E-learning Tools and Web-resources for Teaching Reconfigurable Systems

Valery Sklyarov and Iouliia Skliarova
University of Aveiro, Department of Electronics and
Telecommunications/IEETA, 3810-193 Aveiro, Portugal
skl@det.ua.pt, iouliia@det.ua.pt
WWW home page: http://www.ieeta.pt/~skl/
http://www.ieeta.pt/~iouliia/

Abstract. The paper shows that the teaching of reconfigurable systems is a complex and difficult process requiring many novel methods and tools to be considered within the same course. This process can be greatly simplified with the aid of the information resources and e-learning tools that have been developed and incorporated into an integrated methodology that includes original evaluation and motivation methods. This methodology has been successfully applied to the teaching of reconfigurable systems for approximately 10 years. The paper discusses the impact of this methodology on such important components of higher education as research activity, postgraduate learning, and training for engineering work in industry.

1 Introduction

The functionality of a reconfigurable system can be customized on the fly and altered, if required, after the design steps have been completed. This feature opens up a vast range of possibilities for such very useful techniques as the reuse and customization of physical components just through reconfiguration, the adaptation to external conditions that are not known in advance, the fulfillment of particular user requirements post-design, etc. It is also very important that the design of high-speed reconfigurable systems can be done through the use of methods and tools that are very similar to software development taking into account the many specific features inherent in hardware circuits.

When field programmable gate arrays (FPGA) were first introduced, they were predominantly used for implementing simple random and glue logic [1]. Nowadays, even undergraduate students are capable of constructing quite complex FPGA-based

systems. Today, advanced research is being intensively performed in the areas of System on Chip (SoC) and Networks on Chip (NoC) [2]. Let us consider some examples. Recent cheap FPGAs from the Xilinx Spartan-3 family [3] contain up to 5,000,000 system gates, which is sufficient for the construction of quite a complicated special purpose computer. A single FPGA chip can be customized for executing a virtually unlimited number of computationally intensive algorithms, and the same chip can be reused for different algorithms. Most advanced FPGAs contain embedded microprocessors, large-volume memories, circuits for multi-purpose synchronization, multipliers, etc. New sophisticated FPGA architectures appear on the market each year and they are used intensively in industry.

The rapid evolution of reconfigurable systems has created a demand for an increasingly large number of engineers who are well-prepared in the relevant areas. An ongoing review of the corresponding curricula is essential in order to incorporate recent advances and ensure that classes are up to date. The curriculum must be sensitive to changes in technology and new developments in pedagogy and should emphasize the importance of lifelong learning [4-6]. Because the domain of reconfigurable systems is very dynamic and many-sided, many topics from different areas must be considered in detail. For example, the majority of design steps are based on methods inherited from software development. This includes the specification of the required behavior, coping with complexity using different strategies (based on the divide and conquer principle, for example), modeling and debugging, etc. On the other hand it is necessary to provide hardware-specific features, such as concurrency, proper timing characteristics, the synchronization techniques needed for satisfying numerous hardware constraints, etc. There is a wide range of tools used throughout the design process, including the synthesis of finite state machines, the use of libraries and intellectual property cores, and so on. In other words, the design methodology can be seen as a very complicated integrated process that requires a wide spectrum of knowledge from different fields. This paper demonstrates how different information resources and E-learning tools can be applied to increase the efficiency and productiveness of teaching, and ultimately to enable the difficulty and complexity discussed above to be managed effectively.

In general, this paper describes a methodology that has been used for teaching reconfigurable systems at the Department of Electronics and Telecommunications of Aveiro University (Portugal). Some results in this area have already been shown in [7]. In 1997 the first optional discipline in this area entitled *Advanced Digital Systems* was introduced. Today three disciplines are included in the pedagogical plans: *Reconfigurable Computing*, *Reconfigurable Digital Systems*, and *Advanced Reconfigurable Systems*. They are also planned to be taught within the new curriculum according to the Bologna proposal [8].

