Introducing Programmable Logic to Undergraduate Engineering Students in a Digital Electronics Course

Elías Todorovich, *Member, IEEE*, José A. Marone, and Martín Vázquez

***Abstract*—Due to significant technological advances and industry requirements, many universities have introduced programmable logic and hardware description languages into undergraduate engineering curricula. This has led to a number of logistical and didactical challenges, in particular for computer science students. In this paper, the integration of some programmable logic concepts into an introductory digital electronics course is presented. The proposed optional lab develops a printed circuit board that implements a programmable logic block. Another contribution is the collaborative problem-solving methodology used to achieve this goal. Surveys completed by the students, and their final grades, show that the lab has improved the quality of their education and has contributed to a successful integration of programmable logic concepts in an introductory digital electronics course. Because of its demands on students' time and effort, the lab favors the most motivated students. This suggests future research on a proposal for a lab that would be feasible within the time constraints for even the least motivated students.**

***Index Terms*—Active teaching, collaborative problem solving, digital electronics, engineering education, programmable logic.**

I. Introduction

THE IMPORTANCE of programmable logic in a computer engineering (CE), electrical engineering (EE), or even computer science (CS) curriculum is clear [1]. Students have to learn a number of new concepts and vocabulary and acquire complex skills in the laboratory in a short period of time. For this reason, it is helpful for students to learn the basic concepts of programmable logic early in their studies. This paper focuses on a successful experience that introduces the core concepts of a typical fine-grain RAM-based field programmable gate array (FPGA) in a second-year course on digital electronics.

The specific goal is the development of a didactic modular board with the basic functionality of a Xilinx-like slice, previously known as configurable logic block (CLB) [2]. The laboratory presented in this paper is called CLB-ED. The student's goal from the perspective of the digital electronics course is to develop a basic printed circuit board (PCB). This lab begins with circuit understanding, circuit schematic development, simulation, and PCB design—including manual routing and soldering—and concludes with board testing. To the best of the authors' knowledge, similar studies or teaching approaches covering programmable logic concepts in an introductory course have not been published.

During the 2007, 2008, and 2009 academic years, this teaching experience was presented at the Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, to CS students as an optional lab. This lab introduces concepts on programmable logic with the intention of improving the teaching efficiency in later courses that teach or use programmable logic.

This laboratory work provided students with valuable hands-on experience. However, additional time was needed to teach it, and there was an extra monetary cost both for the university and for the student. Thus, it was important to determine whether the approach had a significant impact on the academic performance of the students. This paper presents the results of the impact of this laboratory work on the digital electronics course and in the third- and fourth-year courses on computer architecture. The teaching experience in developing the CLB-ED boards is described in detail. Finally, the question of whether the students learned more about programmable logic is addressed.

II. Previous Work

Using programmable logic is a common practice in most universities worldwide. One of the earliest references is [3]. Since the 1980s, programmable logic devices (PLDs), together with a complete development methodology, have been used in computer architecture classes for CS and EE students to build working computers at Cornell University, Ithaca, NY. As usual with programmable logic, these labs minimize repetitive work and permit simulation before construction while ensuring that the student designs can be built, debugged, and operated in a single semester.

It is clear that CS students have a weaker background than CE students for understanding programmable logic. Teaching hardware description languages (HDL) is an issue as well. Some authors, for example [4], claim that even CS students can receive adequate training on digital design with programmable logic in a single semester if the course is well planned and adequate support material is provided, e.g., predesigned I/O modules.

One of the benefits of programmable logic is that it improves active learning techniques. For example, one of those techniques was successfully applied at the Technical University of Lodz, Lodz, Poland, to teach both discrete and programmable

Manuscript received February 07, 2011; revised April 28, 2011 and June 19, 2011; accepted August 01, 2011. Date of publication October 13, 2011; date of current version May 01, 2012. This work was supported in part by the Agencia Nacional de Promoción Científica y Tecnológica, Argentina, under Project PICT-2009-0041.

E. Todorovich and M. Vázquez are with the Instituto de Investigación en Tecnología Informática Avanzada (INTIA), Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil B7001BBO, Argentina (e-mail: etodorov@exa.unicen.edu.ar; mvazquez@exa.unicen.edu.ar;).

J. A. Marone is with the Instituto de Investigación en Tecnología Informática Avanzada (INTIA), Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil B7001BBO, Argentina, and also with the Consejo Nacional de Investigaciones Científicas y Técnicas (CONICET), Caba A1033AAJ, Argentina (e-mail: marone@exa.unicen.edu.ar).

Digital Object Identifier 10.1109/TE.2011.2169065

0018-9359/$26.00 © 2011 IEEE

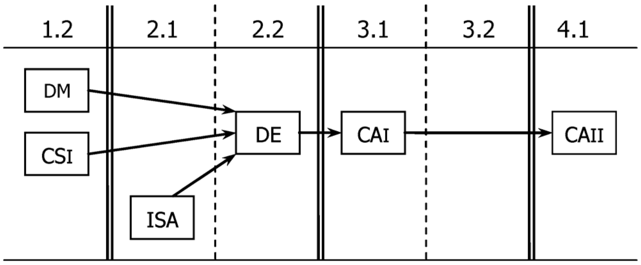

Fig. 1. Sequence of courses.

design techniques with the same board [5]. Furthermore, the authors show that these complex PLD (CPLD) boards can help EE and CS students bridge the gap between the discrete and programmable logic approaches.

