AC 2009-761: AN FPGA-BASED EMBEDDED-SYSTEM DESIGN LABORATORY FOR THE UNDERGRADUATE COMPUTER ENGINEERING CURRICULUM

John Bowles, University of South Carolina

John Bowles is an Associate Professor in the Computer Science and Engineering Department at the University of South Carolina where he teaches and does research in reliable system design. Previously he was employed by NCR Corporation and Bell Laboratories. He has a BS in Engineering Science from the University of Virginia, an MS in Applied Mathematics from the University of Michigan, and a Ph.D. in Computer Science from Rutgers University.

Gang Quan, University of South Carolina

Gang Quan is an assistant professor in the Department of Computer Science & Engineering at the University of South Carolina. He received his PhD from the University of Notre Dame. He is the recipient of an NSF Faculty Early Career Award. His research interests include real-time systems, embedded computing, power-/thermal-aware design, electronic design automation, advanced computer architecture, and reconfigurable computing.

An FPGA-Based Embedded System Design Laboratory for the Undergraduate Computer Engineering Curriculum

Abstract

The primary focus of this project is the development of FPGA-based materials and practices for an undergraduate embedded system design laboratory. FPGA-based devices are especially well suited for building application-specific systems in an undergraduate embedded system design course due to their comparatively low cost, shorter design cycles and reusability. The laboratory platform uses an FPGA as the hardware substrate onto which students configure and subsequently reconfigure IP core modules using modern embedded system development tools and processes. The course materials are based on exemplary materials presented in recent Xilinx XUP Professor Workshops on embedded system design and are engrained with state-of-the-art concepts and technology applied to emerging methodologies for embedded system design. Student assessments of the course and their own learning have been exceptionally positive.

1. Introduction

FPGA (Field Programmable Gate Array)-based devices are especially well suited for building application-specific systems in an undergraduate embedded system design course. FPGA-based designs have a much shorter design cycle, lower cost, and a smoother learning curve than traditional System-On-A-Chip[1] technologies. In addition, the devices are programmable and reprogrammable, which makes them reusable throughout the lab practices and excellent devices for investigating different design alternatives. FPGA devices are also becoming increasingly popular in industrial embedded system designs, therefore, learning to use the tools and design processes for FPGA based embedded systems provides students with skills and experiences that can be readily applied when they begin to compete in the global labor force.

The primary focus of this project is the development of FPGA-based materials and practices for an undergraduate embedded system design laboratory. The laboratory platform uses an FPGA as the hardware substrate onto which students configure and subsequently reconfigure Intellectual Property (IP) core modules using modern embedded system development tools and processes. The course materials are based on exemplary materials presented in recent Xilinx XUP Professor Workshops on embedded system design and are engrained with state-of-the-art concepts and technology applied to emerging methodologies for embedded system design.

1.1. The Lab Environment

Traditionally, the use of FPGAs in the undergraduate curriculum has been pretty much limited to the design and testing of digital circuits instead of the development of more advanced embedded systems[6]. There are two reasons for this: *insufficient hardware capability* and *lack of an appropriate embedded system design environment*. Earlier generations of FPGA devices were quite limited in terms of the available hardware programmable resources such as control logic blocks and they could not accommodate complex components such as processors—which are the most critical component in embedded systems. However, thanks to advances in IC technology, new generations of FPGA devices have significantly greater hardware resources and computing power. Embedded processors can be easily implemented either as a software core or as a dedicated hardware component inside a single FPGA device. The embedded system design environment has also been greatly improved. The Xilinx Embedded Development Kit (EDK)[11], together with Xilinx ISE[12], integrates a wide variety of design tools, Intellectual Properties[2], libraries, wizards, hardware/software generators, and documentation into a unified design environment. This greatly facilitates FPGA-based embedded system design.

We selected the Xilinx XUP Virtex-II Pro development board[15] (shown in Figure 1) as the hardware platform for our embedded system lab practices. At a cost of less than $300 (university price), this board provides an advanced hardware platform that enables the creation of complex applications. It is built around the Xilinx Virtex-II Pro FPGA device, with two built-in, 32-bit PowerPC 405 processors, 13,969 slices and more than 2M bytes of block RAM available. The dual processor cores provide an excellent opportunity to teach and experiment with some advanced embedded systems topics such as embedded distributed computing. Flexible MicroBlaze 32-bit and PicoBlaze 8-bit soft processors[16] can also be easily implemented with the reconfigurable fabric. The board also has a wide variety of peripheral chips available, including video input/output, audio CODEC, Ethernet, a multi-gigabit transceiver, serial port, external memory interface, and flash memory. The peripheral adapters and ports of this board are particularly attractive for project prototyping and are the focus of several lab projects.

