Teaching VLSI Design Considering Future Industrial Requirements

Matthias Hanke, Thomas Schuster, Tim Kranich, Mladen Berekovic
Institut für Datentechnik
Technische Universitaet Braunschweig
Braunschweig, Germany
{hanke | schuster | kranich | berekovic}@ida.ing.tu-bs.de

Ralf Goettsche
High Performance Computing
Intel GmbH Braunschweig
Braunschweig, Germany
ralf.goettsche@intel.com

Abstract—**In 1999, the University of Braunschweig introduced a new study program dedicated to embedded systems design called "Computer and Communication System Engineering". It combines traditional educations in computer science and microelectronic engineering. VLSI Design I is one course of this program that works hand in hand with industrial partners. To integrate practical experiences into VLSI education the teaching combines tightly coupled lecture, tutorial and lab to guarantee an ideal learning process. All have the scope on integrated circuits design.**

I. Introduction

Embedded systems increasingly ease our everyday life. Simultaneously their user becomes less aware of them due to their advanced integration. Cell phones, vehicles, medical equipment or entertainment devices unrecognizably rely on digital circuits.

In the past the amount of transistors per area has doubled about every two years [1] leading to a tremendous amount which is no longer manageable by traditional approaches. Thus the development of embedded systems composed of tightly coupled microelectronic devices and multiple software layers has evolved into more platform based design methodologies [2]. This improves the designer's ability of creating more complex systems but does not allow to forget about the transistor's physics as basic element of a circuit. Thus modern education needs to cover a vast range of topics from microelectronic engineering to computer science. Computer and Communication System Engineering is a study program that combines industrial and academic needs of both fields [3].

The philosophy of the VLSI design I course as part of the study program is to tightly couple the contents of lectures, tutorials and labs (Figure 1) supported by industrial partners. This realizes a high level of coherence for the course's contents. Theoretical knowledge is imparted in the lecture. A tutorial supplements it by accentuated exercises. Both events are completed by a lab which includes practical implementations and tool usage. It addresses students from all semesters and thus brings together bachelor as well as master students. The course has been developed in close cooperation with Intel Braunschweig to consider the industrial aspects, technology trends and practical relevant topics. The target is to educate engineers with a sound awareness of technology physics who are able to utilize state-of-the-art design tools.

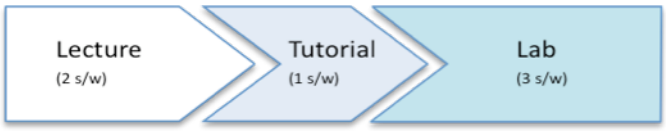

Figure 1: Interlocking of the course's sessions

II. Industrial Interests

Industry partners have dedicated interest in the education of future engineers [4]. Thus they cooperate with universities in creating new study programs, e.g. [5], or in funding professorships. Additionally, they create education programs to support teaching.

Intel believes that students everywhere deserve the skills necessary to succeed in a knowledge-based economy. That is why Intel provides a holistic blend of technologies, programs, and professional development resources for teachers, which can empower countries around the world. With improved teaching and learning environments, students have a superior chance to develop the required skills. Therefore, Intel setup the education initiative [6].

To support education in chip engineering, Intel GmbH Germany sponsored a professorship that takes care of the physical basics for system design. For a close collaboration and exchange, the professorship stands in a direct dialog with Intel GmbH Braunschweig to keep the lectures updated and to orient the lab at actual practices in industries.

III. Course Concept

Future engineers need to have a background in a wide range of topics. It starts from the basic physics of semiconductor materials, continues with the design of circuits including methodologies and ends up in the usage of tools and the handling of operating systems.

Materials and structure of transistors define its capacity, resistance and thus switching behavior. Depending on the technologies complexity different amounts of metal layers can be applied for signal routing. All physical properties of a

manufacturers process technology are represented by tables in a technology file. This file is the target for electronic design automation (EDA) tools that map a register transfer level (RTL) description of a circuit design to gate level. The courses aim is to take student through the whole design process from the physical structure of semiconductors to design methodologies.

Theoretical basics are built up during the lectures. They offer the audience the background they need to have to understand the exercises of tutorials and labs. Tutorials offer guided example exercises to theoretical problems discussed in the lecture. Thus participants get an impression for practical impacts of theoretically discussed lecture content. The Labs communicate hands on skills to connect knowledge from lecture and tutorial with practical tasks with the help of modern EDA tools. As the course program builds up competence from the ground up it demands hardly any prerequisites from its participants. Thus, it allows to address students independently from their study progress. As a consequence, it supports knowledge exchange among the audience.

IV. LECTURES

As the main part of the course the lecture offers the theoretical background of VLSI. It takes two sessions of each 45 minutes a week.