Recent work presents an interesting means for teaching digital design utilizing programmable logic kits. In [6], a lab-at-home approach is proposed at the Universidad de la República in Montevideo, Montevideo, Uruguay. EE students taking an introductory digital design course study, design, and test digital circuits at home using a low-cost programmable logic kit provided by the university. Although the idea is simple, teaching time is optimized, thus enhancing the quality of education. The impact of the benefits and costs of providing unlimited access to programmable boards on digital design education is presented in [7]. Those studies were conducted at the undergraduate and graduate levels at three universities in the US and Romania: Rose–Hulman Institute of Technology, Terre Haute, IN; Washington State University, Pullman; and the Technical University of Cluj-Napoca, Cluj-Napoca, Romania. That paper concludes that students improve both their understanding of concepts related to digital systems and their design skills when they have unrestricted access to advanced programmable logic resources including the associated electronic design automation (EDA) tools.

Some authors treat students' motivation toward programmable logic. In [6], the authors focus on the students' point of view. They conclude that the use of programmable logic has helped facilitate the learning process in a digital design course at the University of California, San Diego. The problem of EE students' motivation in performing their laboratory practice is discussed in [8]. One approach to this problem involves incorporating new and innovative technologies; this is often not possible in laboratories or courses that have few resources. That work proposes a method for promoting autonomy and creativity in students while avoiding fully assisted and directed labs.

In the papers referenced above, the students did not have a detailed knowledge of FPGA architectures and probably did not understand most of the techniques for area, delay, and power optimization. This paper describes how basic concepts of programmable logic were taught in a digital electronics course. Furthermore, by means of the proposed lab, basic concepts of programmable logic were introduced early in the CS curriculum, which enhanced the experience and reinforced motivation. Additionally, students in this lab could not rely on memorization of the course material, but had to take a deep approach to learning, thus broadening their learning styles toward active learning and with observable physical results. A study on learning styles and orientations to study is presented in [9]. Students were required to develop their own circuits and were therefore encouraged to gain a better understanding of programmable logic concepts rather than relying on automatic solutions.

TABLE I
Students' Backgrounds

Year	Passed CSI		Passed ISA		Passed DM	
	# st	%	# st	%	# st	%
G07	11	100	10	90	5	45
G07'	82	100	82	100	30	36
G08	16	100	15	93	7	43
G08'	72	100	71	98	30	41
G09	20	100	19	95	20	100
G09'	30	100	30	100	17	56

III. Laboratory Objective and Assignments

A. Students' Background

Typically, students take this class in the second half of their second year. The sequence of courses relevant to digital electronics (DE) is depicted in Fig. 1. At the beginning of the second year, in Introduction to System Architecture (ISA), CS students study number representation, basic programming in assembly language, an introduction to computer organization, and other basic concepts. In Computer Science I (CSI), basic automata theory is introduced. This includes concepts ranging from finite state machines to Turing machines. In Discrete Mathematics (DM), lattice and Boolean algebra theory are taught. Computer Architecture I (CAI) and II (CAII) are third- and fourth-year courses directly related to this course. In CAI, an introduction to computer arithmetic, analysis of performance, instruction set architecture, basic processor architectures, memory hierarchy, and I/O systems are taught. In CAII, parallel processing and its performance and advanced processor architectures are studied in detail.

Table I gives the previous academic backgrounds of the students taking DE in 2007–2009. Students who participated in the optional lab are compared to those who did not. G07, G08, and G09 denote the students who took the optional laboratory, and G07′, G08′, and G09′ indicate those who did not. Table I shows the number and percentage of students who passed CSI, ISA, and DM. In all cases, 100% of the students passed CSI; the students who took the optional lab had a slightly weaker background than the rest of the students with respect to ISA. In G07, G08, and G09, 45%, 43%, and 100% passed DM versus 36%, 41%, and 56% in G07′, G08′, and G09′; that is, the students who took the optional lab had a stronger DM background than the rest of the students.

B. Course Overview

Digital Electronics is the introductory digital logic design course at UNCPBA for students in the CS five-year program [10]. The 15-week course is made up of one 3-h lecture and one 2-h lab session per week. There are seven units in the course, organized as follows:

1) electronics history from the abacus to solid state switches, diodes and transistors;
2) logic gates and basic combinational circuits including modules such as decoders and multiplexers;
3) sequential logic including latches, flip-flops (FF), registers, and counters;
4) read-only memories (ROM) and random access memories (RAM);
5) programmable technologies: programmable ROM (PROM), erasable PROM (EPROM), electrically erasable PROM (E2PROM), programmable logic array (PLA), and programmable array logic (PAL);
6) analog-to-digital and digital-to-analog converters;
7) integrated circuits: modern digital system design, fabrication, testing, very large scale integration (VLSI), and application-specific integrated circuits (ASICs).

The practical content is divided into five labs. Students use Logisim [11] as a schematic capture and simulation tool. Designs are then tested with a virtual protoboard tool [12]. Students have to select commercial components in order to implement the laboratory exercises. In the end, they make use of real protoboards for their designs. For the last optional lab, students employ Kicad [13] in order to design the required PCB. All these tools are free, and some are open-source.

The five labs in the course are organized as follows:

1) logic function to circuits based on AND and OR gate mapping implementation;
2) combinational circuit design;
3) bistables and tristate buffers;
4) sequential circuit design;
5) implementation of a simplified CLB (CLB-ED)

The first 10 weeks are devoted to the first four labs, while the last five weeks are used for the optional lab, which students perform in pairs. This paper focuses on this last optional lab, which is described in detail in Section III-C. With this lab, it is possible to teach students not only how a CLB works and how it can be programmed and reprogrammed, but also modular design, routing delays and glitches, and power consumption issues. This lab can help introduce concepts such as RAM-based FPGA, synthesis from HDLs, logic partitioning, technology mapping, placement and routing, and clock issues like skew.

