

A Portable Lab for the Practical Study of Modern Computer Engineering

Tiago Dias^{*†‡}, Pedro Sampaio^{*} and Pedro Miguens Matutino^{*†}

^{*}Instituto Superior de Engenharia de Lisboa – ISEL, Instituto Politécnico de Lisboa – IPL

Rua Conselheiro Emídio Navarro 1, 1959-007 Lisbon, Portugal

[†]INESC-ID, Rua Alves Redol 9, 1000-029 Lisbon, Portugal

[‡]POLITEC&ID, Estrada de Benfica 529, 1549-020 Lisbon, Portugal

Email: tiago.dias@isel.pt, pedro.sampaio@isel.pt and pedro.miguens@isel.pt

Abstract—This demo paper presents the μ LIC educational platform. μ LIC makes available to the students of Digital Systems and Computer Architecture courses a quite simple, intuitive and portable hardware platform for the realization of their laboratory exercises. μ LIC can also be used in other related courses, such as Embedded Systems or Hardware/Software Co-Design, which allows reducing the time needed to study the manuals of diverse development boards and tools and to concentrate the education towards the core contents of the courses. In addition, the low cost, diminished size, and portable nature of the μ LIC boards enables students to carry a personal unit with them all the time to implement the class exercises also outside the classrooms, as well as hobby projects. In the demo, three typical lab assignments of Digital Systems, Computer Architecture, and Hardware/Software Co-Design courses are used to showcase the μ LIC educational platform: a traffic light controller, a 4-bit Arithmetic and Logic Unit (ALU) and a mini Space Invaders inspired game, respectively.

Index Terms—Computer engineering education, digital systems, educational technology, and programmable logic.

I. INTRODUCTION

Digital Systems and Computer Architecture are still two foundational courses in the curricula of modern Computer Engineering undergraduate courses [1], as well as in the curricula of Electrical Engineering [2] and of most Computer Science BSc degrees [3]. Such courses lead the students from the fundamentals of data representation and number systems, through Boolean algebra, basic logic gates, combinational and sequential logic circuits, and memories, to the basic architecture of a microprocessor and to its assembly language programming. This provides the background into the operation of a typical computer system [4], as well as for other relevant topics for computer engineers, such as Operating Systems, Embedded Systems or Hardware/Software Co-Design.

However, it is a well-known problem that students have difficulty to apprehend many of the underlying concepts, since most undergraduate beginners do not have the necessary logic and formal reasoning to understand the involved levels of abstraction [5]. This not only makes harder the teaching and learning process but also has a great impact in student retention and student motivation for further working in these fields.

To overcome these issues, several different teaching methods and pedagogical strategies have been proposed [6]. These

include the classical paper-and-pencil exercise driven experience, as well as various hands-on approaches based on extensive laboratory practice, such as the design of different complexity hardware circuits and its implementation using either low density logic devices on breadboards (e.g. Transistor-Transistor Logic (TTL) logic chips), simulators or using Hardware Description Languages (HDLs) and Programmable Logic Devices (PLDs), such as Programmable Array Logics (PALs), Complex Programmable Logic Devices (CPLDs) or Field-Programmable Gate Arrays (FPGAs). In fact, this hands-on approach is regarded by almost all modern pedagogical studies as the optimal strategy to support great learning achievement [5]–[7]. Nevertheless, teaching hardware using extensive hands-on experience relies on the availability of suitable hardware platforms and design tools to support the involved laboratory activities.

Nowadays, several programmable logic development kits from different vendors are available on the market. These kits include development boards with FPGAs and/or CPLDs, as well as a set of the most commonly used peripherals, e.g. Light Emitting Diodes (LEDs), seven-segment LED displays, Dual In-line Package (DIP) switches, push-buttons, etc. Moreover, they provide comprehensive documentation and the necessary software tools to support all the stages of the design and implementation process. However, not many of these kits are suitable for higher education for several reasons.

On the one hand, many development kits include state-of-the-art PLDs to provide increased design flexibility and support its use in several different courses. Having a universal platform for the laboratory exercises of multiple courses is a quite desirable feature [8], [9]. This way, students not only can save the time and effort of getting acquainted with the plethora of hardware platforms and software tools used in such courses but also concentrate on the fundamental principles that constitute the main contributions of such courses [10]. However, the use of state-of-the-art PLDs is not trivial and poses several difficulties to the students in the laboratory, especially at the beginning of their courses. In addition, the cost of these kits is usually high, which prevents schools from buying them in great quantities to properly equip their

laboratories. This also prevents students from buying the educational kits or borrowing them from the school, in order to conduct their hands-on laboratory activities outside the classrooms [11].

