Learning Hardware Design by implementing student’s Video-Game on a FPGA

M. Sanchez-Elez¹ and S. Roman¹
¹Dept. Arquitectura de Computadores y Automatica, Universidad Complutense de Madrid, Spain

Abstract - This paper presents a methodology to teach the design of complex hardware systems to Computer Engineering students through CAD tools which allow a rapid prototyping and manufacturing of complex systems. The design of such systems requires a broad scope of knowledge and abilities, both for the students and for the teachers. Challenging each student to implement his/her favorite video-game on a FPGA may be an effective method to help them to acquire the necessary skills to design hardware systems with CAD tools. Each student chooses his/her own video-game as the final course project. We have observed that this enhances the motivation required to face the difficulties of handling complex design concepts together with the use of any CAD tools. This yields a more satisfactory experience for students and an increase in the skills gained. An improvement in academic results is also observed.

Keywords: Logic Design, Hardware Description Language, Reconfigurable Hardware, Education

1 Introduction

The students who are following any of the different computer grades must deal with computer systems from design to implementation, as they will design computing systems and computing components of products, develop and test their prototypes, and implement them to market. We agree with other [1], [2] and [3] who state that these courses are very important for undergraduate curricula in Computer Science and Electrical and Computer Engineering. These graduates need to manage concepts from many different fields.

The goal is that the students may correctly design a complex algorithm into hardware with the help of any of the existing computer-aided design tools [4], [5], [6] and [7]. This means, they should be able to develop the skills that allow an engineer to rapidly generate hardware description language (HDL) that a CAD tool synthesizes into the desired hardware, as pointed by [8].

The students have to deal with parallel execution, signal delay, synchronization, etc, and we have observed that it is tough for them to assimilate these concepts. An additional difficulty is that they need to learn a Hardware Description Language in order to simulate and synthesize their designs. On top of this, the synthesis process is platform dependent.

This paper describes an experiment carried out along several years where we investigate if the development of a complex video-game project, as an alternative to practical examples in the form of guided lab exercises, may be successfully used to teach hardware design. We do not create a video-game focused on explain hardware design concepts, we suggest to students to design a video game, one that they like best.

Our experience is that from the students’ point of view, guided lab exercises seem to be a just a black box with data to collect, which is an approach that is far from real design problems. We suggest that a project based learning (PBL) methodology is more suitable for these courses as it creates a course dynamics that is closer to a professional environment [9].

This paper presents the PBL methodology we use in our courses to fulfill both technical and motivational objectives. The implementation by the students of a complex video-game in hardware not only meets technical design goals, but also highly motivates them in comparison to other type of projects, which is a key aspect in the learning process [10]. Designing and implementing their own video-game as the course project is a key aspect of the methodology, as the students become highly motivated, and therefore more involved and more autonomous in the HW design process. The creation of their own video-game is a challenge related to a reality that is well known for them compared to other possible hardware projects. Most students find it a little hard in the beginning because they have to handle real problems present in hardware design. As their motivation and self-confidence grow along the course, only a minority give up. By the time they are designing their own video-game they become very involved in the course and work really hard. They are autonomous enough to be searching cores that improve their original designs or to be sophisticating the functionalities of the original proposed video-game. This means that the development of their own video-game also helps develop a stimulating and important non-specific cross-skill for engineers as is creativity.

The rest of this paper is organized as follows: Section 2 outlines the educational difficulties present in System Design together with different approaches already published. Section 3 describes the methodology implemented. Section 4
presents details of the last guided laboratory exercise in which students have to make use of all the previous concepts, before implementing their own video game. Section 5 shows experimental results obtained from students grades statistics and several questionnaires.

2 Related Work

In this section the challenges of applying PBL in hardware system design are discussed and related both to already developed educational practices as well as to the knowledge and skills that the EDA industry expects from graduates.

EDA industry has evolved towards the use of CAD tools which allow a rapid prototyping and manufacturing of very complex designs. Hardware design with present software tools do not require such a deep understanding of internal hardware functionalities as it was before [11]. The actual EDA industry requires the management of a wide knowledge in many new areas such as, Hardware Design Languages (HDLs), FPGA-based CAD tools, use of IP cores, soft cores, advanced synchronization techniques, design of virtual systems, etc. [12].

The students who follow these subjects are on their final course, that is, they are about to graduate and start working. Hence, a PBL methodology is convenient for the course development as this is close enough to a real EDA project. The development of a video-game as the contents of the PBL approach is technically as complex as other PBL proposals in this field.

