WSEAS TRANSACTIONS on
ADVANCES in ENGINEERING EDUCATION
Manuscript received Nov. 23, 2007; revised May 21, 2008

A project Based Methodology to Teach a Course on Advanced Digital Systems Design

ANTÓNIO J. ARAÚJO, JOSÉ C. ALVES
Faculdade de Engenharia da Universidade do Porto
Departamento de Engenharia Electrotécnica e de Computadores
Rua Dr. Roberto Frias, 4200 – 465 Porto
PORTUGAL
aja@fe.up.pt, jca@fe.up.pt

Abstract: This paper presents a project based teaching experience in a course on advanced digital systems design, with emphasis on design methodologies and laboratory assignments. The use of FPGA devices as the most suitable implementation technology for teaching digital design is described. The course structure, oriented to the development of real working digital systems, challenges the students and increases their motivation. This improves the learning process and the productivity of classes. The laboratory works are deployed on a FPGA based platform that implements a real-time video processing chain, from a low cost digital camera to a standard desktop monitor. Examples of proposed laboratory projects in a recent course edition are presented and their evaluation results are analysed.

Key–Words: Digital systems design, FPGA, design methodologies, teaching digital systems

1 Introduction

This paper presents a course on advanced digital systems design taught at Faculty of Engineering of the University of Porto (FEUP), Portugal. The course is taught in the 4th year of the Integrated Master's Degree (Bologna-compliant graduation) in Electrical and Computer Engineering (ECE) and it is attended by students of the Telecommunications area, specialization in microelectronics and embedded systems. The course is called Digital Systems Design (DSD), EEC0055 [1], and addresses the front-end design phase of complex digital systems for implementation on microelectronic technologies. This is the first in a stream of two complementary courses on digital design and it is focused on the digital design perspective at the logic and upper abstraction levels. Following this, the course VLSI Circuit Design (EEC0056 [1]), concentrates on the physical design aspects of digital systems in current silicon-based Integrated Circuit (IC) fabrication technologies.

Student participation is widely recognized as being a key point in effective learning. It is also known that the traditional lectures can be augmented with learning experiences that involve and challenge the students [2, 3, 4]. This is particularly true in technological areas, where a key aspect for a successful learning is the contact of the students with laboratory equipments. In a laboratory, students can interact with the results of their assignments, diagnose failures that might occur and apply their skills in an effective manner. By this way, a laboratory is a fundamental framework to provide the future designers with their professional competence. Important professional skills can be acquired, e.g. teamwork, creativity, learning from mistakes and the ability to design using specific methodologies [5].

The technology and Electronic Design Automation (EDA) tools employed by industry in the design of digital hardware evolve rapidly. Well-prepared engineers, who are able to produce actual designs and adapt to this dynamic world, are in demand. Thus, the training of a digital hardware engineer should be more focused on acquiring designing skills rather than on learning a specific language or software tool. However, proposals such as [6, 7, 8] are too focused on the details of a particular Hardware Description Language (HDL) and do not answer to the necessities of the industry [9]. Nowadays, many proposals are being made to try to reduce the gap between the educational community and industry [10, 11, 12, 13]. To learn digital design, students should practice with design methodologies and tools as close as possible to those used in industry.

This paper describes the teaching methodology used in the DSD course at FEUP, that is focused on the development of real working digital systems. Motivation of the students increase if they can see, touch and play with the final results of their projects, going

far beyond computer simulations and model abstractions. This aspect has a key role contributing to an effective strategy of active learning.

After these considerations the remaining sections are organized as follows. Section 2 discusses key aspects about alternative implementation technologies that are more appropriate to teach an advanced digital systems course. Section 3 presents an overview about the main courses taught currently at FEUP on the area of digital systems. Section 4 describes the goals of the DSD course, the current course programme, the teaching methodology and the laboratory assignments. Section 5 describes a laboratory infrastructure used by the students to experiment their projects and presents some recently implemented projects. The student's grades obtained in the final project are also summarized. Finally, section 6 draws the main conclusions that can be extracted from the main issues discussed in the paper.

2 Considerations about implementation technologies

To practice the complete *design-verification-physical implementation* design cycle for a complex digital system within a 14-week course, the typical duration of a semester in terms of given classes, it is necessary to adopt a convenient implementation technology that allow the students to see and experiment the results of their projects in the laboratory. Although FEUP, as a member of the European EUROPRACTICE programme [14], has easy access to the fabrication of integrated circuits prototypes, the typical 2-3 month turnaround time invalidates the utilisation of this solution as the target technology. Besides, the relative high cost of IC fabrication represents a huge limitation for practical design iterations, which is inevitable to happen during the learning process.