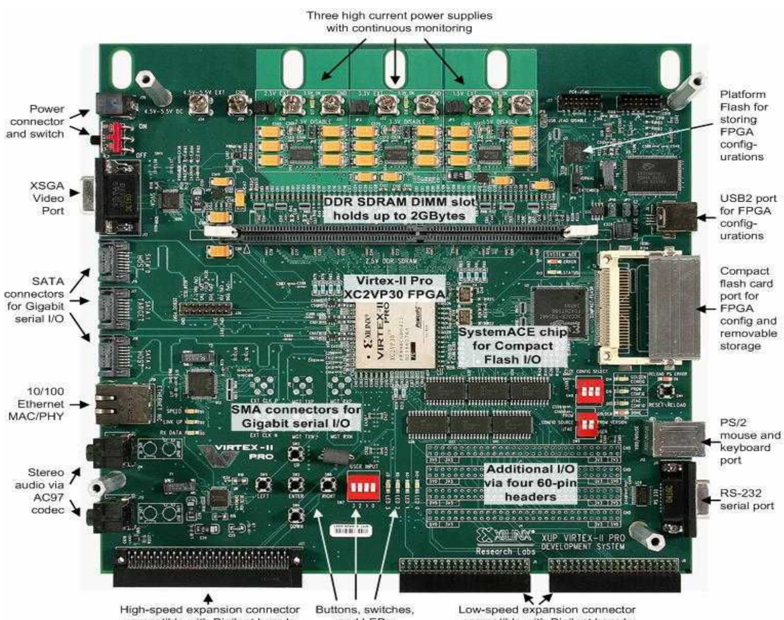

Figure 1. Xilinx XUP Virtex-II Pro Development Board[15]

The Xilinx embedded system development and implementation software packages, EDK and ISE, form the software environment for our experimental platform. This environment provides a comprehensive suite of software tools that support the entire embedded system design process; they range from low level logic design, simulation, and synthesis tools to high level cross-platform compilation, hardware/software debug, and co-simulation tools. This environment is highly intuitive; by abstracting and automating the design process it enables a smooth learning curve for students. We found that the students in our lab, even with no formal knowledge of hardware design languages such as VHDL or Verilog, and with little direct oversight from the instructor and teaching assistant, were able to use this software package and board to design a number of very interesting projects.

We choose the Xilinx hardware and software not only because Xilinx, as one of the leading FPGA companies, has established a proven record in both industry and academia, but also because Xilinx has a superior university program: the Xilinx University Program (XUP)[13]. In addition to offering free (or highly discounted) software packages and hardware platforms, XUP also offers extensive university support by providing hands-on workshops, example courses, a news group, free technical support and other services. Specifically, XUP has developed, and makes available to universities, a complete set of exemplary materials including presentation slides and detailed instructions for hands-on lab projects through its professor workshops[17] on embedded system design. These projects served as the starting point for developing our lab materials for the undergraduate students.

2. The FPGA-Based Embedded System Design Laboratory Course

We launched the FPGA-based embedded system design course during the Spring 2008 semester with 12 students and, as of this writing (Spring 2009 semester), we are teaching the course for the second time with 19 students. This report is mainly based on the 2008 class but some changes for the 2009 class, based largely on student feedback, are also noted.

The course covers six main topics in embedded system design:
1. Introduction to embedded system and design methodology;
2. Embedded hardware and IP (Intellectual-Property)-based embedded hardware design for SOC (System-On-a-Chip);
3. Embedded software design;
4. Embedded system debugging;
5. Embedded operating systems; and
6. Advanced topics.

The course centers around five lab projects intended to help students learn about embedded system design and the Xilinx ISE design tools and a student project. The labs are:

1. **Simple Hardware Design Using Xilinx ISE Tools.** This lab introduces the design of an FPGA with ISE™ software and takes the student through the step-by-step design process.