On the other hand, low-cost development kits seldom provide all the hardware resources required to design useful and motivating lab assignments, since they feature low logic capacity PLDs and very few peripherals. To make matters worse, these low-cost kits offer none or very limited expansion capabilities.

In any case, the vast majority of the existing development kits are based on FPGAs, which many authors consider to be inappropriate for the initial courses of an undergraduate degree, for three main reasons. Firstly, FPGAs exhibit an intrinsically high complexity. Secondly, the high logic capacity of FPGAs does not require students to take a modular design approach. Finally, the use of FPGA development boards is not regarded as a purely hands-on experience because it does not involve wiring. Conversely, CPLDs are much simpler and cheaper PLDs. Moreover, they can be easily used with a classic breadboard to construct modular demonstrative circuits, which studies show students feel are satisfying and engaging [12].

This paper presents μ LIC [13], an educational platform that was developed at Instituto Superior de Engenharia de Lisboa (ISEL), the school of engineering of the IPL - Politécnico de Lisboa in Portugal, to jointly tackle all these issues. This demo showcases its usage with three exercises: the implementation of a traffic light controller, the design of an ALU, and the realization of a mini Space Invaders inspired game. These examples are taken out of lab assignments of two different courses taught at the BSc in Computer Science and Computer Engineering at ISEL, namely Logic and Digital Systems and Informatics and Computer Laboratory.

The rest of the paper is organized as follows. Section II introduces the μ LIC educational platform, while section III presents the software tools that are used to implement, test and deploy the designs in the CPLD. Then, the demo exercises are described in section IV. Finally, section V presents some concluding remarks.

II. THE μ LIC EDUCATIONAL PLATFORM

μ LIC is an educational platform based on a Xilinx CPLD that is intended to be used in multiple courses in the curricula of Computer Engineering undergraduate courses, such as Digital Systems, Computer Architecture, Embedded Systems and Hardware/Software Co-Design. The hardware component consists of the μ LIC χ development board and four expansion boards, as shown in Fig. 1. The platform also provides a custom device manager library that allows easy access to the user designs loaded into the CPLD from an external device, such as a Personal Computer (PC) or microcontroller. All these aspects are detailed in the following sections.

A. The μ LIC χ development board

μ LIC χ is a complete USB-powered hardware platform that enables the development and prototyping of low and medium

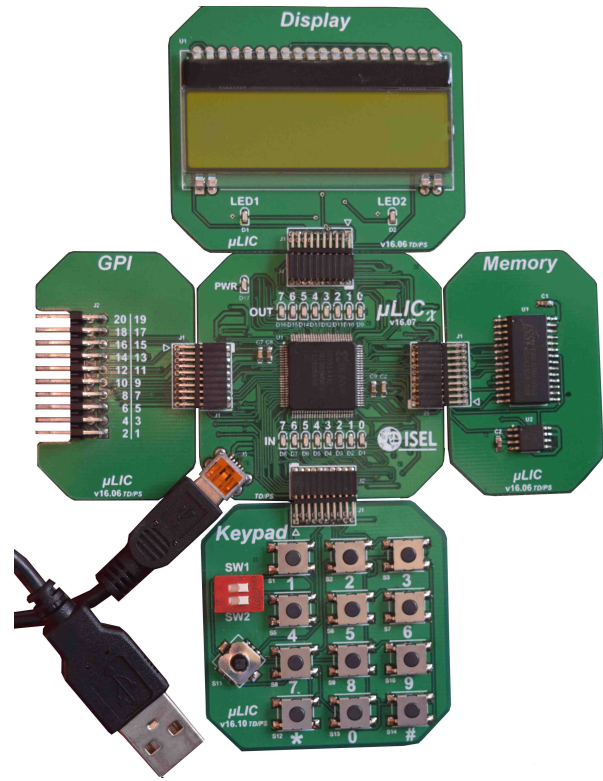


Fig. 1: The μ LIC hardware component.

complexity designs using a Xilinx XC95144XL CPLD [14], as depicted in Fig. 2. Therefore, it is quite suitable to support teaching courses in digital and computer system design.

In choosing the CPLD, we were most concerned with three issues. First of all, we wanted students to become familiar with CPLDs and Electronic Design Automation (EDA) tools adopted by the industry, so that they can take advantage of such knowledge later in their work after graduation.

Then, we wanted that the CPLD could be used with a conventional breadboard kit using simple wiring, in order to students feel a real hands-on experience when implementing their designs. In addition, this helps the students to get handy with good practice of testing and debugging.