Field-Programmable Gate Array (FPGA) is now a mature technology for the implementation of digital systems [15, 16, 17]. These are completely fabricated commercial ICs that can be configured an unlimited number of times by the end user in a matter of seconds using low cost equipment. It allows the rapid implementation of multi-million gate-equivalent digital systems with no real costs of silicon fabrication, providing high-performance digital systems competing, in same cases, with custom made ICs. Although the emphasis of the DSD course is not FPGA-targeted design, this technology was adopted in the course as the implementation media for the laboratory projects due to various reasons: fast design cycle with no costs of design iterations, support for high-performance and complex digital systems and availability of a large variety of commercial development platforms that include various peripherals and interfaces (e.g. RAM, analogue-digital signal converters, network interfaces, etc.). According to published experiences [18, 19, 20, 21, 22, 23], many other universities around the world have adopted FPGA circuits to teach their digital systems design courses.

Before the introduction of this course in 1998, advanced digital design was taught in a course focused on Application Specific Integrated Circuit (ASIC) design. However, the relatively long turnaround time of ASICs did not allow the experimentation of the systems developed during the course. FPGAs shorten drastically the design-implementation cycle and are thus a convenient technology for teaching digital systems design in a practical perspective.

By the given reasons and considering the context of the DSD course in the ECE graduation programme, the FPGA technology was chosen to implement the main laboratory projects.

3 The area of digital systems at FEUP

The courses currently offered at FEUP by the ECE programme in the area of electronic digital systems assure the coverage of the different phases of designing digital systems. Using a Y-chart [24], figure 1 illustrates the distribution of the subjects that represent the central themes taught in those main courses.

The design at the system level is addressed in courses as Digital Signal Processing, Principles of Telecommunications (3rd year), Telecommunication Systems and Image Processing (4th year). These courses manipulate abstract models and algorithms in the areas of corresponding applications. Emphasis is on the algorithmic level rather than on real implementations. Laboratory works are based mainly on computer simulations and pre-assembled experiments. The contents of these courses allows to place them in the Y-chart over the axis that represents the behavioural domain and in the highest abstraction level, the system level.

Another group of courses address topics on conventional processors and computers: Microprocessors (basic architecture, low level programming), Microprocessors Based Systems (embedded systems with microprocessors, peripherals, interfacing) and Computer Architectures (study of conventional computers). These can be placed in the Y-chart along the structural domain, at the system level and partially on the RTL.

The design at the physical level is done in the courses of Electronics III and VLSI Circuit Design.

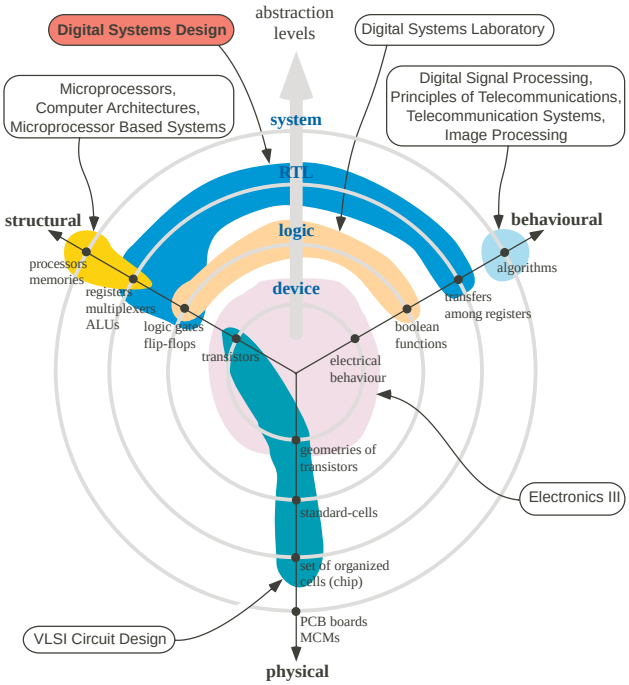

Figure 1: Courses on digital systems area at FEUP.

The first, Electronics III, covers the fundamental aspects of the production technology of integrated circuits with emphasis on the CMOS process. The operation, implementation and characterization of fundamental logical blocks are studied. This course can be located on the three representation domains in the lowest abstraction level, the physical domain. On the other hand, VLSI Circuit Design focuses the design of integrated digital circuits according to all of the levels of the physical domain, covering all of the steps of the *back-end* design phase, targeting an ASIC in CMOS process.

Digital Systems Laboratory is an introductory course on logic design. It is located on the 1st semester and its main purpose is to introduce fundamental concepts that are required by posterior courses on the area. Theoretical principles and practical aspects of the analysis and synthesis of digital systems are taught. This can be located in the Y-chart over the logic level, covering essentially the structural and behavioural domains.

