CHIPDESIGN – From Theory to Real World

Guillermo Payá-Vayá, Thomas Jambor, Konstantin Septinus, Sebastian Hesselbarth,
Holger Flatt, Marc Freisfeld, Peter Pirsch
Institute of Microelectronic Systems
Leibniz University Hannover, Germany
{guipava, jambor, pirsch}@ims.uni-hannover.de

Abstract

This paper presents our experiences and describes the details of a project-oriented ASIC design course held at Leibniz University Hannover. Our approach for this curriculum is to bring a real project (AVR instruction compatible microcontroller) into the classroom and introduce a professional standard cell based chip-design to the students. Moreover, the students are responsible for the organization of the overall project, improving their social skills and the ability to manage conflicts. The feedback shows that this approach is implementable and fulfills the required needs for both, industry and university curriculum.

Categories and Subject Descriptors:
B.7 [Hardware]: Integrated Circuits

Keywords:
Integrated circuits, computer architecture education.

1. Introduction

Complex microelectronic systems often comprise microprocessors and microcontrollers. Therefore, system engineers need comprehensive knowledge in computer architecture and chip design. Within the last two decades, a variety of different material on such digital systems has been introduced in electrical engineering and computer science curricula [1]. Unfortunately, the practical training at universities often does not meet the expected requirements.

Computer science educators are still facing fundamental questions like how "real-world problems" can be integrated into university programs. Since 1999, the Institute of Microelectronic Systems offers the project-oriented ASIC design course "*ChipDesign*" [2], which is attended by an average of 22 students per year. Our project provides a suitable solution to the question above.

Firstly, it offers the students a unique teamwork experience, comprehensive and practical knowledge. Then, the fabricated chip itself is another important motivation in return for a successful work.

Letting students solve issues autonomously in small groups in order to enhance teamwork skills is a common educational approach, which can be found in all kind of schools, colleges and universities. However, when it comes to teach chip design this way, the available range of university curricula gets smaller. One of the main reasons might be that some commonly used electronic design automation (EDA) tools require high expenses and experience. In order to avoid this issue, many courses which include HDL digital design are often limited to FPGA-examples [3,4,5]. In our *ChipDesign* project, the complete chip design professional flow is facilitated to the students, allowing them to get in touch with similar situations that an engineer is confronted with in industry. Some projects on a similar basis can be found in universities from United States, one example from the mid-nineties is [6].

Until 2002, the students' project focus within *ChipDesign* was the development of a Least Cost Router ASIC (LCR). The LCR determines the cheapest long distance provider for a given telephone number and dials the number automatically, including the provider access code [7]. In order to introduce some computer architecture aspects into the project, the LCR design was replaced by an AVR instruction compatible microcontroller [8], hereafter called *IMS-micro*. The

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, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
WCAE '07, June 9, 2007 San Diego, CA
Copyright 2007 ACM 978-1-59593-797-1/07/0006…$5.00

new project increased the overall complexity and allowed the students to design commonly used interfaces, such as I2C or UART.

The new design requires adapting some relevant parts of the project organization. On the one hand, the increased complexity of the new design requires higher motivated students. Therefore, a pre-selection test is used to choose the most prepared students. Moreover, additional short tests during the project encourage the students to learn the basics of ASIC design. On the other hand, the students take on more responsibility within the *IMS-micro* development. The new project provides a chance to perform design optimizations depending on the students' motivation.

This paper concentrates on two aspects. First, we present an advanced chip-design project, held at the Leibniz University Hannover, Germany, making emphasis on educational aspects and describing the technical background taught during the course. Second, a critical evaluation is presented, e.g. which are the strengths and the weaknesses and how to improve the quality of the course.

The paper is organized in five sections. Section 2 describes the course organization including educational aspects. Section 3 presents the new project and makes emphasis on the technical background required by the students in order to develop the project. In section 4, the new project is evaluated. Finally, conclusions are given in section 5.

