Teaching Future Verification Engineers: The Forgotten Side of Logic Design

Fusun Ozguner, Duane Marhefka, Joanne DeGroat
The Ohio State University
Department of Electrical Engineering
ozguner@ee.eng.ohio-state.edu
marhefka.2@osu.edu
degroat@ee.eng.ohio-state.edu

Bruce Wile, Jennifer Stofer, Lyle Hanrahan
IBM Corporation
bwile@us.ibm.com
stofer@us.ibm.com
lyleh@us.ibm.com

ABSTRACT

This paper describes a senior/graduate level course in hardware logic verification being offered by The Ohio State University in cooperation with IBM. The need for the course is established through the growing importance of logic verification to users of custom logic designs. We discuss the short-term and long-term goals for the course, and describe the course content and format. The course relies heavily on lab projects to illustrate the main concepts. Three projects and a final project review are described.

1. INTRODUCTION

Thumbing through an engineering magazine or trade journal, it's easy to find articles that begin with declarative statements about the impact of logic verification on today's chip design. Many articles (and design leaders) state that 70 percent [1][3] of the engineering effort is spent on verifying the function of the chips. Other sources site the 3-to-1 head count ratio between verification engineers and logic designers. The bottom line is that the industry now recognizes that verifying logical correctness of the design of hardware systems is a major gate to time-to-market.

Those engineers skilled in functional verification are at a premium within the industry. In the past, verification head count was achieved by requiring new engineering talent to first work in verification prior to becoming logic designers. But this verification-staffing model fails under today's design pressures. Instead, verification is recognized as a critical skill with a career path of its own. Companies that execute well in their design process are placing very high value on verification engineers who have a breadth of knowledge of key verification techniques. These verification engineers have the experience of multiple design cycles, often having learned difficult lessons from design passes that have not been functionally correct.

At the same time, engineers out of our universities are not well versed in verification. Certainly, very few engineering graduates are familiar with verification methodology, let alone the idea that verification has a career path in industry. Techniques such as pseudo-random based verification, model checking, and simulation coverage feedback are not learned until encountered at work. So as our industry adapts to the verification challenges of complex, multi-million gate chips, our infrastructure to educate engineers must adapt as well.

Universities such as The Ohio State University (OSU) are stepping up to this challenge. In the course, "Functional Verification of Hardware Designs," The Ohio State University presents a wide suite of verification techniques to senior/graduate level engineering candidates. The course tracks the evolution of verification and introduces the fundamental verification techniques. Students utilize the methods through in-depth simulation projects, where they are challenged to find both simple and obscure bugs in hardware designs.

This paper describes the course content of OSU's "Functional Verification of Hardware Designs." The course goals, infrastructure, and projects are discussed, along with future growth areas for functional verification education.

2. GOALS

From a business perspective, the short term goal of teaching the fundamental concepts of logic verification to electrical and computer engineering students is to create a pool of informed engineering graduates with some initial training from which industry can draw to help staff this critical hardware development area. A large percentage of electrical and computer engineering graduates who are interested in digital logic development are only interested in the design aspect of the development cycle. The mismatch between verification staffing requirements and graduates' skills and expectations results in job interviews where the student first has to be educated on what verification is, why it is important, and what makes it a challenging career choice. Students often confuse verification with manufacturing test or with the Spice analysis done on a senior year design project. New verification engineers in industry are faced with a large amount of learning that could have been started at the university. The intention of the verification course is to educate the students so that they can make a more informed career choice, to provide some initial training on verification, and to evaluate their aptitude for verification before accepting a job.

The long-term business goal is to seed academia with the need for education and research focused on logic verification. Potential university research topics associated with verifying the rapidly increasing complexity of logic designs include verification of

DAC 2001, June 18-22, 2001, Las Vegas, Nevada, USA.

asynchronous boundaries, the value of design for verification, and the usefulness of various coverage metrics. This research would provide more in-depth training for graduate students and create a forum to investigate topics with long-term influence on verification methodology.

The academic goals of the verification course are drawn from the fundamental elements of a verification project. The students should understand the problem to be solved, experience current verification methods, use industry verification tools, face challenges that make logic verification a difficult engineering problem, and be shown areas of future development. To accomplish this in a single term undergraduate/graduate course the intent of the lectures and labs is to have the students:

- Use an industry simulator
- Understand and use a test bench
- Write a test plan
- Write test cases, managing stimulus and checking response
- Debug and understand different types of logic bugs
- Determine when enough verification has been done
- Handle a configurable design
- Make a final project presentation for management review to release the design

Additional topics will be introduced to the student such as formal verification, design for verification, and simulator technology.