The design at the RTL level is covered in the course of Digital Systems Design. The main topics are digital system modelling at the RTL level, with standard hardware description languages, functional validation and RTL synthesis. It is also included a reduced component of physical design, that is fundamental to perform the back-end phase for the FPGA. It is still established the bridge for the design at the system level, with the study of mechanisms used in the high-level synthesis process and of problems associated to the construction of complete datapaths and arithmetic operators. This way, the focus of the course locates on the RTL level, from the behavioural to structural axes.

The depicted areas in figure 1, associated to the nuclear areas focused by each course, cover in a complete way all of the representation domains and abstraction levels, although with reduced, but necessary, overlapped areas among some courses. For example, the design task at the logic and RTL levels are both covered by the courses of Digital Systems Laboratory and Digital Systems Design, although with different depths and design perspectives.

4 The DSD course

The aim of DSD course is to supply the students with technological skills and methodological aspects necessary to the design of complex digital systems, considering their implementation in microelectronic technologies or in reconfigurable digital systems. For that, it is intended to teach them the necessary competences to design digital systems using a structured approach. The learning process is based on the development of digital hardware projects at different abstraction levels, using HDLs and industry grade design tools.

Next section details the design process of digital hardware. The course programme is introduced in section 4.2, the teaching methodology is described in section 4.3 and the laboratory assignments are presented in section 4.4.

4.1 The design process

The digital systems design process is centred in three main tasks: modelling, validation and synthesis. These tasks can occur at different abstraction levels and representation domains, depending on the complexity of the project. Figure 2 represents a simplified design flow diagram illustrating the relationship among them.

At the system level, the starting point is generally the study and evaluation of an algorithm to be used in the system under design. Behavioural models are used to choose and evaluate a resolution method for the problem. The modelling and validation process is based on high-level modelling and simulation tools, e.g. Matlab/Simulink and programming languages. The synthesis of a model at the system level consists on translating it to a representation in a lower abstraction level, i.e. the RTL level, based on blocks that implement functional elements at this level.

Designing at the RTL level consists on building and validating a RTL model, in a behavioural or struc-

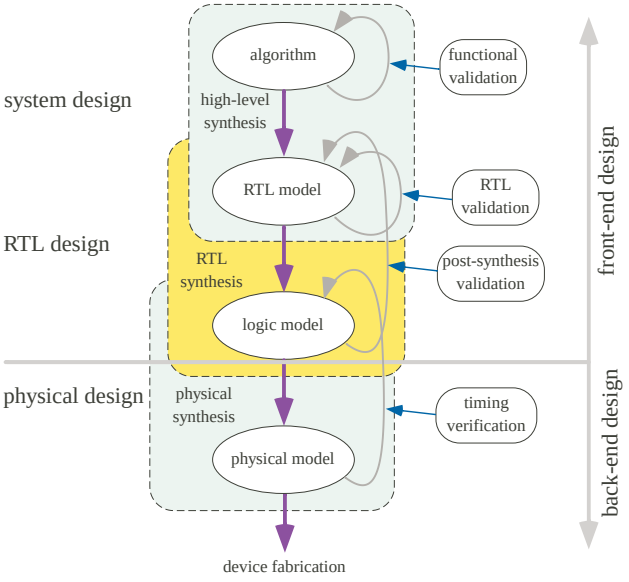

Figure 2: Digital design tasks, abstraction levels and representation domains.

tural mode, and translating it to a structural representation at the logic level through automatic synthesis. This constitutes an important task when designing complex digital systems and is supported by simulation and synthesis software tools that are essential in this activity. The result of this process is a model that represents a netlist of available logic elements of the target technology, each one corresponding to a physical cell on that technology. This phase and the previous one complete the *front-end* design phase where an algorithm is converted into a digital system. The DSD course acts mainly at this level.

The translation of that structural model into a physical representation is usually known as the back-end and is specific of the target technology. It involves the spatial organization of blocks over the physical area of the target device, the placement and routing of cells, and the extraction of physical characteristics.

4.2 Course programme

The course covers the following main topics:

- Structured design of digital systems: design flow, top-down and bottom-up design methodologies, modelling at different abstraction stages, modularity and hierarchy concepts;
- Hardware description languages: modelling, verification and synthesis perspectives, Verilog HDL;
- Design verification: functional validation, logical verification and temporal analysis, and techniques for writing testbenches;
- Synthesis of digital systems: RTL and behavioural synthesis, and RTL synthesis under temporal and physical constraints;
- Synchronous digital systems: clock generation, management and distribution;
- Reconfigurable implementation technologies: FPGAs and combined FPGA/microprocessor systems;
- Control and datapath synthesis: synthesis of control units and custom processing units, architectures of both fixed-point and floating-point arithmetic operators;
- Low power: design principles and techniques oriented for the reduction of power consumption in digital circuits.

As far as possible, these topics are maintained independent of the implementation technology.

