Multimedia Tools for Teaching Reconfigurable Systems.

Conference Paper · January 2006

Source: DBLP

2 authors:

Valery Sklyarov
University of Aveiro

Iouliia Skliarova
University of Aveiro

MULTIMEDIA TOOLS FOR TEACHING RECONFIGURABLE SYSTEMS

Valery Sklyarov, Iouliia Skliarova [1])

Abstract
Multimedia tools provide significant assistance in vast variety of different areas and one of them is education. The paper shows that such tools are especially important for constantly updating engineering courses which present recent advances in microelectronic industry. One of such courses is dedicated to learning reconfigurable systems based on field programmable gate arrays (FPGA) where many novel methods and tools have to be incorporated within the same course. The relevant pedagogical process can be greatly simplified with the aid of multimedia resources and e-learning tools that have been developed and included into an integrated methodology successfully applied to the teaching of reconfigurable systems for more than 10 years. The paper discusses the impact of this methodology on attaining the required knowledge and experience with particular emphasis on course oriented tutorials, language templates and Internet resources.

1. Introduction

Tremendous progress in the scope of field programmable gate array (FPGA) technology has made it possible to advance configurable microchips from simple gate arrays that appeared on the market in the mid-1980s to platform FPGAs containing more than 10 million system gates and targeted to the design of very complicated electronic systems. Developing such systems on the basis of high capacity FPGAs involves vast variety of design tools and requires a large number of well-prepared engineers in the relevant areas [12]. Thus, new trends must be appropriately reflected in the respective pedagogical activity. The information revolution has focused considerable attention on educational needs relative to information science and engineering [1]. This tendency has taken place due to valuable assistance provided by different types of multimedia, such as the Internet, mobile video conferencing, interactive tools for intensive education, etc. The paper demonstrates how some of these resources have been successfully employed for teaching reconfigurable systems and describes course-oriented multimedia tools that have been developed.

Note that FPGAs themselves are widely used for mobile computing and multimedia. For example, Xylon Company combines Xilinx FPGAs of Spartan-3 family with the logicBRICKS IP cores library [9] allowing to customize quickly system designs running on generic FPGA development platforms into specialized multimedia products. Xilinx multimedia, video and imaging systems solutions provide the programmable hardware platforms, design tools, IP and reference designs needed to develop real-time video and image processing systems for a wide diversity of applications, such as video broadcasting and video conferencing, surveillance cameras, medical

[1] Department of Electronics, Telecommunications and Informatics, IEETA, University of Aveiro, 3810-193 Aveiro, Portugal skl@det.ua.pt, iouliia@det.ua.pt

imaging, home gateway and digital TV [17]. The Xilinx Virtex-4 programmable technology enables the developers to rapidly implement state-of-the-art DSP systems with high performance. Using FPGA-based reconfigurable processors for computation-intensive multimedia functions was considered in [11] reporting significant reduction in the number of clock cycles. Announced in 2006 Xilinx Virtex-5 FPGAs are a programmable alternative to custom ASIC technology and offer the best solution for addressing the needs of designers in the scope of high-performance logic, DSP, and embedded systems with unprecedented logic, hard/soft microprocessor, and connectivity capabilities [16]. This gives an ideal platform for mobile and multimedia applications. The Virtex-5 microchips are built upon advanced 65nm triple-oxide technology with speed up on average 30 percent higher and with capacity increased 65 percent over previous generation 90-nm FPGAs. Practically unlimited capabilities of recently appeared on the market reconfigurable devices for the design of complex systems can be seen from the following example: one of the recent FPGA XC5VLX330 [16] (Virtex-5 family) contains 25920 configurable logic blocks (CLBs), 192 DSP slices, 10 368 Kb of block RAM (including 18 Kb and 36 Kb blocks), and 6 devices for advanced clock management. The plenary talk by Mike Butts in FPL (Field-Programmable Logic and Applications)'2003 entitled "Molecular Electronics: All chips will be reconfigurable" reports that future project densities are likely to be upwards of 100 billion devices per square centimeter and argues that cheap molecular-scale reconfigurable logic, memory, and interconnect are likely to become the predominant digital technology a decade hence. The advances and promising applications of reconfigurable systems given above clearly demonstrate future prospects of FPGA technology and its challenging potentialities for both industrial needs and research activity.

