Training Design Methodology Skills at the Master's Level

Per Larsson-Edefors and Kjell Jeppson
Chalmers University of Technology
41296 Gothenburg, Sweden
{perla, jeppson}@chalmers.se

Abstract- This paper reports on the learning environment that we have set up in our master's program on embedded electronic system design for training the students' electronic system design skills. In particular we report on a new course entitled *Methods for Electronic System Design and Verification* that emphasizes the design methodology skills, that is, the process of design and verification. This new course has as prerequisite two introduction courses of the master's program, and together these three courses prepare the students for taking on the spring design project and the second year master thesis project. As a result of this continuous and coherent methodology training, design skills are reinforced and the students become confident in undertaking complex tasks, such as the master thesis project.

I. Introduction

The *Embedded Electronic System Design* (EESD) master's program at Chalmers University of Technology offers courses from system design and computer architecture to digital and analog circuit design. As an important constituent of the EESD program, the course entitled *Methods for Electronic System Design and Verification* (Methods for short) unconventionally emphasizes methodology, that is, the process of design and verification, rather than learning templates of architectures and circuits.

The motivation for the Methods course rests on the fact that electronic system designers are forced to make use of Electronic Design Automation (EDA) tools to manage design complexity and meet, for example, strict timing, power dissipation and time-to-market budgets. To apply the right EDA tools in the right context and in the right sequence has become a methodological challenge that rivals traditional design challenges intrinsic in logic and circuit design. The Methods course was created in response to this need. This paper will outline the key aspects of the Methods course, its role in the EESD program, and the skills training that is integrated throughout the EESD program.

II. The Methods Course

In the following, we will review goals, pedagogical concept, feedback and examination of the Methods course.

A. Learning Outcomes

Since the process of design and verification of complex systems is critically dependent on communication between different groups of engineers, the learning outcomes of the course do not only pertain to skills in mastering EDA tools and design flows, but also to skills in oral and written communication. The overall outcomes for the course are the following: Upon completion of the Methods course, the student will be able to …

- describe the algorithmic principles of a number of important EDA techniques, such as behavioral and logic synthesis, logic simulation, static timing analysis, and power analysis.
- describe contemporary EDA flows and their fundamental weaknesses and strengths.
- apply appropriate EDA tools to electronic system design and verification problems.
- identify what design phases EDA tools are good at handling and, conversely, in what situations a designer needs to be wary of the output result produced by the EDA tools.
- critically and systematically integrate knowledge, to model, simulate, predict and evaluate features of digital ASIC design flows, also with limited or incomplete information.
- communicate his/her conclusions, and the knowledge and rationale underpinning these, clearly and unambiguously.

B. Pedagogical Concept

To approach the learning outcomes, the pedagogical concept of the Methods course rests on three cornerstones:

- Lectures supplying the design and verification context of advanced electronic systems containing software and hardware.
- Computer lab exercises offering comprehensive hands-on training on industrially relevant design and verification problems using state-of-the-art EDA tools.
- Term paper work—an active study into research-level texts—ensuring that each student can focus on an appropriate EDA area and at the same time obtain training in reading research papers.

The lectures cover methodology topics from embedded processor design and verification to design for manufacturing of integrated circuits. The course emphasis, however, is on the RTL level and the subsequent physical implementation phases. Thus, functional verification, test bench implementation, synthesis, and timing and power analysis are key lecture topics. The lectures are supported by

ISBN: 978-2-35500-019-5

two text books available to the students as eBooks [1] and online lecture material.

The lectures are given in a top down order, in terms of abstraction level, to facilitate the computer lab exercises. For example, when the students are developing test benches and perform extensive logic simulations, the accompanying lectures are on logic simulation and functional verification.

The computer lab exercises revolve around the design and verification of an ALU for a processor. The exercises are based on Cadence tools, in particular the Encounter system, including RTL Compiler and NCSIM. Prior to the exercises, the students are obliged to plan their ALU implementation by defining a block schematic as well as a timing plan. This visual definition of the implementation has proven to be very important as this guides the students when defining the VHDL code for the ALU as well as the test bench.