4.3 Teaching methodology

The topics in the DSD programme are taught based on a set of slides presented during the lectures. The teaching materials are made available to the students before the start of the course, allowing them to make better use of in-class time. Whenever opportune, exercises are solved to exemplify or clarify some particular aspects. The evaluation of the knowledge acquired by the students on the theoretical classes is done by two mini-exams, accounting for 40% of the final grade.

The practical classes are used to present some tutorials about development tools and, mainly, to develop laboratory projects that exercise the design methodologies and concepts presented in the theoretical classes. The laboratory projects are detailed in the next section and contribute with 60% for the final grade. This reveals the strong component of the hands-on practice in te course.

By the reasons described in Section 2, the laboratory assignments are addressed to use reconfigurable circuits to implement the developed digital systems. Presently, Xilinx FPGAs are used for that purpose with the ISE development tools [25] and the HDL used is Verilog [26]. The design approach consists on using the top-down methodology and performing the entire design cycle, i. e., functional verification, automatic synthesis and implementation, on a Xilinx

Spartan-3 FPGA contained in a Digilent [27] prototyping board.

The first stage of the design cycle, the functional verification, is essential for a successful design. Wrong models that not satisfy the project specifications can not result in correct hardware solutions. As part of the teaching methodology, the students are alerted to this fact and to the importance of the functional verification of a digital system as a way to guarantee its correctness. To put this in practice, the students are encouraged to exchange their verification models among them. This is a very useful practice to avoid incomplete design verification due to a wrong or biased interpretation of the specifications. If the model under checking is wrong, due to a mistake on the interpretation of the project specification, then the testbench also tends to be wrong.

Assessment is most commonly associated with testing whether students have achieved the learning outcomes of the course. However, used imaginatively, assessment techniques can be used as a part of the learning experience to help foster a deeper engagement with the material under study. The assignments of the DSD course are done with groups of two students. It is known [2, 28] that the teamwork-based approach is very advantageous. This enables students to experiment and acquire the skills that they will need in their future jobs. Some of these important skills are interpersonal communication, problem-solving, leadership, negotiation and time management. Consequently, teamwork used in a context of active methodologies provides an useful and more significant learning, with positive effects on the academic performance of students, motivation and their attitudes towards learning [29, 30]. Some of these advantages have also been underscored by the student's majority, who consider group activities and active methodologies to be more interesting, productive and easier to learn than traditional teaching.

The organization and the contents of the laboratory assignments are described following.

4.4 Laboratory assignments

The DSD course has a strong hands-on practical component based on three main laboratory assignments. The practical classes are planned to allow the execution of these projects in articulation with the theoretical classes. By this way the background topics are guaranteed. This set of three assignments allows the students to progressively contact with the design flow and methodologies, as well as the corresponding tools, by using an incremental approach as will be shown later.

The weight of the laboratory assignments on the final grade are summarized in table 1. The effort is an estimate of the time required to complete the project, including the time during classes and the work extra classes.

Laboratory	1	2	3
Duration (weeks)	2	3	8
Effort (hours)	7	15	40
Weight (%)	10	10	40

Table 1: Duration and evaluation of the laboratory projects.

Following sections detail the laboratory projects, presenting their objectives and their corresponding pedagogical contributions.

4.4.1 Lab 1 – Modelling and functional verification

The first project face the students with modelling and functional verification of digital systems. It is expected that the students familiarize with the Verilog HDL and the ModelSim simulator. For this purpose typically a control-dominated synchronous digital system is designed. The main issues addressed on this assignment are the correct understanding of RTL modelling of both sequential and combinational logic circuits, building basic testbenches with automatic verification procedures, practice with the front-end design stage through the design cycle and verify the project against specifications.

This first assignment is initiated after a tutorial about the digital simulator ModelSim. The specification is given to the students in the class and in the same class they should answer to a short questionnaire. This intends to motivate them to clarify what and how they should do. At the end of this session the students have detailed the organization of their systems and can then start with the HDL modelling phase.

4.4.2 Lab 2 – Synthesis and implementation

The second project is intended to obtain a circuit, implementing and experimenting it in a FPGA. For the first time the students go through all the design flow stages: modelling, functional validation, synthesis, implementation and final experimentation. A basic ISE project is supplied with the necessary resources to implement it in the Spartan-3 prototyping board. The proposed project consists in a well defined block that has to be integrated in the top level design. With this approach, it is intended that the students concentrate on the circuit to be developed, once this second

project still usually reveals some flaws of the students at the modelling level. The reason for that is the improper use of HDL constructions that not synthesize properly. This is one of the main difficulties when students start working with a HDL, because they tend to confuse a HDL with a programming language. Several authors relate this problem [31].

This approach still allows the students to understand the overall organization of a complete ISE project and learn how to define design constraints, like assigning signals to the input/output pins of the FPGA circuit, and setting the maximum delay for some critical signals. Additionally, the students familiarize with the remaining design tools.