2. Course Description

The course *ChipDesign* extends the predominantly theoretical study on a university by practical aspects. In addition, an appropriate learning approach is used to introduce the students to the issues of a "real" project.

2.1 Personal and institutional requirements

The course is geared towards students of electrical engineering and computer science. They study in 3rd – 5th year and have a bachelor or a comparable degree. For this reason, the students have some knowledge in digital circuits and computer architectures. Moreover, some of them have basic knowledge of electronic design automation tools. In general, it is assumed to have a heterogeneous target group.

Different aspects motivate the students to attend the *ChipDesign* project. The course provides an insight into the development of integrated circuits and could be a base for a future occupational activity. This applies to students with emphasis on microelectronics particularly. An additional motivation is to manufacture the successfully designed chip.

The supervision of the course is guaranteed by seven members of the research staff (tutors). They have deep knowledge and experience in chip design, which result from their study and research activity. The university provides the equipment (computer, EDA tools, communication platform, group room), which is used in other research projects also. During the course, the students are separated in different groups. Each group is supervised by one of these tutors.

Since the last years, a student's pre-selection is done by testing the basic knowledge of the students. This test also provides extra information for dividing the students into different teams. The successful work of each team is crucial to finish the *IMS-micro* project.

2.2 Course goals

The focus of the course is the improvement of technical and social skills. During the course, the students learn and work in teams containing a maximum of four persons. Each team is responsible for a subtask of the overall project. Based on the assigned responsibility, we cultivate social skills of the students and the ability to manage conflicts in a productive way. Each student is obliged to structure his activity, because he has to cooperate with his own team members, other students and the tutors. For this reason, the students increase their personal responsibility and planning ability ("jig-saw method").

Finally, in the range of technical skills, we focus on system design architecture, implementation with a hardware description language, verification strategies and basic knowledge of the backend.

2.3 Learning approach

We aim our goals by the approach of self-regulated learning [9]. Three important blocks complete our learning approach: knowledge, student process and project (see Figure 1). The students are treated as "active constructive participants in the learning process". They actively construct their own concepts and knowledge.

In the first block, called knowledge, the members of the research staff extend the student's background knowledge offering seminars and tutorials, and organizing supervised group meetings. Our project attempts to reduce the pure cognitive and consume based courses as much as possible.

In the second block, the students' process is divided into six phases in the style of [10]. This approach is known as "complete activity". In context of this approach, the students pass through the following phases: Information, Planning, Decision, Action,

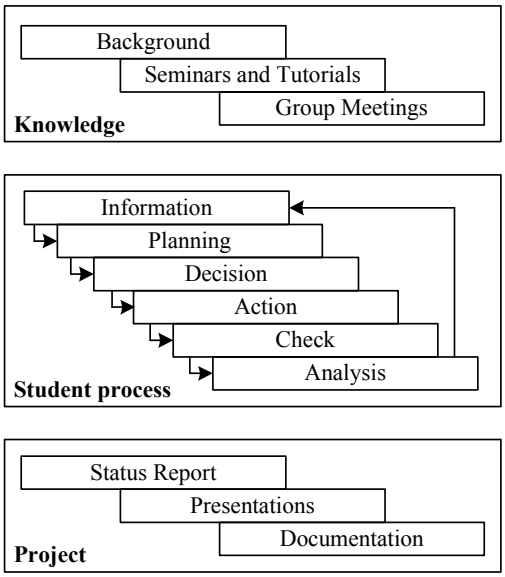

Figure 1. Learning approach diagram.

Checking and Analysis (Evaluation). The student process is cyclic, which means, after analyzing the results, the process can restart using the issues detected in the previous process. As in all managing processes, we could speak of "controlling". In these phases, the students learn and work by themselves, in partner or teamwork.

Finally, the different groups present their results to the other students and the staff members by giving technical presentations, status reports and documentation.

