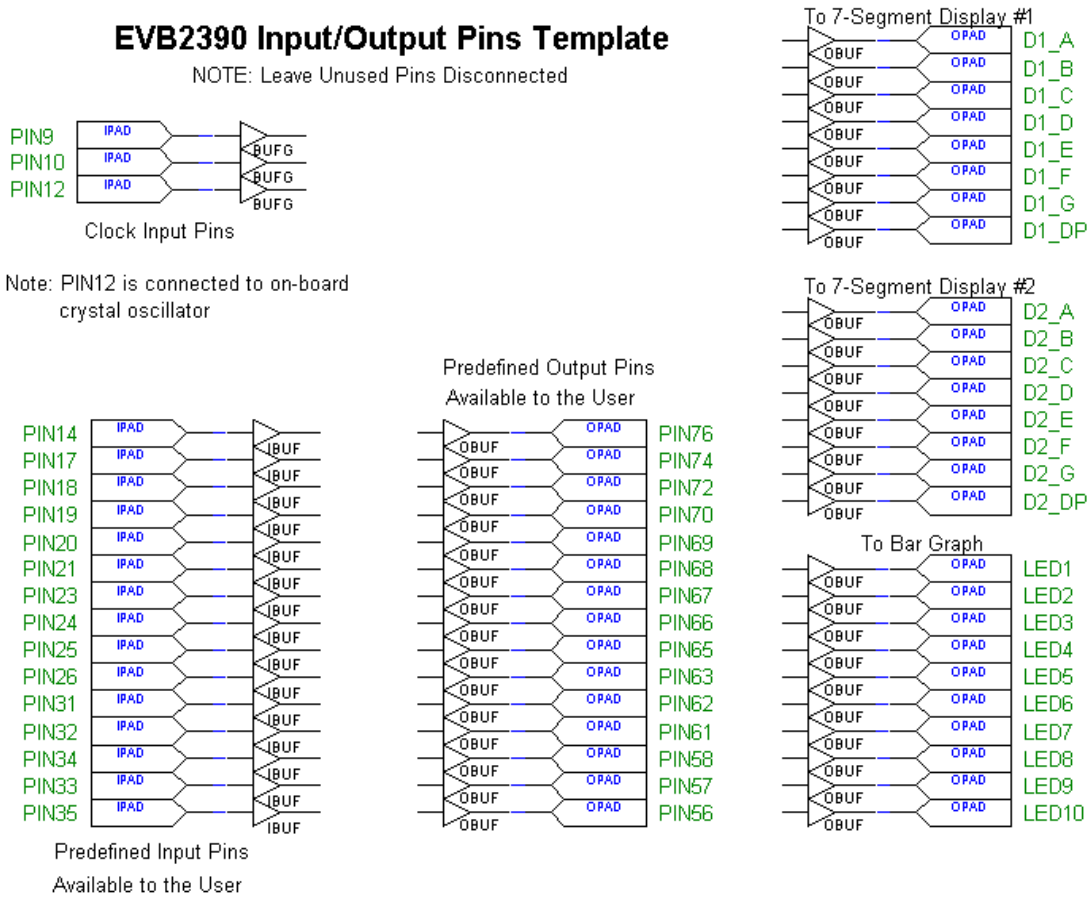
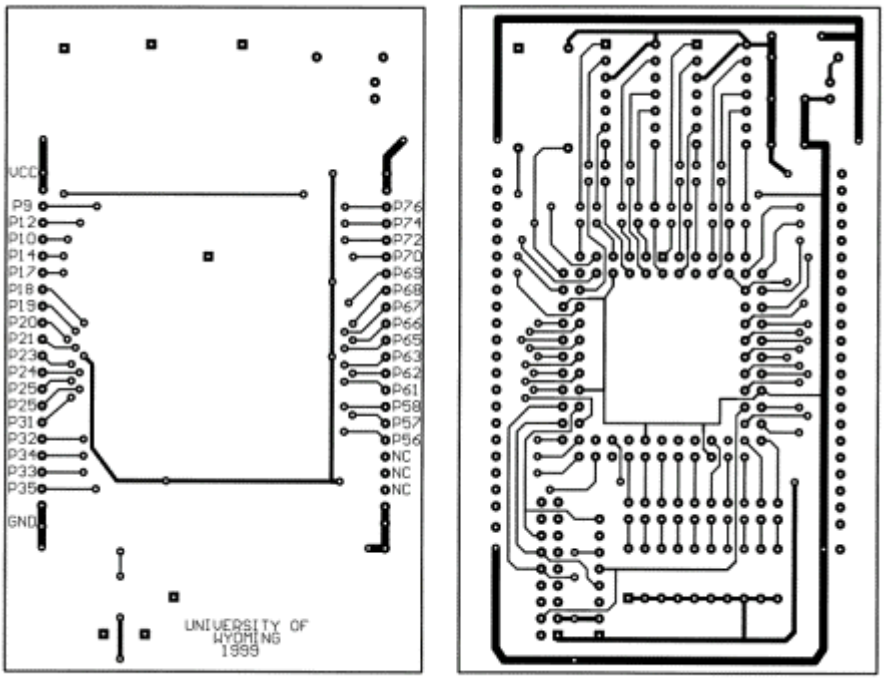
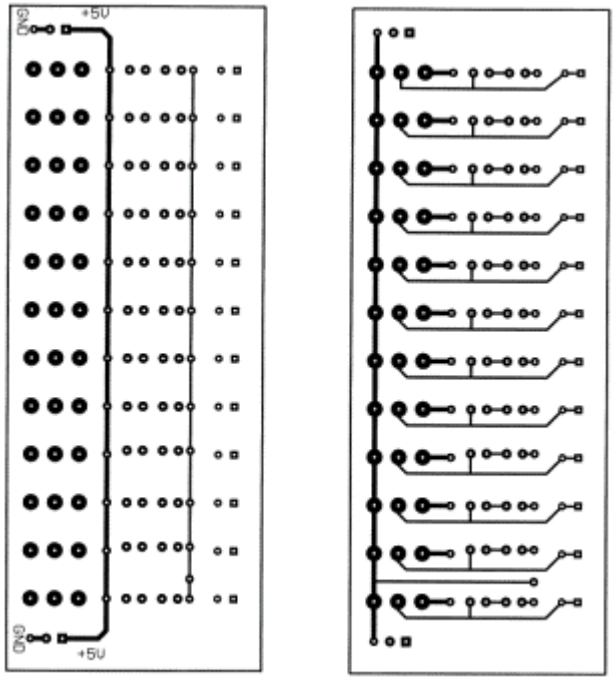