2. **System-on-chip Hardware Design Using Xilinx EDK Tools.** This lab shows the student how to use the Xilinx Embedded Development Kits (EDK) to create the hardware system for a typical embedded system. It includes three mini-labs that guide the student through the process of hardware development for a processor system using the Xilinx Platform Studio (XPS) and shows the student how to add Intellectual Properties (IPs) and implement the design.

3. **Embedded Software Development.** This lab guides the student through the process of writing a basic software application. This lab includes writing a basic software application

to access peripherals devices, adding a timer and interrupt controller, and then using the timer module to generate periodic events.

4. **Embedded System Debugging.** This lab focuses on the process of debugging the processor system. It uses the Xilinx debugging tools, Xilinx Microprocessor Debugger (XMD) and ChipScope, to investigate both the system hardware and software.

5. **Real-Time Operating Systems.** In this lab students use the Xilinx Xilkernel, a Real Time Operating System (RTOS) to develop multi-threaded programs. The RTOS provides multi-tasking operations and guarantees the timeliness of the applications instead of emphasizing system throughput. It provides a set of core modules and POSIX style functions that students can customize to implement higher level applications.

We are also developing lab materials that illustrate advanced topics, such as the application of multi-core technology, found in today's embedded systems. We have developed several labs, that include software for up to four processor cores. For example, one lab is to design a matrix multiplication program with its execution distributed in different cores. We have also built new clock drivers such that the working frequency for each core can be dynamically updated.

Each lab consists of several mini-labs that illustrate different concepts within the main topic. The students also design, implement, and present to the class a project of their own interests. All the lab work is done in teams of two or three students. The course had one final exam but no homework other than the labs. Based on student feedback after the first (2008) course we shortened or significantly revised the class lectures on processor design, pipelining and memory hierarchy, network models, device drivers, and inter-process communication. We also developed sets of homework for the Spring 2009 semester to give students additional practice in these areas.

3. Student Background

At the beginning of the semester a survey was done to determine the students' background and preparation for the course. The results are summarized in Table 1. A majority of the students were from the Computer Engineering (CE) major; this is not surprising since the course is required in the CE curriculum and most of them are interested in the topic. There were also a few Electrical Engineering (EE) majors who could take it as an elective. There were no Computer Science (CS) majors who could also take the course as an elective. In terms of prerequisites, most of the students can program in C/C++ and assembly language but they are not necessarily proficient. Most have also taken courses in digital logic design, computer organization, and operating systems. The survey also indicated that a majority of students had little prior knowledge or experience in embedded systems, hardware design, or electronic design automation tools.

Table 1. Student Background and Preparation

Student Major	Computer Engineering: 91.7% Electrical Engineering: 8.3%			
Reasons for taking this course	Degree requirement: 25% Purely self interest: 8.3% Both degree requirement and self interest: 58.3%			
Academic background				
Questions: To what degree are you familiar with the following knowledge and skills?				
	Have no idea	**Heard about it**	**Learned it before (but am not comfortable with it)**	**I am an expert on this**
C/C++		8.3%	41.66%	41.67%
Assembly programming	8.3%		75%	8.3%
Digital logic design			33.3%	66.7%
Computer organization			33.3%	66.7%
Operating systems		8.3%	33.3%	66.7%
Hardware design language	25%	33.3%	33.3%	
Embedded system and design	33.3%	33.3%	25%	
FPGA and design	58.3%	16.6%	16.7%	
Design tools (Xilinx, Cadence, Synopsis, etc)	66.7%	16.6%	25%	
Development board	50%	16.6%	25%	

4. Student Performance

An assessment of student performance in the course was based on the students' project reports for the five instructional labs and their final exams. The results are summarized in Table 2. A student met a specific lab objective completely (100%) if:

1. His/her team completed and demonstrated the corresponding lab successfully (60%);
2. The answers to the corresponding questions in the lab were correct (20%);
3. The lab reports were complete and well written (20%).

In the final exam, a student met a specific objective (100%) if his/her answers for the corresponding problems were correct.

As shown in Table 2, we found that students did much better in their lab assignments than on the final exam. One reason was that students felt more motivated to build something rather than prepare for a test. In addition, as some students suggested, having some individual written homework would have helped them to prepare better for the close-book final exam. We started giving additional homework assignments to complement the labs this spring.