2 Programs, Theoretical Classes and Labs

The primary objective for the group of courses on reconfigurable systems is to provide experience in the design flow shown in Fig. 1, which is based on either system-level or hardware description languages. Basically the design flow splits into two courses. The first course is dedicated to hardware-oriented tools (**V**ery High Speed Integrated Circuits **H**ardware **D**escription **L**anguage – VHDL [9], in particular) and the subsequent course considers the design flow based on system-

level tools (the Handel-C language [10], in particular). The reasons for choosing these languages are explained in [7].

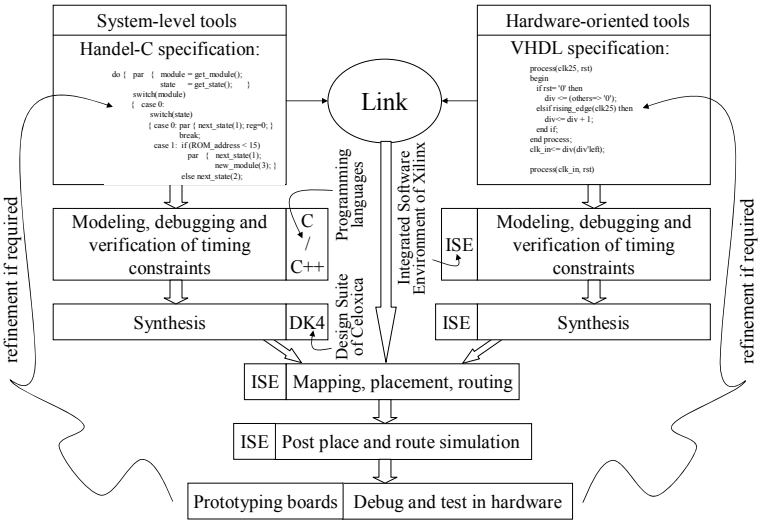

Fig. 1. The design flow for reconfigurable systems

Fig. 2 shows the material that is required as a base for the disciplines on reconfigurable systems (see the left-hand part of Fig. 2) and the knowledge that has to be acquired after completing the two courses (see the right-hand part of Fig. 2). Consequently, the material that is given in the theoretical classes covers the set of topics listed on the right-hand part of Fig. 2. The first course (see the right-hand part of Fig. 1) is given over a 14 week period with 3 hours of theoretical and 2 hours of practical classes per week. The second course (see the left-hand part of Fig. 1) is also given over 14 weeks with 2 hours of theoretical and 2 hours of practical classes per week. Laboratory work is based on a variety of prototyping boards (namely, RC10, RC100, RC200 [10], TE-XC2Se [11], DETIUA-S3 [12]) and one of these (DETIUA-S3) [12] has been developed at the department. After finishing the courses, the students acquire experience in the design of small and medium size reconfigurable systems based on recent commercial FPGAs that are currently recommended for future applications. The majority of the students present a quite complicated mini-project that is used for final evaluation; this will be discussed in more detail in section 4. During laboratory work, the following types of FPGA-based circuits have to be designed (see also [7] for additional details):

- For communications with peripheral devices, such as pushbuttons, dip-switches, LEDs, segment displays, LCD, mouse, keyboard, VGA-monitor, etc.;
- For interfaces with standard microchips, such as static RAM, flash RAM, etc.;
- For communications through standard ports, such as RS232 and parallel;
- For executing arithmetical and logical computations, implementing various algorithms (for example, solving combinatorial problems), etc.

As can be seen from the previous discussion, the material of the group of disciplines considered presumes very intensive study of new methods and tools on the part of the students and hard practical work with digital systems that must be tested and demonstrated in real hardware. In fact, a very large and diverse amount of material needs to be assembled within the courses and this gives rise to a serious problem. Indeed, it is necessary to provide a great deal of new material that is mostly unfamiliar to the students within a limited time period. The subsequent sections of this paper discuss how the information resources and e-learning tools that have been developed make this problem manageable.

