Integrating Reconfigurable Logic in the First Digital Logic Course

Guillermo A. Vera, Jorge Parra, Craig Kief, Marios Pattichis and Howard Pollard
Department of Electrical and Computer Engineering, The University of New Mexico,
Albuquerque, New Mexico 87131-1356
{alonzo,jeparra,kiefc,pattichis,pollard}@ece.unm.edu

Abstract - **Programmable logic popularity among the industry has increased due to its capability to shorten lead-time while also increasing design flexibility. This has led to the need for engineers with experience in programmable devices and hardware description languages such as VHDL. In response to this, many universities are introducing this topic into undergraduate engineering curricula; facing a number of logistical and didactical challenges.**

Programmable logic was introduced in the undergraduate curriculum at the University of New Mexico (UNM) in Fall 2000. For this purpose, a laboratory with programmable devices development boards was built and a series of laboratory exercises were developed. Our laboratories are built on Complex Programmable Logic Devices (CPLDs) and Field Programmable Gate Arrays (FPGA) devices. The laboratory exercises use the Xilinx ISE development environment. In order to provide the students with resources for better analysis and understanding of programmable logic designs, the labs also use other tools such as XPower, Floorplanner and Sysgen.

This paper discusses the challenges that were faced, and how our experiences differ from previously reported similar efforts. It explains the process followed to select the software and hardware that best suited this task. It also compiles the laboratory exercises and explains how they address basic concepts learned on digital logic courses. All the developed material is currently freely available online.

Index Terms – Digital logic course, Engineering curricula, Programmable logic, VHDL.

INTRODUCTION

The introduction of programmable logic in the undergraduate course work of electrical and computer engineering students has become a must in recent years due to strong industry demand. Even though the fact that programmable logic can improve effectiveness in digital logic teaching has long ago been established [1] and several universities are already on this path, the introduction of programmable logic technology in a first course on digital logic remains a challenge.

In spite of the difficulties, there are a number of important benefits in teaching these tools into a first digital logic course. In particular, the use of an integrated logic design tool exposes students to the full design and implementation cycle and also to the use of hierarchical methods for top-down design. It provides students with the skills to employ the reconfigurable logic tools for digital circuit analysis, to examine, understand and debug reconfigurable logic implementations.

It has been recognized (see [2]-[3]) that the use of 7400 discrete elements into laboratory sessions is inefficient and might generate frustration because of the time needed to build and debug projects. Adequate hardware to replace 7400 logic has been a concern since the early times of programmable logic [4]-[5]. Programmable devices use in digital electronics labs have been reported as early as 1988 [6]-[8]. Experience using prototyping platforms from different developers Xilinx® and Altera® were reported in [9]-[14] Additional experiences in building custom prototyping platforms were also reported in [15]-[17]. In the case of software, experiences using Xilinx and Altera tools were reported in [9]-[14] while others provide their own custom-built software [17].

Pedagogical approaches reported in [2]-[3], [18]-[19] agree on the use of programmable logic to foster logical thinking, encourage creativity, and avoid inherent limitations of the old discrete logic approach. A deeper study on the pedagogical needs that programmable logic can fill in terms of student curricula and how it can be introduced in a laboratory classroom is reported in [20]. In [18], a gradual introduction to programmable logic is proposed, having as a goal the exposure of the students to capabilities and power of this technology rather than attempting to achieve a great level of expertise. References [12]-[14], [21]-[22] describe the use of programmable logic devices through different stages of undergraduate curricula. It is also worth mentioning that reported efforts [23]-[24] on taking laboratories to a distance learning environment are based on custom software and third party systems in order to provide required hardware to the students. Examples of other Internet-based laboratories can be found at [25]-[27].

UNM, in collaboration with the Xilinx University Program (XUP), has developed a number of digital logic labs for the first course in digital logic. With respect to previously reported experiences, these labs are an alternative based on reliable, inexpensive software and hardware. Also, all source materials are available from www.ece.unm.edu/vhdl.

