The article has been published in IEEE Transactions on Education, Volume: 54, Issue 4, November 2011, pp.652-656

Teaching Discrete and Programmable Logic Design Techniques Using a Single Laboratory Board

Piotr Dębiec and Marcin Byczuk

***Abstract*—Programmable logic devices are used at many universities in introductory digital logic laboratories, where kits containing a single high capacity programmable logic device (PLD) replace "standard" sets containing breadboards, wires, and small- or medium-scale integration (SSI/MSI) chips. From the pedagogical point of view, two problems arise in these laboratories. Firstly, students have some difficulties in understanding the link between discrete and programmable design techniques. Secondly, using only a single PLD, students are not automatically forced to learn modular design skills, which is one of the required learning outcomes in Electrical Engineering (EE) curricula. To overcome these problems an innovative laboratory is proposed in the paper. Each training board contains four complex PLDs (CPLDs) and peripherals. In the initial sessions students implement designs in which CPLDs are programmed as standard SSI/MSI chips and interconnected to construct the target circuit. Then an active learning technique is used: students form groups of any size, and can set themselves more complex problems and choose their design approach. The**

proposed laboratory was set up at the Technical University of Lodz, Poland, and the first classes were given in the 2008/09 academic year. The instructors' insights, and design reports and feedback from students were gathered and analyzed. The results showed the effectiveness of the novel approach. Students were deeply involved in solving self-imposed problems. They learned "engineering imagination", modular design and teamwork skills. Some groups went far beyond the requirements set by the instructor. However, an additional conclusion is that "standard" approach should not be completely eliminated.

***Index Terms*—Educational technology, electronics engineering education, laboratories, logic design, programmable logic devices**

I. Introduction

The Programmable Logic Device (PLD) market has been the most rapidly developing sector of the very large scale integration (VLSI) semiconductor device industry for the last several years [1]. This fact has encouraged universities to incorporate PLD training boards in electrical engineering (EE) and computer science (CS) education and to replace discrete logic with PLDs even in introductory digital design laboratories.

This tendency is especially beneficial for CS students, who generally want to perform programming tasks and software simulations rather than make hardware connections.

Using PLD design and simulation software, they are able to develop quite complex systems using procedure-oriented hardware description languages (HDLs) such as Verilog HDL or VHDL. Using PLD boards, students can implement their designs quickly and enjoy interacting with hardware. This methodology can result in an improvement in learning outcomes [2].

The combination of the very short time needed for hardware implementation, the lack of wiring problems encountered, and well-prepared lecture and laboratory materials, allow teachers to run a state-of-the-art digital design course in a single academic term and receive positive and constructive feedback from CS students [3], [4].

An innovative approach to teaching digital design is to enhance distance learning with PLD hardware kits. The low-cost kits can be lent to students, or they can buy them for about $100. This teaching method has a high level of acceptance among students and teachers, optimizes teaching time and gives better final exam scores. The main drawback for students is that the boards are too expensive [2], [5].

As the research results show, incorporating PLD hardware in introductory digital design courses gives CS students a better understanding of the subject area than they acquire in simulation-only laboratories. However, replacing discrete logic laboratories with PLD-based ones for EE students may change the learning outcomes of the class. First, students get much more skilled in “programming” than in hardware modular design. They don’t place chips in breadboards, they don’t make connections between logic blocks, and they use logic probes only occasionally. Instead, they describe designs mainly in HDLs and perform a series of compilation, pin assignment, simulation, and chip programming processes.

Of course, modern introductory digital design laboratories should move towards HDLs and PLDs [6]. However, from pedagogical point of view simple gate-level designs should also be included in these laboratories to give students a better understanding of basic concepts of digital systems. These two different concepts, of gate-oriented and PLD-based designs, are usually taught using two different types of laboratory workstations. To smooth the transition between, some teachers have their students perform the intermediate step of reimplementating gate-oriented designs as PLDs [7].

