

A Prototyping System for Mobile Devices

Valery Sklyarov

University of Aveiro, Department of Electronics, Telecommunications and Informatics, IEETA

+351234401539

skl@det.ua.pt

Iouliia Skliarova

3810-193 Aveiro, Portugal

+351234370355

iouliia@det.ua.pt

Manuel Almeida

+351234370500

manuel.almeida@
ieeta.pt

Bruno Pimentel

+351234370500

pimentel@ieeta.pt

ABSTRACT

The paper discusses an FPGA-based prototyping system with both wired and wireless programming and data exchange facilities. The system can be efficiently used for different types of portable devices. For example, it can supply additional interfaces to mobile computers; can be seen as a hardware accelerator for solving computationally complex problems or serve as a base for portable embedded applications. The system includes an extendable core prototyping board, development tools and supplementary software. The board is designed in such a way that it provides direct support for reconfiguration and self-reconfiguration. It can also be employed for virtual prototyping systems that require more resources than the capacity of the onboard FPGA. The supplementary software includes all necessary drivers and interfaces to interact with the board. The development tools are targeted to design space exploration and they consist of hardware description language libraries, design templates, IP cores, software models and multimedia applications. The latter are mainly used for different types of experiments and for hardware/software partitioning.

Categories and Subject Descriptors

C.3 [Computer Systems Organization]: Special-Purpose and Application-Based Systems.

General Terms

Performance, Design, Experimentation.

Keywords

Reconfigurable systems, Prototyping board, Mobile devices, Remote reconfiguration.

1. INTRODUCTION

Contemporary field-programmable gate arrays (FPGA) are composed of programmable logic cells, memories, arithmetical devices, processors, circuits for advanced synchronization, etc. It is possible to change both the functionality of the majority of

FPGA components and interconnections between the components through reprogramming the relevant chip. This opportunity opens practically unlimited capabilities of FPGA-based systems in mobile computing and embedded hardware [1-4]. FPGAs can be seen as key blocks in reconfigurable computing, which can be defined as a methodology of using programmable logic devices in a system design such that the hardware-based logic can be changed and optimized to perform various tasks. Reconfigurable computing is computer processing with highly flexible computing fabrics. The principal difference when compared to using ordinary microprocessors is the ability to make substantial changes to the data path itself in addition to the control flow.

Mobile computing systems are such systems that may be easily moved physically and whose computing capabilities may be used while they are being moved [5]. As a rule, mobile devices are small and thus they have limited functional and computational resources. One possible way to extend functional capabilities is very similar to the general approach to the design of reconfigurable systems proposed more than 45 years ago [6] defining a new technique to evolve higher performance computing from any general purpose computer. In this particular case reconfiguration enables the designers to optimize hardware resources for executing vast varieties of operations through customization of these resources. The same idea may be applied to mobile computing allowing to extend functional and computational capabilities through reconfiguration and virtualization of mobile hardware in such a way that the same available physical resources may be retargeted to solving different problems and establishing different external interfaces. FPGAs can be seen as an ideal platform for such purposes because they possess a number of very useful distinguishing features both in implemented architectures and in functional capabilities [7] and combine high computation demands with dynamic task sets [8]. The most important of these features are listed below:

- Configurability that makes it possible to consider such systems as soft application-specific integrated circuits, i.e. soft hardware. It means, in particular, that reconfigurable hardware can be designed much like software for general purpose computers and thus, hardware of mobile devices can be customized quickly and easily.
- Reconfigurable devices are hardware circuits that can be designed, physically implemented and tested remotely. Indeed, physical implementation requires just uploading a bitstream, i.e. a binary file, to FPGA reconfiguration memory, which can easily be done through any available wireless interface. This is a very interesting feature for mobile computing.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

IWCMC'07, August 12-16, 2007, Honolulu, Hawaii, USA.
Copyright 2007 ACM 978-1-59593-695-0/07/0008...\$5.00.