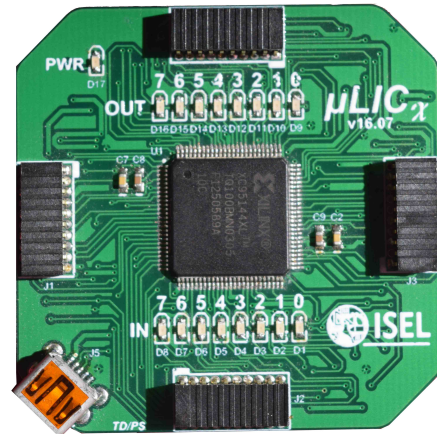
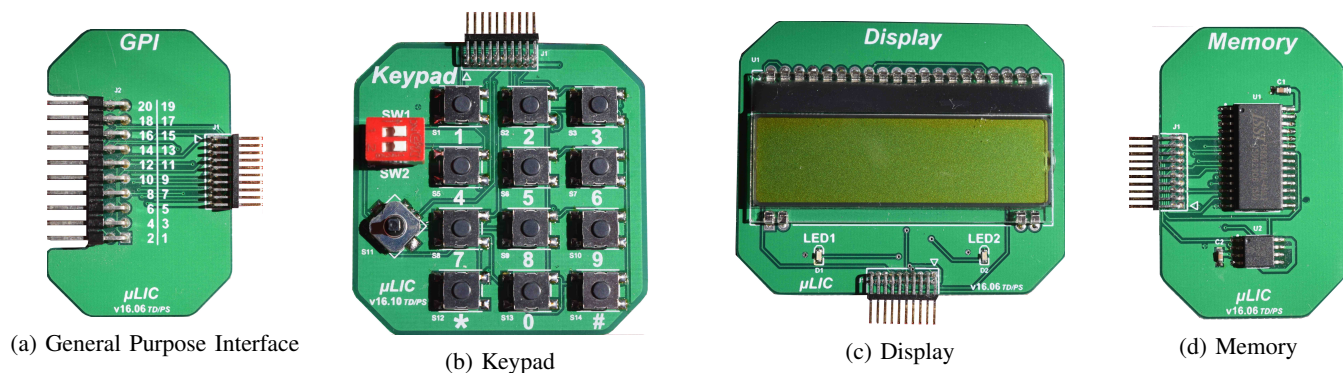


Fig. 2: The μ LIC χ development board.

Fig. 3: The μ LIC expansion boards.

Finally, we wanted the ability to use the CPLD in conjunction with TTL compatible parts, which use 5 Volt power and 5 Volt signals. This way, lab activities can also make use of standard off-the-shelf peripherals (e.g. Liquid Crystal Display (LCD), I2C/SPI sensors, audio codecs or VGA encoders, as well as the 74-series Integrated Circuits (ICs) that are still available at many university educational labs.

Xilinx is the leading developer of CPLD/FPGA, offers commercial EDA tools for free to university students, including a logic simulator, and makes available plenty hardware libraries and documentation. Furthermore, its XC95144XL CPLD is suitable for designing medium capacity digital systems and includes a single Input/Output (I/O) bank that supports both the 3.3 Volt Low Voltage CMOS (LVCMOS) and the Low Voltage TTL (LVTTL) standards. Therefore, all the I/Os are fully 5 Volt tolerant even though the core power supply is 3.3 Volt.

Besides the CPLD, the μ LIC χ development board features on-board user-settable clocking circuitry, 16 LEDs, and a user-defined push-button with no board-specific function. Nevertheless, it can be used as the default reset for the CPLD logic, due to being an input to the XC95144XL CPLD GSR pin. The board also provides four general purpose I/O mezzanine connectors with 2×10 5 V tolerant pins to interface with external functions or devices, such as low-density logic devices (e.g. TTL logic chips on breadboards).

μ LIC χ also makes available two 8-bit parallel ports to support data transfers between the user designs loaded into the CPLD and external devices, such as microcontrollers or a PC. Both ports are implemented using a FT4232H IC from the vendor FTDI [15], which operates as a USB to a triple bit-bang channel converter, i.e. two parallel ports plus a JTAG port. The data transfers are accomplished using also the FT4232H IC, which integrates a USB 2.0 PHY and is capable of implementing the entire USB protocol, together with the onboard USB type-B connector, and a standard USB cable to connect the μ LIC χ to the external device.

The same components are also used for the CPLD in-system programming. In this case, a standard USB cable connects the USB port on the board to a USB port of a PC running the Elementary CPLD USB-Programmer (eCUP) software,

presented in section III. The Joint Test Action Group (JTAG) chain is normally mastered by the FTDI FT4232H IC.

With this set of peripherals, a quite reduced dimension, $60\text{ mm} \times 60\text{ mm}$, and being USB powered and in-system programmable, μ LIC χ can be used as a portable hardware platform.