As occurred in the previous project, this project starts after a tutorial about the ISE package.

4.4.3 Lab 3 – Final project

The two previous laboratory assignments assume an introductory role in the use of the design tools and give emphasis on the taught design methodologies. The third project is a more complex project where the students have to conceive and develop core parts of a given design, exploring and evaluating alternatives to meet realistic design constraints and optimize some quality criteria. In the past few years several different projects were proposed, spanning diverse application areas such as audio coding and processing, software radio, processor design and real-time video processing. Although the proposed projects are simplified to be compatible with the student's skills and the available time, they result in interesting demonstrators and motivate the students into an active learning process.

The design of a concrete application and the used methodologies and tools, challenges the students and involve them in a working environment close to what they will find in industry.

5 Developed projects

This section describes the laboratory infrastructure built for the DSD course. It has been used since two years ago by the students to experiment their projects. An overview of some recently implemented projects is also included. The section ends presenting the evaluation parameters used in the third project.

5.1 Laboratory infrastructure

This infrastructure was developed and implemented two years ago and since then it is being used successfully. It consists on a video processing chain that receives data from a OmniVision 7120 [32] monochromatic camera, perform some image processing operation and supplies the resultant signals to the board that interfaces with the VGA monitor. This basic real-time video processing system supports 30 frames per second, uses 8 bits per pixel and is built on a XILINX Spartan-3 FPGA board. The FPGA implements the video processing system and additional control and interfacing modules. This includes a RS-232 connection with a PC, an I2C interface with the camera and the generation of the VGA synchronization signals. Figure 3 shows the composition of the infrastructure and the physical resources.

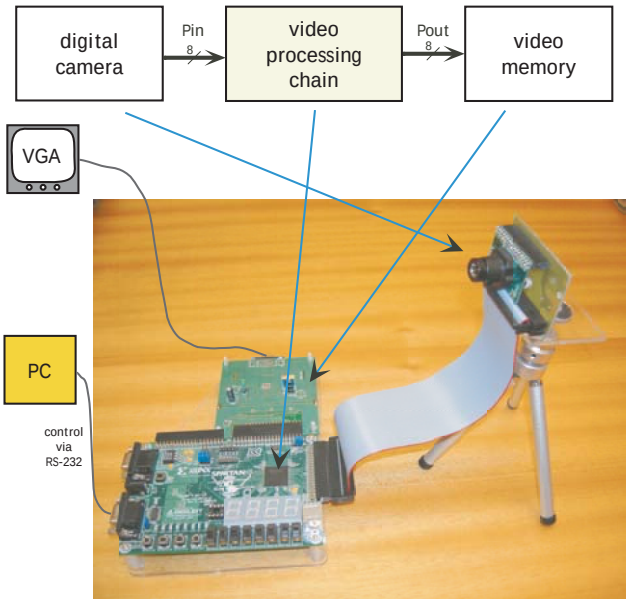

Figure 3: Laboratory infrastructure.

An ISE project with all the blocks that support the implementation of image processing operations is normally given to the students including a module that exemplifies how to integrate in the system the blocks they have to develop. Figure 4 shows a structural view of the given top-level design, with emphasis on the video processing module inserted on the chain.

One of the blocks included in the given ISE design, synchronize the signals coming from the digital camera with the 100 MHz system clock. The pixel values are generated at 12.5 MHz, and for each one the correspondent coordinates are also generated, considering that $(0, 0)$ corresponds to the upper left corner of the image. Besides that, horizontal and vertical synchronism signals are also produced. The system still includes an interface with the digital camera I2C bus, allowing the configuration of several parameters of the camera by setting proper configuration registers, e.g., it is possible to define the image scan mode, choosing between interlaced or progressive scan, etc.

Various image processing operations, normally known by the 4th year graduation students, can be

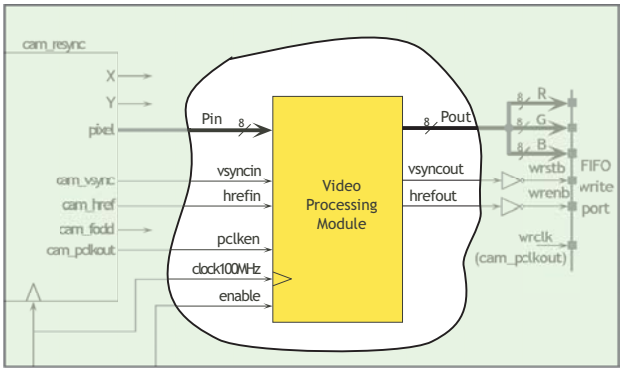

Figure 4: Insertion of a video processing module on the given design.

implemented. For all of them, the implementation methodology consists in defining an architecture that satisfies the required specification, its modelling and functional verification, its integration on the system, and finally the synthesis and implementation of the entire design.