This paper presents a novel laboratory board (CPLD board) which can be used to train students in both types of design concept. The main aim was to put a set of CPLD devices on a single board. The devices can be used as standard or non-standard digital blocks and can be quickly interconnected using ribbon cables. All inter-device signals are continuously probed using a set of light-emitting diodes (LEDs) placed on the board. These features allow students to practice discrete logic design.

For complex designs, the logic capacity limitation of a single CPLD means that students must break the system into separate modules (CPLDs), and interface these using serial or parallel buses. This modular design skill is very important for EE students. Additionally, a number of standard peripherals are placed on the board to make this introductory laboratory more attractive.

The structure of the paper is the following. Section II presents a brief description of the innovative board, and the motivation for developing it. In Section III, the teaching methodology for digital logic design for EE and CS curricula is presented, along with a list of sample design tasks completed in the CPLD laboratory in academic years 2008/09 and 2009/2010. Section IV presents student feedback. A summary and final conclusions are given in Section V.

II. CPLD Board Description

Various programmable logic development kits are available on the market. Almost all PLD manufacturers offer their own versions, of more or less use in higher education. For example, the DE2 board developed by Terasic Technologies has a field-programmable gate array (FPGA) - the Altera Cyclone II integrated circuit (IC) - and features high logic capacity, modern peripherals and comprehensive learning documentation. A number of universities use these boards successfully in their laboratories [8]. However, the kits available that include the DE2 contain only one PLD chip, and thus do not force students to take a modular design approach; they are thus not very suitable for EE students in introductory laboratories. This drove the eventual decision to build an original CPLD-based laboratory.

The general structure of the board, Fig. 1, consists of four main blocks: CPLD ICs with a Joint Test Action Group (JTAG) programmer, a universal bus, a number of peripherals, and the microcontroller, which controls some of the elements on the board. Brief descriptions of these blocks are presented in the following subsections.