Table 2. Student Performance

Course Objectives	Student Performance					
	Lab 1	Lab 2	Lab 3	Lab 4	Lab 5	Final Exam
Embedded system and design challenges	100%	100%	100%	100%	100%	62.71%
Embedded hardware design	94.13%	93.21%				77.24%
Embedded software development			88.58%			64.25%
Embedded devices and hardware/software interaction		93.24%	96.32%			72.34%
Embedded system debug				94.17%		80.06%
Embedded operating systems					81.53%	44.67%
Technical report writing	100%	94.56%	98.71%	98.79%	90.42%	

5. Student Projects and Productivity

Student productivity was evaluated based on the students' final projects and intermediate reports. The students developed and implemented five final projects, each by a separate group:

1. **Two-Player Pong Game.** In this project, an 8x8 LED matrix was developed by the students and connected to the XUP Virtex-II Pro board to serve as the display for the Pong game. The dip switches, push buttons, and LEDs on the board were used as the controllers for the game.
2. **WebServer Using Ethernet MAC.** This project implemented a simple Web server running on one of the PowerPC cores inside the Xilinx Virtex-II Pro FPGA. A laptop was connected to the board to test the network server.
3. **"Simon Says" Game.** This project implemented two versions of the traditional game "Simon Says". The program generates a random sequence displaying one LED at a time, and the player must accurately repeat the sequence by pressing corresponding buttons for the LEDs. Both a software version (running on the PowerPC inside the FPGA) and a hardware version (using the FPGA fabric) were implemented.
4. **MP3 Player.** This project was intended to read an MP3 file from a compact flash card, and play it through the AC97 codec card.
5. **MPEG player.** This project was intended to read an MPEG file from a compact flash card, and decode it using an MPEG decoder running on the PowerPC.

The first three projects were completely implemented and successfully demonstrated. The last two projects are significantly more difficult and the students were only partially successful (e.g., they were able to read the files from the compact flash card but not correctly decode and play it). (A large part of the difficulty that only became apparent late in the project is that the MP3 and MPEG decoder specifications are not freely available.) All but the first team used sample projects (from the web or sources provided by the Xilinx XUP program) as templates. However, considering the complexity of the commercial development tools and boards, the effort to implement these projects was not trivial by any means. After completing the five instructional labs, the students showed much greater confidence in using the Xilinx ISE tools and development board.

6. Evaluations

We developed two questionnaires to collect background information on students at the beginning of the course and feedback on the labs at the end. The College of Engineering and Computing also has a standard course evaluation form which we used to obtain students' views of the course. This data, together with the students' project reports and final exam results, were used to evaluate the effectiveness of the teaching materials in terms of student performance, capability, and satisfaction.

In an end-of-semester survey, we collected student feedback on four aspects of the course:

1. New knowledge learned from this course (Table 3);
2. New skills and experiences obtained through this course (Table 4);
3. Enhancements to the knowledge learned in other related courses (Table 5); and
4. Student views of the course materials and workload (Table 6).

These results are summarized in Tables 3 through 6.

Table 3. Student Assessment of Knowledge and Concepts Learned from the Course

Question: Through this course, I have learned about _____________ to my expectations.					
Answers	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
FPGA devices	27.27%	63.63%	9.09%		
FPGA design flow	18.18%	72.72%	9.09%		
Hardware design language	27.27%	54.54%	18.18%		
The embedded system and its design challenges	72.73%	27.27%			
Processors and embedded devices	81.82%	18.18%			
Hardware/software interface	45.45%	54.54%			
Embedded hardware architecture design	63.63%	36.36%			
Embedded software design process	54.54%	36.36%	9.09%		
Cross-compiler and its role in embedded software design	27.27%	45.45%	27.27%		
Embedded hardware debugging	27.27%	54.54%	18.18%		
Embedded software debugging	45.45%	45.45%	9.09%		
Embedded hardware/software co-debugging	27.27%	45.45%	27.27%		
Process/thread in embedded systems	45.45%	45.45%	18.18%		

Table 3 (continued).