There are several papers [13], [14] which use games as part of the engineering education. These proposals replace the traditional lecture-based passive learning methodology with an active multi-sensory experiential learning methodology. We would like to emphasize that this is not what we are suggesting in this paper. The authors mentioned are using a game as a simulation environment, while we are focusing on the hardware implementation of a video game in a FPGA, in the place of traditional circuit design laboratory exercises.

Most published works about learning methodologies using game projects are inspired by software courses [15], [16]. In these works games are integrated in three ways: traditional exercises can be replaced by games motivating the students to put extra effort in doing the exercises [17], [18]; games can be used within a traditional classroom lecture to improve the participation through knowledge-based multiplayer games [19]; or game development projects can be used in computer science or software engineering courses to learn specific software skills [20].

The last paper mentioned presents a methodology that is focused on the need to engage students in the course, which is also our purpose in this work, as well as to help students face the real problems of a software complex project. Developing a complex game design as a useful educative tool is quite clear for a software design course, but it is not so obvious for hardware design courses. Therefore the convenience of a paper such as this, in which we show that complex video-game developing is as useful in hardware design courses as it is for software design courses.

Some author who also deal with the challenges of teaching hardware design today still hold a traditional point of view of hardware design which they apply to a new hardware target (FPGA), such as [21], or develop complex concepts by using an FPGA-based tool as the main support[22]. In addition, most digital design texts dealing with HDLs [23], [24] present a classical view of digital design concepts merely translated into HDL. They commonly propose the structural design of a simple microcontroller as final project, which adds no significant understanding to the former schematic approach.

The proposal of [25] is also interesting, although these authors make separate simulation and synthesis projects. These are complementary aspects of a whole design process and should be analyzed together from the beginning of the course, since a correct simulation does not guarantee a correct implementation on the target FPGA.

Other authors have shifted from the traditional methods and dedicate half of the semester to implementing a complex final project. Such is the case of [12, 26], in which a video-game is a possible choice for the students. The main difference with the methodology presented in this paper is that in our case the design of their own video-game is the baseline of the course development, being in the mind of the students that the final goal is to create it, which helps maintain an interest and deep understanding from the beginning until the end.

The work presented in [27] uses PBL in a very attractive way, although they use VIP technology, which requires sophisticated laboratory equipment.
Some authors use video-games as the baseline of the course [28], [29] and [30], but in these cases the video-game is not the project itself, it is a tool created by the teachers to help students acquire concepts. These games follow the drill-and-practice model, emphasizing rote memorization and failing to capture the interest of students [31]. The most common approach implies that problems are embedded in a game narrative, and solving them is part of the game [32]. They have turned the whole learning setting into a game, on the idea that playing and solving problems share many learning features. However, in the methodology explained in this paper the problem to be solved is implementing the game itself, which has to be developed by the students.

3 Methodology

The methodology assumes that the students are already familiar with: laws of Boolean algebra; binary number representation; basic combinational logic design, basic gates, multi-level digital design; combinational building blocks; basic sequential building block and Finite State Machine (FSM) design. A basic knowledge of VHDL is also desirable, so a 4 - 6 hours VHDL tutorial [33] is given to the students, regardless of whether or not they have studied it in previous courses.

The key focus of the methodology is on creating the expectation that at the end of the course, once they have mastered certain design challenges, the student are ready to implement their favorite console video-game on a FPGA. All the design created along the course may be used to create the video-game. The project is developed using VHDL as design language, Spartan 3 [34] as prototyping platform and Xilinx ISE [4] as CAD tool.

The syllabus of the course is the following:

2. FPGAs blocks and VHDL:
   a. Configurable Logic Block Architecture
   b. Other configurable HW: Block RAMs
   c. Spartan 3 Family
   d. Prototyping boards and the Xilinx ISE Design Suite.
3. Advanced VHDL
   a. Register Transfer (RT) level specification of digital systems using VHDL.
   b. Logic - RT level design techniques.
   c. Logic - RT level analysis of digital systems.
   d. Synthesis for FPGAs.
4. I/O Interfaces
   a. The XESS boards [35]
   b. PS2 Key-board hardware
   c. VGA controller
   d. Music generation

The Fig. 1 describes in a week scheme the syllabus of the course.