B. Expansion boards

The functionality of the μ LIC χ development board can be extended by using the four expansion boards depicted in Fig. 3. Each one of these boards makes available a different set of peripherals and can be connected to any of the four mezzanine connectors available on μ LIC χ . It is through these mezzanine connectors that the CPLD on the μ LIC χ board communicates with the expansion boards.

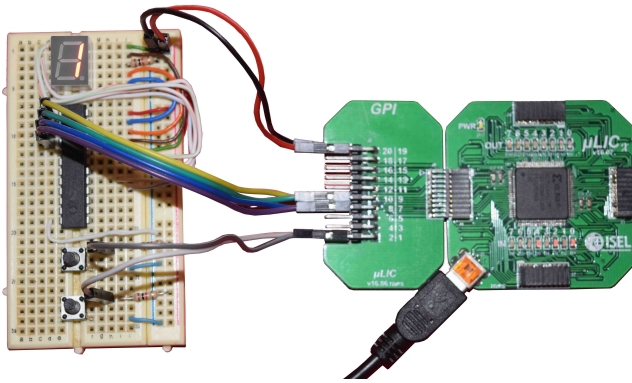
1) *General Purpose Interface (GPI)*: This expansion board provides open access to 17 general-purpose pins on the mezzanine connection to the CPLD on the μ LIC χ board. In addition, it makes available the on-board clock signal, VCC and Ground on another three pins.

The GPI board is useful for circuit debugging, practicing basic electronics and connecting other parts directly to the CPLD, as shown in Fig. 4. Since the pins connect directly to the device, it is responsibility of the user to adhere to the electrical standards of the CPLD.

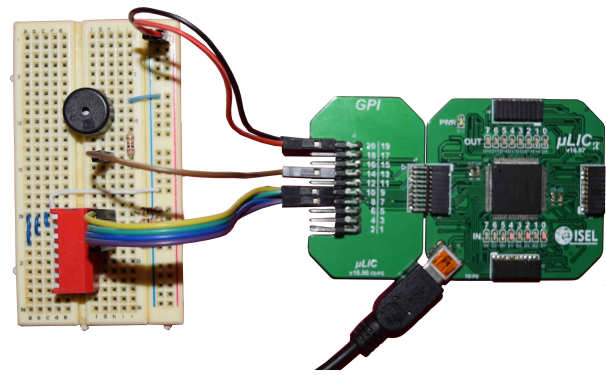
2) *Keypad*: The Keypad expansion board includes a DIP switch with two positions, a 2-axis joystick and 12 push-buttons, which can be used individually or as a 3×4 keypad. The pins of these peripherals are directly connected to 10 general-purpose pins of the CPLD on the μ LIC χ board through the mezzanine connection.

3) *Display*: The Display expansion board provides a convenient interface to two LEDs and a 2×16 character LCD with LED backlight. The LCD controller can be configured to implement the standard 8-bit and 4-bit parallel interface protocols [16], as well as the four wire Serial Peripheral Interface (SPI) protocol [17]. Such configuration is set using the CPLD pins.

Although this board can be useful for circuit debugging, it is mostly intended to implement user interfaces, either alone or combined with the Keypad expansion board.



(a) 7-segment LED display, TTL compatible IC, and push-buttons.



(b) Buzzer and DIP switches.

Fig. 4: Using the GPI to connect low-density logic devices and peripherals to the μLIC_X development board.

4) *Memory*: This expansion board also provides a convenient interface to the CPLD but for two different memory modules: a serial EEPROM [18] and a static RAM [19].

The EEPROM has a capacity of 1024 Kbit and supports byte-level and page-level functions. Moreover, it features Page, Sector and Chip erase functions, typically associated with Flash-based products. It is accessed via a simple four wire SPI protocol.

The RAM is a very high-speed and low power memory with a capacity of 32 KB. However, a single page of 64×4 -bits is available for the CPLD at a time. Such restriction results from pin count limitations on the mezzanine connection and on the CPLD device, whose I/O pins are shared between the four mezzanine connectors.

C. The μLIC Device Manager Library

The μLIC Device Manager Library provides a Java language interface for user applications running on a PC, or other programmable computing devices, to access the input and output parallel ports of the μLIC_X development board and, thus, exchange data with the designs loaded into the CPLD.

This library consists of a very simple Application Programming Interface (API) based on a single class (`PIOPort` – Parallel Input/Output Port) with only two static methods, as shown in Listing 1, which were built and archived into a Java ARchive (JAR) file.

The `in()` method reads back the levels of the 8 bits of the input port, while the `out()` method sets the levels of the 8 bits of the output port to `value`. Listing 2 presents a very simple Java program that exemplifies the use of this API.