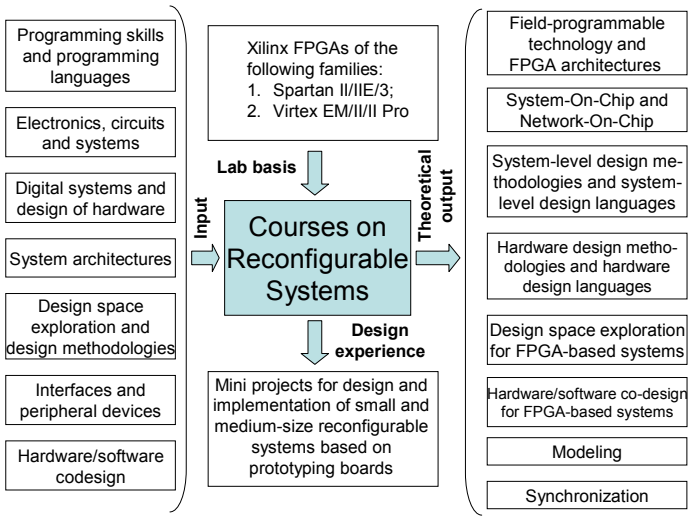

Fig. 2. The input (basis) and output (acquired experience) for the group of courses

3 Information Resources and E-learning Tools

Fig. 3 classifies the basic information resources that have been created and intensively used for the group of disciplines under consideration. They are divided into the following key segments [7]:

1. Technical papers and Web resources, which include exhaustive materials for labs and mini-projects. If possible, the source is given in Portuguese, otherwise in English. Most of the materials are provided in both English and Portuguese, which essentially simplifies dissemination of the information internationally. Note that many other universities have already used the information resources developed in this context at Aveiro. Besides, there are a number of international students attending the courses who don't speak Portuguese.
2. Documentation and supplementary materials that present all the details necessary for communication between FPGA-based circuits and external devices. They include the specification of interfaces, timing diagrams, instruction sets and programming modes for peripheral and other controllers,

information about device drivers, etc. This segment also contains detailed information about architectures and the utilization of FPGAs and CPLDs (complex programmable logic devices). The materials that are required mostly can be downloaded directly from the course site, and less important materials can be referenced through the relevant Internet addresses.

3. Examples and design templates that enable students to reduce the design time significantly. This has been achieved because of the following reasons:
   a. Examples have been selected in such a way that it is possible for new circuits to be developed by analogy with a given prototype.
   b. The generalized design templates enable students to select and customize a template for a particular circuit. For example, the design template for a hierarchical finite state machine (FSM) [13] specifies the basic functionality of fundamental blocks such as stacks and combinational circuits with control facilities at two levels, for state transitions within the active FSM and between different concurrent or sequential FSMs.
   c. Certain parts of examples can be reused in student projects. As a rule, such parts describe auxiliary circuits that are used for debugging, visualizing results, etc.
4. Tutorials and design diagrams explain the various design scenarios. They make it easier to understand the algorithm to be implemented or the primary functionality of the designed system, and can be classified as follows:
   a. Tutorials that demonstrate various scenarios within the respective computer-aided design (CAD) environment and explain the use of the prototyping boards.
   b. Tutorials that explain language constructions that can be synthesized and their distinctive features.
   c. Tutorials that illustrate different modes of interaction with typical peripheral devices through widely-used standard interfaces.
   d. Tutorials that permit different design methods to be understood. For example, they explain how to describe reprogrammable FSMs, how to construct a hierarchical FSM, how to execute recursive algorithms, etc.
   e. Tutorials that explain advanced synchronization techniques.
5. Student mini-projects, which have made it possible to realize an evolutionary approach for the group of disciplines related to reconfigurable systems. Current student projects are based on previously-developed student projects as well as other resources indicated above that are well-documented and available for the future. Thus, specific requirements have to be established for the projects to make their repeated reuse straightforward. This permits the complexity of the projects suggested in each new pedagogical year to be increased. As a result, small size projects that were proposed to students starting in 1997 have been evolved into medium size projects that are presently close to real-world problems in engineering design. Finally, it enables the department to reduce significantly the lead time for training practical engineers for industrial enterprises. Note that mini-projects provided the basis for many final year projects developed in different application areas involving FPGA-targeted design flows.