- FPGAs introduce a new distinguishing computing paradigm and eliminate the necessity for traditional von Neumann architecture although such architecture can be used if required.
- Parallelism becomes the most important technique allowing system performance to be significantly increased. This is also very important for mobile computing because many sequential circuits can be replaced with functionally equivalent combinational circuits working in parallel allowing to speed up diverse operations significantly.
- FPGA-based systems open practically unlimited opportunities for design space exploration, comparing alternative and competitive implementations, etc. Thus, a set of experiments with mobile devices can easily be performed.
- They easily permit any required external interface to be established. Thus, the interfaces might be customizable, which is very important for mobile applications.
- Since FPGAs can be configured not only statically but although dynamically we can construct virtual systems that involve more resources than the resources available on an FPGA chip. Indeed, since dynamic reconfiguration makes it possible to change functionality of FPGA during run time, we can partition a complex system into subsystems, functioning sequentially. As soon as one subsystem has completed the required task, hardware can be retargeted to the subsequent subsystem through reconfiguration of the same FPGA. This is a very challenging feature for mobile applications.

The paper suggests an FPGA-based prototyping system, which permits to solve numerous problems in mobile computing because it can be seen as a movable, portable, remotely accessible computing system with customizable computing functions supported by the developed design tools. The system possesses the following characteristics:

- The core FPGA can be configured using both wired (USB) and wireless (Bluetooth) interfaces;
- The developed software/hardware components provide support for both dynamic onboard reconfiguration and remote wireless reconfiguration and/or interaction;
- The design process is supported by numerous developed tools, such as hardware description language (VHDL, in particular) libraries, design templates, intellectual property (IP) cores, software models and multimedia applications.

The remainder of this paper is organized in six sections. Section 2 describes potential applications of the developed prototyping system. Section 3 discusses hardware, software and remote facilities of the system. Section 4 considers the proposed development tools. Section 5 demonstrates the capabilities of dynamic reconfiguration and self-reconfiguration. Section 6 is devoted to an example illustrating how to use the developed prototyping board in an embedded system controlling an assembly line. The conclusion is in section 7.

2. POTENTIAL APPLICATIONS

Mobile devices became an important driver for FPGA products and technologies because of the following reasons [1]:

- Portable devices are highly power-dependant, which requires processing per microwatt to be optimized;
- Autonomous mobile devices used in some applications (such as medical, space, etc.), where repair by human is for some reason impossible, may need either autonomous or remote maintenance. This requires (remote) reconfigurability and adaptability;
- Sometimes we want to apply new design techniques, such as evolvable hardware, i.e. possibility to construct a circuit with a given specification, whose structure is previously unknown. This requires (remote) self-reconfigurability;
- For a number of applications we want to use the same hardware resources for solving different problems. Reconfigurability is the easiest way to provide this opportunity.

The considered prototyping system has been designed in such a way, which permits to satisfy the indicated above requirements as close as possible. Satisfying the first requirement has been achieved by allowing to use low power FPGAs. Other requirements have been fulfilled through the implemented support for static and dynamic reconfiguration.

Figure 1 demonstrates potential applications of the developed system, which can be either connected to a portable device in order to extend its capabilities (such as support for additional interfaces – Figure 1,a, accelerating computationally intensive algorithms – Figure 1,b, etc.), or can be seen as a portable computational unit for embedded applications (Figure 1, c). It should be noted that the primary objective of the system is rapid prototyping. It means that using this system we are able to verify a vast variety of useful extensions (i.e. hardware circuits enabling mobile devices to expand their functions and interfaces), which will be ready for manufacturing in commercial miniature products considered as accessories for mobile devices.

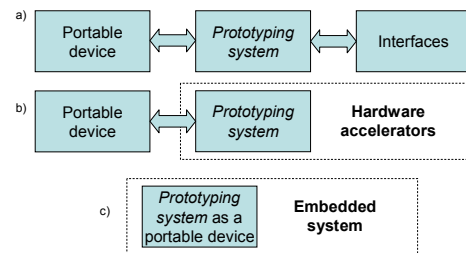


Figure 1. Potential applications of the developed prototyping system.