The μLIC Device Manager Library was implemented for both the Windows, Linux, and Mac operating systems by using the device manager provided by the vendor FTDI [15] and the Java Native Interface framework [20], as depicted in Fig. 5.

```
int PIOPort.in();
void PIOPort.out(int value);
```

Listing 1: The μLIC device manager library API.

```
import isel.leic.pioport.*;

public class Example {

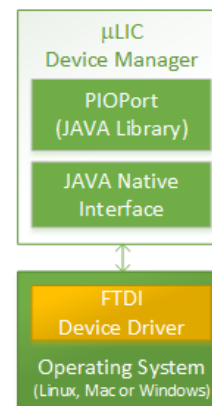
    public static final char HIGH = 1;
    public static final char LOW = 0;

    public static final char LED_PIN = 2;
    public static final char PB_PIN = 6;

    public static void main(String[] args){
        while (true) {
            if ((PIOPort.in() >> PB_PIN) & 1) == HIGH){
                PIOPort.out(HIGH << LED_PIN);
            } else {
                PIOPort.out(LOW << LED_PIN);
            }
        }
    }
}
```

Listing 2: Program that turns a LED on/off by pressing/releasing a push-button.

The beneficial features of the μLIC Device Manager Library include board ease of use, shortened development time, and communication hardware and protocols abstraction. In general, this library makes it easier for students to get their Java programs up and running in the shortest length of time.

Fig. 5: μLIC Device Manager Library software organization.

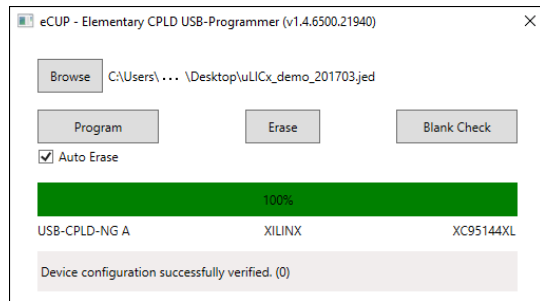


Fig. 6: Main view of the eCUP software.

III. DEVELOPMENT TOOLS

The software development tools for the $\mu\text{LIC}\chi$ development board are of two kinds: a commercial EDA tool from Xilinx and the eCUP academic software.

The Xilinx ISE Design Suite v14.7 WebPACK Edition is used to conduct all the stages of the development cycle of a digital circuit. Using this freeware tool, logic circuits can be described both with a schematic or a HDL, such as Verilog or Very High Speed Integrated Circuit HDL (VHDL). Moreover, such descriptions can be simulated or compiled to produce JEDEC image files that are used to program the XC95144XL CPLD. In fact, one important advantage of this EDA software is the possibility to freely perform simulation before the actual hardware implementation, by using the ISim simulator.

The eCUP software is used to conduct the CPLD programming task. This tool was developed in-house for two specific reasons. On the one hand, the high cost of the JTAG programming cables from Xilinx. On the other hand, the need to properly command the on-board FTDI USB converter (see section II-A) so that the JTAG chain is properly mastered and the CPLD correctly programmed. Using this alternative way of programming, configuration files can be transferred to the CPLD on the $\mu\text{LIC}\chi$ development board at any time using a standard USB cable and the eCUP software with significant savings to the users.

Figure 6 depicts the main view of the free eCUP academic software.

IV. DEMONSTRATION

The purpose of the demo is to show the advantages of using the $\mu\text{LIC}\chi$ educational platform to conduct some laboratory activities in computer engineering, computer science and electrical engineering undergraduate courses. In particular, the platform is used to showcase some typical lab assignments of three different courses, namely Digital Systems, Computer Architecture, and Hardware/Software Co-Design.

A. Digital systems courses

Digital Systems is an introductory course in most BSc degrees in computer engineering, computer science and electrical engineering. Typically, this course covers the fundamentals of digital logic, such as Boolean algebra, and the theoretical aspects of digital design, including combinatorial and sequential logic, standard modules such as decoders and multiplexers,

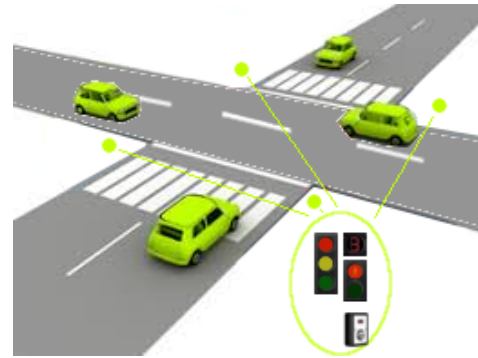


Fig. 7: Street intersection traffic/pedestrian lights mapping.

and system design. Laboratory sessions are an integral part of these courses, being structured in several lab assignments or activities that help the students assimilate the course topics.