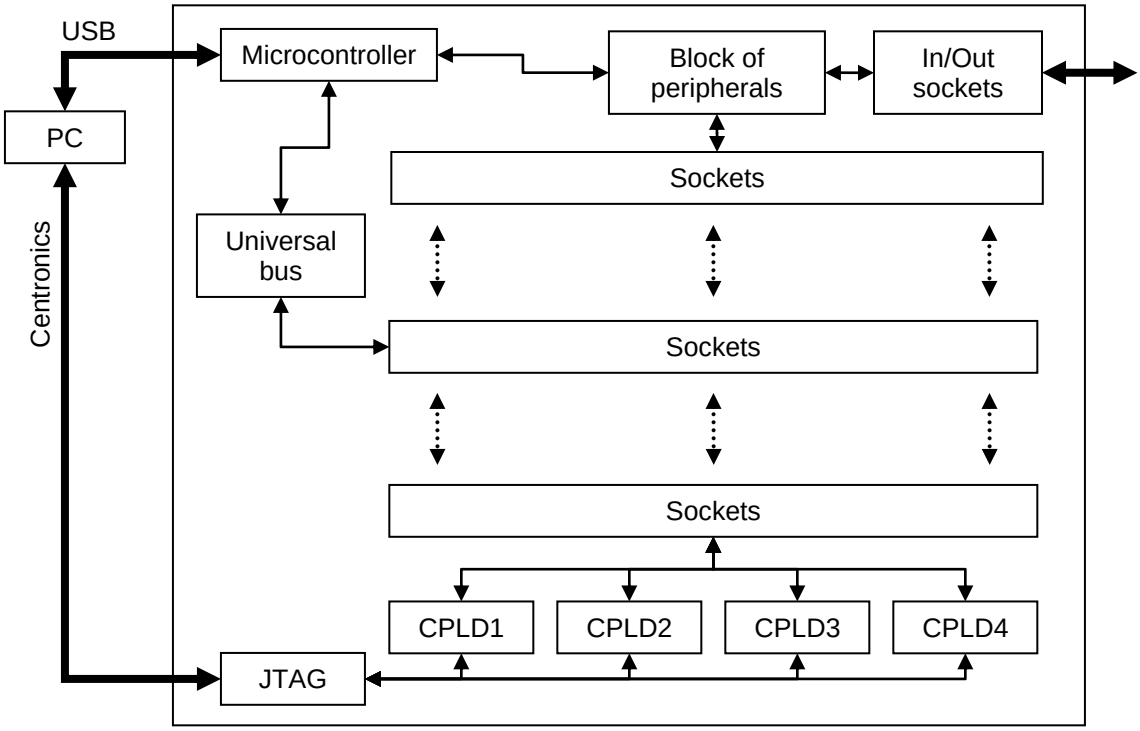

Fig. 1. Block diagram of CPLD board.

A. Universal Bus

Since the main aim of the CPLD board design was to ensure a large number of flexible connections between the peripherals and CPLD devices, a quantity of sockets and auniversal bus block occupy a large part of the board area, Fig. 2. Such a solution forces students to make the appropriate connections between logic blocks, using ribbon cables and single wires. The universal bus block consists of eight 8-bit buses. Each bus contains four sockets so that inter-block connections are possible. Innovative four-state LED probes were designed and connected to each bus line; these probes indicate either logic 0, logic 1, high impedance, or logic level contention, thus enabling more detailed testing and debugging than would two-state probes.

B. CPLD Block

The CPLD Block consists of four M4A5-64/32 chips, provided free of charge by Lattice Semiconductors within the framework of their university program. A deliberate choice was made to use four ICs with lower logic capacity, instead of a single IC with high logic capacity, so that students are forced to break complex problems into simpler ones, thus developing their “engineering imagination.” All CPLDs are serially connected with the JTAG programmer used for in-system programming (ISP). All the CPLD's input-output (I/O) pins are accessible via sockets and students can connected these with other CPLDs and peripherals either directly or by using a bus.

C. Peripherals

The peripheral block consists of a large number of standard devices widely used in digital systems, namely: 24 bistable and eight monostable switches with debouncing; a 4x4 matrix keypad; two binary code decimal (BCD) to 7-segment decoders; two 4-digit, 7-segment LED displays (one with independent control and one with multiplexed control); an alphanumeric liquid crystal display (LCD); a universal asynchronous receiver/transmitter (UART) with voltage level converter and DB9 socket (which can be used to connect to another CPLD board, PC workstation, or any other device equipped with RS232 port); two independent 32kB random access memories (RAMs), which allow for design of a processor with Harvard architecture; an I^2C bus connecting electrically-erasable programmable read-only memory (EEPROM); a real time clock (RTC); a digital-to-analog converter (DAC); a buffered PS/2 connection to PC keyboard or mouse; a clock signal generator; another DAC with accompanying devices (an analog comparator, an acoustic amplifier with loudspeaker, and composite video signal

formation circuitry with RCA output socket); and 1-Wire bus with iButton reader. Given this variety of peripherals, both simple and advanced designs, such as microprocessor systems, can be implemented. Additional external peripherals can be connected to the board via an extension socket.

D. Microcontroller

To improve the flexibility of the laboratory set, a microcontroller with the appropriate software was used to control some board functions and parameters. The board can be connected to a PC via a Universal Serial Bus (USB) cable and a series of commands can be sent to the board using any terminal emulator (e.g., Microsoft HyperTerminal) Functions available to students: changing the frequency of clock signals; writing or reading of any RAM module; writing or reading of I^2C peripherals; configuration of a debouncing module; and monitoring of the universal bus states.