There are four computer lab exercises that must be carried out in sequence (the learning outcomes for each lab are shown in Table I):

- Lab 1: ALU implementation and verification
- Lab 2: Timing-driven synthesis
- Lab 3: Timing closure and power analysis
- Lab 4: Place and route

C. Feedback

The learning outcomes are assessed in several different, complementing ways. To make the students develop skills in design, implementation, verification *and* communication, it is important to give them feedback continuously throughout the course. Students receive feedback …

- after the initial planning phase (week 2 of 7), where they define an RTL block design and a timing plan.
- after the ALU block and its test bench have been developed and verified (week 3) using the test-vector reference provided.
- after each of the four lab exercises, by way of the PingPong learning platform [2]. The students write log book entries explaining the progress made and the results obtained in their lab work. Subsequently, within a week, the teacher writes a comment to each entry. Also, each entry is graded. Conciseness is a key factor to make this feedback feasible and thus the students are required to use less than 400 words per entry.
- after the oral presentation associated with the term paper work (weeks 5 and 6).
- after the term paper draft has been submitted (week 7).

D. Examination

The total grade of the course depends on the lab work (60%) and the term paper work (40%). The first part combines grades given for initial planning, ALU RTL code and test bench, log book entries, and the final 4-page lab report. The term paper work is evaluated based on the oral presentation, the activity in discussions, and the final 4-page term paper.

III. The EESD Master's Program

The Methods course is one in a series of courses within the EESD master's program, which in turn is based on its predecessor program *Integrated Electronic System Design* which has been described in detail at previous EWME workshops [3, 4]. Due to the evolvement of Swedish industry, which is becoming more systems oriented, we considered it necessary to shift focus from ASIC circuit-design level to embedded electronic system design[1].

A. Overall Program Organization

The EESD program offers a range of courses, from electronic system design and computer architecture, to digital and analog integrated circuit design. The overall structure of this two-year program is laid out in Fig. 1. The academic year is divided into four quarters and in each quarter students take on two 7.5 ECTS courses in parallel. A dominant feature of the first year is the 15 ECTS design project where students, in groups, are designing key blocks of embedded systems; for example, embedded processors for audio decoding to circuit designs of class-D amplifiers.

TABLE I
Learning Outcomes of Computer Lab Exercises

Lab	After the lab exercise is done the student will be able to…
1	• design, implement, verify an ALU (and other units of such complexity) and its test bench, and debug mistakes inevitable in this process. • describe what is functional verification and logic simulation. • describe the purpose of test benches for functional verification. • perform basic functional verification using an industrial EDA tool for logic simulation. • elaborate on the issue of verification coverage in logic simulation; that is, how many test vectors does it really take to know that we have verified a design.
2	• describe what is ASIC cell-based synthesis, that is, logic synthesis and technology mapping. • describe what is static timing analysis (STA). • describe what is the difference between synthesizable and non-synthesizable/behavioral hardware descriptions. • perform basic logic synthesis; from hardware description language level to generic gate level. • perform basic technology mapping; from generic gate level to standard cells of a library. • perform basic timing-constrained synthesis and carry out subsequent STA analysis. • perform functional verification of synthesized implementations to verify implementation quality. • elaborate on what is the relation between initial ALU code and synthesized Verilog code.
3	• describe what is timing closure in the context of a synthesis flow. • perform a design respin in an ASIC design flow to fulfil timing requirements. • describe what is the impact on power dissipation of, on the one hand, timing constraints and, one the other hand, clock rates. • describe what is the impact on power dissipation of power analysis methodology, either using general signal switching probabilities or actual test vectors. • perform power analysis using signal statistics from a logic simulation.
4	• describe the main features of a place-and-route flow. • describe the connection between custom layout, and placement and routing of standard cells. • perform the basic steps of place and route. • describe what steps the place-and-route software handles efficiently. • describe what steps the place-and-route software does not handle efficiently, and for which the designer's understanding helps to improve design significantly.

[1] This process will be discussed elsewhere.