2.4 Scheduling

During the *ChipDesign* project the students participate in different tutorials in order to improve their skills in all the phases of a standard cell based design flow (see Table 1).

3. The Project

3.1 Design Concept

The first project proposed in *ChipDesign* was a Least Cost Router (LCR) [2]. In Germany, different telephone providers can be selected for long distance calls by dialing a provider access code prior to the telephone number. For this project, the students had to design and implement a controller, which automatically chooses the cheapest telephone provider, depending on time and day. The LCR project comprises a keyboard interface, display controller, synthesizer for a loudspeaker to output dual tone multi-frequency signals and a central control unit (CCU). The students had to design and implement the CCU and integrate it into the complete system.

In 2003, we decided to replace the LCR by an 8-bit RISC microcontroller (*IMS-micro*). The *IMS-micro* design project allows education of ASIC design as well as system level aspects, e.g. computer architecture and low-level software programming.

Table 1. *ChipDesign* project scheduling. The different seminars / tutorials have duration of 90 to 120 minutes.

Week	Seminars / Tutorials	Design Phase
W0	Microcontroller basics I	
W1	Verilog-HDL tutorial	Phase A - Module Implementation
W2	Writing Testbenches Functional Verification	
W3	Logic synthesis tutorial	
W4	Assembler programming Microcontroller basics II	Phase A - Module Verification
W5	Formal Verification	
W6		
W7	Phase A - Presentation	Phase A - Module Synthesis
W8		Phase B - System Verification - System Synthesis
W9	Layout tutorial	
W10	Backend tutorial	
W11		
W12		Phase B - System Emulation
W13	Phase B - Final presentation	Phase B - System Layout
W14	Project Status	Phase C - Optimization

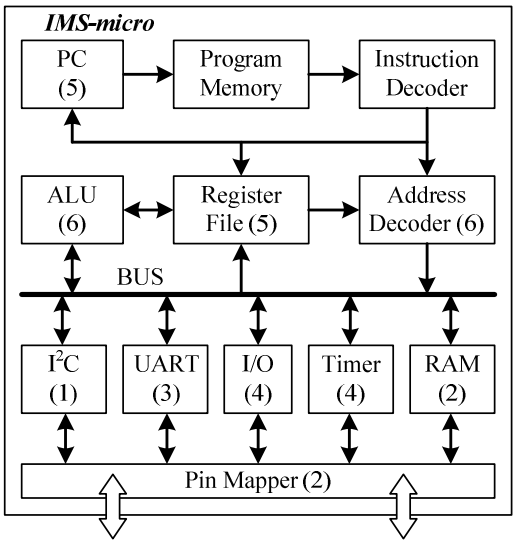

Figure 2. *IMS-micro* block diagram. The numbers in brackets refer to the student group number shown in Table 2 (Group#).

The *IMS-micro* (see Figure 2) is an ATMEL AVR 90S8515 [8] instruction set compatible microcontroller. This compatibility brings the possibility to use all free software for this widespread microcontroller available in the Internet. Moreover, the ATMEL AVR-Studio tool suite, which contains a simulator, assembler and C-compiler, can be used for implementation and start up the microcontroller.

According to Table 1, the project is divided into three different phases: A, B and C. In each phase, the students are separated into different groups, which should cooperate with the others. At the same time, the students participate in different seminars (see Table 1) to learn design, implementation and verification of an ASIC from specification to physical layout, while using state-of-art software tools.

Table 2. The *IMS-micro* is partitioned into six equally important groups of modules, which are closely connected.