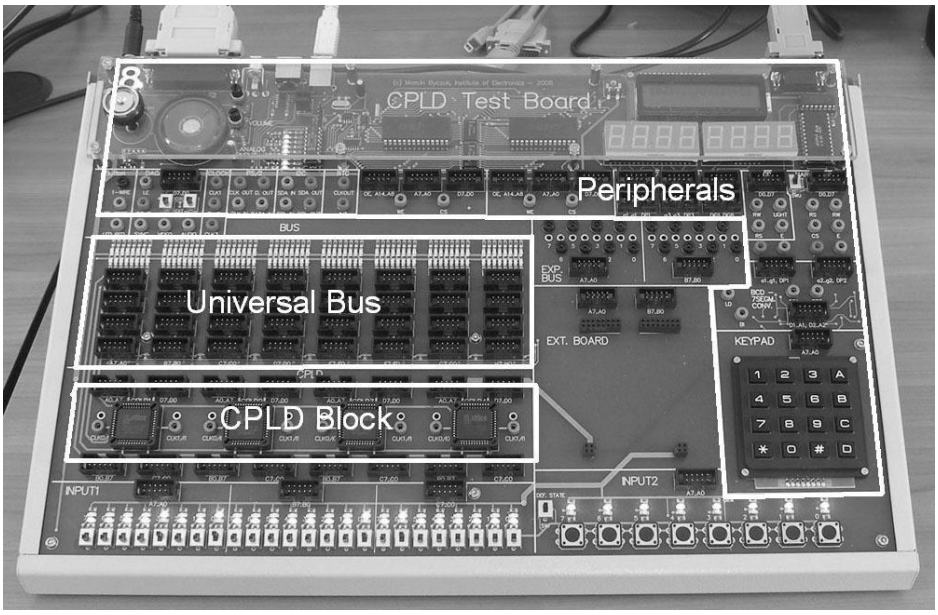

Fig. 2. Main blocks of the CPLD board (the microcontroller block is mounted on the other side of the printed circuit board using surface mount technology).

The software used for the laboratory is an industry-level design tool - ispLEVER Classic [9] from Lattice Semiconductor, one of the three biggest suppliers of CPLD ICs [10].

III. Teaching Methodology

The CPLD board presented was developed in the Institute of Electronics (IE), Technical University of Lodz (TUL), Poland, to complement or perhaps soon to replace a set of six discrete logic boards (DL boards), which were also developed in IE TUL as an alternative to breadboards. The DL boards contain standard SSI/MSI devices, which are soldered and partially interconnected (for example, all power pins are connected to a power supply socket, clock inputs are fed with a common clock signal, some select or input lines of MUXes are connected to the appropriate signals). This feature speeds up the implementation process, but requires a short but important phase of board circuit analysis. After discovering what the existing electrical links are, students make the crucial final connections using only a small number of wires, which are already fixed at one end to the appropriate nodes on the board. Thus the number of potential wiring errors, which is the main drawback of using breadboards [11], was substantially limited.

Both types of boards are used by about 250 students every year, in a number of required courses across two faculties: the Faculty of Electrical, Electronic, Computer and Control Engineering (EECCE), for EE students, and the International Faculty of Engineering (IFE), for CS and EE students.

The methodology and research results presented here refer to the courses run at IFE, where CS and EE students learn the fundamentals of digital design in their third semester, attending 30 hours of lecture and 15 hours of laboratory. The objective of the laboratory is to practice, quickly and effectively, the principles presented in the lecture. Therefore the laboratory sessions are performed using DL boards and each student is expected to solve the same set of problems, for instance designing a 4-input multiplexer (MUX), a 2-to-4 line decoder/demultiplexer using gates, combinational circuits using multiplexers, an iterated circuit (decoders, adder, comparator, multiplier), flip-flops and latches, and simple finite state machines (FSMs), and performing an analysis of a general microcomputer structure (central processing unit, memory, timing and control circuitry). After completion of the course students earn four European Credit Transfer and Accumulation System credits (ECTS-credits) [12].