There are some important features of the prototyping system, which are listed below:

1. Extension capabilities (see Figure 2), which permit to verify interactions with numerous standard and application-specific devices, such as that are shown in Figure 2. A number of extension mini boards have been designed and manufactured for such purposes.
2. Testing and comparing alternative and competitive implementations. The developed system permits to store in the onboard flash memory up to seven autonomous bitstreams and any of them keeps projects for the entire FPGA and can be uploaded to the FPGA at any time. Besides of this, there exist an opportunity of wireless uploading from a host computer. Such

techniques are especially important for mobile embedded systems when we want to test different types of functionality.

3. Support for dynamic reconfiguration, which is implemented in such a way that the system components enable the FPGA to be reconfigured sequentially in accordance with a pre-defined arbitration strategy.

4. Support for self-reconfiguration making it possible to construct adaptive systems and to eliminate some problems through auto-repairing.

5. Support for multimedia making it possible to interact with virtual hardware, modeled in software of a host computer, through the relevant visual interface. For example, peripheral devices shown in the right-top corner of Figure 3 behave much like similar physical devices shown in Figure 2. Another example is an assembly line presented in the left-top corner of Figure 3, which makes it possible to verify numerous physical operations. Such multimedia applications enable the user to interact with the board distantly establishing necessary connections through the Internet.

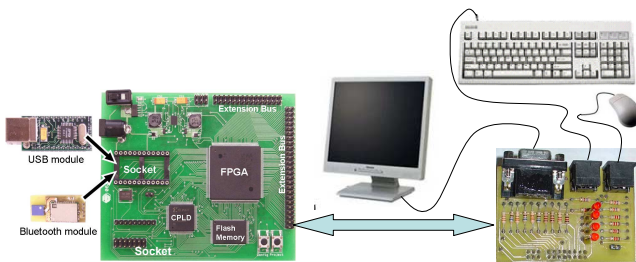


Figure 2. Extension capabilities.

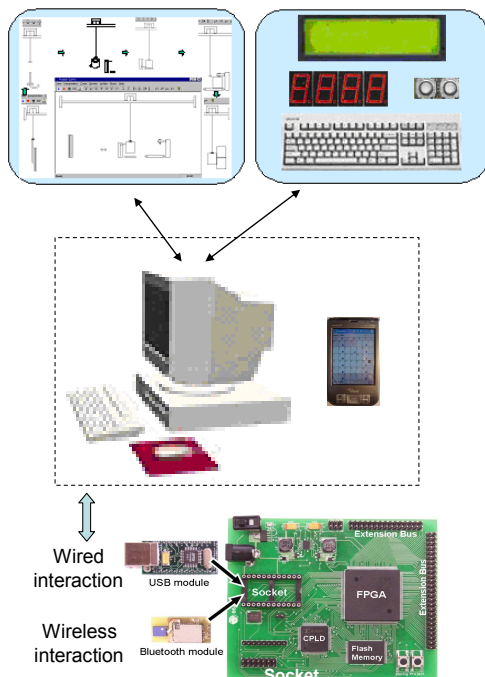


Figure 3. Multimedia capabilities.

3. PROTOTYPING SYSTEM ARCHITECTURE

3.1 Hardware

The basic architecture of the developed prototyping system [9] (see Figure 4) is organized in such a way that permits to provide the following features:

- Powering and programming the board from a host computer through USB port. If necessary an external power source can also be used;
- Keeping bitstreams for the FPGA in the board's flash memory, which permits to use the board as an autonomous device without any connection to the host computer and only external powering has to be provided;
- Keeping more than one bitstream in the flash memory for dynamic reconfiguration of the FPGA. The capacity of the selected flash memory permits to store up to 8 bitstreams, seven of which can be used for user projects;
- User-friendly software interface for programming the board and data exchange;
- Extension connectors for interacting with application-specific plug-in devices.

The flash memory is divided into two logical sections. The first section contains a bitstream that has to be pre-loaded to FPGA in order to allow transferring application-specific bitstreams (up to seven) from a host computer to the second section and vice versa. The second logical section is used to store application-specific bitstreams for subsequent quick loading into the FPGA.