Developing digital systems on the basis of high capacity FPGAs requires the use of computer-aided design (CAD) tools. In fact the electronic design automation business has profoundly influenced the integrated circuit business and vice-versa, e.g., in the scope of design methodology, verification, libraries, and intellectual property (IP) [10]. Traditionally, FPGA-targeted CAD systems support schematic and hardware description language-based design flows involving model-specific tools (such as the synthesis of finite state machines from a graphical specification) and IP core generators based on parameterization or templates. Recently, commercial CAD tools allowing digital circuits to be synthesized from system level specification languages (such as Handel-C [5] and SystemC [15]) have appeared on the market. Thus, the domain of reconfigurable systems design is very dynamic and many-sided. Evidently an ongoing review of the universities' curricula is necessary in order to incorporate the recent advances in FPGA architectures, design methods, and CAD tools. In other words, the curriculum must be sensitive to changes in technology and new developments in pedagogy and should emphasize the importance of lifelong learning [6]. Significant assistance in this direction can be provided through extensive use of multimedia tools. It is important to note that on the one hand multimedia tools are very helpful for teaching reconfigurable systems and on the other hand reconfigurable systems themselves provide a very significant base for the design of multimedia applications.

The remainder of the paper is organized in five sections. Section 2 discusses problems that are common for complex engineering courses and shows how to provide significant assistance in solving these problems with the aid of carefully organized information resources. Section 3 demonstrates the importance of animated tutorials for teaching reconfigurable systems. Section 4 lists the information resources available through the Internet for the considered courses. Section 5 characterizes some student FPGA-based projects in the scope of multimedia and mobile computing. The conclusion is given in section 6.

2. Brief courses' overview (problems and solutions)

This section presents the pedagogical strategy that has been adapted at the Department of Electronics, Telecommunications and Informatics of Aveiro University (Portugal). There are two following each other courses on reconfigurable systems included in the curricula on computers and electronic engineering, which will be continued in the future curricula proposed and approved in accordance with the Bologna proposals aimed at reform of high education in Europe. The first course is based on hardware description languages and it includes the following basic topics:

- FPGA architectures with examples;
- Static and dynamic reconfiguration;
- FPGA targeted CAD systems with the detailed studying of Xilinx integrated software environment (ISE);
- Modeling FPGA-based systems in general purpose software (in C/C++ language, in particular);
- Hardware description languages (HDL) for specification, modeling and synthesis of FPGA-based systems (VHDL is used as a primary HDL);
- VHDL in details;
- Using language templates, hardware templates and design libraries;
- Using IP cores;
- Finite state machines (FSM), hierarchical FSMs, parallel FSMs, reconfigurable FSMs;
- Design of interfaces with FPGA-based systems;
- Communication with typical peripheral devices, such as mouse, keyboard, VGA monitor, liquid crystal display (LCD) panel, static RAM, flash memory, etc.
- Practical applications chosen depending on specialization of the students.

The second course is based on system-level specification languages and it includes the following basic topics:

- Introduction to system-level specification and design for reconfigurable devices;
- System-level specification languages (on examples of Handel-C and SystemC);
- Design suit DK of Celoxica;
- Handel-C in details;
- Relationship between Handel-C and general-purpose description languages (C/C++, in particular);
- Modeling and debugging at different levels of abstraction;
- Relationship between Handel-C and HDL (VHDL, in particular). Interface between Handel-C and VHDL projects (design methods based on Handel-C and VHDL in such a way that one part of a project is described in Handel-C and another part in VHDL);
- Hierarchical finite state machines (HFSMs) allowing recursive functions to be described in Handel-C. Comparison of recursive and iterative algorithms implemented in hardware.
- Communication with typical peripheral devices, such as: mouse, keyboard, VGA monitor, LCD panel, static RAM, flash memory, etc.; device drivers;
- Practical applications chosen depending on specialization of the students. For example, in scope of mobile computing and multimedia the following projects have been proposed: games, FPGA-based GPS devices, interaction between FPGAs and mobile computers through Bluetooth interface; FPGA-based systems for processing maps in cartography, communications using Ethernet protocol, data compression/decompression, etc.