In the following semester EE students attend another laboratory (30 hours, two ECTS-credits) where they design more complex systems and use CPLD boards. An active learning technique has been applied in this course [13]. Students are given as many “degrees of freedom” as possible: they can form groups of any size, they can set their own design problems, they freely choose their design approach (schematic, VHDL, Verilog HDL, or ABEL HDL), and they do not have to attend each laboratory session because much of the programming and simulation work can be performed at home. The list of problem tasks proposed by a student group must be approved by the instructor. A

bigger group must solve more problems, or more complex problems, to receive the same grade as a small group. For each problem, the group elects a leader, who distributes sub-problems among group members, coordinates the teamwork, and performs the final tests. Intergroup communication is also encouraged. The first goal of this methodology was to give students an opportunity to tackle problems that interested them and to match the complexity of problems with the students' abilities. Moreover, the variety of educational backgrounds at IFE and the flexibility of the CPLD boards encourage the instructor to pose different sets of problems for each student group. The second goal was to make use of students' teamwork skills.

Students are expected to write laboratory reports in electronic form; these should detail the functional requirements, block diagrams, design methodologies, final test results, and the contribution of each group member. The project source files must also be attached. To complete the course each student in the group has to present and discuss a part of the project selected by the instructor, and answer some questions on it.

During the first sessions students are asked to re-implement some selected SSI/MSI circuits from discrete logic laboratory, for instance an iterative adder, iterative comparator, traffic light controller, or other FSMs, using CPLDs as appropriate modules. Then students designed, simulated, and implemented a number of self-selected problems. The most interesting projects implemented in the 2008/09 and 2009/2010 academic years were:

- Simple terminal (keypad and LCD display controller),
- Scrolling text advertisement using RAM and LCD display,
- Random number generator (a shift register with a special feedback),
- Simple calculator with addition and multiplication,

- “Master mind” game using keypad and LCD display,

- Traffic light controller,

- Simple piano with eight sounds,

- Percussion automata,

- Audio player (samples loaded into RAM and then read in infinite loop),

- Arabic to Roman numbers converter (range 0-999),

- 8-bit von Neuman microprocessor system with move, add, OR, AND, XOR, conditional jump, and unconditional jump operations,

- Generator of a binary image (a pattern stored in RAM) in the PAL television standard.

The designs were developed in groups of two to five students.

IV. Teacher Insights and Feedback from Students

To evaluate the innovative teaching methodology, teachers’ insights and a student feedback were collected.

The instructors’ observations were that during all the laboratory sessions students at all levels were equally deeply involved in performing self-imposed tasks. The designs were mostly concerned with the keypad, loudspeaker, and displays, because the majority of students enjoyed feeling and testing the results with their senses (i.e., they were of “sensor” learning type [14]). The students were impatient and wanted to obtain the results very quickly. That is probably why no group dealt with the I^2C and 1-Wire buses. Using these buses would have required a lot more work before any data could be sent to the peripherals to obtain a spectacular effect.

Another daunting task for the students was software debugging. The main role of the instructor was to assist them in finding errors in the design source files. Some groups asked for extra time in the laboratory to correct these errors.

A wide variety of design approaches were used by students to obtain their final circuits. Some groups confined themselves to schematic entry, using simple logic blocks from the standard library. Other groups combined standard blocks with dedicated blocks described with ABEL HDL. Yet other groups preferred Verilog or VHDL, and described their designs using the behavioral modeling scheme of these languages.

The approaches chosen by the students reflected the complexity of the target circuits – simple designs were described with schematic and/or ABEL approaches, whereas complex designs forced students to use VHDL or Verilog. Even the lower level groups fulfilled the requirements set by the instructor, whereas more advanced groups went much further.

To elicit student feedback, a survey of SSI/MSI laboratory (DL boards) and CPLD laboratory (CPLD boards) was conducted among EE students. All 50 students willingly completed the survey forms, which contained a series of statements. Similarly to the survey presented in [2], one of five possible scores was given for each statement. The scores ranged from -2 to +2 with -2 indicating strong disagreement with the statement, 0 indicating a neutral attitude, and +2 indicating strong agreement with the statement. Table I presents the survey results, where the numbers of students who assigned the given score to the given statement are listed in columns denoted by **-2**, **-1**, **0**, **+1**, and **+2**. Column **Sum** contains the total scores, which are the weighted sums of the individual scores.