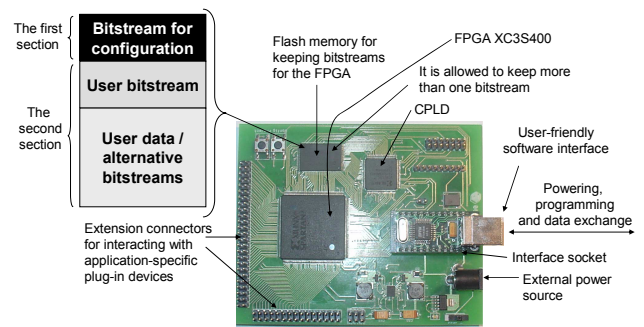


Figure 4. Basic architecture of the prototyping system.

The board contains a powerful FPGA XC3S400 [10] of Xilinx Spartan-3 family. For minimizing the reconfiguration time, a parallel reconfiguration mode has been employed.

The CPLD (see Figure 4) is needed for controlling the flash memory because during configuration the FPGA cannot execute this task. The CPLD generates also an initial reset signal for FPGA circuits as soon as a new configuration is completed.

Extension connectors permit to attach any application-specific external hardware, which enables the designer to optimize resources, to improve performance and to extend the functionality (see Figure 2).

An interface socket enables either wired USB or wireless Bluetooth module to be connected.

3.2 Software

The developed software [9] (Prototyping Board Manager - PBM), provides a convenient user-friendly interface and powerful design tools. PBM supports the following three modes:

1. Autonomous experiments with different single bitstream projects without connection to the host computer.
2. Dynamic reconfiguration of the FPGA, which permits to implement circuits that involve more resources than the resources available in the FPGA through run-time reconfiguration.
3. Programming FPGAs installed on additional extension boards. In this case, the core FPGA is considered to be a controller (manager) for a run-time reconfigurable system, which is composed of multiple FPGAs. In particular, such mode has been used for self-reconfiguration purposes.

The developed software can collaborate with commercial CAD systems in such a way that PBM supplies all kinds of low-level functionality, device drivers, interface and debugging facilities and CAD systems make it possible FPGA based circuits to be designed.

For example, user projects can completely be managed in Xilinx ISE [10] or in any similar environment, which finally generates a bitstream that is ready for downloading to FPGA. System-level specification tools (such as Celoxica DK design suite [11]) can also be used.

3.3 Remote capabilities

USB controller can be replaced with a Bluetooth module [9] permitting both configuration of the board and data exchange (note that in this case an external power source is required). The developed software/hardware tools make it possible to recognize the type of the attached module automatically. A Bluetooth module interacts with the FPGA through serial protocol (8 bit data, 115200 baud-rate, no parity bit and one stop bit).

Remote capabilities have been tested for a number of practical applications, namely:

- Reconfiguration of the onboard FPGA for experiments with different projects;
- Modifying the functionality of dynamically reconfigurable finite state machines (FSM) in such a way that the FPGA configuration bitstream is not changed and altering the functionality of the implemented FSM is achieved through reloading RAM blocks that are primary components of the FSM;
- Interactions with multimedia tools, such as that are shown in Figure 3, etc.

4. DEVELOPMENT TOOLS

The development tools include hardware description language libraries (VHDL libraries, in particular), design templates, IP cores, software models, and multimedia tools.

The libraries are considered to be language fragments for specifying typical operations. They can be just copied to more

complicated projects in order to supply the desired functionality. Figure 5 shows one potential example. Suppose it is necessary to design a circuit, which receives data from a keyboard, executes some operations over the received data and displays the results on a VGA monitor. The primary objective of the project is implementation of the operations. All the other blocks are considered to be auxiliary, because they just supply input and output facilities. Using predefined, pre-tested language fragments for such blocks make it possible to simplify the design process significantly.

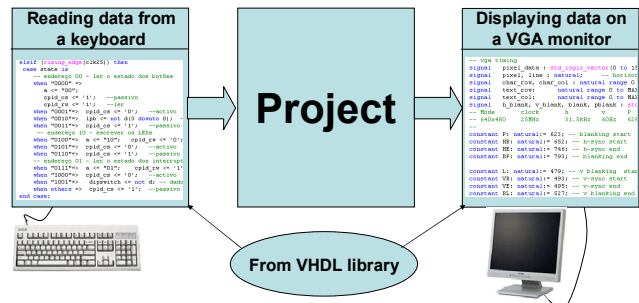


Figure 5. Using language library fragments.