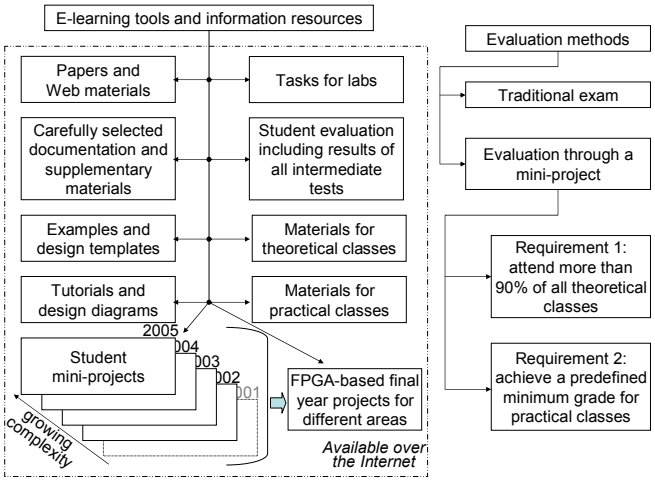

Fig. 3. Information resources and evaluation methods for the group of disciplines on the design of reconfigurable systems

In addition to the key elements considered above, the available resources include traditional supporting materials for classes and the results of different types of evaluation (see Fig. 3).

Special attention should be paid to tutorials because they cover practically all the directions in FPGA-based design considered above to produce the required output knowledge and experience (see Fig. 2). Each tutorial is organized as a set of graphical animated slides developed as a Microsoft PowerPoint presentation. Fig. 4 gives an example that demonstrates how teaching materials are presented. One theoretical class is dedicated to comparing iterative and recursive algorithms that might be used for hardware design as well as for software development. The class is organized as a set of topics that explain:

1. The structure and execution of recursive and iterative algorithms;
2. The specification of recursive and iterative algorithms for implementing such algorithms in hardware;
3. Models of control units that enable recursive and iterative algorithms to be executed;
4. Synthesis and implementation issues for practical CAD systems (such as Xilinx ISE [3] or Celoxica tools [10]).

Each topic is supported by the relevant part of the tutorial. For example, the left-hand part of Fig. 4 illustrates how a binary tree for recursive sorting can be constructed that implements the algorithm [14]. Animated graphics show how the tree is constructed incrementally and how each input value either creates or modifies a node. Being able to step forwards and backwards through the tree creation process significantly eases understanding of the algorithm. The middle part of Fig. 4 demonstrates how hardware-oriented flow-charts for recursive and iterative algorithms are built. Once again, all the required steps are shown sequentially with the opportunity to return to any intermediate step. Finally, control unit models for recursive and iterative implementations are explained (see the right-hand part of Fig. 4) with all the necessary language constructions.

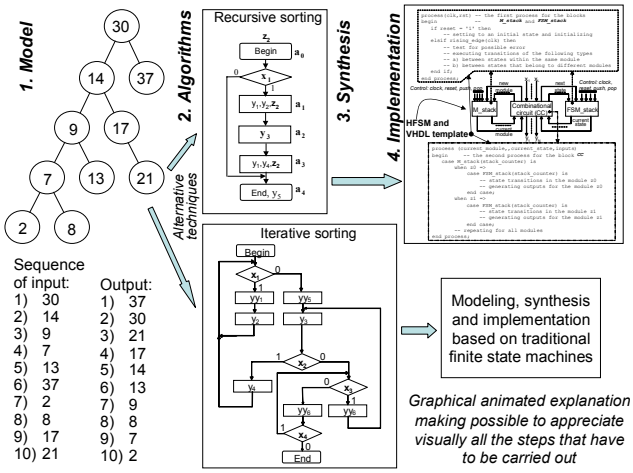

Fig. 4. Explanation of the fundamental features needed for implementing recursive and iterative algorithms in hardware

To maximize the effectiveness of the classes, students have access to all the available materials through a central point, the University e-learning Website [15]. Access to the Website is password protected, but the password can be supplied on request. In part, the materials described above are available at [16]. The data are organized so that students can rapidly find and download any required item (such as a tutorial, an example, a design template, etc.). The "Theoretical classes" and "Practical classes" sections (see also Fig. 3) are updated during a semester, and they include lectures and references to the relevant tutorials, examples, projects, and other material. For example, the tutorials alone contain more than 1000 animated slides that cover practically all of the material that is required for the classes.