Group#	Module	Description
-	Instruction Decoder	Given module
6	Arithmetic&Logic Unit	Data path for calculations
	Address Decoder	Address management
5	Register File	Register management
	PCIRQ	Program counter management
4	Timer	2 timers for time management
	I/O-Ports	Port configuration
3	UART	External RS232-like interface
2	External SRAM	Interface controller
	Internal SRAM	Interface controller
	Pin Mapper	I/O-Port arbitration
1	I2C	External I2C interface

3.2. Phase A. Module Implementation

At the beginning of the *ChipDesign* project, the students receive a basic specification of the *IMS-micro*, including the instruction set to implement the microcontroller. The different modules (see Table 2 and Figure 2) of the *IMS-micro* are specified by describing their functionality in a technical way and indicating the possible input and output signals. The specification given to the students is not complete. This causes some bugs after connecting the different modules that the students should identify and fix. Students must work in teams in order to get the system running.

During this phase, the students have to assimilate all the information given and implement the different modules in Verilog-HDL language [11]. The students also participate in the "*Verilog-HDL tutorial*" (W1) (see Table 1).

The *ChipDesign* seminar "*Writing test-benches. Functional verification*" (W2) gives some basic knowledge about verification strategies [12]. After this seminar, the students define their own verification environment depending on the complexity of the module to check. For example, a non-complex test-bench written in Verilog is enough to evaluate the UART module functionality; while a random stimuli generation is required to check all possible status-flags combinations of the ALU module (different kind of operations with different inputs operand should be considered).

Finally, the "*Logic synthesis tutorial*" (W3) focuses the process of converting a design represented with a hardware description language (HDL), e.g. Verilog, into an optimized gate-level circuit. The tutorial explains which kind of logic circuit (e.g. adders, multiplexers, registers…) is being inferred, when using different HDL structures. The standard cell technology is also introduced together with some considerations about design constraints, such as timing or area.

The goals during this phase are to improve skills in Verilog and get familiar with an HDL simulator. At the end of this phase, each group should provide the verified Verilog code for the modules assigned.

3.3. Phase B. Top Integration, Synthesis, Emulation and Backend Level

During Phase B, the students have to deal with sub-module integration, top-level synthesis and global verification. Additionally, they do the backend (physical layout) using the verified gate-level netlist. Communication between the students should increase, due to possible bugs detected in some modules implemented previously. Therefore, a web-based bug list is publicized and managed by the students of the *ChipDesign* project. All students have access to this list, in order to introduce more detected bugs or add, comment or remove the status of a particular bug. In this phase, the students are again divided into following teams:

- **Top-level integration and system verification.** Students create a top-level design to integrate all the sub-modules of the previous phase. Moreover, a simulation environment is mandatory to start verifying the system. The verification consists of running different assembler programs created by group 2 on the HDL simulator (Mentor Modesim [13]).

- **Assembler programming.** This team implements different applications in assembler language in order to

check the correct functionality of the modules implemented in the *IMS-micro* with ATMEL AVR Studio.

- **Top-level synthesis.** During Phase A, the students learned how to synthesize small designs. Now the students have to synthesize the complete *IMS-micro* by using the Verilog files provided by top-level integration team with the Synopsys Design Compiler tool. Moreover, errors detected during the synthesis, i.e. combinational loops, have to be identified inside the Verilog code and reported to the top-level integration team (controlling). The critical path also has to be identified and reported for possible design optimizations.

- **In-circuit emulation.** *ChipDesign* offers an additional FPGA-based verification approach. Simulation only allows the students to run small programs in feasible time. For simulation-based verification, it is difficult to model behavior of peripherals attached to the *IMS-micro*. Therefore, the students map the *IMS-micro* to an FPGA allocated in an emulation system by using Xilinx ISE tools. The emulation system is attached to an external development board (see Figure 3). This board, which contains external SRAM memories, an RS-232 connector and some LEDs, is used to test complex programs and special functions, like booting the *IMS-micro* from a Host PC via UART or I2C.