TABLE I

SURVEY RESULTS

	Statement	-2	-1	0	+1	+2	Sum
1.	The SSI/MSI laboratory allowed me better to understand the lecture topics.	0	3	3	17	27	**68**
2.	The SSI/MSI laboratory should be run using CPLD boards.	17	13	13	4	3	**-37**
3.	The SSI/MSI laboratory should still be run using SSI/MSI boards.	1	2	9	16	22	**56**
4.	The CPLD laboratory was interesting to me.	0	2	2	15	31	**75**
5.	I learnt a lot in the CPLD laboratory.	0	3	6	18	23	**61**
6.	Letting students choose how many students to have in a working group is a good idea.	0	1	9	11	29	**68**
7.	Giving students a free choice of the design tasks is a good idea.	0	0	3	8	39	**86**
8.	The CPLD laboratory materials published on the Internet were helpful.	1	2	11	19	17	**49**
9.	Working with CPLD boards I was able to learn more than by using only simulation software.	0	0	6	6	38	**82**
10.	I don't like debugging designs.	2	4	17	16	11	**30**
11.	I prefer hardware testing to simulations.	0	0	9	14	27	**68**
12.	Hardware tests allowed me to find errors faster than by using simulations.	0	5	14	13	18	**44**
13.	The ispLEVER design environment is not easy to use.	6	10	8	20	6	**10**
14.	The experience of using Lattice ispLEVER software is worth adding to my resumé.	1	2	15	22	11	**39**
15.	CPLD boards are easy to operate.	3	4	12	22	9	**30**
16.	Instead of four smaller CPLDs, one large capacity IC should be used on the CPLD boards.	10	13	13	8	6	**-13**
17.	The necessity of using several CPLDs taught me modular design.	1	0	9	26	14	**52**
18.	I would like to perform further experiments with CPLD boards.	0	1	6	23	20	**62**

The results clearly indicate that implementing designs in hardware is much more interesting and enjoyable for students than working only with software (Statements 4, 9, 11, and 12). The majority of students felt that they had learned a lot in both laboratories (Statements 1 and 5). In addition, they would not like to implement SSI/MSI designs using CPLD boards in the initial learning phase (Statements 2 and 3). This finding is very important for instructors who teach fundamentals of digital design. Replacing of

SSI/MSI boards with PLD boards, even with boards containing several PLDs, would not be a good idea in introductory laboratories. Nevertheless, the constructed boards can bridge the gap between the discrete and programmable logic approaches [7], and can also be used in later semesters to teach complex designs (Statement 18).

The score for Statement 17 and the instructors' experience show that the boards are very useful in teaching modular design. The desire for a single high capacity CPLD expressed by 30% of students (Statement 16) probably results from the difficulty of modular decomposition of a design problem and the impatience to "see" the results quickly.

Lattice ispLEVER Classic software did not receive a good score partially because its error and warning messages were not very informative, and there were some problems with synchronization of the source files with the intermediate files.

The "freedom teaching" strategy was largely accepted by students (Statements 6 and 7), as giving them an opportunity to perform at their best. The majority of final grades in CPLD laboratory were either good or very good.

V. Summary and Conclusion

The main objective of developing the innovative laboratory was to teach discrete and programmable design techniques with a single board. The board contains four independent CPLDs, which can be easily programmed as standard 74xxx devices, using library items built into design software; these can then be quickly interconnected on the board using ribbon cables.

Practical experience and feedback from EE students show that replacing standard SSI/MSI boards with these CPLD boards would not be a good solution from the pedagogical point of view, so discrete logic boards should still be in use in introductory laboratories.

Having only a single semester to teach fundamentals of digital design, a common situation in the case of CS students [4], at least two or three initial designs should be implemented using SSI/MSI boards to establish an understanding of the principles, before switching to CPLD boards where much design work is done "behind the scenes."

Allowing students to define their own design problems, to form groups freely, and to choose their design approach (schematic, ABEL, Verilog, VHDL) themselves has resulted in very satisfactory learning outcomes. The innovative boards are expandable, flexible, allow for practice of modular design skills and can be used in more advanced courses, such as Computer Architecture.