Answers	Strongly agree	Agree	Neutral	Disagree	Strongly disagree
Multi-tasking in embedded systems	36.36%	45.45%	18.18%		
Scheduling in embedded systems	54.54%	36.36%	9.09%		
Inter-process communication in embedded systems	45.45%	27.27%	27.27%		
Real-time operating systems	54.54%	45.45%			

Table 4. Student Assessment of New Skills and Experiences Obtained in the Course

Question: I have gained the following skills and experiences to my expectations.					
Answer	**Strongly agree**	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Use of commercial embedded design tool	54.54%	27.27%	18.18%		
Use of commercial embedded design board	63.63%	36.36%			
IP-based Embedded system hardware design	36.36%	45.45%	9.09%	9.09%	
Embedded software development	54.54%	36.36%	9.09%		
Embedded System debugging	72.72%	27.27%			
FPGA development	18.18%	72.72%		9.09%	
VHDL programming	18.18%	54.54%	27.27%		
Using C-program to drive hardware	36.36%	63.63%			
Industry development process	45.45%	45.45%	9.09%		
Team work	63.63%	27.27%	9.09%		
Technical report writing	81.81%	18.18%			

Table 5. Student Assessment of Enhancement of Knowledge Learned in Other Courses

Question: I know about ________________ better through this course.					
Answer	**Strongly agree**	**Agree**	**Neutral**	**Disagree**	**Strongly Disagree**
Digital logic design	27.27%	54.54%	9.09%	9.09%	
Computer organization	54.54%	36.36%	9.09%		
Operating systems	54.54%	27.27%	18.18%	9.09%	
Compiler	36.36%	45.45%	9.09%	9.09	
Software Programming	27.27%	27.27%	27.27%	9.09%	9.09%

Table 6. Student Assessment of Course Work Load and Course Materials

	Strongly agree	Agree	Neutral	Disagree	Strongly Disagree
Teaching contents closely coupled with labs	45.45%	54.54%			
Breadth and depth of topics and contents are about right	54.54%	45.45%			
Lectures and course slides are easy to follow	63.63	36.36%			
Lab instructions are easy to follow	27.27%	63.63%			9.09%
More labs should be assigned	18.18%	18.18%	36.36%	27.27%	

Overall, as shown in Tables 3 to 6, students were very satisfied with the course and felt that they learned a great deal. The student response to the question: "Overall, how would you rate this course?" on the College of Engineering and Computing course evaluation was also very positive. The average score was 4.13 on a scale of 1 to 5 with 1 being "poor" and 5 "excellent". This response was higher than both the college (4.05) and department (3.87) averages, which is unusual for a junior-level course. (The average includes graduate courses which are almost always rated higher than undergraduate courses.)

The student responses also indicated that this course had positive impacts on enhancing what they learned in other courses. The students were also satisfied with the breadth and depth of the topics on embedded system design covered in the course. Opinions on the workload were evenly spread with roughly a third of the students wanting more, another third neutral and the remaining third wanting less work.

Comments from students were also very positive about the hands-on experience with commercial development tools and boards. Ten of the 11 students in the class explicitly listed this as the main strength of the course. Several students complained the schedule was a little tight, and they would have liked more time for the final project. We are taking this feedback into consideration as we refine the schedule and materials for the spring 2009 class.

7. Problems and Other issues

Using the commercial development tools and test boards provides abundant opportunities for students to explore their own interests. This is advantageous in cultivating the students' enthusiasm for the class and more generally for their future profession. However, there is also a downside with regard to adopting these tools and boards in a classroom setting.

First, students tend to focus and spend significant time on learning the details of the development tools and boards but pay less attention to the principles of embedded system design. In addition, neither the instructor nor the teaching assistant gets enough practice to become as skillful in using the tools and boards as a technician in industry. As a result, both the instructor and the teaching assistant have to spend a great deal of time and effort mastering the details of some of the usage techniques. Even so, some students still get frustrated when their particular problem cannot be solved immediately by the instructor or TA. In addition, focusing too much on the mechanics of using the development tools in the course would compromise the instructor's efforts to introduce the concepts and principles of embedded system design. How to strike a balance between use of the development tools and the core components of embedded system design for both the students and instructors is not a problem for which we have simple answers at present.

Second, the lab materials we developed have a software version compatibility problem. Commercial development software tools evolve very rapidly due to both market needs and the nature of computing technology. The advantage is that we are able to use newly available functions and IPs to simplify our new designs (such as for multi-core processors). However, the Xilinx tools we used do not guarantee complete compatibility, and sometimes the differences between versions of the software can be substantial. This was the case for Xilinx ISE 9.1 and the current version Xilinx ISE 10.1. As a result, some of the labs we developed with an older version of the software needed major modifications to be used with the newer version of the software.