- **Backend.** In the "*layout and backend tutorial*" (W9 and W10), the students learn how to translate a gate level netlist of an example circuit into a chip layout, which is ready for fabrication. They go through a simplified flow containing topics like floor planning, cell placement and routing. In addition, the fundamentals of the layout verification, like design rule check and layout versus schematic check are shown. All these steps are done with industrially used tools, like Cadence Silicon Ensemble. The backend of the *IMS-micro* is performed by a small group. This group requires a high intensive supervision due to the responsibility for finishing the design process. The result is a verified layout that subsequently will be transferred to the foundry.

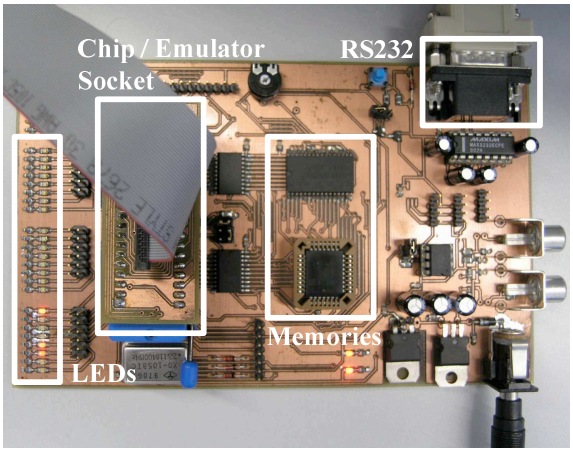

Figure 3. *IMS-micro* evaluation board. The *IMS-micro* chip is substituted by an FPGA-based system. The board is also used to test the manufactured chips.

- **Technical documentation.** Documentation is an essential part of creating technical systems, which is often neglected by the developers. The students learn how to write technical documents using common office tools.

3.4. Phase C. Intensive Verification and Optimization

At the end of *ChipDesign* project (Phase B), the *IMS-micro* project is fully verified and ready for manufacturing. In case of finishing Phase B before the end of *ChipDesign*, the students are motivated to identify the critical paths inside the *IMS-micro* design and perform architectural modifications to improve the performance.

Another thing attractive for improvement is the verification strategy. On the one hand, code coverage measures [13] can be analyzed in detail during the *IMS-micro* verification in order to check whether all possible states have been covered. On the other hand, automatic assembler test programs can be generated to reach all these states. The students use functional coverage measures [12] to guide the test program generation.

3.5 ASIC manufacturing and testing

Since 2003, the *IMS-micro* is designed within the *ChipDesign* project. All these years (except in 2004), a chip was manufactured by Austriamicrosystems AG. In 2006, *IMS-micro* was produced using a 0.35 μm CMOS technology [14] (see Figure 4). One of the main goals for *ChipDesign* is, that students should design and accomplish the project by themselves. Figure 5 shows the evolution of the *IMS-micro* project for the last three years in terms of hardware parameters (area and critical path). In addition, Table 3 shows common bugs introduced by the students during the *IMS-micro* implementation in the last four years.

All the manufactured chips can be tested by using the *IMS-micro* evaluation board (see Figure 3) and