VHDL libraries of this type have been developed for the following devices: VGA monitors, keyboards, mice, pushbuttons, dipswitches, LEDs, LCD panels, touch panels, parallel interfaces, serial interfaces, USB, Bluetooth, analog to digital converters, and some others [12].

A design template is a skeletal code, which can be customized somehow using predefined rules. A number of templates have been developed to provide support for hierarchical, recursive and parallel execution of various algorithms. The details can be found in [13,14].

Let us consider an example. Suppose we want to design a simple reprogrammable control system (such as that can be used for a robot), which includes a reconfigurable FSM and an execution unit. The prototyping board is considered to be a remotely-controlled device embedded in the robot. The design templates described in [15] allow for constructing an FSM with predefined constraints (such as the maximum number of inputs that have to be examined during one clock cycle, etc.) whose behavior might be changed through reloading FSM RAM blocks. This can be done remotely (see subsection 3.3) using wireless interface. Thus, the available design templates [15] permit to construct the FSM-based reprogrammable control unit. Remote board interface enables us to change the robot functionality distantly through modifying the content of FSM RAM blocks.

The IP cores present customizable projects for entire design problems. For example, recursive traversing a binary tree can be seen as a common problem for many practical applications, such as recursive sorting [16], combinatorial algorithms [17], etc. Customization is achieved through some generic parameters such as the size of stacks, registers, RAMs, etc. The details of the respective implementations for the developed IP cores can be found in [13,17].

The software models permit to explore the problem in general-purpose computers before the respective implementation in mobile hardware. Experiments with software models using

debugging facilities enable the designer to understand the problem and to shorten the time for the relevant FPGA-based project.

Such software models have been developed for the following three areas: solving combinatorial search problems, such as graph coloring, Boolean satisfiability, etc.; data compression/decompression; and operations over data, such as sorting considered above.

The developed multimedia tools make it possible to model FPGA-based systems partially in software and partially in hardware. The software/hardware parts communicate through established interface and they are constructed in such a way that we can explore systems with either more hardware or more software. In other words, the boundary between software and hardware is rather flexible.

One of the most important components of hardware/software co-simulators (such as that is shown in Figure 3) is a virtual visual sub-system, which enables the designer to verify the functionality of the developed system in a visual mode using a computer monitor (see the top part of Figure 3). A multimedia library enables the circuit designers constructing different visual objects on the screen and establishing different communication mechanisms between software of the computer and the attached hardware (additional details can be found in [18]).

Such types of hardware/software co-simulators are very promising in numerous areas such as: virtual design space exploration, rapid dissemination of different models and methods in the scope of hardware design, comparing alternative and competitive circuit implementations using the Internet facilities, education, engineering training, etc. It is important to note that the considered technique permits to get a valuable assistance from different types of multimedia, such as the Internet, mobile video conferencing, interactive tools for intensive education, etc.

5. RECONFIGURATION AND SELF-RECONFIGURATION

Figure 6 demonstrates the basic steps of the method [19] that can be used for dynamic FPGA reconfiguration. There exists also an opportunity to construct a multi-board dynamically reconfigurable system in such a way that each board can be autonomously reconfigured. This possibility enables the designer to build very complicated systems composed of circuits that can be flexibly modified and updated during run-time.

Using this technique we can implement a system that requires some hardware resources R_c , on available hardware that has resources R_h , where $R_c > R_h$. Applying the method for uploading the bitstreams to the flash memory allows to construct adaptive mobile embedded systems, which might change their functionality dependently on external events that can be analyzed by the host computer. The latter permits to form and to download the proper configuration in response to external events.

Similarly, this problem can be solved by an arbiter in a set of connected and communicating prototyping boards, in such a way that one board is considered to be an arbiter and the other boards are responsible for executing different operations of the considered problem. The developed software and hardware permits to execute the following tasks:

1. Activation/suspending of different boards;
2. Reloading bitstreams in different boards from the relevant flash memories (this task is managed by an arbiter);
3. Reloading the onboard flash memories from the host computer through either USB or Bluetooth interface (any bitstream can be uploaded individually to any flash memory).
4. Implementing self-reconfiguration strategy when an arbiter communicates with the remaining boards and with the host computer, which provides new bitstreams. In other words, dependently on some events the arbiter knows which FPGA has to be reconfigured and how it has to be done.