 ISBN: 978-2-35500-019-5

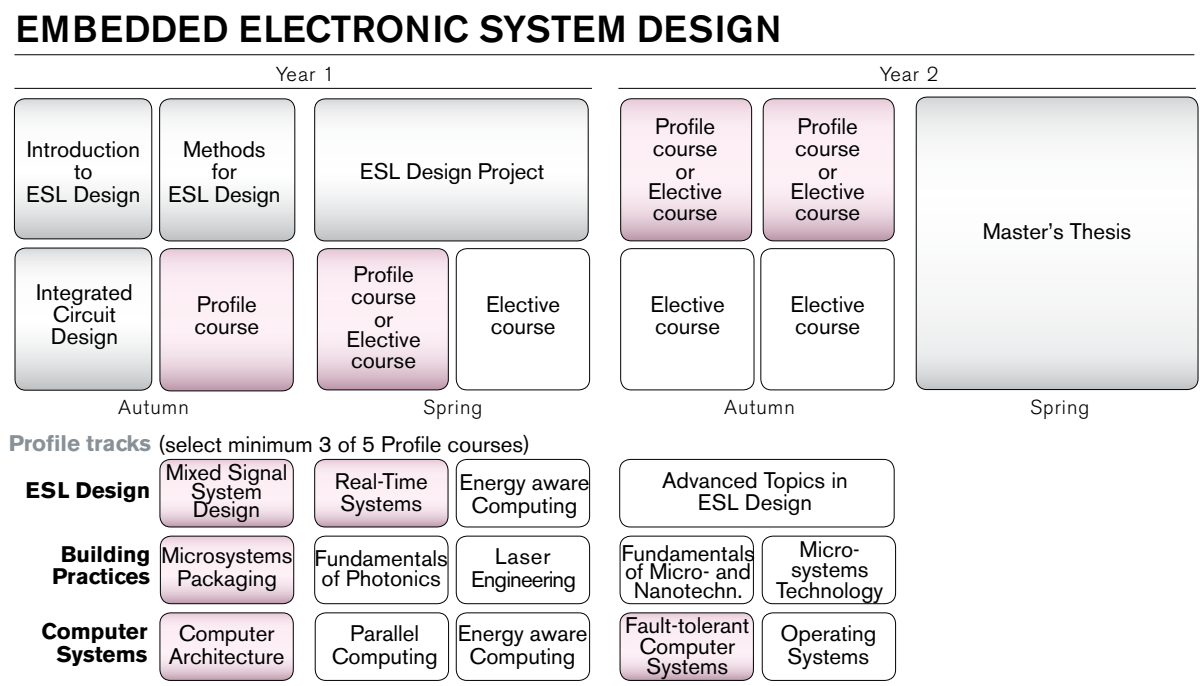

Fig. 1. EESD program plan.

The project is located in the spring term and since the project work is demanding, there are a number of pre-requisites to start the project course. The students fulfill many of the prerequisites by taking the mandatory, preparatory courses of the fall term. However, some project prerequisites are offered in elective courses, allowing the students to select elective courses based on what technical area they want to pursue in the project course.

During the first quarter two mandatory courses are offered as a portal to the program. One of these courses is the *Introduction to Electronic System Design* course which takes on the system level part of the design flow in a top-down approach. The other one is the *Introduction to Integrated Circuit Design* course that takes on the circuit level part of the design flow in a bottom-up approach.

The intention of the program and course planning is that the top-down and bottom-up approaches of the portal courses meet at a sub-system level with a complexity corresponding to advanced adder design. Together these courses allow for an exposition of cost-performance trade-offs between different technology platforms and between software and hardware[2]. Also, the two portal courses provide the platform from which the mandatory Methods course continues the training of the students.

For example, whereas *Introduction to Electronic System Design* emphasizes hardware implementation skills using VHDL, the Methods course also covers concepts like verification, test benches, and metrics such as coverage. Furthermore, whereas *Introduction to Integrated Circuit Design* emphasizes circuit implementations skills all the way up to standard cell development, the Methods course uses standard cells for cell-based synthesis and place and route, and associates the timing and power models of foundry standard cells with the analysis done at the circuit level.

B. Program Learning Outcomes

The aim of the EESD master's program is to educate engineers that can *conceive* complex embedded system specifications, *design* such systems at the computer architecture/electronic system level and *implement* and *verify* such systems using state-of-the-art integrated circuit and packaging technologies and EDA tools. The program provides a deep coverage of the methodologies critical to this process.

EESD graduates must be qualified to work as productive engineers in industrial teams designing and building state-of-the-art embedded electronic products, or to undertake graduate studies leading to a doctorate in the area of electronic system design. In particular this means that EESD graduates are …

1. proficient in the basic trade of *conceiving, designing, implementing,* and *verifying* complex embedded electronic systems; from software for embedded electronic systems to hardware for integrated circuits.
2. proficient in the use of various state-of-the-art computer-aided design tools used in industry.
3. aware of the *fundamental limitations* of both the *design tools* and *methodologies,* and the *technology platforms* that represent current best practice.
4. able to take on new technical challenges and to generate technical advancements in embedded system design or building practices.
5. able to carry out qualified industrial tasks within given constraints by applying *suitable methods,* also when in

[2] During the fall, students also have the opportunity to choose one of three semi-elective courses: *Computer Architecture, Mixed Signal System Design,* and *Introduction to Microsystems Packaging.*

an industrial context technical aspects might be secondary to constraints associated with economy and environment.