The main difficulty in teaching reconfigurable systems is the multidisciplinary nature of such systems. Even very simple FPGA-targeted projects require knowledge from multiple areas. Let us consider an example of a recursive sorting algorithm, described in C language in [8]. The complete task for the second course students is formulated as follows:

1. Describe the algorithm [8] in C++ using class templates in order to be able to sort different types of data, namely unsigned integers and strings;
2. After modeling in C++ and understanding all the details, describe the same algorithm in Handel-C and:
    - 2.1. Use HFSMs to implement recursive calls;
    - 2.2. Debug the project at a system level;
    - 2.3. Use device drivers to communicate with a keyboard and a VGA monitor. This is needed for entering initial data and for visualizing the results;
3. Describe any chosen by the students module in VHDL and establish links between the VHDL and Handel-C modules within the same project;
    - 3.1. Simulate the VHDL module at the functional level.
    - 3.2. Compile Handel-C modules to EDIF (electronic design interchange format) in the Celoxica DK environment.
    - 3.3. Complete the project in the Xilinx ISE.
4. Implement the project in an FPGA and test the results in physical circuits.
5. Design reconfigurable HFSM for the considered circuit based on embedded to FPGA RAM blocks. Provide for customization of the FPGA-based system through reconfiguration in such a way that either unsigned integers or strings could be sorted.

This example clearly demonstrates that different types of expertise are strongly required. Indeed, to fulfill the point 1 it is necessary to have knowledge in object-oriented programming to be able to prepare the proper C++ code, which will look something like the code given in [14, pp. 204-206]. Point 2 above requires experience in Handel-C, which possesses lots of differences comparing with C language. Besides, VHDL has to be used further and it has many differences comparing with C/C++ and Handel-C. The remainder points above necessitate familiarity with many FPGA-oriented topics, such as how to implement recursive calls in hardware (note that neither VHDL nor Handel-C support recursive calls), how to reconfigure circuits, etc. Therefore the integration of a large amount of materials is required within the course. To cope with this problem, instructors need to provide more intensive learning techniques that allow more knowledge to be dispersed within the same time slot. This task has been achieved through the systematic use of multimedia approach to learning. Our experience has shown that one of the most important components significantly assisting in rapid accumulation of the required knowledge is a course-oriented set of animated tutorials, which will be considered in the next section.

3. The role of animated tutorials

Tutorials play very important role in the considered courses. They cover practically all topics included in the respective curriculum and enable the students to simplify learning finally allowing the required knowledge and experience to be acquired. Each tutorial is organized as a set of graphical animated slides developed as a Microsoft PowerPoint presentation. The basic idea and organization of the tutorials will be demonstrated on an example. One of the elaborated topics is dedicated to a hardware circuit, which provides a low-level control of a VGA monitor. Figure 1 shows the basic architecture of such circuit. It is composed of a VGA controller, which generates signals for: a) horizontal and vertical synchronization (HS and VS respectively); and b) pixel colors

(R, G and B). It interacts with VGA memory, which keeps background image and foreground data that have to be visualized on the screen. These data are supplied by the Input Data block, which communicates with physical peripheral devices, such as a keyboard or dipswitches. Suppose the required data are composed of three fields: operand A, operand B and an operation # (such as addition, subtraction, multiplication or division). The result R of the selected operation # $\in$ {+,-,*,/} has to be displayed on the monitor screen as A # B = R (see figure 1). Symbol Memory is a storage, which keeps graphical images of each displayed character in the form of an array with the size of 16x16 pixels.