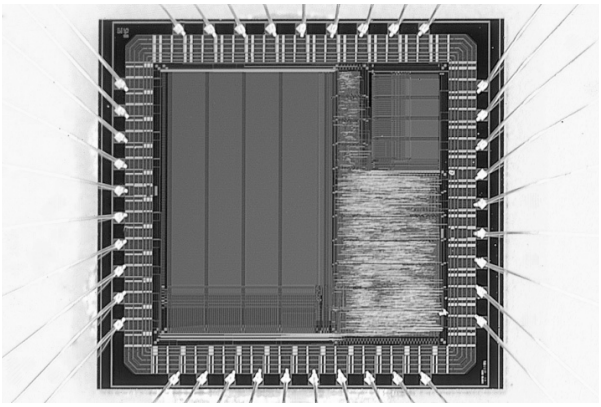

Figure 4. 2006 *IMS-micro* 0.35 μm CMOS chip

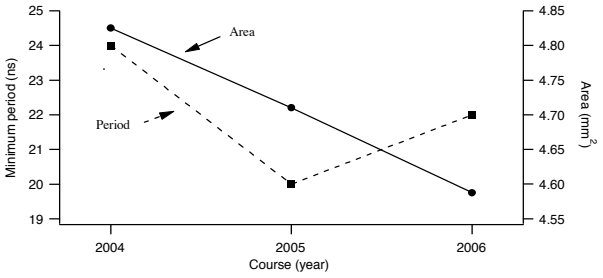

Figure 5. *IMS-micro* design evolution in terms of synthesis-based hardware parameter estimation: area (in mm^2 without data and instruction memories) and worst-case period (in ns).

running test C programs on the microcontroller. To support the chip testing, a workstation is connected to the evaluation board using the RS232 port. Since 2003, all the *IMS-micro* manufactured chips are working properly with a maximum running frequency of 60 MHz.

4. Evaluation

The evaluation is based on student and staff feedback taken after each course. The results show our experiences gained through the years of this project.

Deciding for a new design was a big effort, but the last years have shown that the *IMS-micro* is more suited to teach students all aspects of the design process. While the LCR is a dedicated hardware with limited functionality, the *IMS-micro* is more complex and flexible. The LCR is clearly structured with modules of similar complexity. Almost all modules are FSM-based with strong dependencies to other modules. The *IMS-micro* is a more sophisticated design with fewer dependencies and inhomogeneous module complexity. Therefore, building modules for

Table 3. List of common bugs detected during the *IMS-micro* project implementation.

Bug	Module	Description
unintended latches introduced	any	The HDL compiler introduces latches for signals driven from conditional statements not covering all cases. Use of default assignments recommended. This is not specific for *IMS-micro*.
16-bit zero flag	ALU	The 8-bit ALU supports 16-bit the ADDW/SUBW instructions being executed in two cycles. In contrast to 8-bit operations executed in one cycle, the zero flag must only be set when both partial operations' results are zero. This bug is quickly found as it causes the startup code of gcc programs to fail.
H flag	ALU	The AVR I-set specifies to preserve the H (half carry) flag during an ADDW instruction. In *IMS-micro* the 2-cycle ADDW is decoded as first ADD and then ADDW; as ADD alters the H flag, it must be restored when ADDW is signalized to the ALU. This bug does not affect any software we tested, but it is discovered during verification.
debug code left	boot ROM	*IMS-micro* features a small boot ROM with code to boot via UART or from I2C flash chip. For more transparency during emulation, this boot code is extended with debug outputs to LEDs; before finishing verification, these extensions must be removed.
reset of register file contents	register file	The AVR I-set specifies not to clear the GP registers upon reset. The C compiler startup code clears registers by XORing them with themselves, causing simulation to fail as X'es cannot be removed this way. This is not a real bug, but a verification issue.

the *IMS-micro* requires knowledge of the whole architecture, but the acceptance is higher since it is a practical design.

The change in the supervision concept results in better communication between students and tutors. While in the first years, two team leaders per group were assigned, now only one is responsible for each group, which reduces personal requirements.

In the first years, the only prerequisite for students was to be in a technical course program. In order to limit total group sizes, students of mature years were preferred. Since only "years of study" is not a good measurement for motivation and skills of students, assignment to groups was random and heterogeneous. A great improvement has been achieved by pre-selecting participants by written exams. Testing skills and motivation by exams allows a grade-rated group

assignment. Students who fail the exam can be rejected, those who pass get an access to the project. However, personnel requirements are higher due to compilation and evaluation of exams. Project group sizes are smaller and each student has to cope with more work, which makes him to take on more responsibility. On the other hand, smaller groups allow tighter cooperation in and between groups. Despite all benefits of smaller groups, it has to be pointed out that pre-selection tends to create idealized groups, which might be different in real project teams.