The cornerstones of this methodology are the following:

1. At the beginning of the course the instructors present an overall guideline of the theoretical concepts required for the implementation of different hardware modules, all of which may be combined to create any classical video-game.
2. The different hardware modules that the students develop have been specifically designed to present increasing challenges in order to capacitate them to finally implement their own video-game. These modules are based on the develop of different Algorithmic State Machine (ASM):
   a. ASM with counters: the teacher provides a guideline of the VHDL code for a synthesizable ASM and a basic counter. The students have to modify this code to implement a traffic light system.
   b. Use of the FPGA internal memories: the VHDL code for a RAM is given together with the analysis of its synthesis process into a BRAM. Students are requested to reuse this code to implement a FIFO.
   c. Input interfaces (PS2 Keyboard): the teacher gives a theoretical introduction on hardware interfaces and details on the PS2. Students are requested to reuse this code to implement a FIFO.
   d. Output interfaces (VGA): a VHDL VGA controller is given so that students may paint several geometric figures on a PC screen.
   e. Guided implementation of an easy video-game Fig. 2. The students implement an Arkanoid video-game completely guied by...
the instructors.

3. Midterm group discussion: students are requested to suggest a video-game that may be implemented with the hardware modules developed along the course. This should be presented as a realistic project, that is, they have to evaluate the difficulty of the game by identifying the necessary hardware elements, the amount of reusable code and deadlines. In some cases the teacher addresses the students to either simplify or increase the complexity of their projects, in order to meet the educative goals and deadlines of the course.

The implementation of any video-game choses by the students fulfill the common topics in advanced logic design courses: model, simulate, synthesize and optimize complex subsystems, using in most cases a register transfer level courses: model, simulate, synthesize and optimize complex subsystems, using in most cases a register transfer level (RTL). Therefore, for the students who have successfully passed the course the acquired skills are:

1. Correct implementation of flip-flops and registers: when the students attempt to paint different figures and colors on the VGA screen, if their if-else VHDL code is not correct the RGB signal is latched and the colors are not the expected.
2. Correct implementation of a ASM: when the students create the ASM for the moving ball, they learn that the ball does not correctly bounce if the tool has not inferred the ASM associated.
3. Understand the differences between the simulation and the implementation of their designs.
4. Handle several clock domains: the design of the video-game implies using different clock signals for: the VGA controller, the movement of the ball, the video-game controller and the keyboard.
5. I/O management.
6. Use of other partners IP cores: the students feel motivated to enhance their games with new functionalities; as they have to comply with deadlines they are encouraged to make use of existing IP cores such as sound controller, screen backgrounds … and/or reuse modules designed by their classmates.
7. Be aware of the amount of area used by their designs: the students feel inclined to make complex visual designs, which result in a high use of FPGA resources. Thus, the synthesis process becomes slow and in some cases the complete design does not fit the FPGA.
8. Develop self-confidence as hardware designers: once the game is completed, the students themselves realize that they have acquired the skills for hardware design and this motivates them to continue learning.

4 Case Study

The easy video-game implementation for the students as first project in order to train for develop their own video-game is the arkanoid. One of the most successful video-game to be implemented in a FPGA is the classical arkanoid, as it is very popular among the students and therefore the majority of them easily understand its operation. In addition, the game is more sophisticated than a tennis game, which highly motivates students as it poses the challenge to implement a fairly complex video-game.

At this point of the course, the students are familiar with the CAD tool and they have implemented the hardware modules described above (BRAMs, PS2 interface, VGA controller) and the instructors give them seven steps they need to follow to correctly implement this game:

1. Paint the video-game edges on the screen, this means to paint a square.
2. Implement a four-state machine in VHDL to emulate the ball movement. The ball bounces off the edges at a 45º angle. The instructor gives the students a VHDL template to create this ASM (Algorithm 1). PX and PY stand for the ball positions on the screen. Each time that the VGA horizontal and vertical counters equal the ball position, the rgb signal value has to be equal to the color of the ball. BallClock stands for the speed of the ball (this clock is slower than the refresh rate of the screen). However, the clock for the ASM is the FPGA main clock.
3. Create the slide-bar. What size is it? How does it move? The students have to decide how many pixels the bar moves with each keystroke.
4. Modify the original ball-ASM in order to emulate the ball bouncing off the slide-bar.
5. Create a matrix of different color rectangles using a BRAM; this could be done as an extension of the VGA controller lab session.
6. Erase color rectangles. Create a VHDL code that changes the color of a rectangle, stored in a BRAM, to background color each time the ball reaches a rectangle.
7. The ball has to bounce off the rectangles if they are not already deleted. When the ball bounces, the rectangle is erased.