Fig. 2 EVB2390 Template Schematic Diagram



TOP BOTTOM

Fig. 3 PC Layout for EVB2390 Board



TOP BOTTOM

Fig. 4 PC Layout for Switch Board

III. Lab Philosophy and Goals

In the first couple of labs, the students are designing simple combinational logic circuits in the Xilinx Foundation Series software and simulating them to make sure they work correctly. They then wire those circuits using TTL and CMOS integrated circuits; to verify they will work using traditional hardware chips. In the next set of labs, the students design a BCD to 7-segment decoder, add-subtract circuit and a min-max circuit, which are all, combined to make an ALU (Arithmetic Logic Unit). These are word problems where students must develop a truth table and use minimization techniques learned in class. The following are specific teaching goals to be accomplished:

1. Implementing design on breadboards using SSI and MSI chips (This is done in the first part of the lab)
2. Computer aided simulation of their design
3. Learn how to use schematic capture
4. Learn how to implement word type problems into hardware
5. Learn about concept of hierarchical design
6. State machine design and macrotiming diagrams
7. Microtiming diagrams in simulation tools and on oscilloscopes.
8. Introduction to Hardware Descriptive Languages -HDL (how much easier is to do it with HDL)

IV. Laboratory Assignments

Labs begin using conventional design techniques such as schematic capture and as labs progress, this approach is being replaced with hardware descriptive language. The Xilinx Foundation Series package can interpret both Verilog and VHDL, and we have decided to use VHDL as it is a more powerful language. Our initial concern was that this language has too steep of a learning curve. It seems that students did not have any difficulty with VHDL primarily because they have received several other programming examples in VHDL and the task was to modify or combine existing code. We have chosen this approach since the main goal in this class was to demonstrate to the students the advantages of HDL, not to make them experts in this language. In our curriculum, we have another course solely devoted to HDL, which follows the basic digital course.