After being processed, the image is sent to an external board that includes a set of dual-port FIFO memories and a high speed DAC. These memories are then read, synchronously with the VGA synchronism signals, and applied to a digital-analogue converter that produces the analogue signal to the VGA monitor.

Complementing this hardware infrastructure, it is also supplied a PC application to interface the FPGA circuit, allowing a bi-directional communication via the RS-232. This tool facilitates the communication with the circuit implemented in the FPGA, either for control and configuration and also, in some cases, for debbuging purposes.

5.2 Examples of proposed projects

Some of the laboratory projects proposed in recent DSD course editions are briefly presented. Excepting the first, the students have to integrate some functions in the video processing system, adding modules to a given ISE project. For that, they practice the whole design flow, where one of the most time-consuming and simultaneously most important tasks is the design validation by functional simulation.

5.2.1 Serial-parallel programmable interface

This is an example of the first laboratory (*Lab1*), dedicated to modelling and verification. It consists in the development of a programmable bidirectional data interface, with 4 input ports and 4 output ports, 16-bits wide. This programmable interface implements a command interpreter allowing to read or write data on any of their IO ports.

The emphasis of this project is on the design of a finite state machine, that controls the data interface, and on the familiarization with Verilog modelling and functional verification.

The resulting module was later joined to an UART unit producing a very useful module, allowing a basic communication via serial port between a PC and a circuit on the FPGA. This is the same module included in the ISE project that implements the basic video processing system. Figure 5 illustrates its composition and usage. This communication module

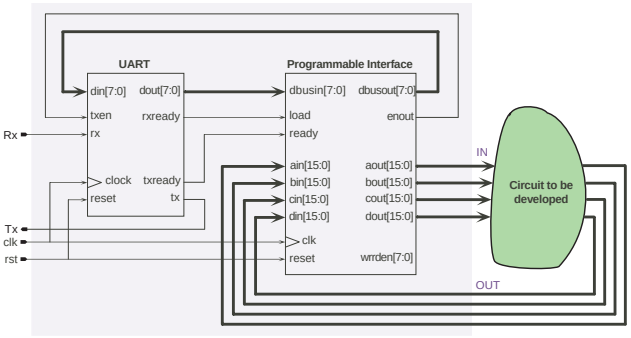

Figure 5: Serial/parallel interface for bi-directional communication between the PC and a circuit in the FPGA.

has been widely used in the students projects due to its simple but effective and flexible usage. In fact, it facilitates data exchange between a PC and a circuit under development. For example, it can be used to supply input data to a circuit under development by writing on a certain output port, or to observe the outputs of that circuit by reading the input ports where it is connected. Besides that, it can be used to debug purposes, helping students to identify and to correct certain failures, increasing this way their productivity. All the projects over the digital camera infrastructure used this data interface.

5.2.2 Brightness control

This project was proposed as a *Lab2* project. It aims to develop a module to control the brightness of an image. To experiment it in the video processing infrastructure it must be added to the base processing chain to then implement it. In spite of the simplicity of this project, that basically deals with signed arithmetic operations with saturation, the emphasis of this project is on the integration of the system and on the understanding of how the entire system operate. These aspects are important, because it is the first time that the

students really implement a circuit in the FPGA.

The brightness of the image can be changed by adding a number to each input pixel. According to the signal of this number, positive or negative, the brightness is increased or decreased, respectively. In both cases the resulting pixel value should saturate on the admissible limits, i. e., 0 (black) and 255 (white). The number used to control the brightness is modified by the end-user by pressing two buttons. One of them causes its increment by 10 units, when it is actuated, and the other one causes the decrement of the same value. In this operation, the saturation of that value must be also considered. It is required that these operations are performed synchronously with the clock of the processing system.

5.2.3 Real-time histogram

This project consisted on the determination and visualization of the real-time histogram correspondent to the captured image. To define the visualization place, the coordinates are defined using the PC communication interface (section 5.1). The counting of the pixel's frequencies on a frame is saved in the FPGA internal memory to allow the visualization of the current histogram within the next frame. Actually, this was the principal challenge to the students as well as the normalization of the histogram to allow a convenient representation scale. This task requires an integer division but the students should solve the problem by implementing a convenient approximation method avoiding the implementation of a divider.

This laboratory project and the next two were proposed as final (*Lab3*) projects.

5.2.4 Motion detection

A basic technique for motion detection is based on the difference of two consecutive video frames. This was applied in a project that consisted in the implementation of a motion detection based on the video image system. It also allows the end user setting threshold levels that establish when it is considered motion. To help the end user having movement perception, the level of movement is presented as a bar over the original image. The size of that bar is proportional to the detected movement. It is also possible to specify in run-time the coordinates of the window where the motion should be detected. That region is highlighted to differentiate it in the whole image on the screen.