REFERENCES

[1] J. Cassel. (2010, June). Market Watch: A Forgettable Year for Memory Chip Makers. [Online]. Available: http://www.isuppli.com/marketwatch/Pages/De-fault.aspx

[2] Y. Zhu, T. Weng and C.-K. Cheng, "Enhancing Learning Effectiveness in Digital Design Courses Through the Use of Programmable Logic Boards," *IEEE Trans. Edu.*, vol. 52, no. 1, pp. 151–156, Feb. 2009.

[3] R. J. Kapadia, "Teaching and Learning Styles in Engineering Education," in *Proc. 38. ASEE/IEEE Frontiers in Education Conf.*, Saratoga Springs, NY, 2008, pp. T4B1–

T4B4.

[4] J. N. Amaral, P. Berube and P. Mehta, “Teaching Digital Design to Computing Science Students in a Single Academic Term,” *IEEE Trans. Edu.*, vol. 48, no. 1, pp. 127–132, Feb. 2005.

[5] J. P. Oliver and F. Haim, “Lab at Home: Hardware Kits for a Digital Design Lab,” *IEEE Trans. Edu.*, vol. 52, no. 1, pp. 46–51, Feb. 2009.

[6] K. Nickels, “What are the “Fundamentals” of Modern Digital Logic Design?,” in *Proc. 2005 ASEE Gulf-Southwest Annu. Conf.,* Texas A&M University, College Station, TX.

[7] K. Nickels, F. Aminian and J. P. Giolma, “Bridging the gap between discrete and programmable logic in introductory digital logic laboratories,” in *Proc. 2001 American Society for Engineering Education Annu. Conf. & Expo.*, New Mexico.

[8] Terasic Technologies Inc. (2010, July). Altera DE2 - Development and Education Board. [Online]. Available: http://www.terasic.com.tw/cgi-bin/page/archive.pl?Language=English&No=30

[9] Lattice Semiconductor Corp. (2010, July). ispLEVER Classic Software. [Online]. Available: http://www.latticesemi.com/products/designsoftware/isplever/ispleverclassic/index.cfm

[10] Lattice Semiconductor Corp. (2010, July). Programmable Logic Market Share. [Online]. Available: http://www.latticesemi.com/corporate/about/marketposition.cfm?source=sidebar

[11] K. Nickels, “Pros and Cons of replacing discrete logic with programmable logic in introductory digital logic courses,” in *Proc. American Society for Engineering Education Annu. Conf.*, St. Louis, 2000.

[12] European Commission, Education & Training (2011, January). ECTS User's Guide. [Online]. Available: http://ec.europa.eu/education/lifelong-learning-policy/doc/ects/guide_en.pdf

[13] K. Nickels, "Do's and Don'ts of Introducing Active Learning Techniques," in *Proc. American Society for Engineering Education Annu. Conf.*, St. Louis, 2000.

[14] R. M. Felder and L. K. Silverman, "Learning and Teaching Styles in Engineering Education," *J. Eng. Educ.*, vol. 78, no. 7, pp. 674–681, 1988.

Piotr Dębiec received his M.S. and Ph.D. degrees in Electronics Engineering from the Faculty of Electrical, Electronic, Computer and Control Engineering (EECCE), Technical University of Lodz (TUL), Poland, in 1989 and 1999, respectively.

He is currently an Assistant Professor at TUL. His teaching interests include digital systems, hardware description languages, computer networks, and databases. His research interests include cellular neural networks, image processing, digital design, and integration of academic information systems.

Marcin Byczuk received his M.S. and Ph.D. degrees in Electronics Engineering from the Faculty of Electrical, Electronic, Computer and Control Engineering (EECCE), Technical University of Lodz (TUL), Poland, in 2002 and 2007, respectively.

He is currently an Assistant Professor at TUL. His teaching interests include digital systems, digital signal processing, low level programming and code optimization. His research interests include human-computer communication, mixed signal systems, real-time systems and audio processing.