These designs are downloaded, using modern technology, into the Xilinx 95108 CPLD, to verify they work. Then they combine the min-max circuit, add-subtract circuit, AND's, OR's and X-OR's into one circuit and design decode logic to control it. The design is then placed that in the chip to create an ALU (Arithmetic Logic Unit). The same design is implemented using VHDL, and downloaded onto the chip to verify that you can synthesize code into a chip and it will actually create working hardware. Lastly, several labs pertain to sequential logic design and its implementation on the CPLD chip. Samples of such designs are synchronous counters and stepper motor controllers.

Laboratory is designed for 12 units:

Lab 1 **Binary Arithmetic**

This is first exposure of the Xilinx software to the students. The instructor supplies the actual design, but the students have to do experiments with how the carry-in affects the results. They are also learning how to simulate this circuit. This lab is relatively easy for Computer Science (COSC) students but quite challenging for Electrical Engineering (EE) students.

Lab 2. **Implementation of Boolean Equations**

Students are learning how to draw schematics using digital primitives such as AND, OR, and NOT. In this case, students must not only simulate the digital design, they have to build and wire the circuit on a breadboard. In contrast to the Lab 1 this laboratory is relatively easy for EE students and quite challenging for COSC students. In many cases for the COSC students this is the first experience in their life with an electronic laboratory, where they have to wire something and test it.

Lab 3. **Design of a Combinational Logic Circuit**

This is a "word problem" to be implemented in hardware. Again the students have to enter their design into the schematic capture and simulate it. Once they get a working schematic, the students wire the design to verify it works using actual hardware.

Lab 4. **Combinational Logic Reduction**

A decoder for the 7-segment display is a more complex logic design where logic reduction is also required. This lab requires significant pre-lab preparation to design the decoder using Karnaugh maps. The idea of hierarchy in designs is also introduced in this lab.

Lab 5. **Add/Subtract Implementation**

The students must design a one-bit full adder circuit. The student then take the full adder and using the idea of a hierarchy, create a four-bit add/subtract circuit. In this lab, the students are implementing their design into CPLD chip. This way they are gaining more confidence that they can create more complex designs.

Lab 6. **Add/Subtract Implementation using VHDL**

In the add/subtract lab the students are given the VHDL code of an adder and implement this code on the chip to verify the code performs the same as the schematic. This is first exposure of the concept of hardware descriptive language to the students. The students also realize how much more efficient VHDL is.

Lab 7. **Design Data Path Circuits using VHDL**

This lab provides the first chance the students get to create their own VHDL circuit. Instead of using the schematic capture approach, students enter their design using behavioral VHDL code for a MINMAX circuit. The VHDL is converted into a hierarchical macro and the remaining design is done with a traditional schematic approach.

Lab 8. **Arithmetic and Logic Unit Design**

In this lab, students are designing the core of a simple computer. Depending on the Op-code, different operations must be performed in their ALU. This lab allows them to understand the operation of a small part of a computer and after completing this long and difficult lab they have the satisfaction that they have designed and implemented a part of primitive computer, on the CPLD chip. It also shows the students that HDL can be combined with schematics in a complex hierarchical design.

Lab 9. **Arithmetic and Logic Unit Design VHDL**

This is basically the same design as in Lab 8, but everything is done with the VHDL language. Students are learning how much easier using VHDL is. After all of the difficulty they have had in the previous lab with a traditional schematic approach. Students like this lab very much and many of them continue on to take the elective course about hardware descriptive language.

Lab 10. **Bistable Memory Devices**

Simple latch circuits are being constructed here such as S-R and D. Internal structure of these latches are analyzed. First they have to build latches out of digital primitives, then using edge-triggered flip-flops, the students construct a ripple counter. This lab is the first exposure to synchronous design.

Lab 11. **Stepper Motor Controller**

The purpose of this lab is to design a state machine that controls a stepper motor. During state machine designs, students often have a real problem of figuring out why they have to study this concept and what the practical implications are. This is a very practical design where after completion they can actually observe and control movement of stepper motor with their design.

Lab 12. **Synchronous Counter Design**

Synchronous design is very important part of modern digital design and students are again exposed to this concept here. They have to observe different timing diagrams including micro-timing and macro-timing. For that purpose, they have to use both simulation tools and oscilloscopes.