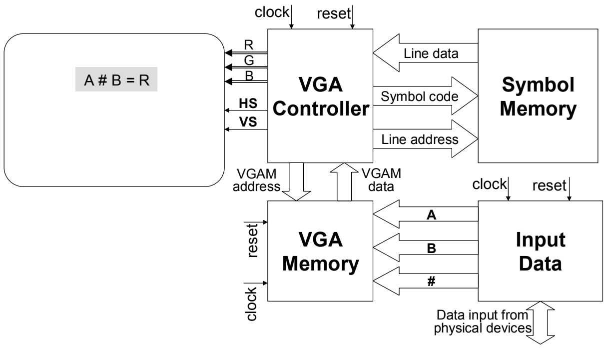

Fig. 1. Architecture of a circuit controlling a VGA monitor

There are several following aspects that have to be understood:

- The VGA monitor has to be properly synchronized and the pixels have to be accurately displayed in the correct screen positions. This is a job of the VGA controller and it is necessary to understand how this block forms dynamically changing image in accordance with input data.
- Input data are received from an external peripheral device, such as a keyboard. Thus, it is necessary to understand interaction mechanisms with the chosen peripheral device. In fact, various devices might be used and it is very difficult to remember all necessary interfaces and other relevant issues. Thus, a supplementary information is strongly required.
- Background and foreground images have to be stored in the block called VGA memory with a simple memory controller, which mixes background image and the displayed data. Suppose that VGA memory is built from embedded to FPGA RAM blocks. Thus, it is necessary to understand how to work with this memory. Once again many options might be suggested to students, such as: using RAM blocks; using FPGA distributed memory; using external static RAM, etc. In the last option numerous available static RAM microchips require knowledge of the respective interfaces. Therefore, once again, the proper information resources with fast search capabilities would undoubtedly provide a very significant assistance.
- Input data are received in the form of ASCII codes. Hence it is necessary to understand how to convert such codes to an array of 16x16 pixels representing picture of the relevant characters on the screen. This is done with the aid of Symbol memory block (see figure 1) and it is necessary to understand how it works.
- After all blocks illustrated in figure 1 have been designed it is necessary to provide the proper synchronization of the entire circuit composed of the considered blocks.
- Since the functionality of the circuit has to be described in VHDL all the points above require the appropriate experience in this language.
- There are a number of auxiliary issues, such as familiarity with different design scenarios in CAD environments, configuring FPGAs and the relevant software/hardware tools, simulation, verification methods for physical FPGA-based circuits, etc.

The considered above task has to be completed within 2 practical classes (2 hours each one). Thus, many different types of expertise have to be demonstrated within a very limited time slot and students must work intensively. Figure 2 shows information and multimedia resources, which provide very significant assistance allowing this problem to be successfully solved. Three tutorials on the left-hand side of figure 2 enable the students to understand how to construct the required blocks. Each tutorial is organized as a set of graphical animated slides, which demonstrate all the involved processes in dynamics. Each slide from the VGA tutorial is composed of many pictures displaying sequentially. Animated graphics shows how each line of pixels on the screen is being constructed continuously step by step from left to right and how each pixel of this line is being displayed. Being able to step forward and backward through the line of pixels creation process significantly simplifies understanding of how to provide the proper control. Indeed, all the required steps are shown sequentially with the opportunity to return to any intermediate step and this is very helpful to students. Similar slides explain how to display each particular character on the screen, i.e. how to search for a pixel matrix based on a given ASCII code and how the character composed of those pixels will be displayed dynamically. Our experience has shown that this is not an easy matter and the relevant material has to be properly understood by the students.