The rest of this paper is organized into three Sections. In Section 2, the objectives for the task of introducing reconfigurable logic into the first digital logic course are outlined. In Section 3, the latest editions of the labs given at the University of New Mexico are described. In Section 4, a detailed description of lab assessment results is given. In Section 5, ongoing educational efforts are summarized.

DIGITAL LOGIC COURSE OBJECTIVES

At UNM, the first digital logic course is ECE 238L (Computer Logic Design). Its main topics include Boolean algebra and logic gates, combinational and sequential circuit design and analysis, and an introduction to computer design. There are several text books that cover these topics, (i.e. [28-32]). ECE 238L is based on [29] which covers these topics in the first ten chapters. Topics are supplemented by laboratory sessions where students gain practice by using actual hardware and learn the details involved in the design of digital systems.

Laboratory sessions based on programmable logic and the inclusion of (CPLDs) and (FPGAs) in class lectures started in 2001. XUP supported this effort through software and hardware donations to establish the school's first programmable logic laboratory. This fact also reinforced our decision to use [29] as text book since it provides a student edition of Xilinx design tools. Since its inception, instructional material has been developed and presented online at the department's website [33].

A prerequisite for a digital logic course with reconfigurable logic tools is that the students must have taken a prior programming course. This requirement facilitates the introduction of a Hardware Description Language (HDL) earlier in the course. We summarize our objectives for teaching reconfigurable tools in the first course in digital logic below:

- Early introduction of an integrated logic design tool.
- Reconfigurable logic tools for digital circuit analysis.
- Reconfigurable logic tools for digital circuit implementation.
- Hierarchical models for top-down design.

We will address each objective separately, as it follows.

I. Early Introduction of an Integrated Logic Design Tool

Early introduction of logic design tools aims at familiarizing students with the design hierarchy as soon as possible. Great care must be taken to ensure that fundamental concepts are emphasized, while at the same time, the students are exposed to reconfigurable logic tools, and the complete design and implementation process. Thus, the process can start with traditional pencil and paper circuits with just a few gates that students can recognize, and then walk them through the whole design, implementation, and testing phases.

Our experience suggests that when this is done at the very first lab, it tends to be very popular with the students. The students feel empowered that they have a powerful, industry quality tool that can help them accomplish very complicated tasks. In turn, this tends to motivate the students for doing more complicated projects. The next step on this process is to design and implementation using an HDL.

II.Reconfigurable Logic Tools for Logic Circuit Analysis

The use of reconfigurable logic tools in the teaching of digital logic circuit analysis has some clear benefits. Students are able to simulate and test their designs, and to verify their implementations. Verification is accomplished through the use of an HDL test bench code. In addition, some specialized tools can be used to help measuring power consumption for their design, and also determine the highest frequency for the target architecture. This exposes students to some modern design issues. The use of these tools can help students to visualize the clocks and their outputs in their own design.

III.Reconfigurable Logic Tools for Examining Digital Circuit Implementation

The use of reconfigurable tools is often criticized for not allowing students to see the most basic elements of their design. To circumvent this problem, there are a number of reconfigurable logic tools that allow students to peruse into the hardware implementation and to examine exactly how their design was implemented. We believe that this point should be greatly emphasized in the labs of an introductory course. It is important to make the students feel in control of their design, and that their design description was mapped to real hardware.

IV.Hierarchical Models for Top-Down Design

Perhaps the most powerful argument in favor of the use of reconfigurable logic tools is in hierarchical design. Pre-built logic circuits can be easily combined and allow for the implementation of more complicated circuits.

When using hierarchical design, we strongly caution against the use of any components that were not taught in regular class lectures. Our basic philosophy is that the students should be in a position to actually design the blocks that they are using.

DIGITAL LOGIC LAB IMPLEMENTATION

Clearly, for an early introduction to Computer Aided Design to be successful, we must commit a number of computational resources, staff support, and fees for software licenses. In summary, to build the laboratory, software, hardware and teaching material (laboratory manuals and procedures) were required.