By following these seven steps, the students gradually approach the implementation of the arkanoid video-game in an easy.

The students develop the different hardware components required for the final design by means of progressive laboratory exercises. IP cores are also integrated in the project for some parts of the design. This integration of IP cores and gradually developed parts makes the process accessible to all students in a reasonable lapse of time and is also closer to a real development of a complex commercial system.

For instance, the keyboard controller module is developed during week 7 of the course as a practical exercise which consists of implementing a 22-bit shift register, and a with-select VHDL structure that determines which key has been pressed. The VGA controller is developed during week 8 of
the course. The VHDL code of the controller itself is given to
them and their task is to assign values to the rgb signal in
order to display a relatively complicated geometrical figure
on the screen.

As a result, when students are faced with the challenge

Algorithm 1. VHDL Template for ball movement
to implement the Arkanoid game, they have already
developed the skills and modules they need for the task are
may rely upon these to face new challenges such as the
management of memory contents and its representation on
the screen.

Each rectangle is linked to a different memory word so they
need to learn how translate pixels into memory addresses by
concatenating the x and y parameters of the pixel. By
eliminating the two least significant bits of y and the four
least of x, a single memory address may be linked to a 16 x 4
rectangle. A double port memory is required so that it may be
read both from the VGA controller as from the game
controller.

The game controller also has to translate the ball position into
memory addresses. The ball position (px, py) is used to
calculate if the ball is encountering a rectangle. If the memory
content associated to the rectangle is that of the screen, then
the ball continues its movement. However, if the memory
content is one of the possible colors of the rectangles, the ball
has to bounce and the color stored at that memory location
has to be changed into the screen color (the rectangle
disappears).

Students are taught how this memory functionality may be
implemented with the Block RAM modules in the FPGA with
a read port connected to the rgb signal and another read/write
port associated to the ball movement.

5 Results

We started using this methodology on academic year
2010-2011. The courses are held at Universidad Complutense
de Madrid (www.ucm.es/english). It is an elective subject for
students in their last undergraduate course, with an average of
20 students per class.

<table>
<thead>
<tr>
<th>Grade/Year</th>
<th>2009/10</th>
<th>2010/11</th>
<th>2011/12</th>
<th>2012/13</th>
</tr>
</thead>
<tbody>
<tr>
<td>A+</td>
<td>3%</td>
<td>7%</td>
<td>17%</td>
<td>13%</td>
</tr>
<tr>
<td>A</td>
<td>4%</td>
<td>23%</td>
<td>22%</td>
<td>40%</td>
</tr>
<tr>
<td>B and C</td>
<td>23%</td>
<td>40%</td>
<td>50%</td>
<td>40%</td>
</tr>
<tr>
<td>D</td>
<td>67%</td>
<td>23%</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>E and F</td>
<td>3%</td>
<td>7%</td>
<td>11%</td>
<td>7%</td>
</tr>
</tbody>
</table>

The results obtained by the students are shown in (Table 1).
Column 2009/10 presents the results obtained when the PBL
approach was not centered in the development of a video-
game by the students. The other three columns follow the
PBL approach with a video-game as the target
implementation. Along these last three years we have
observed that, on average, 90% of students successfully
complete the course. It is also remarkable that more than 30% of
the students achieved an A or A+ grade. The good results
obtained are explained by taking into account that each
school year, the instructors have been refining the practical
exercises and the whole methodology. We have also seen
increasing motivation from the beginning of the semester in
the students enrolling this course along these years.
explanation for this is that a majority choose this course because their fellow students, who have already done it, recommend it to them (we have also noticed an increase in the number of students enrolling the course).

Table 2: Results at the University Quality Assurance in Higher Education Program

<table>
<thead>
<tr>
<th>Methodology and Scheduling (max 30)</th>
<th>Score (2010/11)</th>
<th>Score (2012/13)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>24.9</td>
<td>21.8</td>
</tr>
<tr>
<td>Course Development (max 30)</td>
<td>26.4</td>
<td>27.3</td>
</tr>
<tr>
<td>Results (max 40)</td>
<td>35.3</td>
<td>34.2</td>
</tr>
<tr>
<td>Total (max 100)</td>
<td>86.6</td>
<td>83.3</td>
</tr>
</tbody>
</table>