Cost aspects of the chips' manufacturing process introduce to the impact of physical basics to the final product. Namely these are e.g. cost of a die, dies per wafer or die yield. Formulas for the calculation of all of them are deduced stressing the importance of physical devices for the final product. Additionally, the lecture offers a broad overview on the development of the semiconductor market.

This leads to the structure of transistors. Explanations of semiconductor materials treat different devices and their behavior including the meaning of their characteristic plot. Detailed description of internal electrical fields an their influence on electric charges gives a deep insight into the transistor.

Electrical resistance and capacitance have a major impact on signal delays and shapes on a chip. Depending mathematical models are taught as well as interconnect strategies. Furthermore the lecture gives information on how to calculate signal delays.

After explaining structures and behaviors of the chip's elements the audience is introduced to the manufacturing process. Elucidations to oxidation and diffusion techniques lead to standard cell design. This includes design rules and is followed by packaging methods like bonding and flip chip.

Combining transistors to complex gates is introduced by the CMOS inverter as basic element for more complicated functions. The content goes from logical equations over the development of stick diagrams to routing strategies.

Additionally the lecture informs about design methodologies (Figure 2) to show the most efficient way of managing the product development process. This is highly connected to the standard cell design flow from RTL code to synthesis explained on state of the art EDA tools. Implied by this is the introduction to fanout, buffering and critical paths.

Furthermore there is focus on programmable logic arrays showing the potentials of reconfigurable hardware. Especially these devices are connected to the term of intellectual property (IP). Students learn how to differ hard and soft IP going along with the allocation to design hierarchies.

Correct hardware verification becomes increasingly important with growing design size and complexity. Thus finally the lecture gives information of modern test and self-test methods of designs.

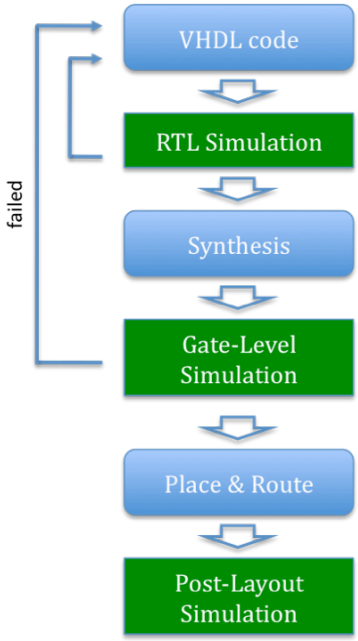

Figure 2: Typical ASIC design flow

This lecture content supports the understanding for designing and optimizing digital integrated circuits with respect to different quality metrics namely cost, speed, power and reliability.

V. TUTORIALS

Supplementary a dedicated tutorial of one session a week offers practical exercises on selected topics from the lecture. It covers essential calculations essential for VLSI system.

Final product cost is an important factor influencing the success of a design in market. Thus Students learn how to practically estimate the package costs for chip development from the theoretically treated formulas of the lecture including e.g. the calculation of yield and process complexity parameters. Along with this goes the understanding of certain design attributes' cost impact e.g. chip size and utilized technology.

Embedded circuit are built up by transistors. As follows an introduction to the transistor plot explains the physical behavior of the basic circuits' element. From given physical parameters and formulas from the lecture characteristic values have to be calculated. These are used to reconstruct the plot for a transistor. Additional exercises practice how to calculate

saturation current or how leakage varies according to temperature.

Employing De Morgan's rule students learn how to derive the logic equations for CMOS circuits design. They are enabled to transform equation into transistor-level schematics and vice versa. Furthermore they are introduced to the creation and comprehension of stick diagrams.

Interconnect modeling is the first step to be done before applying mathematical models on a circuit. Exercises on Elmore delay show how to derive timing from the created models. This clearifies the influence of wires' length on signals.

Additionally logical effort notation is covered by the handling of formulas for path and gate on example circuits. Propagation delays are calculated on the example of a NAND gate. Delay on path level is calculated on a circuit consisting of different NOR and NAND gates.

These exercises offer practical experience to the theoretical contents of the lecture and thus deepen the taught subjects. They build a bridge to the contents treated in the labs.

VI. LABS

The lab closely follows lecture and tutorial. With its three sessions a week it forms a major part of the course. Its aim is to give a deep insight in the practical design process of VLSI systems. Students are guided through the whole system design process starting at RTL and reaching GDSII including place and route.

Operating system for the labs is RedHat Enterprise Linux [7]. All utilized tools are installed on virtual machines (VM). According to this the system for all work stations is only set up once and can be employed on nearly every host operating system. This enables to have a clean system on the start of a lab just by copying the VMs on the computers' hard disks. Furthermore the installation works independent from other installations. Result is a very low maintenance effort especially when working in a computer pool that is used by other work groups either that have own dedicated demands on the system configuration.