This demo addresses the design and implementation of algorithmic state machines [21], one of the lab assignments that has been found to be mandatory in all the evaluated courses. In particular, the $\mu\text{LIC}\chi$ development board is used to implement a traffic light controller simulating the typical operation that can be found at a street intersection between a main road and a side road, as shown in Fig. 7.

In this 4-way intersection, with an east-west road crossing a north-south road, there is a red-amber-green traffic light for each of the four directions. In the east-west direction, there are also two pedestrian crossings with two pedestrian walk lights. Besides the red-green lights, these pedestrian walk lights include one pedestrian button and one countdown timer. The pedestrian buttons are simulated by push-buttons, while the countdown timers are implemented using a seven segment display.

In this demo, eight LEDs of the $\mu\text{LIC}\chi$ development board are used to show the state of all the traffic lights. The GPI expansion board is used to connect the two push-buttons and one seven segment display to the traffic light controller implemented in the CPLD installed in the $\mu\text{LIC}\chi$ development board. To demonstrate the ability to use the $\mu\text{LIC}\chi$ development board in conjunction with TTL parts, an IC is used to perform the Binary-coded decimal (BCD) to seven segment decoding. This setup is depicted in Fig. 4a.

B. Computer architecture courses

Computer architecture is another important introductory course in the aforementioned undergraduate programs. Among other topics, this course usually introduces the principles of computer organization and Central Processing Unit (CPU) design, the basic architecture concepts, such as instruction set design, and the memory and I/O subsystems, together with assembly language programming. Due to the involved levels of abstraction, lab work is also a fundamental part of the apprenticeship. In this demo, the $\mu\text{LIC}\chi$ development board is used also to accomplish one of the most common lab assignments in computer architecture courses (and also in some digital systems courses): the design of a basic ALU.

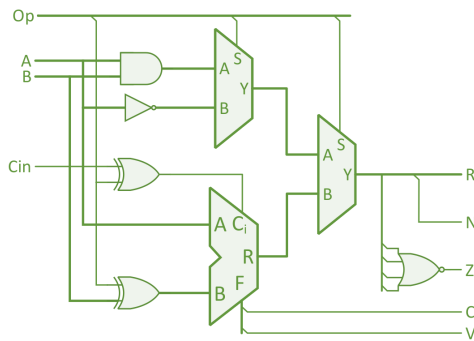


Fig. 8: Block diagram of the showcased ALU.

The ALU is one of the essential parts of a CPU and consists of a combinatorial digital circuit that performs arithmetic and bitwise operations on integer binary numbers. The inputs to the ALU are the data to be operated on, named operands, and a code indicating the operation to be performed. The outputs of the ALU are the result of the performed operation and the related status data. This data conveys information that can be used to assess the correctness of the computed results or to make decisions.

The showcased ALU operates on two 4-bit operands (A and B) and implements four different operations, as shown in Fig. 8: addition, subtraction, bitwise AND and bitwise NOT. The operation to be performed is determined by a 2-bit binary code (Op). Besides the 4-bit result (R), the ALU also generates four active high status bits to state whether such result is zero (Z) or negative (N), if an arithmetic carry/borrow was generated (C), or if an arithmetic overflow has occurred (V).

The ALU demo uses the 16 LEDs of the $\mu\text{LIC}\chi$ development board to show the binary codes of both the two operands and the operation result, and the value of the four status bits. Since the $\mu\text{LIC}\chi$ development board includes a single push-button, the GPI expansion board is also used in this demo to connect a set of DIP switches to the CPLD. The switches are used to determine the value of the two operands and the code of the operation to be performed.

C. Hardware/software co-design courses

Hardware/software co-design investigates the concurrent design, or co-design, of tightly coupled hardware and software components of an embedded system [22]. The main goal is to exploit the synergy of hardware and software, in order to optimize and/or satisfy design constraints such as cost, performance, and energy consumption.