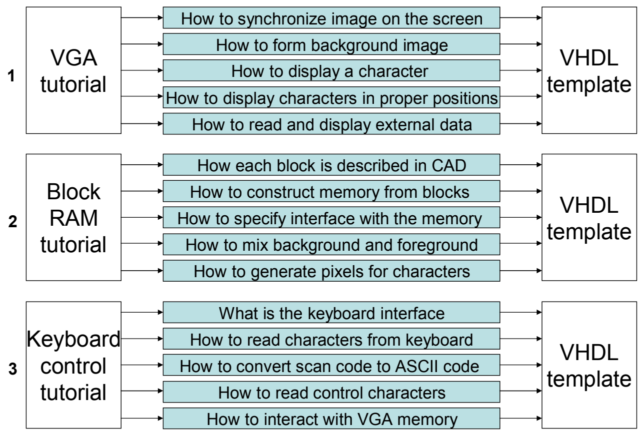

Fig. 2. Using tutorials and VHDL templates

Three blocks with VHDL templates on the right-hand side of figure 2 enable the students to understand how to specify the required functionality in VHDL. For example, the following VHDL code helps to establish horizontal synchronization:

```
horizontal_sync: process(clk, rst)
begin
  if rst = '1' then pixel <= 0;
  elsif rising_edge(clk) then
       if pixel = BP then pixel <= 0;
       else pixel <= pixel + 1;
       end if;
  end if;
end process horizontal_sync;

horiz_blank: h_blank <= '1' when pixel > P else '0';  -- blanking
HS <= '0' when (pixel >= HB) and (pixel < HE) else '1';
```

Here `clk` is a clock signal with frequency 25 MHz, `rst` is a reset signal and `BP`, `P`, `HB` and `HE` are constants, such as `constant BP: natural:= 793;` indicating the end of a blanking period. Establishing the correct synchronization based on such pieces of code is much easier than making this job from scratch. Generally speaking, VHDL template is a skeletal code, which can be customized and used as a fragment of a project. For example, for image synchronization the following VHDL templates have been given to students:

- Horizontal synchronization;
- Vertical synchronization;
- Horizontal text management;
- Vertical text management;
- Interaction with VGA memory (for background and foreground images).

A similar technique has been adapted for hardware templates. The latter are considered to be circuits with reconfigurable elements (such as RAM blocks), which enable the student to customize the circuit for particular applications. An example of such a circuit is a reconfigurable finite state machine.

On the whole the tutorials and VHDL/hardware templates explain various design scenarios in detail and make it easier to understand either the algorithm to be implemented or the primarily functionality of the designed system. The tutorials can be divided in the following groups [12]:

- Tutorials that demonstrate various scenarios within the respective CAD environment and explain the use of the prototyping boards/systems.
- Tutorials that explain language constructions that can be synthesized and their distinctive features.
- Tutorials that illustrate different modes of interaction with typical peripheral devices through widely-used standard interfaces.
- Tutorials that permit different design methods to be understood.
- Tutorials that explain advanced synchronization techniques.

Note that the majority of these tutorials are available online from [3].

4. Internet resources

The developed information resources, available through the Internet [3], are divided into the following key segments:

- Technical papers, which include exhaustive materials for labs and mini-projects;
- Tutorials and VHDL/hardware templates;
- Documentation and supplementary materials, which 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 FPGA and CPLD (complex programmable logic devices). The most required materials can be downloaded directly from the course site and less important materials can be referenced through the relevant Internet addresses;
- IP cores for implementing search algorithms over binary/ternary matrices (such as that are used for solving the Boolean satisfiability or matrix covering problems) and for data compression/decompression;
- Student semester and final year projects. Note that availability of these projects makes possible to implement an evolutionary technique where current student projects are based on the previously developed student projects;

- Examples and design templates that enable the students to reduce the development time significantly. This has been achieved because of the following reasons:
  - Examples have been selected in such a way that it is possible for new circuits to be developed by analogy with a given prototype;
  - The available generalized design templates enable the students to select and customize a template for a particular circuit. For example, the hardware template for an HFSM specifies the basic functionality of fundamental blocks (such as stacks and combinational circuits with control facilities at two-levels, namely for state transitions within the active FSM and between different concurrent or sequential FSMs).
  - Certain parts of the examples can be reused in student projects. As a rule, such parts describe auxiliary circuits that are employed for debugging, visualizing the results, etc.