I. Software

There are two main streams of commercially available software tools for digital design and programmable logic:

Altera and Xilinx. As stated by [19], [21] the main key issues to consider for software selection are; *performance, complexity, suitability* and *affordability*. Xilinx software does satisfy reasonable expectations in all four key issues. Xilinx tools were chosen due to their availability through the Xilinx University Program and the fact that the current textbook [29] for the digital design class provides a copy of the software. Over the years, Xilinx tools have demonstrated to be didactic and intuitive, making students learning experience easier.

The current series loaded to every laboratory machine in the department is as follows:

- Xilinx ISE 7.1.4
- Xilinx EDK 7.1.2
- Xilinx Sysgen 7.1
- Mentor Graphics ModelSim 5.7 and Leonardo Spectrum
- Matlab Release 14 (with Simulink)

II. Hardware

In terms of hardware, the laboratory needed reliable and long lasting equipment, able to withstand student's errors. A CPLD [34] and an FPGA [35] board from Digilent (www.digilentinc.com) were selected (see Figure 1). Both boards provide state-of-the-art devices with a basic set of inputs/outputs available for the students. Having both technologies available in the laboratory helps students to understand some of the basic principles of programmable logic technology [36].

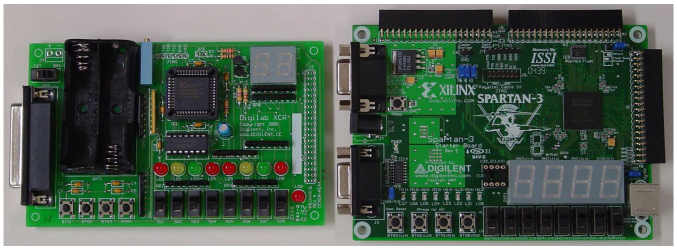

FIGURE 1
DIGILENT COOLRUNNER CPLD (LEFT) AND DIGILENT SPARTAN3 FPGA (RIGHT) PLATFORMS.

III.Laboratory manuals

The laboratory manuals currently in use are a result of five years of evolution, during which software, hardware and course requirements have changed. The last five generations of previous lab assignments are posted in our WEB-site at http://www.eece.unm.edu/vhdl. The material is structured to allow a professor to customize the lab sequence and content of each session.

We expect to continue providing the most current (as well as the previous) versions of our labs. The basic set of labs is build upon eight lab manuals briefly described in the next sections.

- *Lab 1 and 2: Introduction to VHDL, ISE and ModelSim*

In this first lab, the main objective is to have students become familiar with the Integrated Software Environment (ISE) design flow. Students are presented with a simple logic schematic shown in Figure 2.

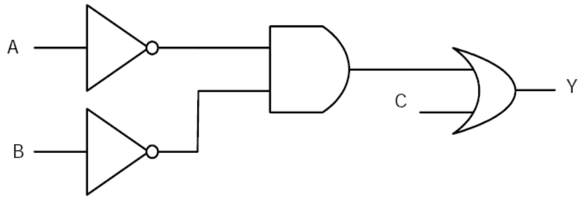

FIGURE 2
FIRST LOGIC DIAGRAM PRESENTED TO STUDENTS FOR IMPLEMENTATION USING VHDL.

By the second week in the course, students can recognize inverters, AND gates, and OR gates. They are given the VHDL description of the circuit and the test bench for its testing. The lab manuals provide GUI (Graphical User Interface) screen shots and step-by-step procedures for how to develop a new project in ISE, how to add the hardware description code and to simulate the resulting project. Figure 3 shows a sample of the simulation result for Figure 3 that the students will have to analyze.

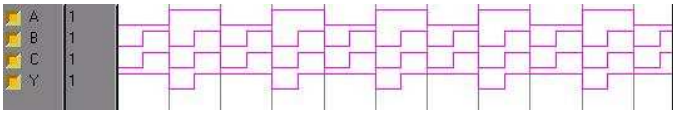

FIGURE 3
MODELSIM SIMULATION FOR FIRST LOGIC DIAGRAM

Using the obtained simulation results, the students can compare the predicted results from pencil and paper calculations with the result of simulating the code. At this stage, they begin to grasp the concept that digital circuits can be described based solely on their behavior.