5.2.5 Bi-dimensional convolution

A 2D convolution was also implemented in a recent laboratory assignment. A 3×3 matrix was used in the convolution operator. Choosing appropriate values to that 9 coefficients result various filtering effects, e.g. edge detection, blur, emboss, sharpen, etc.

The principal challenges were the arithmetic involved and the pixels storage. In fact, the students needed to worry about the allocation and scheduling of the arithmetic operators. The pixels storage, consisting in saving the line of the current pixel and the previous two, requested to think how to access the eight neighbours of the current pixel from the adopted storage solution. Besides that, the effect of the image borders must be solved. These situations were problematic due to the FPGA available space and system timing constraints to allow the real-time system operation.

5.3 Obtained results

The largest extension and complexity of the last project, *Lab3*, justify its greater weight comparing to the other course assignments. The weight of this project is normally 40% to 50% of the final grade. The evaluation is made considering aspects like the organization and quality of the Verilog code of both developed modules and testbenches, the produced datasheet about the developed system including a brief characterization and the results of the implementation, and obviously considering the fundamental aspect that is the correct functionality of the system. These features were graded in four levels.

A measure commonly used to quantify the size of a hardware implementation is the gate count of the circuit. A circuit characterized by occupying n equivalent gates means that is a circuit with a size similar to n 2-input NAND logic gates. The optimization of the system implementation reached by the student teams, measured by the gate count, is also valued serving to distinguish them. The following equation was used to do that.

$$\left(1 - \frac{GC - GC_{min}}{GC_{max} - GC_{min}}\right) \times 5\%$$

GC is the gate count obtained by a team, GC_{min} is the minimum gate count among all the design teams, i. e., the best design solution in terms of this feature, and GC_{max} is the maximum gate count reached by a design team. By this way, considering they have time do to that, the teams can really compete among them. In spite of the low grade of this evaluation component, the students exhibit a competitive attitude that is stimulated by that factor.

Table 2 presents the obtained results on the third laboratory project (*Lab3*) in the last DSD occurrence. The mean results were obtained considering the four-level grading concerning the enumerated aspects of each student's project. The average class evaluation

	Weight (%)	Results (%)
Organization	15	78
Implementation	60	68
Datasheet	20	59
Gate-count	5	49

Table 2: Evaluation results of *Lab3* in 2007/08.

was about 65%. In spite of being a medium result, due to the imposed demand, we understood that most of the students reached a good level of maturity relatively to the methodologies and taught techniques.

6 Conclusion

The project based teaching methodology presented on this paper introduces an effective way of teaching advanced digital systems design to students in electrical and computer engineering, opening new directions in engineering education.

FPGAs are currently the main implementation technology of digital systems when a fast design cycle is required. This paper has presented its utilisation in the laboratory assignments of a digital systems design course in the ECE graduation programme at FEUP. The fast design cycle of this technology and the possibility to reconfigure the circuits allow the students to perform unlimited design iterations without additional costs. By this way they can easily explore the solutions space of a digital system implementation considering the consumed resources and its performance. These features improve the productivity of the classes and the learning process results more appellative and attractive.

This paper presented a project driven digital systems design course considering the implementation of such projects in FPGA devices. Its main goal is to answer to the necessities of the industry, which demands engineers trained in the art of digital design. Thus, the course focuses on teaching design methodologies instead on teaching specific languages and tools.

The teaching methodology and the laboratory projects of the course were presented. Each assignment given to students has a clear objective because of the limited time budget. Industry-grade tools are used so that the students work in an environment as similar as possible to the one in industry. With this approach, the students acquire valuable knowledge and skills on an actual technology that assumes a role with growing importance in the development of digital hardware. Besides that, FPGA based prototyping impels a design approach based on HDLs that favors the design portability to other target technologies.

References:

[1] `http://www.fe.up.pt`.

[2] Juan A. Marin-Garcia and Jaime Lloret. Improving teamwork with university engineering students. The effect of an assessment method to prevent shirking. *WSEAS Transactions on Advances in Engineering Education*, 5(1), January 2008.

[3] Saumil Merchant, Gregory D. Peterson, and Don Bouldin. Improving embedded systems education: Laboratory enhancements using programmable systems on chip. In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, June 2005.

[4] Norio Ishii, Yuri Suzuki, Hironobu Fujiyoshi, Takashi Fujii, and Masanori Kozawa. Designing effective learning environments for creativity. *WSEAS Transactions on Advances in Engineering Education*, 3(10), October 2006.

[5] Lyle D. Feisel and Albert J. Rosa. The role of the laboratory in undergraduate engineering education. *Journal of Engineering Education*, pages 121–130, January 2005.

[6] Zainalabedin Navabi. The impact of hardware description languages on the education of hardware design. In *Proceedings of the 8th IEEE University/Government/Industry Symposium*, pages 235–239, Westborough, MA, USA, June 1989.