The complexity of the involved topics difficulties the teaching and learning process and often has a great impact in student participation. Hence, courses addressing these concepts and techniques involve extensive laboratory practice and are project-oriented to support greater learning achievement. Also, most projects consist of real-world problems that instructors try to make challenging and fun for the students, in order to keep them engaged and motivated. As such, the project component of these courses quite often consists in designing video games [23]. In this demo, the complete $\mu\text{LIC}\chi$ educa-

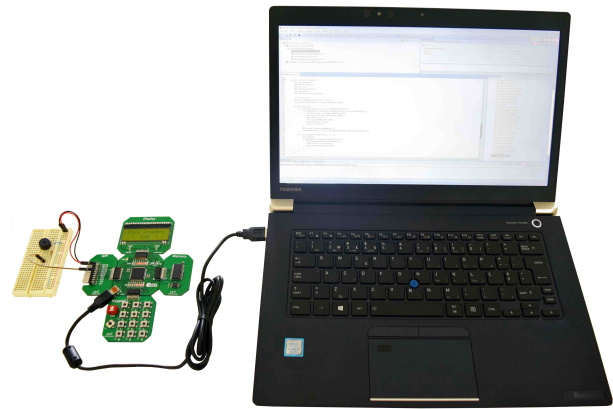
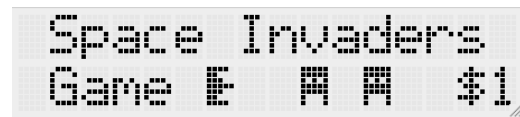


Fig. 9: Setup of the mini Space Invaders game.



(a) Welcome screen.



(b) Horde of invaders.

Fig. 10: LCD messages of the mini Space Invaders game.

tional platform is showcased using one of such game-inspired projects: mini Space Invaders.

Mini Space Invaders is a shooting game inspired in the 1978 arcade game created by Tomohiro Nishikado [24]. In this game, the aliens are represented by numbers in the range 0 to 9 and the horde of invaders consists of a string of numbers that moves horizontally, from right to left, in a LCD. The player aims the laser cannon to intercept the closest approaching alien by using a keypad to choose a number in the range 0 to 9. The player defeats such alien and earns points by shooting the laser cannon when properly aimed. As more aliens are defeated, the speed of the aliens' movement is increased. Defeating all the aliens on the LCD brings another wave that is more difficult, a loop which can continue endlessly. The alien invasion is declared successful and the game ends whenever the aliens reach the players laser cannon.

In this demo, the game logic is implemented by a Java program running in a PC that uses the μLIC Device Manager library to interact with the game console. The game console is implemented using the $\mu\text{LIC}\chi$ development board and all its expansion boards, as shown in Fig. 9. As it can be seen in Fig. 10, the LCD in the Display board presents the game evolution and some status information, such as the player's running score and number of lives. The Keypad and Memory boards implement the required 11 keys keypad with hardware buffer. Game sounds are generated using an external buzzer, which is connected to the CPLD on the $\mu\text{LIC}\chi$ development board using the GPI board.

V. CONCLUSIONS

This demo paper showcases the μ LIC educational platform. The platform comprehends a development board based on a Xilinx TTL-tolerant CPLD, named μ LIC χ , and four expansion boards that provide the most commonly used peripherals when teaching digital circuits design. The estimated manufacturing cost of μ LIC χ is 20 Euros, which makes it a very low cost PLD board. The cost of its four expansion boards is also very low, having been estimated in 18 Euros. The μ LIC educational platform also includes a freeware commercial EDA, a custom CPLD programmer, and a device manager library that allows easy access to the user designs loaded into the CPLD from an external device. By using free software and hardware boards that are very low cost, present diminished sizes and are completely USB-powered, the μ LIC educational platform can be used as a portable lab infrastructure to support the lab work of multiple courses in the curricula of BSc degrees in computer engineering, computer science and electrical engineering, both inside the classrooms and at home.

μ LIC has been used with great success since 2017 in the Informatics and Computer Laboratory course [25], which is taught at ISEL in the second curricular semester of the BSc degree in Computer Science and Computer Engineering. The feedback from the students is quite encouraging, showing positive support for the platform. Furthermore, the student behavior in the laboratory and the grades distribution revealed that the majority of the students met the learning outcomes of the course more easily and consolidated the fundamental concepts underlying the design of modern digital systems by using the μ LIC educational platform.

The future work is focused on two main goals. On the one hand, the definition of the best strategy to incorporate the teaching of VHDL in a course named Logic and Digital Systems, which is the introductory digital systems course taught in the first semester of the aforementioned BSc degree program. This must be carefully planned due to the complexity of VHDL and the fact that undergraduate students are known to have difficulty in handling its abstractions [26]. On the other hand, the design of the complete set of laboratory exercises for the Logic and Digital Systems and Computer Architecture courses, taught also in the second semester of the same undergraduate program. This will allow establishing the μ LIC educational platform as the unified lab platform for all the courses in the Hardware and Computer Architecture scientific domain.

ACKNOWLEDGMENTS

This work was supported by national funds through FCT, Fundação para a Ciência e a Tecnologia, under project UIDB/50021/2020.