The second lab project is just an extension of the first one. It allows students to develop a slightly larger project where they write some of their own descriptions. Although this is not a hardware description language course, surveys have shown that students want to write more of their own description. Once again, the simulation helps the students verify that they are correctly achieving the behavioral description.

- *Lab 3: Computer Arithmetic and IP Cores*

Design hierarchy is introduced with Core Generator (COREGEN) in the third lab. This tool offers students selectable pieces of intellectual property (IP) that are prepackaged and optimized for a specific device. Students are taken through the process of implementing a two's complementer and then are expected to implement a subtractor.

- *Lab 4 and Lab 5: Sequence Detectors and Counters*

In the sixth week, the students are introduced to sequential circuits. Laboratory 4 is an introduction to sequence detectors and CPLD devices. Where as the previous projects had stopped at synthesis (translating VHDL to electronic design interchange format (EDIF), this project continues on to implementation (creating a file that can be downloaded to the programmable chip). The students then use the buttons and LEDs on the CPLD board to step through a state machine's sequence and visualize the results on the LEDs.

An important concept introduced in this lab is the *behavioral* description. Students will realize that there is more than one way to describe a behavior, and that some ways are more efficient than others. In order to allow the students to relate their behavioral description of the system with a physical implementation, they are introduced to an ISE feature that allows them to visualize an RTL schematic of the synthesized design as shown in Figure 4.

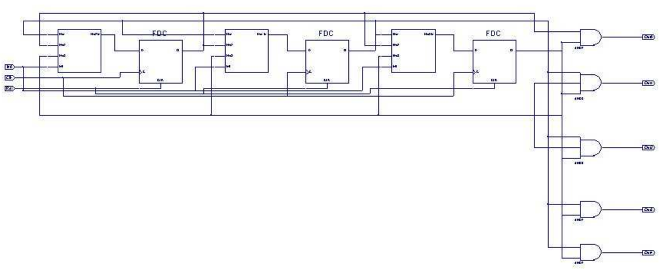

FIGURE 4
RTL VIEW OF A SIMPLE SEQUENCE DETECTOR.

Laboratory 5 continues with the exploration of sequential circuits and CPLD devices by introducing counters. This time, the students will use on-board oscillators and clock divisors to clock their counter sequences through the on-board seven-segment display.

- *Lab 6: Accumulators*

This laboratory is a transition from CPLD to FPGA devices. The many differences between these two device types are explored. The most important fact that students realize is the need to switch to FPGA devices to perform large-scale, register-intensive projects. CPLDs are designed for small-scale low-power applications.

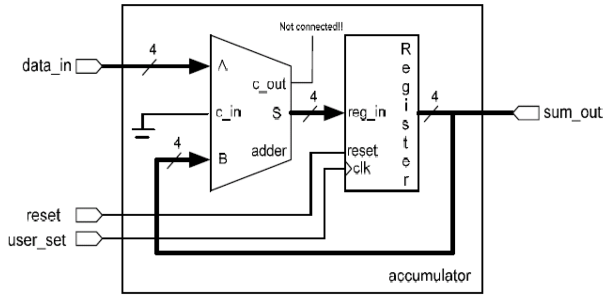

FIGURE 5
ACCUMULATOR DESIGN FOR LAB 6.

Students are required to build a 4-bit ripple carry adder using a bottom up design flow. In this way, they will first have to build a half adder, then a one-bit full adder, and finally the 4-bit full adder. Once the adder is designed, compiled and simulated, the students are asked to add a register to build an accumulator as shown in Figure 5.

Another important concept introduced is the *structural* description. They are instructed to draw schematics and then describe structurally their design. It also introduces modular design, since they will use small blocks to build bigger, more complex blocks and finally, a system.