C. CLB-ED Module Description

Fig. 2 shows a simplified version of the old Xilinx Virtex slice [2]. Both upper and lower halves include a four-input function generator, carry logic, and an FF. The output from the function generator drives both a combinational output and the D input of the FF. In addition, the Virtex slice contains hardwired multiplexers (MUX F5 and F6) that combine function generators to provide functions of five or six inputs enabling area optimizations. Even this simplified example could be too complex for a CS second-year student to implement the circuit with discrete components and develop the corresponding PCB. For example, as Virtex function generators are implemented as four-input lookup tables (LUTs), they can be configured as a 16 × 1-bit synchronous RAM or a 16-bit shift register.

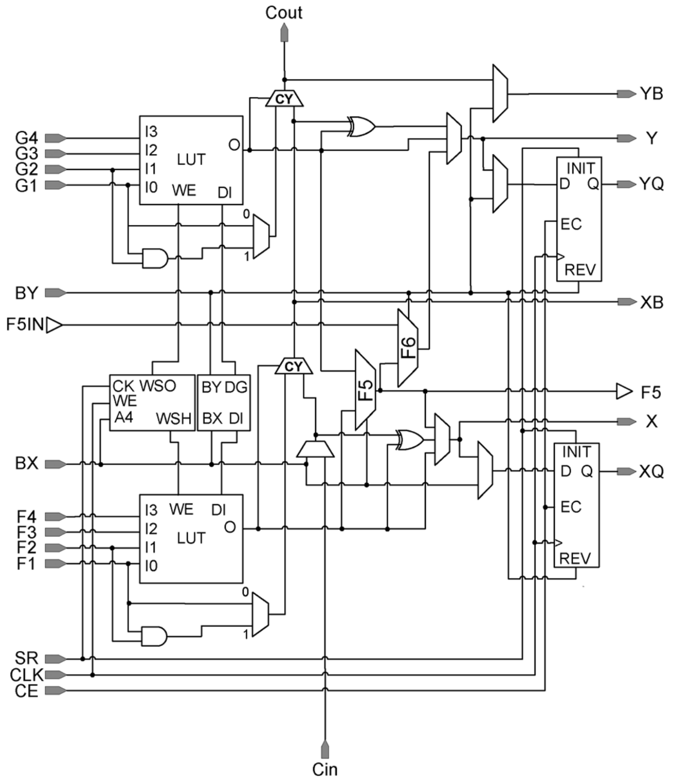

Fig. 2. Logic diagram of a Xilinx Virtex Slice (from [2]).

According to the fundamental objectives of engineering instructional laboratories developed in the ABET/Sloan Foundation colloquy [14], upon completing the proposed lab, the students will be able to do the following:

1) design, analyze, and build simple PCBs from given specifications using specific EDA tools and methodologies;
2) configure and evaluate the resulting programmable logic boards with different alternative combinational and sequential circuits observing the impact on area and routing;
3) demonstrate appropriate levels of independent thought by solving specific development and implementation problems on programmable logic;
4) communicate effectively their partial and final results by means of short presentations and technical reports;
5) work in teams assigning tasks, monitoring progress, and meeting deadlines.

In order to reach the learning objectives, the students have to develop a PCB with simplified functionality as shown in Fig. 3. In this case, the function generators have only three inputs. Fundamental functionality, such as programmability, embedded registers, fast carry logic, and additional multiplexers to provide four- and five-input functions, is taught. Although one CLB-ED module can implement a number of functions, by combining these boards, the students can build more complex circuits like N-bit adders, registers, and counters as shown in the examples that follow.

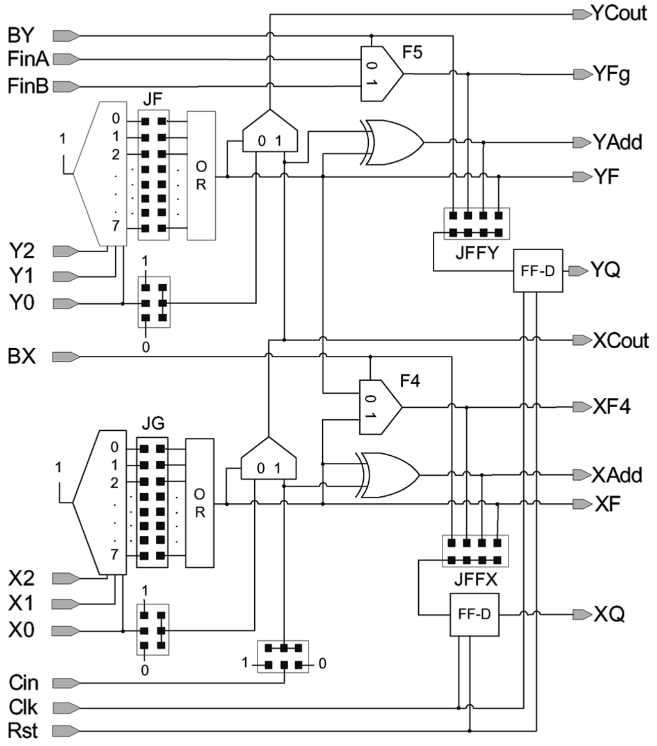

Fig. 3. CLB-ED circuit.