Finally, although we used, and are still using, Xilkernel, the real-time kernel developed by Xilinx, in our labs, we are not satisfied with it. We found that as a micro-kernel, Xilkernel supports only a very basic set of functions and we found it very tedious and time consuming to develop labs for more advanced topics such as the multi-core programming and real-time scheduling topics we wanted to pursue. After an initial survey, we considered changing to the Nucleus OS from Mentor Graphics (MG). Productivity was our major concern in this choice; Nucleus is a commercial real-time operating system that supports multi-core type architectures and we hoped using it would significantly reduce our workload to develop the applications. MG also has a higher education program making their products available to academia at a much lower price than their general market price. However, when we obtained the development tools from MG, we discovered two problems. First, Nucleus works only with Xilinx ISE & EDK 8.2, while our multi-core labs were developed in Xilinx ISE 10.1. In addition it was not possible for us to migrate back to version 8.2 since our multi-core labs employ new IPs, available only in Xilinx ISE 10.1. (This is the same software compatibility problem mentioned above.) And second, MG's higher education program did not make the Nucleus OS license available with its development tools. As a result, we are still researching other alternatives, such as using uClinux to achieve our goals here.

8. Conclusions

Overall, the use of commercial embedded system development tools, the use of the Xilinx Embedded Development Kit (EDK), and materials based on the Xilinx XUP Professor Workshops on embedded system design for an undergraduate embedded system design laboratory has proven to be a successful approach for modernizing what was previously a fairly traditional embedded system design course. Students are learning to use the tools effectively, if not expertly, they are learning important design concepts and principles, and they enjoy doing it.

9. Acknowledgement

Work on this project was supported in part by the National Science Foundation, Course Curriculum and Laboratory Improvement (CCLI) program, Award # 0633641, "CCLI: Novel Instructional Material Development for Embedded System Education in the Undergraduate Computer Engineering Curriculum" at the University of South Carolina.

Bibliography

1. Chang H., L. Cooke, M. Hunt, G. Martin, A. J. McNelly, Lee Todd, *Surviving the SOC Revolution: a Guide to Platform-Based Design*, Kluwer Academic Publishers, Norwell, MA, USA, 1999

2. Chilton, J., "IP Market Challenges: Today and the Next Generation" IP-based SOC Design Workshop, 2002

3. Nucleus RTOS, http://www.mentor.com/products/embedded_software/nucleus_rtos/

4. POSIX, http://en.wikipedia.org/wiki/POSIX

5. Troxel, D., "Introductory Digital Systems Laboratory", Massachusetts Institute of Technology, 2003.

6. Wolf, W., *FPGA-Based System Design*, Prentice Hall, 2004

7. Xilinx, "Spartan and Spartan-XL Families FPGA", http://www.xilinx.com/bvdocs/ publications/ds060.pdf.

8. Xilinx, "Virtex-II Pro Interactive Datasheet", http://www.xilinx.com/applications/ web_ds_v2pro/ .

9. Xilinx, ChipScope Pro, http://www.xilinx.com/ise/optional_prod/cspro.htm/ .

10. Xilinx, Power Consumption Tools, .http://www.xilinx.com/products/design_resources/design_tool/grouping/power_tools.htm .

11. Xilinx, Xilinx Embedded Development Kit, http://www.xilinx.com/products/ .

12. Xilinx, Xilinx Integrated Software Environment, http://www.xilinx.com/products/ .

13. Xilinx, Xilinx University Program (XUP), http://www.xilinx.com/univ/overview.html, 2004.

14. Xilnix Xilkernel**,** http://www.xilinx.com/ise/embedded/edk91i_docs/xilkernel_v3_00_a.pdf .

15. XUP Virtex-II Pro Development System, http://www.xilinx.com/univ/XUPV2P/Documentation/XUPV2P_User_Guide.pdf .

16. XUP, EDK Work Flow with MicroBlaze, http://www.xilinx.com/univ/xup/labsfnd/downld.htm , Xilinx, 2004.

17. XUP, Professor Workshops & Seminars, http://www.xilinx.com/univ/uwkshp1.htm , Xilinx, 2004.

18. Zhang, Y. and Quan, G., "Enhancing the Adaptivity for Multi-Core Embedded Systems with Dynamic Performance Scaling in FPGA", *International workshop on Adaptive and Reconfigurable Embedded Systems*, 2008.