Table 3: Student’s Perceptions and Opinions of the Course

<table>
<thead>
<tr>
<th>Perception</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>I have been pleasantly surprised</td>
<td>33 %</td>
</tr>
<tr>
<td>I liked it</td>
<td>57 %</td>
</tr>
<tr>
<td>I found it boring</td>
<td>10 %</td>
</tr>
<tr>
<td>I feel disappointed</td>
<td>0 %</td>
</tr>
<tr>
<td>Which of these options reflects your perception of the course?</td>
<td></td>
</tr>
<tr>
<td>The organization and development of the practical sessions</td>
<td></td>
</tr>
<tr>
<td>Was interesting and strengthened my interest in HW design</td>
<td>48 %</td>
</tr>
<tr>
<td>Was interesting but required a lot of effort</td>
<td>27 %</td>
</tr>
<tr>
<td>I managed to comply with the course goals by investing a lot of effort</td>
<td>5 %</td>
</tr>
<tr>
<td>I think this methodology demands too much effort</td>
<td>0 %</td>
</tr>
<tr>
<td>My perception about the design skills acquired is that</td>
<td></td>
</tr>
<tr>
<td>I feel capable of designing any other complex system</td>
<td>38 %</td>
</tr>
<tr>
<td>I feel capable of designing systems of a similar nature</td>
<td>52 %</td>
</tr>
<tr>
<td>I feel capable of designing systems of a lower complexity</td>
<td>10 %</td>
</tr>
<tr>
<td>I still do not feel capable of facing the design of a complex system</td>
<td>0 %</td>
</tr>
<tr>
<td>I feel that choosing this course for my curriculum</td>
<td></td>
</tr>
<tr>
<td>Is not significant</td>
<td>0 %</td>
</tr>
<tr>
<td>Was interesting but I do not think will help me find a job</td>
<td>14 %</td>
</tr>
<tr>
<td>Will provide more professional opportunities for me</td>
<td>62 %</td>
</tr>
<tr>
<td>Will help me find the type of job I am interested in</td>
<td>24 %</td>
</tr>
</tbody>
</table>

Moreover, during academic year 2010-2011 and 2012-2013 we enrolled this course in a university program meant to evaluate teaching performance. Within this program, the results of teaching activities are translated into terms of progress in student learning and in the assessment expressed as perceptions or opinions by students, graduates, academic leaders and the academic staff. This evaluation is coordinated within a program of the Spanish Agency in Quality Assurance in Higher Education, ANECA. This agency is a full member of European Association for Quality Assurance in Higher Education (ENQA) and a full member of the International Network for Quality Assurance Agencies in Higher Education (INQAAHE). The results obtained for this course in this program are shown in Table 2. This evaluation has special mention for being among the 5% top rated of Universidad Complutense de Madrid. Additionally, the instructors of the 2011 - 2012 course have carried out a specific test about students’ perceptions and opinions. The results are shown in (Table 3). This test shows that students are satisfied with the course and feel confident on their skills for designing complex hardware, although for some of them the effort required was significant. Most of them realize that designing hardware systems is a realistic professional path to be considered, a perception that is confirmed when they meet colleagues who are already working in companies that use the same CAD tools and similar hardware targets.

Examples of video-games implemented by the students are:

- Pac-Man (Fig. 3)
- Mario Bros
- Tetris (just four figures)
- Piano learning
- Tron video-game
- Flappy Bird
- Space Invaders

Fig. 3. Pac-Man screenshot.

6 Conclusion

The evolution of hardware design processes in EDA industry requires that the former approach for the teaching of hardware design be revised. The use of CAD tools, which are platform dependent, together with the complexity of the hardware design concepts involved make both the teaching and learning quite a hard task.

In this respect we have found many interesting approaches based on the development of a course project similar to real industry projects. However, these types of projects are still too far from the students realities. Implementing a complex video-game on a FPGA triggers students’ motivation. The methodology presented has been designed to channel the students’ energy through a milestones journey which leads to the video-game implementation. At the end of the journey they have acquired the hardware design skills required for engineering industry today. The results of the anonymous
questionnaires from our students support our hypothesis as well.

Moreover, the final outcome is so attractive for them, and the motivation to round off the video-game becomes so strong that many students exceed the undergraduate course goals. This explains why the three years that the video-game has been used as a course project there is an increase in the B, A and A+ marks.

To sum up, it is remarkable that in comparison to the classical course project, the percentage of students that invest the minimum effort to pass the subject (D mark) has decreased and even disappeared. This clearly supports our hypothesis that the motivational aspect of the course project is a key factor in the success of a PBL methodology. Moreover, this methodology is economically sustainable as only a low-priced Xilinx’s Spartan 3 platform is required.

7 References