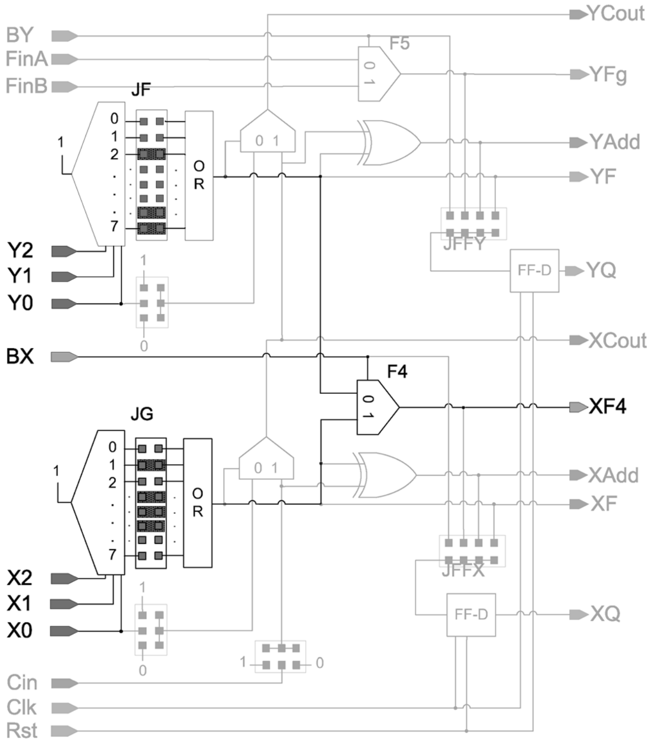

Fig. 5. Four-variable function configuration.

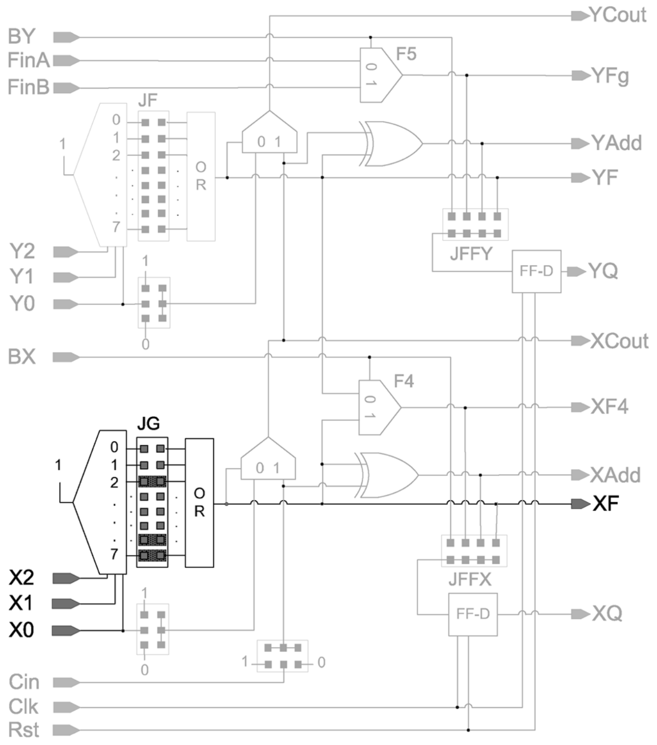

Fig. 4. Three-variable function configuration.

TABLE II
TRUTH TABLE FOR THE THREE-VARIABLE CONFIGURATION EXAMPLE

CLB-ED Pin	X2	X1	X0	XF
Variable	a	b	c	$f()$
F Function Inputs	0	0	0	0
	0	0	1	0
	0	1	0	1
	0	1	1	0
	1	0	0	0
	1	0	1	0
	1	1	0	1
	1	1	1	1

1) Three-Input Function Implementation: Fig. 4 shows the implementation of a three-input logic function. The jumpers JG map the function values as they are presented in a truth table. In this example, the function $\overline{a}b\overline{c} + ab\overline{c} + abc$ is shown (Table II). The result is available in the output connector XF. Now, it is easy for the students to understand how every three-variable logic function can be implemented with this board.

2) Four-Input Function Implementation: Fig. 5 shows how a four-input logic function can be implemented using MUX F4. This MUX is similar to MUX F5 in Virtex and Spartan FPGAs.

3) Five-Input Function Implementation: The key here is the use of an additional multiplexer (F5) controlled by the input BY. This MUX F5 in the CLB-ED board has the same goal as MUX F6 in Virtex and Spartan FPGAs. The students verify that with this additional multiplexer, two instead of three boards are needed. By studying four- and five-variable logic function implementations, it is clear why additional logic is added to the basic LUT+FF+MUX CLB configuration.