A new tool is introduced in this lab: Floorplanner®. This tool allows the students to view and edit location constraints and visualize the inner workings of an FPGA and its components. It also gives the students a feeling on how their design is actually implemented in hardware. Students use the Floorplanner to explore how inputs come into the FPGAs through the Input/Output Blocks (IOBs), and routed to Configurable Logic Blocks (CLBs) and finally out to other IOB as shown in Figure 6.

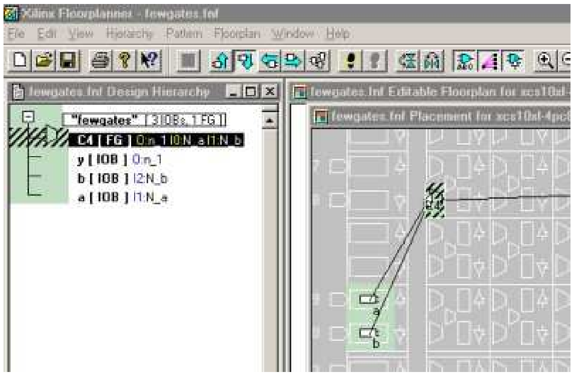
Xilinx Floorplanner - fewgates.fnf
File Edit View Hierarchy Pattern Floorplan Window Help
fewgates.fnf Design Hierarchy
fewgates.fnf Editable Floorplan for xcs10xl-
fewgates.fnf Placement for xcs10xl-4pc
"fewgates" [3 IOBs, 1 FG]
C4 [FG] 0:n_1 0:N_a I1:N_b
y [IOB] O:n_1
b [IOB] I2:N_b
a [IOB] I1:N_a
a
b

FIGURE 6
FLOORPLANNER DISPLAYING TWO INPUTS AND ONE OUTPUT.

Students are also introduced to the *Xpower* tool. Xpower is a component of ISE that allows one to estimate the power consumption of a design. Power consumption is an important issue for designers to be exposed to during an introductory course.

- *Lab 7: Introduction to System Generator*

This lab is used to introduce, at a basic level, a different tool for entry design: System Generator (SysGen). SysGen is a Matlab blockset provided by Xilinx that allows users to build and simulate hardware in the Simulink environment [37]. Since the students are already familiar with Matlab at this stage of their studies, the SysGen learning curve is small. Although this tool might abstract completely the students from the code-based hardware description task, it does provide a comprehensive environment for modular design. It also provides a graphical way to handle input stimuli and to visualize outputs. This lab guides the students through the design of a simple Add/Substractor block as shown in Figure 7. By the end, the student will estimate resource consumption (number of CLBs, Block RAM (BRAMs) and IOBs) and performance (frequency of operation). This allows them to compare the new design to the previous, vhdl-described designs, in terms of performance and resources.

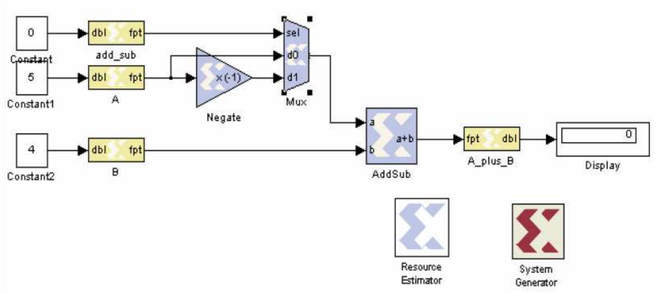

FIGURE 7
SIMPLE SYSGEN MODEL.

- *Lab 8: Final Project*

In this final lab the students are asked to build a major system. Through the years, several projects have been proposed. For details please refer to the lab website history at http://www.eece.unm.edu/vhdl/past_versions.htm.

These projects are meant to challenge the student to make decisions such as what device to use (CPLD or FPGA) and what abstraction level to describe the system on (i.e. behavioral / structural description). It also teaches team work and modular design.

LAB ASSESSMENT

The effectiveness of the strategies followed in the ECE 238L course were measured based on student surveys. These surveys asked for feedback on methodology, unusual difficulties in completing the projects, time taken to complete the projects, and alternatives for improving the lab.