5. Conclusions

In this paper, advances in ASIC design education at the Leibniz University Hannover are presented. The project is separated in educational courses and practical work. After the project, students should be ready to design integrated circuits of higher complexity. The students have the opportunity to learn the whole integrated circuit design flow.

The project improves the social skills of the students by giving them all the responsibilities for the organization of the overall project. The students should resolve all possible conflicts themselves, because initially the research staff (tutors) only gives support in technical issues.

One way to increase the attractiveness of the project is to use an AVR compatible microcontroller as a design objective. This design aims to introduce system-level aspects in the course seminars, like computer architecture and low-level assembler programming. However, it introduces additional factors that should be evaluated, i.e. modifications in the supervision concept due to the new design structure, more communication between the students and high motivation due to the increase of the design complexity.

This kind of project has an elevated cost. The university can only offer this project when the requirements of equipment, EDA-tools and supervisors are fulfilled. Moreover, the university pays the manufactured chips.

Currently, *Chipdesign* project is part of the curriculum and the improvements concentrate on introducing a web-based communication infrastructure.

Acknowledgements

The authors acknowledge the contribution and participation of all those who, during these years, helped to realize this course in the way it was. The authors also want to thank N. Nolte, H. Kluβmann and P. Leppelt for their contributions to this paper.

References

[1] H. Hedberg, J. Rodrigues, F. Kristensen, H. Svensson, M. Kamuf, and V. Owall, “Teaching digital asic design to students with heterogeneous previous knowledge” in Microelectronic Systems Education, 2005. (MSE ’05). Proceedings. 2005 IEEE Int. Conf. on, 2005, pp.15–16.

[2] M. Kulaczewski, S. Zimmerman, E. Barke, and P. Pirsch, “Chipdesign: a novel project-oriented microelectronics course,” in Microelectronic Systems Education, 2001. Proceedings. 2001 Int. Conf. on, 2001, pp. 71–72.

[3] F. Ahamed and F. Scarpino, “An educational digital communications project using FPGAs to implement a BPSK detector” Education, IEEE Transactions on, vol. 48, no. 1, pp. 191–197, 2005.

[4] J. Amaral, P. Berube, and P. Mehta, “Teaching digital design to computing science students in a single academic term” Education, IEEE Transactions on, vol. 48, no. 1, pp. 127–132, 2005.

[5] V. Sklyarov and I. Skliarova, “Teaching reconfigurable systems: methods, tools, tutorials, and projects,” Education, IEEE Transactions on, vol. 48, no. 2, pp. 290–300, 2005.

[6] T. Chen, “From system design to IC design in 14 weeks - teamwork makes it possible” Education, IEEE Transactions on, vol. 36, no. 1, pp. 137–140, 1993.

[7] J. Jachalsky, M. Kulaczewski, and P. Pirsch, “Project management and verification-the key problems of student chip design courses” in Frontiers in Education, 2002. FIE 2002. 32nd Annual, vol. 3, 2002, pp. S2D–12–S2D–16 vol.3.

[8] Atmel Corporation: 8-bit AVR Microcontroller with Programmable Flash. AT90S8515 Data Sheet. September 2001.

[9] Wolters, C.A.; Pintrich, P.R.; Karabenick, S.A. Assessing Academic Self-regulated Learning Conference on Indicators of Positive Development: Definitions, Measures, and Prospective Validity. March 2003

[10] Dörnyei, Z.; Ottò, I. Motivation in action: A process model of L2 motivation Working Papers in Applied Linguistics, 1998.

[11] Thomas, D.E.; Moorby, P.R. : The Verilog Hardware Description Language. Kluwer Academic Publishers, June 2002.

[12] Bergeron, J. “Writing Testbenches - Functional Verification of HDL models”. Kluwer Academic Publishers. 2001.

[13] Mentor Graphics Corp.: ModelSim SE User’s Manual. Version 6.0c. January 2005.

[14] AMS lib (http://www.austriamicrosystems.com/)