4 Evaluation Methods and Criteria

The right-hand part of Fig. 3 shows the evaluation methods used (see also [7]). These are:

1. Traditional examination at the end of the course;
2. Evaluation through individual mini-projects that are allowed if and only if the following mandatory requirements have been satisfied:
   a. Students must attend practical and theoretical classes. Every absence is taken into account except where an acceptable justification document has been presented. Students must not miss more than 10% of the theoretical classes. Every task in the practical classes must be completed before the deadline established for the task (usually the end of the proper practical class with some exceptions for complex design problems).
   b. Students must achieve at least predefined minimum grade for practical classes (which is 14 within a grade range of 1 to 20).

Our experience has shown that 80% of the students were able to be evaluated through projects. It should be noted that this has been achieved in spite of the fact that the criteria for evaluation through mini-projects have been toughened compared

with [7]. The best student projects are submitted for publication in a Portuguese magazine on electronics and telecommunications. At present there are more than 30 student publications devoted to different areas of FPGA-based design.

A mini-project option for the final evaluation can be chosen at the beginning of a semester. The requirements that are agreed upon for each mini-project have the following basic characteristics: 1) the project is individual, 2) the project involves using the majority of methods and tools considered within the relevant discipline, and 3) the project is complete and relatively complicated. The time available for each mini-project is 13-15 weeks. At the end of this period the students have to be able to explain all the methods, language constructions, and automation tools used in the design. The development process for mini-projects has been organized as a sequence of the following steps.

- Formulation and discussion with the student of the design problem, which usually requires a few iterations;
- After the student understands the problem, he/she will begin the design with an opportunity to reuse blocks that have been developed previously;
- Periodical discussion of intermediate stages and results;
- Delivery of the final report;
- Public presentation and evaluation of the project.

It should be noted that mini-projects cover a wide range of potential applications [7]. Many of them may be needed for other projects that have to be completed outside of the courses. This approach provides an additional motivation for the students because reconfigurable systems can be linked with a similar work within other disciplines in which a particular student may be very interested.

Note that evaluation through mini-projects becomes very difficult or even impossible when the number of students exceeds a certain value. Our experience shows that this value is 40-45 students for 2 teachers responsible for the discipline. In this case the impact of laboratory classes on final evaluation is considered to be very important. A variety of methods have been put into practice. For example, good results have been obtained using the following approach in 2005/2006. At the beginning of each practical class a small project is proposed to the students, which has to be finished and presented for evaluation at the end of the same class. As a rule, not all the students can cope with this problem, and such students are allowed to finish the work within a given additional time period (one week maximum). If the student succeeds, a positive grade (10 or more in the range of 1 to 20) is guaranteed, but it will be significantly lower than the grades for students who completed the work during the class. Before the practical classes, relevant (but not the same) examples are considered and explained in theoretical classes. This approach achieves two important results: the students seek to attend theoretical classes (that are not obligatory in the University) because by doing so their subsequent work in the practical classes is significantly simplified; and the students work very hard in the practical classes.

5 Relationship with Research

During 2003-2006, three Ph.D. theses have been successfully completed in the scope of reconfigurable system design. Currently, there are two Ph.D. students working in this area. In 2003-2005 the student papers were accepted and presented in the major conference on reconfigurable systems – FPL – *Field Programmable Logic and*

Applications. Student papers have also been presented in a number of other conferences, such as *Euromicro Symposium on Digital Systems Design*, *Engineering of Reconfigurable Systems and Algorithms*, etc.

In 2002, the Advanced Microelectronics Engineering (AME) course was opened to provide additional training for graduate students for the Portuguese microelectronic industry. The majority of the AME attendees in Aveiro University were found to be former students of the disciplines considered in this paper. The former students of AME are currently working in industrial enterprises dealing with FPGA and ASIC-based design.