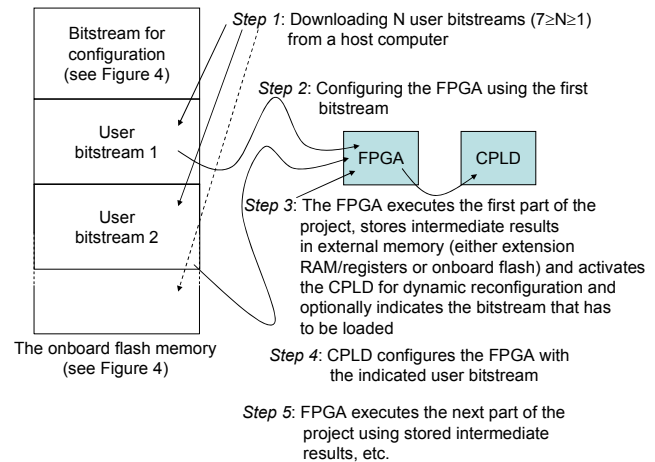


Figure 6. Reconfiguration of the FPGA.

6. EXAMPLE

Let us consider an example demonstrating how to use the developed prototyping system in an embedded system controlling an assembly line (see Figure 7). The core FPGA implements an FSM with dynamically modifiable functionality (see the top-right corner of Figure 7), which has been constructed on the basis of the developed design template. All necessary details about the template are given in [15]. In order to change the circuit functionality it is necessary to reprogram the FSM RAM blocks and it can be done through reloading them from the onboard flash memory. The latter can be employed for reprogramming the FPGA or for some other purposes. The following three developed tools can be used in this example:

1. A design template for the reconfigurable FSM;
2. Software that enables the user to verify the FSM functionality and to generate files for the RAM blocks, shown in Figure 7;
3. A multimedia application that permits to model the assembly line using such tools that are shown in Figure 3.

Finally, we can test different control algorithms for the system in Figure 7 and any of such algorithms can either be copied from the flash memory or uploaded from the host computer.

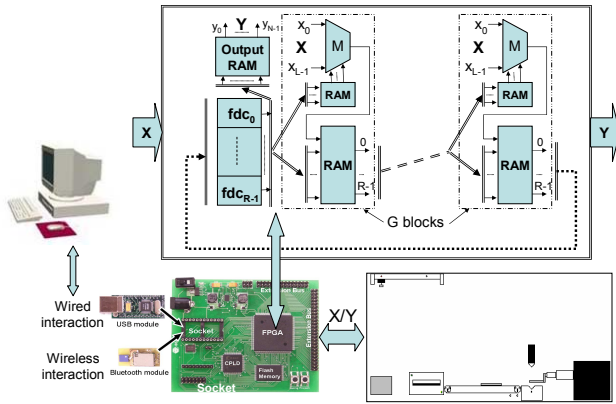


Figure 7. Controlling an assembly line.

7. CONCLUSION

The paper describes a portable prototyping system, which can be used for mobile computing and embedded applications. The system includes a low-power FPGA-based extendable prototyping board, supplementary software and development tools. The primary distinguishing features of the system are capabilities of dynamic reconfiguration (including self-reconfiguration) and numerous development tools significantly simplifying synthesis, verification and modeling steps for different design problems. The core prototyping board sets up wired (USB) and wireless (Bluetooth) interfaces with the host computer. The designed software permits to configure the board and to execute a number of useful supplementary functions. The developed multimedia tools allow virtual design space to be explored.