Student feedback has been used to modify the labs to their current state. Repeatedly, students reacted positively to the fact that they could develop (and modify) projects dynamically, and the capability of changing hardware by modifying the underlining source description. Second, when given the choice of completing the projects using traditional (7400) logic and programmable logic, students choose programmable logic.

However, as mentioned in [19], students were led to false associations between hardware implementations and software coding. To minimize this misrepresentation, Xilinx's Floorplanner was included as a laboratory tool that allowed students to unveil the details of the hardware implementation of the described circuit on a reconfigurable device.

Emphasis on verification of the execution of the circuits implemented by the students and individual measurements of characteristics of these circuits such as maximum operating frequency and power consumption also contributed to widen the separation of the paradigm between software and hardware.

It was also found that students benefited from a design methodology that flowed from hand-drawn diagrams and circuits, to VHDL coding, VHDL test bench verification and finally to practical verification using laboratory instruments. This methodology was implemented in a hierarchical bottom-up fashion. It allowed students a smoother transition from the theoretical concepts to the simulation and implementation.

Positive feedback has also been received from other universities during recent Professors Workshops [38]. A final indicator of the acceptance that this effort has obtained among other scholars and institutions is the number of requests that have been made for copies of our hardware configurations and on-line laboratories and tutorials.

FUTURE WORK

Currently, in collaboration with the Xilinx University Program, we are developing a dual-FPGA prototype board for use in more advanced digital logic courses [39]. The prototype is now being used to create new teaching material for more advanced courses. We also plane to develop new teaching material for the first digital logic course, to create a set from which a professor can choose each semesters lab sequence and content.

ACKNOWLEDGMENT

The authors would like to thank the Xilinx University Program and the FMAC effort at UNM for their support to this effort. We would also like to thank K. Isler, T. Brown, J. Hansen, D. J. Ortley, L. Nilhen, J. Wright and M. Dubuque from UNM for the support and time committed to this effort. Additionally appreciation would like to be expressed to Clint and Fiona Cole from Digilent Inc.

REFERENCES

[1] Wong D., "Programmable logic design; the application of programmable logic devices to teaching digital systems design", *Int. J. Appl. Eng. Educ*., Oct. 1989, vol 5, No. 2, Pages: 221-228.

[2] Degen G. and Morgida M., "Using programmable logic devices in undergraduate digital design course", *Frontiers in Education Conference*, Now. 1992.

[3] Grower J. and Ugweje O, "Use of PLDs to illustrate fundamentals concepts in switching and logic", *32st ASEE/IEEE Frontiers in Education Conference*, Now. 2002.

[4] Abe T., Iijima J. and Wakatsuki Y., "Implementation of arithmetic algorithms using a PLA", *IEEE Transactions on Education*, Nov. 1989, Vol 32, Issue 3, pp 370-375.

[5] Abe T. and Naraoka M., "A programmable logic array suitable for use in digital system design laboratories", *IEEE Transactions on Education*, Nov. 1992, Vol 35, Issue 4, pp 338-345.

[6] Hedroug N., Abuelyaman E. and Johnson D., "A new approach for teaching computer engineering laboratory using PLDS2", *Frontiers in Education Conference*, 1988, pp. 266-278.

[7] Moss G., "Using programmable logic devices in the digital electronics laboratory", *Frontiers in Education Conference*, 1988, pp. 389-393.

[8] Degen . and Morgida M., "Using programmable logic devices in undergraduate digital design courses", *Frontiers in Education Conference*, 1992, pp. 841.

[9] Areibi S., "A first course in digital design using VHDL and programmable logic", *31st ASEE/IEEE Frontiers in Education Conference*, Oct. 2001.

[10] Ahlgren D., "CPLD-oriented design projects for the first course in digital systems", *Frontiers in Education Conference*, Oct. 1998.

[11] Meyer-Baese U., Digital Signal Processing with Field Programmable Gate Arrays, 2nd edition, Springer 2004.

[12] Hamblen J.," Rapid prototyping using fild-programmable logic devices", *Micro IEEE 2000*, Vol. 20, No. 3, pp. 29-37.