The Lab is not covering the internal structure of digital circuits on the transistor level. This is a very important topic, but it cannot be covered in this course since most of students from COSC haven't ever heard about Ohms or Kirkhoffs laws.

V. Student Reaction

This form of the lab is used for last two semesters. We have received a very positive response from COSC students for whom this is the only hardware oriented course they have taken. In several cases the students like this lab so much that they have changed major from COSC to EE. Several other COSC students have expressed their disappointment that they have not taken this course until they were seniors and therefore a change of their major was not feasible. Most of the COSC students that couldn't change majors continued on in digital design taking the Advanced Digital course where HDL is the primary focus. The most difficult part of the lab for most of the student was getting familiar with the new software. They have no problems downloading to the CPLD since the program takes care of all the commands.

Depending on the students major, there was a clear difference in handling different assignments in the Lab. Problems and solutions for COSC and EE students are addressed separately.

COSC Student Problems

1) The COSC students have had the most problems with the wiring of TTL circuits and the idea of "clean" design using CAD tools. Since the COSC students have not taken a circuits course they haven't ever seen a breadboard. They also have trouble remembering the chips need to be powered up to work properly. The students tend to wire the circuits on the breadboards with wires that are not the proper length and with no color scheme, therefore

making more work for themselves when they have to trouble-shoot their design. They haven't been exposed to voltages so they don't understand what a logical 1 or a logical 0 is. This problem comes into play when they try to simulate the circuit and they don't understand the concept of when the switch is ON it means there is a 1 present on the input.

2) The circuits course teaches the students that neatly drawn circuit diagrams are essential. It also teaches the students how to organize their thoughts before placing components into a CAD type program. Therefore without this circuits course the COSC students tend to just start placing components everywhere on the page which causes wiring to become very confusing again making trouble shooting the design difficult.

3) COSC students haven't ever seen any hardware type design problems. Therefore they tend to have trouble with the concept of design, where the EE student which have taken circuits have seen some type of little design problems.

4) Another problem for the COSC students has to do with the pre-lab and the lab report. In the COSC department, most of the labs they have experienced either don't have a pre-lab or the pre-lab consists of "Read the Lab". The lab reports in the COSC department are usually just hand in your work after you complete the lab and this lab requires a formal written report.

Fixes for COSC Students:

1) The problem of the COSC students never seeing a breadboard before entering this lab can be taken care of by handing out a drawing on how a breadboard is connected. This handout can be seen in Fig. 5.

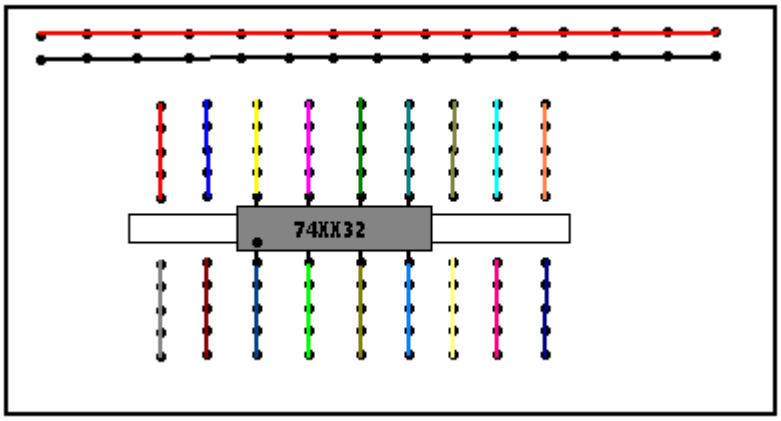


Fig. 5 Breadboard internal connections

The students are also shown how much easier it is to trouble-shoot a design where the wires are cut to the proper lengths and the colors are coded to make sense. The first wiring lab has been done beforehand to show this example. To make the students better understand what a 1 and a 0 are, with respect to the switches, a "1" and a "0" are penciled in above and below the switch for the first wiring lab so the students can understand and hopefully remember what the switch positions are.

2) After doing two or three labs the students catch on to the fact that one must keep a layout very clean. They learn how to use the concept of Hierarchy and what the importance of this concept is.
3) With time the students learn how one must think when doing a hardware type design from a word problem or a given truth table.