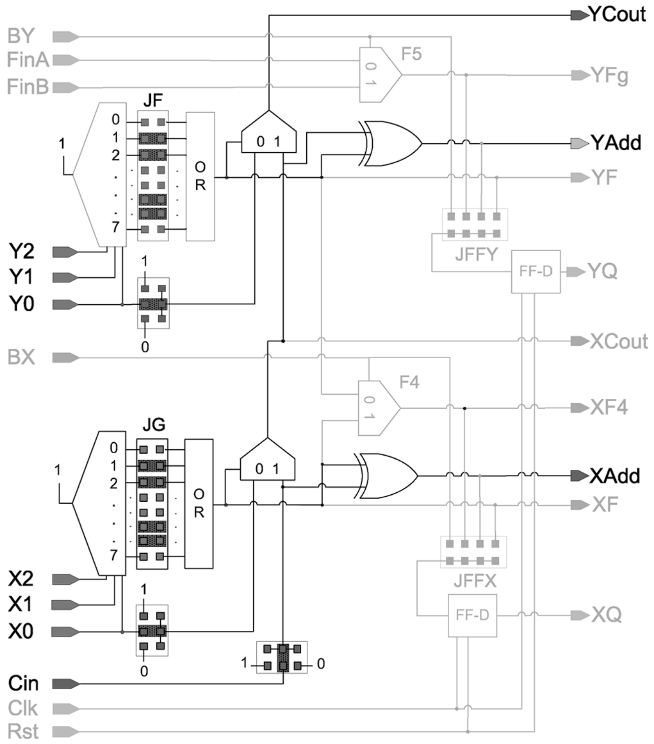

Fig. 6. 2-bit full adder configuration.

4) Adder/Subtractor: FPGAs include specific logic resources in order to accelerate carry propagation. These resources consist of dedicated routings and multiplexers and form the so-called carry chains along all the programmable block columns in FPGAs. These have been available since the late 1990s [15]. Fig. 6 shows the configuration needed to implement a 2-bit full adder/subtractor in a single CLB-ED board. Note that each decoder/jumper/OR (LUT equivalent) block implements the a_i XOR b_i function. The results are presented in the YAdd and XAdd board connectors; the carry is presented in the YCout board connector. In order to develop a 4-bit adder/subtractor, two CLB-ED boards are needed. YCout, in the board that implements the least significant bit portion, is connected to Cin in the second board that implements the most significant bit portion. Students can evaluate the impact of the carry chain circuitry on area and delay.

5) Registers: Bidirectional shift register with parallel load: The register has four modes of operation: inhibit clock (i.e., do nothing), parallel load, shift right, and shift left with "00," "01," "10," and "11" as control commands, s, respectively. In addition to the parallel output q, it has serial input and output. Operations are accomplished synchronously with the rising edge of the clock pulse. Only one CLB-ED board is needed per bit. Both three-input LUT-equivalents implement a 2-to-1 multiplexer, multiplexer F4 completes the required function, and data is loaded into the FF configured by the JFFX jumpers [Fig. 7(a)]. It is possible for students to observe how a basic module can be cascaded [Fig. 7(b) shows a 4-bit register] and to identify routing requirements.

IV. Active Teaching Methodology

Current engineering problems require the cooperation of several groups of experts from different areas. Some technical decisions are significant and require substantial consideration because of the potential economic impact they would have. A collaborative teaching approach may create greater satisfaction for all parties and promote a basis for future problem solving that is respectful and energizing. Students' negotiation and problem-solving skills are also improved. For these reasons, in these experiments a collaborative technique for problem solving is applied. Students only have 2 h per week in person with the staff and four additional hours at home. As the optional lab is mainly developed at home, an electronic forum is used to coordinate the project development. Today, this is a common practice in e-learning and Internet-based distance learning approaches [16].

When a complex project is being developed, it is necessary for participants to discuss and share information, define issues, and share further information in order to obtain a solution. To optimize the time investment, the flow of the discussions is guided by a plan. This plan is divided into three stages and posted at the beginning of the project in the News subforum. The stages are the following.

- *Stage 1:* Goal: To begin the project with adequate content. Study material: Circuit schematic of the required board, report format, and general guidelines. Student feedback: First report, first design (Logisim and Virtual Protoboard), an analysis of available components including datasheets, and OR gate implementation using diodes, pullup and pulldown resistors. Evaluation: A 15-min presentation given by the student.
- *Stage 2:* Goal: To understand routing tools and complete circuit routing. Study material: PCB-ED final schematic, tutorial about PCB deployment, and layout and routing guidelines. Student feedback: Netlist, board layout, board routing, and a second report describing the development. Evaluation: A 20-min presentation given by the student.
- *Stage 3:* Goal: To understand the final assembly and circuit test. Study material: Components, soldering instructions, and PCB techniques. Student feedback: Finished board (assembled and tested by the student), project documentation, and source files. Evaluation: Project presentation.

In the beginning, a sense of community was fostered through tutor initiatives. Later, as the students became more independent and involved, they began interacting directly with other groups' posts and questions via the electronic forum. The tutor then focused on finding the common threads between the various discussions, and weaving them together, to provide overall remarks and summaries on the discussion topics. The didactic model used since 2007 for collaborative problem solving in DE includes sharing perspectives, defining issues, identifying interests, generating options, and proposing objective criteria for decision making. Candidate solutions are combined and summarized, and the options are evaluated to reach a final agreement [17]. In the end, all groups shared the same placed circuit. The differences were in the routing, final implementation details, and PCB quality (Fig. 8).