As students from all stages of their studies are addressed it can not be presupposed that all of them are familiar with the handling of UNIX systems or hardware description languages (HDL). To bring all participants to the same level the lab starts with the very basics by giving an introduction to all needed LINUX commands and a single session to VHDL.

The whole lab is build up upon a simplified 16 Bit VHDL extensively commented MIPS I microprocessor core. Instruction set and pipeline depth are reduced in comparison to a complete implementation. This core allows an immediate start quickly leading to the emphasis of the course as students do not have to write the whole structure theirselves but are enabled to concentrate on the most relevant elements.

The lab is structured by ten dates:

- The center of modern embedded systems usually is a microprocessor. Correct and efficient utilization is a fundamental to create competitive solutions. That is why system designers need to know about the basic properties of their system's processor. Thus students get to know the MIPS instruction set. From the data sheet they learn the different instruction formats i.e. I-, R- and J-instructions. Afterwards they identify the yet implemented ones in the given HDL code.
- Extending the given core to a 32 bit processor (Figure 3) is the first practical task. Therefore datapath, memory access and register file width have to be adjusted. This inlcudes the exchange of the existing 16 bit memory moduel by a given 32 bit module. As result this exercise deepens the understanding of the microprocessor architecture and the VHDL code from the introductory session. It is the premise to get a feeling for hardware description.

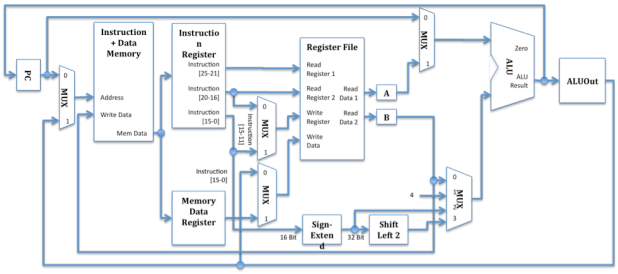

Figure 3: 32 bit MIPS processor [8] for the lab

- Verification of the developed hardware's functionality is a major part of system design to create correctly working products. As follows the students learn to verify the correct behavior of the modified core. They load a given sample binary program into the memory and simulate it. Consequently in the next step the correctness of the implemented changes can be verified by inspecting waveforms created in Mentor Modelsim [8] for the expected results. Detected malfunctions have to be interpreted to be able to make the right corrections in the code.
- Only few instructions are implemented in the given processor strictly limiting its usability. In order to create more complex programs the instruction set is extended by an additional instruction. This helps to get familiar with the structure of VHDL programming language and deepens the understanding of the MIPS processor's instruction handling. In the following the students write their own small binary program strictly sticking to the obliged format from the instruction set documentation. This program to be executed in the MIPS simulation includes the new self created instruction. From the wave forms in connection to the knowledge of the program code it can be concluded when and why certain components of the processor are active.As writing programs in binary is impractical for larger calculations the lab introduces to assembly code. First a multiplication assembly program has to be written that realizes its job by the add instruction. Loop constructs that have to be created for this task can be reused in the following. Afterward a C-code snippet of the Euclidian algorithm has to be translated to assembly code. Cross assembler and linker from open source GNU binutils [10] are used to compile a binary program from it. Both programs are run on the microprocessor. For its verification the results

can be read out from the simulation's wave forms just like for the initial binary program.

- In the following Cadence RTL Compiler [11] is used for gate level synthesis. Cadence tools' behavior is controlled by tool command language (TCL) scripts. After an explanation of the script languages basic elements the exercise is to produce a netlist of the MIPS design for an industrial TSMC 90nm standard cell library. The lab's participants are given a commented script skeleton and may use the tool's built in help to get familiar with necessary commands. They learn how to identify the maximum clock rate of their design and inspect how design size varies with increasing frequency. This allows to get into the usage of TCL scripts to control the tool's behavior, to learn the tool's commands and to get a feeling for the varying design parameters.
- Fundamental for design synthesis is the understanding of the cell library files' content. Included parameters like target temperature or the differences between worst and best case scenarios is extracted from the files and compared to each other. To achieve a better impression of technology libraries' capability the design is synthesized for 60nm and 45nm standard cell libraries as well. Results for maximum clock rate of these models give an impression of the impact of this variance. Maximum achievable frequencies increase with decreasing structure size. These experiments produce standard delay format (SDF) files describing the timing behavior of the components. In contrast to RTL level simulation of the programs in the previous tasks a gate level simulation for the design based on the timing files resulting from synthesis helps to understand how wire delays affect circuit's behavior. Previous timing free simulations worked no matter how high the clock rate had been chosen. Now with increasing frequency malfunctions can be determined.
- Upcoming of mobile devices with their limited access to power supplies make energy concerns an important design goal. According to this the circuit is inspected for its power consumption with the help of Cadence RTL compiler. First Mentor Modelsim is used to create a switching activity file for the design from the program previously created. It contains information of the amount of toggles in the processor from the input program. Afterwards it is fed into power analysis. From the resulting power report students can resolve information about how much different components of the design contribute to total power consumption. Following up the design is equipped with gated clocks and the depending power report shows how this technique reduces designs' power. Results illustrate that the applied mechanisms save up to one third of energy consumption depending on the utilized program.
- As design for test is an essential design goal RTL Compiler is used to insert scanchains. These can be used to test the final hardware after manufacturing. Cadence Encounter Test is used for automatic test pattern generation (ATPG). The resulting test patterns are input for Cadence NC-Sim and could potentially be used for a manufactured chip. A short report gives information about the amount of correct and failed test vector results.
- All previous steps were aiming at bringing the design to a stage where it could be submitted to a manufacturer. Still missing is the final step that maps the design to a GDS II file. For this reason the design eventually undergoes floorplanning, placement, clock tree synthesis and routing in Cadence SoC Encounter (Figure 4).