5. Projects in the scope of mobile computing and multimedia

We have already mentioned at the beginning of the paper that on the one hand multimedia tools are very helpful for teaching reconfigurable systems (this was shown in the previous sections) and on the other hand reconfigurable systems themselves provide a very significant base for the design of multimedia applications (this will be considered in this section).

A number of final year projects were suggested in the scope of processing GPS signals and wireless communications. Two of the completed projects from this scope will be briefly characterized below. The first project had the following objectives:

1. Design of an interface block between a portable GPS device (Geko 201 [7]) and an expert version of the development board RC200 with FPGA XC2V1000-4FG456C [5].
2. Development of software and hardware permitting to copy geographical maps from a host personal computer (PC) to the flash memory available on the board and further visualizing these maps using onboard TFT monitor with touch capabilities.
3. Finding and marking positions on the map using the Geko 201 GPS device.
4. Development of software allowing maps with the details needed for the proper end-user to be constructed in PC.
5. Using some methods from graph theory, such as search for the shortest path enabling us to visit some pre-marked places.
6. Entering different points on the map using the touch panel and making various calculations, such as determining distances between the points, evaluating the time required to arrive to some place, etc.
7. For demonstration purposes a map of the Aveiro University camp was suggested.

The second project extends the capabilities of the first project allowing wireless communication mechanisms between FPGAs and other devices to be established. The primary intent was to design a set of hardware/software blocks, which would permit to support Bluetooth protocols. Figure 3 demonstrates the basic ideas of the project. The project had the following objectives:

1. Design of interface blocks, based on Bluetooth protocols, between FPGAs and external systems. Three of such systems were suggested for elaboration, namely:
   1.1. PC computer with Bluetooth dongle connected through a USB port (see figure 3);
   1.2. FPGA-based development systems, which allow Bluetooth communications to be established. For the considered project the boards RC200 [5] and DETIUA-S3 [2] were used.
   1.3. Bluetooth GPS devices (GPS Receiver BT-338 [4] was proposed as an example).

2. Hardware (FPGA) and software (C++ for PC) implementation of a protocol for wireless communications.
3. Development of debugging tools.

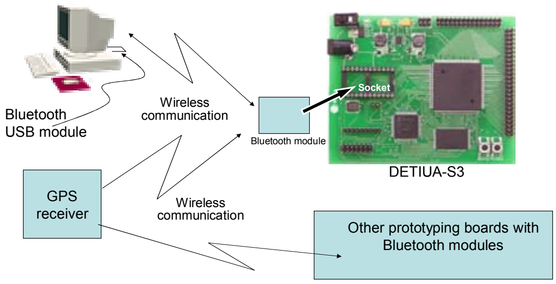

Fig. 3. Using FPGA-based circuits for wireless communications

Hardware circuits for both projects characterized above were synthesized from specifications in Handel-C. Note, that these projects are very useful for mobile and multimedia applications because they give base for the design of very fast portable devices communicating with each others and with host computing systems through the industry standard Bluetooth interface.

Many other examples of semester and final year projects were considered in [12,13]. There are many (more than 30) student publications reporting the results of the projects proposed in the considered courses. A list of references to these publications can be found in [3].

As we can see from the previous discussion, the considered courses in the scope of reconfigurable systems have achieved two following very important objectives:

- The students have acquired a good experience in very important and demanded for practical needs microelectronic technology and design methodologies;
- The students have been able to apply the acquired experience to the scope of their particular interests. It is important to note that the mentioned above interests depend only on the specialization of the students and such specialization does not have any interdependence with the contents of the considered courses. This is because the majority of topics within the courses are application-independent. As a consequence, we can conclude that such courses are very important and helpful for the majority of electronic and computer engineering curricula.