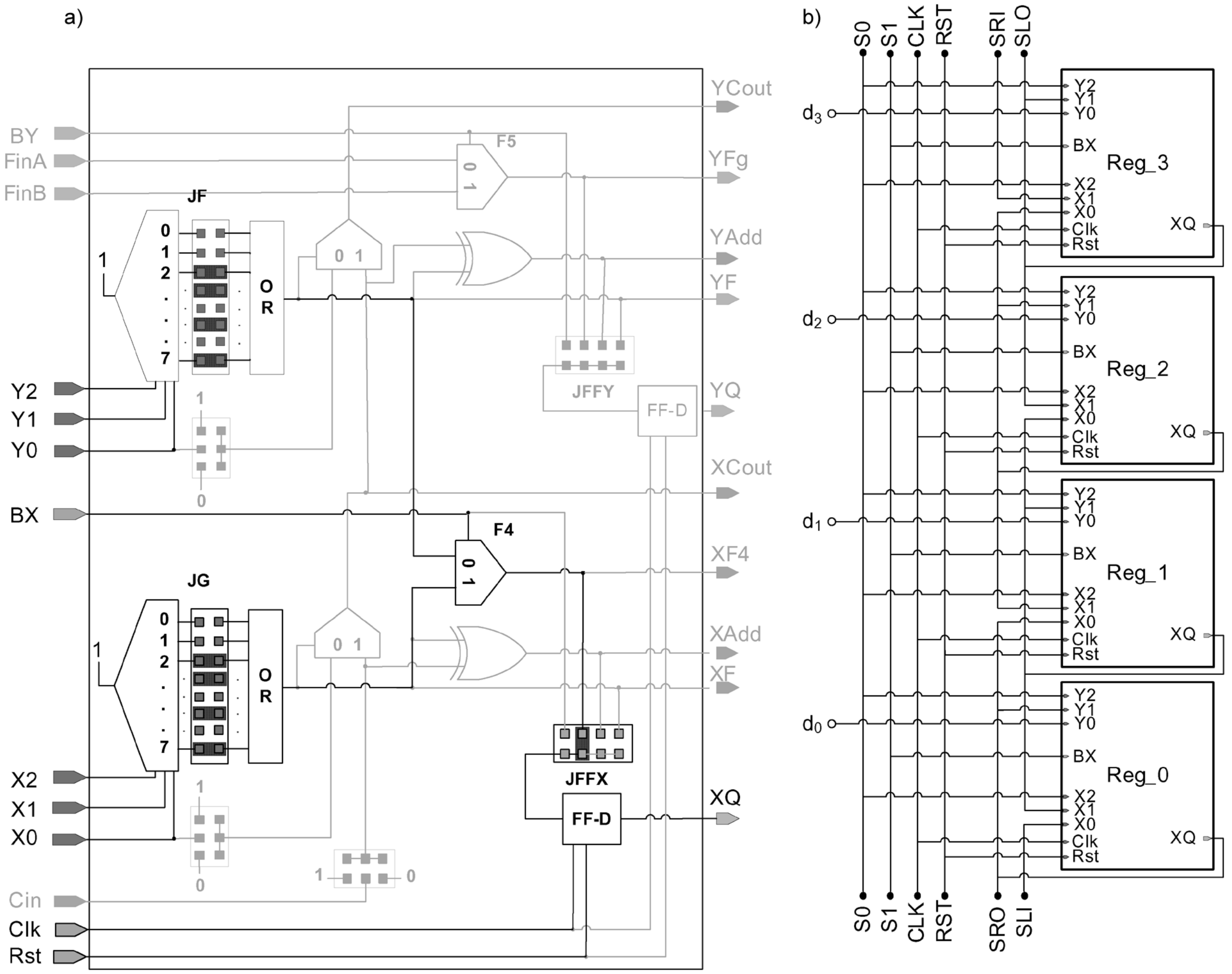

Fig. 7. Bidirectional shift register.

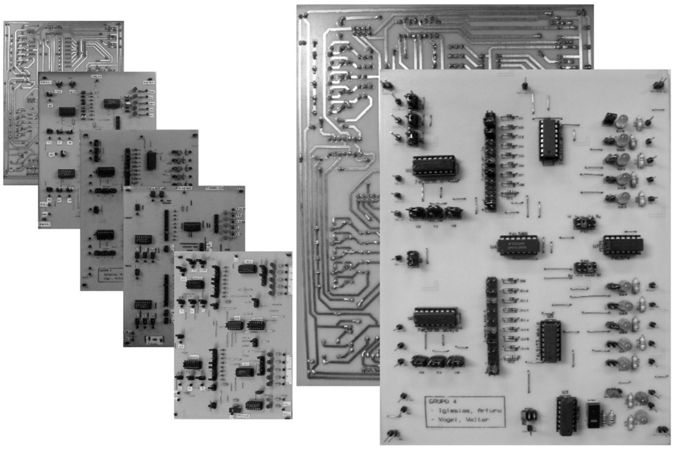

Fig. 8. PCB samples.

V. Survey and Academic Results

A. Survey

At the end of term, the students were asked to fill out a survey in class, assessing their opinions about and experience of the optional labs. In 2007, 8 out of 11 students (73%) completed the survey. In 2008, 11 out of 18 students (61%) responded. Finally, in 2009, 14 out of 20 students (70%) participated. The survey instructed the students to read a series of statements and give each statement a score from 1 to 5: 1 indicated that they strongly disagreed with the statement, 3 indicated they were neutral about the statement, and 5 indicated that they strongly agreed with the statement. Table III summarizes the results; columns 2–4 give the average scores for each analyzed term. Additional space for comments and suggestions was included.