6. able to critically, independently and creatively *identify, formulate* and *solve complex problems* in the field of embedded electronic system design and to make software/hardware trade-offs.
7. able to critically and systematically *integrate knowledge* to model, simulate, predict and evaluate behavior and events, also with limited or incomplete information.
8. able to clearly and unambiguously *communicate* their conclusions and the knowledge and rationale underpinning these conclusions.

Student progression must be considered when designing a master's program and aligning its courses. In this process, program learning outcomes must be divided into course learning outcomes. Vice versa, all course learning outcomes must be mapped to the program learning outcomes. For example, outcomes 3 and 5 above have a tight connection to the Methods course in Sec. II.

The first program learning outcome indicates that the EESD program is based on the CDIO concept [4], or rather on a somewhat modified CDIV concept where "operate" has been replaced by "verify". Inherent to this concept is also the principle of constructive alignment, where students construct meaning from what they do to learn. The teacher assists in this process by aligning the learning activities with the intended learning outcomes. Our mission as teachers is to provide a learning environment that the students find stimulating enough to spend the time and effort to meet the intended learning outcomes. In this process we must align the assessment methods, and criteria for giving feedback on outcome fulfillment, to the suggested learning activities, see Fig. 2.

Fig. 2. Aligning learning outcomes, learning and teaching activities and the assessment. From [5].

C. Program Assessment

In Sweden the 3+2 two-tier bachelor/master's educational structure introduced 2007 adapts both to the structure of the Bologna Declaration and to a long Swedish and Scandinavian tradition of a homogenous one-tier, four to five year education leading to the *civilingenjör* or Diploma Engineer degree. This degree is strongly rooted in industry. The overall learning outcomes for this degree are defined by the Swedish National Agency for Higher Education (Högskoleverket, HSV). In Sweden, HSV is the public authority that oversees higher education institutions.

The learning outcomes defined for the EESD program, as for any master's program, must comply with the overall learning outcomes defined for the five-year civilingenjör education. They also have to comply with the learning outcomes defined for the first three-year bachelor program in electrical engineering (EE) upon which the program is based. HSV also requires us to be able to identify where, when, and how any learning outcome is examined to assure that the student has achieved the intended learning outcome.

D. Overall Assessment of Methodology Skills

As the project is the final course of the first year, it is important to make a comprehensive assessment of the student methodology skills at this stage. There are three assessment mechanisms in the project; two continuous mechanisms (*process* and *product*) and one final report whose overall impression, content, structure and language are evaluated. As far as the continuous mechanisms, the process assessment evaluates the manner in which the students handle project management, by way of log book entries, supervisor meetings and presentations. The product assessment focuses on the technical aspects, such as resulting VHDL code or circuitry, test protocols, and technical documentation. All in all the assessments of the project course cover program learning outcomes 2 and 4-8.

IV. SUMMARY

In this paper we have described a new course that focuses on training design methodology skill. The course is based on an unconventional approach, where emphasis is on design methodology skills, and fits naturally into any embedded electronic system design program. The main consideration behind the course—and behind the EESD master's program as a whole—is student progression. Both courses in a program and lab series within courses must be put together in a coherent way to set up a learning environment where student skills improve gradually. As a result of this process, students gain skills that allow them to undertake large and more complex tasks within project courses and within their final master thesis project.

REFERENCES

[1] L. Scheffer, L. Lavagno, and G. Martin (editors), *EDA for IC System Design, Verification, and Testing*, Taylor and Francis, Boca Raton, FL, 2006.

[2] PingPong, a web-based learning platform for administration of student learning activities. [See link below available January 2012] http://www.ull.uu.se/systems_support/learningplatform

[3] K. Jeppson, L. Peterson, L. Svensson, L. Bengtsson, and P. Larsson-Edefors, A New Master's Program in Integrated Electronic System Design, in Proc. European Workshop of Microelectronics Eduction (EWME), Budapest, May 2008, pp 130-131.

[4] K. Jeppson, L. Peterson, L. Svensson, and P. Larsson-Edefors, Implementing Constructive Alignment in a CDIO-oriented Master's Program in Integrated Electronic System Design, Proc. European Workshop of Microelectronics Eduction (EWME), Darmstadt, May 2010, pp 135-140.

[5] W. Houghton, *Learning and Teaching Theory for Engineering Academics*, Higher Education Academy Engineering Subject Centre, 2004.