The following summary lists the basic scientific results produced by the students (including also former students) in 2003-2006 (i.e. during the last three years):

- Hardware and software for the prototyping board [12] targeted to the educational process and research activity. Information about the core hardware and software is available in [12]. One of the innovations is the implementation of a Bluetooth-based hardware block for the remote downloading of projects and communication with the board, supported by the relevant software tools. This simplifies laboratory work for the students significantly because any student can develop the projects in PC and as soon as it is necessary to debug/test the project in FPGA, he/she can interact with the board remotely.
- Design space exploration and implementation of computationally intensive algorithms, such as those for combinatorial search (for example, for solving the Boolean satisfiability problem [18] and for discovering the minimal cover of binary matrices, for graph coloring).
- Experimental analysis and comparison of different competitive and alternative implementations of computational algorithms, for example, recursive and iterative implementations [17], how embedded memory (or memory with application-targeted access facilities, such as parallel matrix addressing for rows and columns) affects the final complexity and performance of the system.

Acknowledgement

The authors would like to acknowledge Ivor Horton for his very useful comments.

6 Conclusion

An analysis of trends and tendencies in the scope of reconfigurable systems shows that this area is very dynamic and demanded by industry. As a result, a large number of well-prepared engineers are required and this number is expected to increase each year. The paper shows that training of engineers in reconfigurable systems is a complex problem. In order to simplify and increase the effectiveness of the training process, a wide spectrum of information resources and e-learning tools have been developed and successfully used in pedagogical practice. This paper shares the experience in this area acquired during about 10 years of teaching reconfigurable

systems at the department of Electronics and Telecommunications in the University of Aveiro (Portugal).

References

1. S. Hauck, The Roles of FPGA's in Reprogrammable Systems, Proceedings of the IEEE, vol. 86, no. 4, pp. 615-638, Apr. 1998.
2. L. Benini and G. De Micheli, Networks on Chip: A New SoC paradigm, *IEEE Computer*, pp. 70-78, Jan. 2002.
3. Xilinx, Inc., Products and Services, Available: http://www.xilinx.com/.
4. Computing Curricula 2005. Available: http://www.acm.org/education/Draft_5-23-051.pdf.
5. Computer Engineering Curricula 2004. Available: http://www.eng.auburn.edu/ece/CCCE/CCCE-FinalReport-2004Dec12.pdf.
6. *IEEE Transactions on Education*, Guest Editorial, A Vision for ECE Education in 2013 and Beyong, vol. 46, Nº 4, November, 2003.
7. V. Sklyarov and I. Skliarova, Teaching Reconfigurable Systems: Methods, Tools, Tutorials and Projects, *IEEE Transactions on Education*, vol. 48, Nº 2, May, 2005, pp. 290-300.
8. *IEEE Transactions on Education*, Guest Editorial, Undergraduate Engineering Education Challenged by The Bologna Declaration, vol. 48, Nº 2, May, 2005.
9. P.J. Ashenden, *The Student's Guide to VHDL* (Morgan Kaufmann Publishers, Inc., San Francisco, 1998).
10. Celoxica, Software Tools and Development Boards, Available: http://www.celoxica.com/products/default.asp.
11. Spartan-IIE Development Platform, Available: www.trenz-electronic.de.
12. M. Almeida, V. Sklyarov, I. Skliarova, and B. Pimentel, Design Tools for Reconfigurable Embedded Systems, Proceedings of the 2nd International Conference on Embedded Software and Systems, Xi'an, China, 2005, pp. 254-261.
13. V. Sklyarov, FPGA-based implementation of recursive algorithms, *Microprocessors and Microsystems*, Special Issue on FPGAs: Applications and Designs, vol. 28/5-6, pp 197-211, 2004.
14. B.W. Kernighan and D.M. Ritchie, *The C Programming Language* (Prentice Hall, 1988).
15. Available: http://elearning.ua.pt.
16. Available: http://www.ieeta.pt/~iouliia/
17. V. Sklyarov, I. Skliarova, and B. Pimentel, FPGA-based implementation and comparison of recursive and iterative algorithms, Proceeding of FPL, Finland, 2005, pp. 235-240.
18. Skliarova and A. Ferrari, Reconfigurable Hardware SAT Solvers: a Survey of Systems, *IEEE Trans on Computers*, Nov., Vol. 53, Nº 11, 2004, pp. 1449-1461.