Each statement can be above (3.1 to 5.0) or below (1.0 to 2.9) the median value of 3.0 for what the students and staff expected from the course. Survey results show that the students agreed that the overall quality of the course was good (statement 8). In 2007, the students considered the quality of the EDA tools (most of them open-source) and its documentation slightly below the average (statement 2). This feedback was considered and used to improve the materials for the next course. Students felt that the labs helped them to understand the lecture concepts (statement 3). As for the level of difficulty of the labs, the students' evaluations were positive (statement 4 and 7). Most students felt very confident that the hands-on experience with board development was likely to improve their career prospects (statement 6, highest score). On the other hand, they thought that these labs had a moderate impact on future specialization (statement 9), which is reasonable, and beyond instructors' expectations for

TABLE III
SURVEY RESULTS

Statement	Avg. 2007	Avg. 2008	Avg. 2009	
1) Overall quality of the contents presented in the laboratory was good	3.5	3.5	3.6	3.5
2) Overall quality of the EDA tools and documentation was good	2.9	3.5	3.4	3.3
3) Labs helped me to understand the lecture course concepts	3.9	3.6	3.5	3.7
4) The level of difficulty of the labs in general was fair	4.0	3.7	3.6	3.8
5) Working in the digital electronics labs was enjoyable	3.5	3.2	3.9	3.5
6) Developing a PCB will be valuable experience for my résumé	4.6	4.4	4.4	4.5
7) The level of difficulty of the CLB-ED lab was fair	3.7	4.2	3.9	3.9
8) Overall quality of the digital electronics course was good	3.8	3.9	3.5	3.7
9) This motivated me to pursue studies and/or a career in hardware	3.0	3.2	3.1	3.1
	3.7	3.7	3.7	

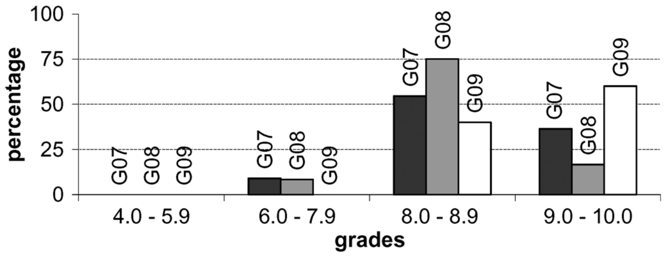

Fig. 9. Final exam score distribution for students who participated in the optional lab.

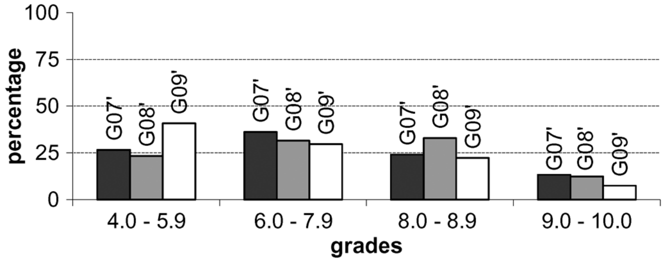

Fig. 10. Final exam score distribution for students who did not select the optional lab.

TABLE IV
STUDENTS' GRADES IN LATER COURSES

Year	Passed CAI			Passed CAII		
	#st	%	Avg. grade	#st	%	Avg. grade
G07	9	81	7.72	7	63	7.21
G07'	67	81	7.25	46	56	7.87
G08	8	50	7.09	7	43	8.28
G08'	57	79	6.15	29	40	7.15
G09	13	65	7.75	7	35	7.82
G09'	18	60	5.97	6	20	7.70
Total: Completed optional lab?						
Yes	30	63	7.56	21	44	7.77
No	142	77	6.14	81	44	7.56

CS students. Finally, in 2007 and 2009 (statement 5), the majority of the students responded that they enjoyed working on the digital electronics labs; the score was slightly lower in 2008, closer to the median value.

B. Final Grades

The digital electronics' final grades for the 2007, 2008, and 2009 terms are shown in Figs. 9 and 10. Students who participated in the optional lab were compared to those who did not, with the former performing significantly better than the latter. Undoubtedly, the most motivated students selected the optional labs.

In order to track the impact of this experience, students' grades in CAI and CAII were studied. In terms of difficulty, the examinations were similar because the professors remained the same, the course content was almost equal, and the tests had the same structure and were graded with the same criteria and methodology. Table IV shows the number and percentage of students who, having passed DE in the 2007, 2008, and 2009 terms, passed CAI and CAII in a later term. It was verified for CAI that the average grades of the students who participated in the optional labs were 23% better than those of the nonparticipants. For CAII, the scores do not show a significant difference; the average grades of the students who participated in the optional labs were only 2% better than those of the nonparticipants. It should be considered that due to typical dropout rates in engineering, significantly fewer students take CAII, a fourth-year course, as compared to DE.

VI. CONCLUSION

The proposed lab was technically, economically, and pedagogically challenging for the staff before its implementation. General results, including both the already available and the improved materials and tools, show that the labs were a successful experience, which repaid the investment in time and resources.

A comprehensive survey was conducted to collect feedback from the students on different aspects of the course. The effect of the proposed active teaching methodology on the optional lab was analyzed. The observations and the survey results indicate that the students considered the hands-on experience with board development very useful for their future professional activities. It became apparent that evaluations of new available tools needed to be done in order to stay current with the offerings from open-source software.

The experience with PCB assembly was excellent. The students were motivated to understand how the real hardware worked as shown by the survey (statements 5 and 6). Nevertheless, to help CS students overcome their inexperience with soldering and electronic circuit assembling, short tutorials were developed. These tutorials were implemented using videos and other digital material so that the students could acquire these skills at home. The students' questions were addressed using the forum.

The final grades were compared, and the results showed that the groups participating in the optional labs obtained better grades. The hands-on experience did make a difference in terms of students' future performance in more advanced courses as well. Although this lab cannot be considered the only reason for these results, it contributed positively to the training and motivation of the more engaged students. Thus, the proposed lab has improved the quality of education. This encouraged the digital electronics staff at UNCPBA to make this lab a requirement in the curriculum. However, considering the time and effort required to complete this lab and the current available time restrictions in the curricula, the staff is working on a new proposal including most of the goals reported here, but optimizing the student time required to complete the work.

