TEACHING HARDWARE DESCRIPTION AND VERIFICATION

Emil Axelsson, Magnus Björk, Mary Sheeran

Department of Computer Science and Engineering
Chalmers University of Technology
Gothenburg, Sweden
{emax,mab,ms}@cs.chalmers.se

1. INTRODUCTION

Since 1999, the formal methods group of the Department of Computing Science at Chalmers University of Technology has given a course on *Hardware Description and Verification* [13]. The course focuses on the use of hardware description languages in design, and on functional verification using simulation, assertion monitoring and formal methods. The course is part of an International Masters Programme in Dependable Computer Systems, although currently the majority of the students are from the 3rd and 4th years of the Chalmers undergraduate programmes in electronics or computer engineering. About 35 students pass the course each year. The emphasis in the course is on practical approaches to hardware description and verification, with students gaining hands-on experience of both commercial and academic tools. In addition, the important concepts and algorithms underlying formal verification are taught. The fact that the course manages to reflect *both* advanced industrial practice and state of the art research is, we feel, its major strength. Here, we emphasise the benefits that the course has brought to our research group.

2. COURSE OUTLINE

The course, which is given in a period of eight weeks (the standard length of courses at Chalmers), is divided into two parts of roughly equal length. We use different hardware description languages in these two parts: VHDL in the first and Lava [5] in the second. Each part contains one lab assignment, which the students solve in pairs, and one take home exam, which must be solved individually. At the end of the course, there is also a written exam with general questions about the course content. To pass the course, students must complete all lab assignments and pass both take home exams (which each count for 25% of the total mark) and the written exam (which counts for 50%).

The VHDL part of the course starts with two lectures introducing VHDL. We know this is very little for such a complex language; however, experience shows that it is enough for the purposes of our course. Our students are mostly either electronic engineering students who have already taken a course on VHDL, or computing science majors who can learn new programming languages relatively easily. Early in the course, we also invite an experienced VHDL programmer from industry to give a guest lecture.

After these introductory lectures, the students start to solve the first lab assignment, which is to implement a stopwatch in VHDL, and to functionally verify it with testing and formal verification tools. At the same time, we start to introduce specification and verification concepts in the lectures. We start with the specification language PSL, which is relatively easy to learn and use [9]. We aim to give practical experience of using real verification tools *and* a good understanding of the important underlying concepts. For this reason, we teach the theory, data-structures and algorithms behind CTL (Computation Tree Logic) model checking. The students get some idea of what is going on under the hood of the verification tool that they are using, and they learn important concepts (such as the use of the BDD data structure) that are more widely applicable. During the 4th week of the course, the students work on a take home exam in VHDL/PSL (as well as attending lectures).

The second part of the course begins with a few lectures introducing Lava, our method and tool for hardware design using a functional language [5]. Lava is a lot smaller than VHDL, but not all students have a suitable background for learning it. The students' expertise in functional programming ranges from zero, for most of the electrical engineering students, to very high for computing science students specialising in programming languages. We have been pleased to discover, though, that the electrical engineering students generally do well in mastering this new approach to circuit description. Claessen and Pace have a similar experience with teaching Lava [4]. The second lab assignment is usually related to the take home exam of the first part, to enable the students to focus more on the new language and less on the circuitry. This also allows them to easily identify pros and cons of each language. Since we want to show current research, we spend a few more lectures on Lava, describing recent work such as adaptive synthesis methods [11]. We also describe some formal techniques used by the Lava system, with particular emphasis on the use of induction and a SAT-solver in functional verification [12].

Literature used in the course includes a VHDL book [1], instructions on Lava [5], and papers about formal verification [10, 7, 8].

Instead of the written exam at the end, we used to have individual oral exams, in which we questioned the students about their solutions to the take home exams, as well as asking general questions about course content. But as the number of students grew we had to switch to a final written exam. We feel that there are pros and cons to both methods of examination. The oral exams allowed us to spend time and effort probing each student's understanding of course content, and the discussions that we had with the best students were often wide-ranging and fascinating. On the other hand, the time limit (which was 25 minutes) excluded some forms of questions – including asking students to write more substantial hardware descriptions in VHDL or Lava. In addition, nervousness marred the performance of a few students, and a week of oral examinations was a gruelling experience for the teachers.

0-7695-2374-9/05 $20.00 © 2005 **IEEE**

3. BENEFITS OF THE COURSE

We view this course as an important part of the interface between our research group (which specialises in hardware design and verification) and its surroundings. Perhaps surprisingly, the course raises the quality of many of our activities and contacts, even outside teaching.

The existence of the course forces us to keep up with trends in industry and with the state of current commercial tools. This prevents us from becoming too academic in our views about tools, and it is why we have taught PSL in recent years. The choice of PSL over (say) System Verilog Assertions was also influenced by the availability of the Verifier tool from the local company Jasper (formerly known as Safelogic). The use of a locally developed tool is motivating for the students, who see that there are job opportunities in Sweden in this area. The company gets to put the tool through its paces and afterwards has access to a source of well-trained staff. In addition, we have, together with a colleague from Jasper Design Automation AB, research activity in the area of PSL semantics [3].