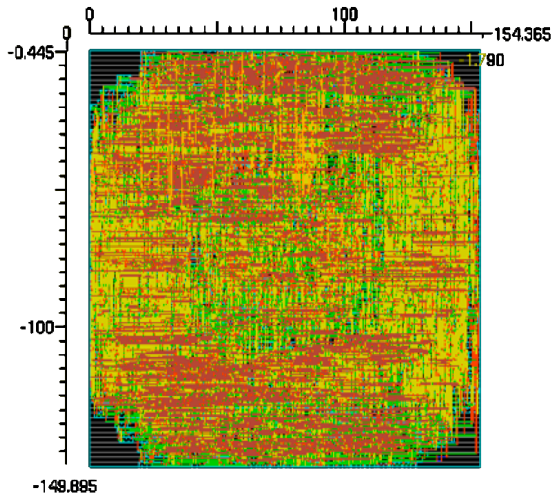

Figure 4: Routed processor from the VLSI lab

VII. Evaluation

Due to the tutorials accompanying the lectures, the supervisors are in direct contact with students solving the given exercises. Problems in the understanding of the lecture's contents can thereby be immediately identified and relieved.

Small homeworks that do not take an effort of more than 15 minutes keep the treated topics in mind and allow to monitor how well the subjects have been learned. In addition the tools' reports give a direct feedback of the participants' advances. Intensive communication during the labs allows to support the students wherever they need help.

After the last sessions an exam evaluates the learned knowledge. It contains the combined topics from lecture, tutorial and lab. Thus a successful examinee has proven knowledge in the whole system development process.

VIII. Conclusion

The tight coupling of lecture, tutorial and lab reaches a high grade of integration concerning theory and practice. Students get involved in the design flow of a complete design from the ground up. They are effectively educated in a wide range of topics. An ample understanding of the underlying principles of embedded system's circuits give the participants of this course a valuable background for their upcoming career as engineer. Efficiency is significantly improved by the tight cooperation with industrial partners.

Acknowledgment

Our thanks go to Intel Braunschweig who are sponsoring our VLSI education program. Cadence and Mentor have provided tools and licenses.

REFERENCES

[1] Gordon Moore, ”Moore’s Law: Raising the Bar“, ISSCC February 2003J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp. 68–73.

[2] D. Densmore, A.Sangiovanni-Vincentelli, R. Passerone, “A platform-based taxonomy for ESL design”, IEEE Des. & Test Comput., Vol. 23, No 5, 2006, pp. 359-374

[3] Mladen Berekovic, Matthias Hanke, Thomas Schuster, Tim Kranich, Rolf Ernst, ”ESL Design in the context of Embedded Systems Education“, WESE'09

[4] Grant Martin, "Industry Needs and Expectations of SoC Design Education", Proceedings of the 2003 International Conference on Microelectronics Systems Education, 2003, p. 146

[5] M. Balakrishnan, B. S. Panwar, "A specialized graduate program in VLSI design tools and technology", IEEE International Conference on Microelectronic Systems Education, 2005. (MSE '05). Proceedings. 2005, pp. 83-84

[6] Intel Corporation, education initiative, http://www.intel.com/intel/education/index.htm?iid=intel_corp+body_education

[7] http://www.redhat.com/

[8] J.L. Hennessy, D.A, Patterson, D. Goldberg, K. Asanovic " Computer architecture: a quantitative approach", Morgan Kaufmann, 2003

[9] http://www.mentor.com

[10] http://www.gnu.org/software/binutils/

[11] http://www.cadence.com