The investigation into the inclusion of programmable logic concepts early in the curriculum will also be continued. All the CE and CS programs are currently participating in a quality assurance process in Argentina according to Higher Education Law 24,521. For this reason, course contents are being reviewed and updated in the 2011 term. It will be interesting to see what the impact of the future version of CLB-ED might be in the context of the new curricula.

ACKNOWLEDGMENT

The authors would like to thank the anonymous reviewers for their helpful comments to improve the quality of the paper.

REFERENCES

[1] N. L. V. Calazans and F. G. Moraes, "Integrating the teaching of computer organization and architecture with digital hardware design early in undergraduate courses," *IEEE Trans. Educ.*, vol. 44, no. 2, pp. 109–119, May 2001.

[2] I. Xilinx, "Virtex 2.5v FPGA detailed functional description," Xilinx, Inc., San Jose, CA, Tech. Rep. DS003-2, Dec. 2002.

[3] G. Brown and N. Vrana, "A computer architecture laboratory course using programmable logic," *IEEE Trans. Educ.*, vol. 38, no. 2, pp. 118–125, May 1995.

[4] J. N. Amaral, P. Berube, and P. Mehta, "Teaching digital design to computing science students in a single academic term," *IEEE Trans. Educ.*, vol. 48, no. 1, pp. 127–132, Feb. 2005.

[5] P. Debiec and M. Byczuk, "Teaching discrete and programmable logic design techniques using a single laboratory board," *IEEE Trans. Educ.*, 2011, to be published.

[6] Y. Zhu, T. Weng, and C.-K. Cheng, "Enhancing learning effectiveness in digital design courses through the use of programmable logic boards," *IEEE Trans. Educ.*, vol. 52, no. 1, pp. 151–156, Feb. 2009.

[7] M. Radu, C. Cole, M. Dabacan, J. Harris, and S. Sexton, "The impact of providing unlimited access to programmable boards in digital design education," *IEEE Trans. Educ.*, vol. 54, no. 2, pp. 174–183, May 2011.

[8] A. Saavedra Montes, H. Bortero Castro, and J. Hernández Riveros, "How to motivate students to work in the laboratory: A new approach for electrical machines laboratory," *IEEE Trans. Educ.*, vol. 53, no. 3, pp. 490–496, Aug. 2010.

[9] R. Felder and R. Brent, "Understanding student differences," *J. Eng. Educ.*, vol. 94, no. 1, pp. 57–72, Jan. 2005.

[10] "Página oficial de la materia electrónica digital," UNCPBA, Tandil, Argentina, 2011 [Online]. Available: http://electronica-digital.alumnos.exa.unicen.edu.ar/,

[11] C. Burch, "Logisim: A graphical tool for designing and simulating logic circuits," 2011 [Online]. Available: http://sourceforge.net/projects/circuit/

[12] A. J. M. de Priego Paz Soldán, "Simulador de construcción de circuitos digitales con escenarios virtuales y tutoriales interactivos," 2011 [Online]. Available: http://www.tourdigital.net/Simuladores/

[13] J.-P. Charras, "Kicad: An open source (GPL) software for the creation of electronic schematic diagrams," 2011 [Online]. Available: http://kicad.sourceforge.net/

[14] L. D. Feisel and A. J. Rosa, "The role of the laboratory in undergraduate engineering education," *J. Eng. Educ.*, vol. 94, pp. 121–130, Jan. 2005.

[15] B. J. New and K. M. Pierce, "Circuit for fast carry and logic," San Jose, CA, Jan. 1996 [Online]. Available: http://www.freepatentsonline.com/5481206.html

[16] W. Rice*, Moodle Teaching Techniques*. Birmingham, U.K.: Packt, 2007.

[17] R. Fisher, W. L. Ury, and B. Patton*, Getting to Yes: Negotiating Agreement Without Giving In*, 2nd ed. New York: Penguin, 1991.

Elías Todorovich (M'10) received the Computer Engineering degree from Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, in 1997, and the Ph.D. degree in computer engineering from the Universidad Autónoma de Madrid (UAM), Madrid, Spain, in 2006.

From 2003 to 2009, he was an Assistant Professor with the Department of Computer Engineering, UAM. He is currently a Professor with the University FASTA of Mar del Plata, Mar del Plata, Argentina, and Adjunct Professor with UNCPBA. His research interests include programmable logic, digital arithmetic, and functional verification.

José A. Marone received the Computer Engineering degree from Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, in 2009, and is currently pursuing the Ph.D. degree in computer engineering at the same university.

Since 2003, he has been part of the INTIA Institute, UNCPBA, were has worked on projects related to embedded systems and programmable logic. His research interests include machine vision, embedded systems, digital electronics, and hardware architectures.

Mr. Marone has obtained a scholarship from CONICET to complete his doctoral studies.

Martín Vázquez received the Computer Engineering degree from Universidad Nacional del Centro de la Provincia de Buenos Aires (UNCPBA), Tandil, Argentina, in 2002, and is currently pursuing the Ph.D. degree in computer engineering at the same university.

He is an Adjunct Professor with the University FASTA of Mar del Plata, Mar del Plata, Argentina, and a Teaching Assistant with UNCPBA. His research interests include FPGA design, digital arithmetic, and embedded systems.