[7] Assim Sagahyroon and Mehran Massoumi. On the use of hardware description languages in teaching VLSI design courses. In *Proceedings of the 26th. ASEE/IEEE Frontiers in Education Conference*, pages 713–716, Salt Lake City, UT, USA, November 1996.

[8] Assim Sagahyroon. From AHPL to VHDL: A course in hardware description languages. *IEEE Transactions on Education*, 43(4):449–454, November 2000.

[9] Don M. Gruenbacher. An active learning/teamwork approach to implementing an integrated circuit design cycle in an advanced hardware description language course. In *Proceedings of the 29th. ASEE/IEEE Frontiers in Education Conference*, San Juan, Puerto Rico, November 1999.

[10] José N. Amaral, Paul Berube, and Paras Mehta. Teaching digital design to computing science students in a single academic term. *IEEE Transactions on Education*, 48(1):127–132, February 2005.

[11] Valery Sklyarov and Iouliia Skliarova. Teaching reconfigurable systems: methods, tools, tutorials and projects. *IEEE Transactions on Education*, 48(2):290–300, May 2005.

[12] Hugo Hedberg, Joachim N. Rodrigues, Fredrik Kristensen, Henrik Svensson, Matthias Kamuf, and Viktor Owall. Teaching digital ASIC design to students with heterogeneous previous knowledge. In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, pages 15–16, Anaheim, CA, USA, June 2005.

[13] Igone Vélez and Juan F. Sevillano. A course to train digital hardware designers for industry. *IEEE Transactions on Education*, 50(3):236–243, August 2007.

[14] `http://www.europractice.com`.

[15] Stephen D. Brown, Robert J. Francis, Jonathan Rose, and Zvonko G. Vranesic. *Field-Programmable Gate Arrays*. Kluwer Academic Publishers, 1992.

[16] R. C. Cofer and Benjamin Harding. *Rapid System Prototyping with FPGAs: Accelerating the Design Process*. Elsevier, 2005.

[17] Esma Alaer, Ali Tangel, and Mehmet Yakut. MIB-16 FPGA based design and implementation of a 16-bit microprocessor for educational use. *WSEAS Transactions on Advances in Engineering Education*, 5(5), May 2008.

[18] Don Bouldin. Impacting education using FPGAs. In *Proceedings of the 18th International Symposium on Parallel and Distributed Processing*, Santa Fe, USA, April 2004.

[19] Stephen A. Edwards. Experiences teaching an FPGA-based embedded systems class. In *Proceedings of the Workshop on Embedded Systems Education (WESE)*, New Jersey, USA, 2005.

[20] Yong-K. Jung. Work in progress – a rapid design methodology for FPGA-based processor platform design education. In *Proceedings of the 35th. ASEE/IEEE Frontiers in Education Conference*, Indianapolis, USA, October 2005.

[21] Dong-Soo Kang, Soo Yun Hwang, Kyoung-Son Jhang, and Kang Yi. A low cost and interactive rapid prototyping platform for digital system design education. In *Proceedings of the 2007 International Conference on Microelectronic Systems Education*, San Diego, CA, USA, June 2007.

[22] Thilo Pionteck. Teaching informatics students the secrets of hardware design. In *Proceedings of the 2007 International Conference on Microelectronic Systems Education*, San Diego, CA, USA, June 2007.

[23] Michael J. Wirthlin. Senior-level embedded system design project using FPGAs. In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, June 2005.

[24] D. Gajski and R. Kuhn. Guest's editors introduction: New VLSI tools. *IEEE Computer*, December 1983.

[25] `http://www.xilinx.com`.

[26] The Institute of Electrical and Electronics Engineers, New York. *IEEE Standard Verilog hardware description language*, September 2001. IEEE Standard 1364-2001.

[27] Xilinx. *Spartan–3 Starter kit board User Guide*, 2005. `http://digilentinc.com/Data/Products/S3BOARD/S3BOARD-rm.pdf`.

[28] Romanas V. Krivickas and Jonas Krivickas. Laboratory instruction in engineering education. *Global Journal of Engineering Education*, 11(2), 2007.

[29] C. W. Holtham, R. R. Melville, and M. S. Sodhi. Designing student groupwork in management education: widening the palette of options. *Journal of Management Education*, 30(6), 2006.

[30] T. Kalliath and M. Laiken. Use of teams in management education. *Journal of Management Education*, 30(6), 2006.

[31] R. James Duckworth. Embedded system design with FPGAs using HDLs (lessons learned and pitfalls to be avoided). In *Proceedings of the 2005 International Conference on Microelectronic Systems Education*, Anaheim, CA, USA, June 2005.

[32] OmniVision Technologies, Inc. OV7120 single-chip CMOS VGA b&w digital camera, 2001. `http://www.ovt.com`.