8. REFERENCES

- [1] International technology roadmap for semiconductors, 2005, System drivers. [Online] Available: <http://www.itrs.net/Links/2005ITRS/SysDrivers2005.pdf>.
- [2] Sklyarov, V. and Skliarova, I. Reconfigurable Systems and their Influence on Mobile and Multimedia Applications, Tutorial in *4th International Conference on Advances in Mobile Computing and Multimedia* (Yogyakarta, Indonesia, December 2006). [Online] Available: <http://www.iiwas.org/conferences/momm2006/tutorials.htm>.
- [3] Nallatech Provides FPGAs for NextCom's Rugged Mobile Computing Systems. [Online] Available: <http://www.embeddedstar.com/press/content/2005/7/embedded18637.html>.
- [4] Plessl, V., Enzler, R., Walder, H., Beutel, J., Platzner, M., Thiele, L., and Tröster, G. The case for reconfigurable hardware in wearable computing. *Personal and Ubiquitous Computing*, 7, 5 (Oct. 2003), 299-308.
- [5] B'Far, R. *Mobile Computing Principles: Designing and Developing Mobile Applications with UML and XML*, Cambridge University Press, 2005.
- [6] Estrin, G. Organization of Computer Systems – The Fixed Plus Variable Structure Computer. In *Proc. Western Joint Computer Conf.* (New York, 1960), 33-40.
- [7] Devlin, N. Reconfigurable Computing Architectures for High Performance Analysis. [Online] Available: <http://conferences.iee.org/medsip/Presentations/day1/Medsip-Reconfigurable-Computing-Architectures.pdf>.
- [8] Steiger, C., Walder, H., and Platzner, M. Operating Systems for Reconfigurable Embedded Platforms: Online Scheduling of Real-Time Tasks. *IEEE Trans. on Computers*, 53, 11 (2004), 1393-1407.
- [9] Almeida, M., Pimentel, B., Sklyarov, V. and Skliarova, I. Design Tools for Rapid Prototyping of Embedded Controllers, In *Proceedings of the 3rd International Conference on Autonomous Robots and Agents (ICARA'2006)*, (Palmerston North, New Zealand, December 2006), 683-688.
- [10] Xilinx, Inc., products and services. [Online] Available: <http://www.xilinx.com/>.
- [11] Celoxica products. [Online] Available: <http://www.celoxica.com>.
- [12] Sklyarov, V., Skliarova, I., Almeida, P. and Almeida, M. Design Tools and Reusable Libraries for FPGA-Based Digital Circuits, In *Euromicro Symposium on Digital System Design* (Belek, Turkey, Sep. 2003), 255-263.
- [13] Sklyarov, V. FPGA-based implementation of recursive algorithms. *Microprocessors and Microsystems, Special Issue on FPGAs: Applications and Designs*, 28, 5-6 (2004), 197-211.
- [14] Sklyarov, V. and Skliarova, I. Hierarchical Specification and Design of Control Systems in Robotics, In *Proceedings of the 3rd International Conference on Autonomous Robots and Agents (ICARA'2006)*, (Palmerston North, New Zealand, December 2006), 623-628.
- [15] Sklyarov, V. and Skliarova, I. Design of Digital Circuits on the Basis of Hardware Templates, In *Proceedings of the International Conference on Embedded Systems and Applications (ESA'2003)*, (H.R. Arabnia, L.T. Yang (eds.), Las Vegas, USA, June 2003), 56-62.
- [16] Kernighan, B.W. and Ritchie, D.M. *The C Programming Language*. Englewood Cliffs, NJ: Prentice-Hall, 1988.
- [17] Skliarova, I. and Sklyarov, V. Design Methods for FPGA-based Implementation of Combinatorial Search Algorithms, In *Proceedings of the International Workshop on SoC and MCSoc Design (IWSOC'2006), 4th International Conference on Advances in Mobile Computing and Multimedia (MoMM'2006)*, (Yogyakarta, Indonesia, December 2006), 359-368.
- [18] Sklyarov, V. Hardware/Software Modeling of FPGA-based Systems. *Parallel Algorithms and Applications*, 17, 1 (2002), 19-39.
- [19] Almeida, M., Sklyarov, V., Skliarova, I., and Pimentel, B. Design Tools for Reconfigurable Embedded Systems. In *Proceedings of the 2nd International Conference on Embedded Software and Systems (ICESS'2005)*, (Xi'an, China, December 2005), 254-261.