EE Student Problems

1) The EE students have problems with the VHDL idea. The EE students love the idea of a CAD type tool since they know how to wire a circuit using a breadboard and realize that a CAD tool is much faster and neater. Therefore most EE students are very reluctant to accept the idea of HDL type design. Most of these students haven't seen any type of a programming language and are intimidated by the idea.
2) The EE students also have a hard time keeping the bigger designs clean.

Fixes for EE Students:

1) The EE students tend to accept the idea that HDL is the only way to do digital design when it is explained to them that in this day and age when designs are so complex that schematic capture is just impossible. This explanation is reinforced when the add/subtract and ALU labs are completed the two different ways.
2) The EE student also learns the importance of Hierarchy in their designs.

VI. Conclusion

There have been several changes made to these labs to try not to overwhelm the COSC students. One of the changes was to do more HDL type design to push the EE students more since that is the way digital design is being done in the industry. With more HDL, the COSC student feel more at home and they tend to catch on quicker. The labs that require wiring TTL chips are kept simple just to demonstrate the concepts of how to wire them. The complex designs are put on programmable logic devices since this is one of the trends of today's industry.

Upon completing the labs, the students have knowledge and skills to design and implement complex digital systems consisting of combinational logic circuits as well as sequential circuits using modern design methods and sophisticated CAD tools. The students' experience includes the ability to use both the schematic editor and the VHDL editor, the skills to extensively simulate, analyze, and trouble-shoot the design and the implementation in silicon.

Bibliography

1. G. Puvvada and M.A. Breuer, "Teaching computer hardware design using commercial CAD tools," *IEEE Trans. Educ.*, vol. 36, pp. 158-163, Feb. 1993.
2. V. Madisetti, A. Gadient, J. Stinson, J. Aylor, R. Klenke, H. Carter, T. Egolf, M. Salinas and T. Taylor, "DARPA's Digital System Design Curriculum and Peer-Reviewed Educational Infrastructure," 1997 ASEE Annual Conference and Exposition, Session 1232.
3. G.M. Brown and N. Vrana, "A computer architecture laboratory course using programmable logic," *IEEE Trans. Educ.*, vol. 38, pp. 118-125, May 1995.
4. C.R. Carroll, "Portable Input/Output Instrument for Interfacing Student Digital System Designs," 1998 ASEE Annual Conference and Exposition, Session 1532.

5. A.B. Grubbs Jr. and A. Anthony, “Implementing PLD Technology in an Introductory Digital Logic Course,” 1996 ASEE Annual Conference and Exposition, Session 3548.
6. C.A. Lipari, C. Sisterna, R. Sundarajaran, and R.W. Nowlin, “VHDL Modeling and Synthesis in the Laboratory,” 1999 ASEE Annual Conference and Exposition, Session 2647.
7. M.E. Parten, “Teaching Digital Design with HDL,” 1997 ASEE Annual Conference and Exposition, Session 1232.

OTTO FUCIK
Otto Fucik (IEEE member) is an Assistant Professor of Electrical and Computer Engineering at the Penn State University at Erie. Before he joined the Penn State he was with the University of Wyoming as a Visiting Professor. He received his M.S. in computer engineering in 1990 and PhD in computer science and engineering in 1997, both from the Technical University of Brno, Czech Republic.

BOGDAN M. WILAMOWSKI
Bogdan M. Wilamowski (IEEE Fellow) is a Professor of Electrical Engineering at the University of Wyoming. He is the treasurer of IEEE Industrial Electronics Society, a member of the IEEE Neural Network Council, an associate editor of IEEE Trans. on Neural Networks and IEEE Trans. on Education. Dr. Wilamowski is the author of 4 textbooks, more than 200 refereed publications, and 27 patents. He received his MS in computer engineering in 1966, PhD in neural computing in 1970, and D.Sc. in integrated circuit design in 1977, all from the Technical University of Gdansk, Poland.

MICHAEL J. MCKENNA
Michael J. McKenna is currently a M.S. student at the University of Wyoming. He received his BS in Electrical Engineering from the University of Wyoming in Laramie, Wyoming. Before returning to graduate school Mr. McKenna spent a summer in Phoenix Arizona at Honeywell Inc. as a test engineer. He teaches several sections of the digital design lab at the University of Wyoming.