REFERENCES

- [1] T. J. T. F. on Computer Engineering Curricula. (2016, Dec.) Computer engineering curricula 2016. Association for Computing Machinery (ACM) and IEEE Computer Society. Los Alamitos, CA.
- [2] K. G. Ricks, D. J. Jackson, and W. A. Stapleton, "An embedded systems curriculum based on the IEEE/ACM model curriculum," *IEEE Trans. Educ.*, vol. 51, no. 2, pp. 262–270, May 2008.
- [3] T. J. T. F. on Computer Curricula. (2013, Dec.) Computer science curricula 2013. Association for Computing Machinery (ACM) and IEEE Computer Society. Los Alamitos, CA.
- [4] D. A. Patterson and J. L. Hennessy, *Computer Organization and Design: The Hardware/Software Interface – MIPS Edition*, 5th ed. Massachusetts, USA: Morgan Kaufmann, Sep. 2013.
- [5] 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.
- [6] H. A. Ochoa and M. V. Shirvaikar, "A survey of digital systems curriculum and pedagogy in electrical and computer engineering programs," in *2018 ASEE Gulf-Southwest Section Annual Meeting*, Apr. 2019, pp. 2183–2187.
- [7] Y. Zhu, T. Weng, and C. 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.
- [8] P. Athanas and C. Patterson, "A holistic approach towards a unified CpE laboratory platform," in *2007 IEEE Int. Conf. on Microelectronic Systems Education*, Jun. 2007, pp. 73–74.
- [9] F. J. Quiles, M. Ortiz, M. Brox, C. D. Moreno, J. Hormigo, and J. Vilalba, "UCORE: Reconfigurable platform for educational purposes," in *2010 Int. Conf. on Reconfigurable Computing and FPGAs*, Dec. 2010, pp. 109–114.
- [10] I. Kastelan, N. Teslic, and M. Temerinac, "Unified learning platform for embedded engineering," in *Embedded Engineering Education*. Springer International Publishing, Jan. 2016, pp. 29–43.
- [11] J. P. Oliver, F. Haim, S. Fernandez, J. Rodriguez, and P. Rolando, "Hardware lab at home possible with ultra low cost boards," in *2005 IEEE Int. Conf. on Microelectronic Systems Education*, Jun. 2005, pp. 19–20.
- [12] J. M. Hill and Y. Yu, "The CPLD provides a third option in the introductory logic circuits course," in *2012 ASEE Annual Conference & Exposition*, Jun. 2012.
- [13] T. Dias, *μ LIC χ Development Board – Reference Manual*, https://www.researchgate.net/publication/338749880_uLIC-X_Development_Board_Reference_Manual, ISEL – IPL, Lisbon, Portugal, Mar. 2017.
- [14] *DS056: XC95144XL High Performance CPLD*, Xilinx, Inc., California, USA, Apr. 2007.
- [15] *D2XX Programmer's Guide*, FTDI, Glasgow, UK, Jun. 2019.
- [16] *HD44780U (LCD-II), Dot Matrix Liquid Crystal Display Controller/Driver*, Hitachi America, Ltd, California, USA, Sep. 1999.
- [17] P. Dhaker, "Introduction to SPI interface," *Analog Dialogue*, vol. 52, no. 3, pp. 49–53, Sep. 2018.
- [18] *25LC1024 Datasheet*, Microchip Technology, Inc., Arizona, USA, May 2010.
- [19] *IS62LV256AL Datasheet*, Integrated Silicon Solution, Inc., California, USA, Jan. 2019.
- [20] JavaSoft. (1997, May) Java native interface specification. Sun Microsystems, Inc. California, USA.
- [21] C. R. Clare, *Designing logic systems using state machines*. McGraw-Hill, Jul. 1973.
- [22] J. Teich, "Hardware/software codesign: The past, the present, and predicting the future," *Proc. IEEE*, vol. 100, pp. 1411–1430, May 2012.
- [23] J. R. Chapman and P. J. Rich, "Does educational gamification improve students' motivation? if so, which game elements work best?" *Journal of Education for Business*, vol. 93, pp. 315–322, May 2018.
- [24] Wikipedia contributors, "Space invaders — Wikipedia, the free encyclopedia," https://en.wikipedia.org/w/index.php?title=Space_Invaders&oldid=945288471, 2020, [Online; accessed 16-March-2020].
- [25] P. M. Matutino, T. Dias, and P. Sampaio, "Teaching hardware/software co-design using a project-based learning strategy," to appear in 14th Conf. on Technology, Teaching and Learning of Electronics (TAEe), Porto, Portugal, Jun. 2020.
- [26] M. S. Nixon, "On a programmable approach to introducing digital design," *IEEE Trans. Educ.*, vol. 40, no. 3, pp. 195–206, Aug. 1997.