6. Conclusion

The paper highlights the importance of multimedia resources for teaching disciplines dedicated to recent advances in microelectronic industry, namely the design and implementation of reconfigurable systems. Due to the great significance of reconfigurable systems for future design space exploration and existing industrial needs, this scope should be properly presented in the pedagogical sphere to provide the knowledge and skills that are needed by future engineers working in these areas. One of the distinctive features inherent in reconfigurable, in general, and FPGA-based systems, in particular, is a many-sided nature, requiring accumulating knowledge from different areas. Thus, some tools stimulating and motivating intensive learning skills have to be discovered. This paper shares an experience in exploiting multimedia approach to learning. It demonstrates numerous information means, the most important of which are course-oriented animated tutorials, language templates and the Internet resources, which provide significant assistance and enable the students to succeed in learning and gaining the relevant experience.

7. References

[1] A Vision for ECE Education in 2013 and Beyond. *IEEE Transactions on Education*, Vol. 46, No. 4, November 2003.

[2] Almeida, M., Sklyarov, V., Skliarova, I., Pimentel, B. Design Tools for Reconfigurable Embedded Systems, *Proceedings of the 2nd International Conference on Embedded Software and Systems, China*, pp 254-261, 2005.

[3] Available at: http://www.ieeta.pt/~iouliia

[4] Bluetooth GPS Receiver BT-338. Available at: http://www.benelec.com.au/pdf/02944308-BT308_Bluetooth_GPS_Receiver_Manual.pdf

[5] Celoxica, Software Tools and Development Boards, Available at: http://www.celoxica.com/products/default.asp

[6] Computing Curricula 2005. The Overview Report for undergraduate degree programs in Computer Engineering, Computer Science, Information Systems, Information Technology, Software Engineering, Computer Science, Available at: http://www.acm.org/education/Draft_5-23-051.pdf

[7] Geko 201 Personal Navigator. Available at: http://www.garmin.com

[8] Kernighan, B.W., Ritchie, D.M. The C Programming Language, Prentice Hall, 1988.

[9] Kovacec, D. A Multimedia Platform for Automotive and Consumer Markets: Xilinx FPGAs and Xylon IP cores shorten design cycles and lower production costs for multimedia applications**.** Xcell Journal Online: http://www.xilinx.com/publications/xcellonline/xcell_53/xc_media53.htm

[10] MacMillen, D., Butts, M., Camposano, R., Hill, D., and Williams, T. W., An Industrial View of Electronic Design Automation, *IEEE Transactions on Computer Aided Design of Integrated Circuits and Systems*, vol. 19, no. 12, pp. 1428-1448, Dec. 2000.

[11] Panainte, E. M., Bertels, K. Vassiliadis, S. Multimedia Reconfigurable Hardware Design Space Exploration. *In proceedings: Parallel and Distributed Computing and Systems*. Cambridge, USA, 2004.

[12] Sklyarov, V., Skliarova, I. Teaching Reconfigurable Systems: Methods, Tools, Tutorials, and Projects. *IEEE Transactions on Education*, vol. 48, n.º 2, pp. 290-300, 2005.

[13] Sklyarov, V., Skliarova, I. E-learning Tools and Web-resources for Teaching Reconfigurable Systems. *In proceedings of 19th World Computer Congress, IFIP International Conference on Education for the 21st Century*, Santiago, Chile, pp. 215-224, August, 2006.

[14] Sklyarov, V. FPGA-based implementation of recursive algorithms, *Microprocessors and Microsystems*, *Special Issue on FPGAs: Applications and Designs*, vol. 28/5-6, pp 197-211, 2004.

[15] SystemC, Available at: http://www.systemc.org/

[16] Virtex-5 LX Platform Overview, May, 2006. Available at: http://direct.xilinx.com/bvdocs/publications/ds100.pdf

[17] Xilinx Delivers Highest Performance DSP Solutions for Multimedia, Video and Imaging Applications. Available at: http://www.fpgajournal.com/news_2005/10/20051024_01.htm