[13] Hall T. and Hamblen J., "System-on-a-programmable-chip development platforms in the classroom", *IEEE Transactions on Education*, 2004, Vo. 47, No. 4, pp. 502-507.

[14] Newman ., Hamblen J. and Hall T., "An introductory digital design course using a low-cost autonomous robot", *IEEE Transactions on Education*, 2002, Vol. 45. No. 3, pp. 289-296.

[15] Zemva A., and Zajc B., "A rapid prototyping environment for teaching digital logic design", *IEEE Transactions on Education*, Nov. 1998, Vol. 41., Issue 4, pp. 342.

[16] Etxebarria I. and Sanchez M., "An educational environment for VHDL hardware description language using the WWW and specific workbench", *31st ASEE/IEEE Frontiers in Education Conference*, Oct. 2001.

[17] Brown G. and Vrana N., "A computer architecture laboratory course using programmable logic", *IEEE Transactions on Education,* May 1995, Vol. 38, Issue 2, pp. 118-125.

[18] Wood S., "Introduction of programmable gate arrays into an introductory logic design course", *Frontiers in Education Conference*, Nov. 1994, session 5B2.

[19] Nixon M. "On a programmable approach to introducing digital design", *IEEE Transactions on Education*, Vol. 40, No. 3, Aug. 1997.

[20] Cole C., "Managing revolutionary change", Master's Thesis, Washington State University, 1999.

[21] Forcer T., Nixon M. and Zwolinski M., "An integrated framework for digital electronics education-programmable logic and IC design tools", in *Engineering Education 2002: Professional Engineering Scenarios* (Ref. No. 2002/056), IEE, Vol. 2, pp. 37-43.

[22] Reid K., "Incorporating VHDL into the digital curriculum", *Frontiers in Education Conference*, 2000, Vol. 2, pp. F2E/7.

[23] Spanias J., Constantinou A. and Bizuneh F., "An online signal processing laboratory", *1st. Signal Processing Education Workshop*, Oct. 2002.

[24] Carpinelli J. and Jaramillo F., "Simulation tools for digital design and computer organization and architecture", *31st ASEE/IEEE Frontiers in Education Conference*, Oct. 2001.

[25] Crabill E., "EE178 laboratory introduction", http://www.engr.sjsy.edu/crabill, retrieved March 1, 2006.

[26] Taylor C. "CSE371 Digital Systems Organization", http://www.seas.upenn.edu/~cse371/Labs/labs.html, retrieved March 1, 2006.

[27] Amaral J. "Computer Organization and Architecture II", http://wwww.cs.ualberta.ca/~amaral/courses/329/labs/index.html, retrieved March 1, 2006.

[28] Wakerly J., "Digital Design, Principles and Practices", Prentice Hall, 2001.

[29] Mano M. and Kime C., "Logic and Computer Design Fnudamentals", Prentice Hall, 2004, third Edition.

[30] Floyd T. "Digital Fundamentals with VHDL", Prentice Hall, 2003.

[31] Balabanian N. and Carlson B., "Digital Logic Design Principles". Wiley, 2001.

[32] Marcovitz A., "Introduction to Logic Design". McGraw Hill, 2002.

[33] "ECE238 Laboratory Website", http://www.eece.unm.edu/vhdl, retrieved March 1, 2006.

[34] "Digilab XCR Reference Manual", Digilent Inc., 2003.

[35] "Spartan-3 Starter kit board User Guide", Digilent Inc., 2005.

[36] PIB 18: "CPLDs vs FPGAs, comparing high-capacity programmable logic", Altera Corp. 1995.

[37] "Xilinx System Generator v8.1 User Guide", http://www.xilinx.com/products/software/sysgen/appdocs/userguide.htm, retrieved March 1, 2006.

[38] "Xilinx professors workshops at UNM", http://www.eece.unm.edu/xup/workshops.htm, retrieved March 1, 2006.

[39] Kief C., "Board of education", Xilinx XCell Journal, winter 2004.