An important aspect of the course has been the inclusion of guest lectures from experienced researchers and designers from industry. These lectures expose both the students and us teachers to insights from the real world, and are often a high point of the course. For an example of a successful such lecture, see Gaisler's lecture on a structured VHDL design method [6]. We have had lecturers from Xilinx, Synopsys and Intel, and have found that companies are happy to allow senior employees to spend time on this task.

At Chalmers, some advanced undergraduate courses, including this one, are classified as "near to research". It is an explicit aim both at Chalmers and nationally in Sweden to have research influence teaching to a greater extent. So, exposing the students to our research, and that of others in the field is a major aim of the course. Thus, we use our own tool, Lava, to teach about the use of a functional programming language in hardware design. The fact that we use the tool for teaching has, in turn, influenced its development, and in particular we have needed to develop a steady stream of new case studies (since old ones are "used up" in take-home exams). This has been very stimulating for our research. One such case study for the course arose when we invited the leader of Chamers' VLSI Design group to lecture about a multiplier design that he and his colleagues had proposed. The second half of the 2 times 45 minutes lecture slot was then devoted to description of the same circuit in Lava – an exercise that has since spawned much work on the description and generation of multipliers in Lava [11]. This has led to an active collaboration with the VLSI group, which has resulted in the layout of a novel multiplier design. As a second example, a current focus in our group is on the design of parallel prefix circuits, and our initial ideas about this topic were worked out during the development of a take-home exam. The layout and fabrication of one of our novel parallel prefix circuits and of a reference design is now an ongoing project in an undergraduate VLSI Design Project course. In addition, the need to explain our research on SAT-based formal verification of hardware to an audience of non-specialists has led to new insights about the algorithms and how they should be presented. Thus, research and teaching feed each other. One result is that the take-home exams, particularly in the Lava part of the course, can be related to research topics, rather than being very standard. They can also be made open-ended, to stretch the best students. This is appreciated by the committed students, some of whom put a great deal of effort into these week-long exams, reporting that this is where they learn most. Some of these enthusiastic students then come to us for Masters Project topics (and indeed Axelsson, one of the authors of this paper, came to doctoral studies by this route, starting with a Lava-related Masters Project [2]). We also find that enthusiastic students tend to contact us later, after they have gone into industry, when they need to find information about related topics (such as the current state of commercial formal verification tools).

A possible criticism of the course is that it tries to fit too much into too short a time. Might it not be better to concentrate on just one of VHDL and Lava? At the end of the course, we always ask the students for their opinion on this, and each year they say that their learning is improved by the exposure to two different approaches, and by being explicitly asked (for example as part of the written exam) to compare and contrast the approaches. So we plan to keep the current course structure.

REFERENCES

[1] Ashenden, P. *The Designer's Guide to VHDL*, 2nd ed, Morgan Kaufmann Publishers, San Franciso, May 2001

[2] Axelsson, E. *Description and Analysis of Multipliers using Lava*, Masters Project Thesis, Chalmers School of Electrical Engineering, 2003

[3] Claessen, K., Mårtensson, J. *An Operational Semantics for Weak PSL*, Formal Methods in Computer Aided Design (FMCAD), 2004

[4] Claessen, K., Pace, G. *An Embedded Language Approach to Teaching Hardware Compilation*, Functional and Declarative Programming in Education, FDPE, 2002

[5] Claessen, K., Sheeran, M. *A Tutorial on Lava: A Hardware Description and Verification System*, April, 2000, http://www.cs.chalmers.se/~koen/Lava/

[6] Gaisler, J. *Lecture on a structured VHDL design method*, 2004, http://www.cs.chalmers.se/Cs/Education/Courses/svh/Slides/structvhdl.pdf

[7] Hu, A. *Formal Hardware Verification with BDDs: An Introduction*, IEEE Pacific Rim Conference on Communications, Computers, and Signal Processing (PACRIM), pp. 677–682, 1997

[8] Kern, C., Greenstreet, M. *Formal Verification in Hardware Design: a Survey*, ACM Transactions on Design Automation of Electronic Systems, Vol 4, Nr 2, pp 123–193, 1999

[9] *PSL Language Reference Manual*, available from http://www.eda.org/vfv/docs/PSL-v1.1.pdf

[10] Seger, C.J. *An introduction to formal hardware verification*, Technical Report, TR-92-13, University of British Columbia, 1992

[11] Sheeran, M. *Generating fast multipliers using clever circuits*, Formal Methods in Computer-Aided Design, FMCAD, Springer LNCS 3312, 2004

[12] Sheeran, M., Singh, S., Stålmarck G. *Checking safety properties using induction and a SAT-solver*, Formal Methods in Computer-Aided Design, FMCAD, Springer LNCS 1954, 2000

[13] *Hardware Description and Verification*, course page: http://www.cs.chalmers.se/Cs/Education/Courses/svh/