3. COURSE STRUCTURE

The course "Functional Verification of Hardware Designs" developed at the Electrical Engineering department at OSU in collaboration with IBM, is a 3 credit hour course with a 2-hour per week lecture and an open lab component. It is open to seniors and graduate students. The prerequisite is a VHDL-based computer design course (Theory and Design of Digital Computers II). The course is unique in that industry-university collaboration and interaction is carried out through the running of the course as well as the development. The IBM Verification Laboratory houses 16 IBM pSeries workstations used for the course projects. The instructors also have identical machines in their offices. Students work on 3 projects with increasing complexity. The projects are a very important component of the course and will be described in more detail in the next section. The course is supported by a teaching assistant, who spent 3 months as an intern with the system verification team at IBM Rochester, developing the projects and also learning to use the tools. The lectures cover concepts such as verification approaches, tools, verification plan and strategies, and architecting test benches. The textbook for the course is "Writing Testbenches: Functional Verification of HDL Models" by J. Bergeron [2]. Some of the lectures are given by guest speakers from IBM. This enables the students to learn from real-world practitioners and interact with industry representatives, before making career decisions. The students learn verification skills by actually verifying designs (with inserted bugs) in the lab. The students also get remote support from IBM engineers. They can send e-mail or interact through a web page.

There are no exams in the course. Grading is based on performance in verifying designs. Students are evaluated throughout the course and they give presentations during the last week of the class to a panel from IBM, which also evaluates their performance.

4. PROJECTS

The designs that the students will verify during the course serve as the course's heart and soul. The art of verification is not a cut and dried science that one can memorize in ten easy lessons, but is a journey of discovery for every engineer who chooses to enter the field. These projects are intended to start the students down that road by showing them how challenging verification can be.

Designing the projects has been challenging in several respects. The first challenge has been to make the designs complex enough to disallow brute force verification without becoming so difficult as to be impossible given the time and resources to be applied. Most design courses today include some verification of the designs, but the designs are typically simple enough that verification can be accomplished by testing all possible inputs. This leads students to believe that verification is not interesting. The second challenge has been to target the inserted bugs to different verification strategies. Including bugs that are only likely to be found if a particular strategy is employed will emphasize that it is generally better to rely on a variety of techniques than to expect one technique to find all types of problems. The final challenge has been how to determine a fair measure of success and how to keep the students from sharing information on bugs found. Once a bug has been identified, creating a test or strategy to find that bug is an infinitely easier problem than thinking to look for bugs where one doesn't know they exist.

The projects are a series of 3 designs developed within IBM for the purpose of education. Each design builds upon the previous design. The first two projects are allocated two weeks apiece and the third project is allocated 4 weeks. The first two designs will be completed individually and the third design will be verified by groups of three students. The course teaching assistant will serve as the "designer" for the projects and provide "fixes" for correctly identified bugs. To mimic actual working conditions bug fixes will not be available immediately upon request, but will be available during regularly scheduled design drops. At least one time during the course an incorrect "fix" will be delivered to emphasize the need to verify that "fixes" are correct and complete.

The first project is a simple calculator, which can accept four commands: add, subtract, shift left, and shift right. The calculator has four input ports and four output ports, but will only accept one command at a time per port. The next command will not be accepted until the first command's output appears on one of the output ports. The command and the first operand appear on the same cycle with the second operand appearing on the cycle immediately following that. The students will be given the design, the test bench, design documentation, and a test plan. They will be expected to write the test cases, execute the tests, debug the failures, and identify the bugs. The bugs include four functional problems and one performance problem.

The second project builds upon the first by adding a two-bit tag to each command and allowing up to four commands to be accepted per port. Commands may be processed out of order. The students will be provided with the design, the test bench, and black box design documentation. Additional micro-architectural design documentation will be available from the teaching assistant upon

request. In addition to the activities performed for the first project, the students are expected to write a test plan and develop a coverage metric for the design.

The third project transforms the design by adding a sixteen register memory, changes all arithmetic commands to read-from and write-to registers, and adds four new commands: fetch, store, branch if zero, and branch if equal. The design allows out of order execution of commands as long as program consistency is maintained. The students will be provided with the design and black box documentation. Additional documentation is again available from the teaching assistant upon request. The students will be expected to write a test plan, implement a stimulus and response methodology, provide for preloading of the registers, implement error injection, and develop a coverage metric for the design.

The final week of the class will be devoted to presentations by the student groups to a panel of verification engineers. The presentations are intended to mimic the function of verification reviews. The students will be expected to present their verification plan, explain their choice of methodology, present the results achieved, and discuss future improvements and lessons learned. The panel will actively participate in the review to help evaluate how well the students have understood the methodologies presented during the course and the tradeoffs they made by choosing a particular path.

5. CONCLUSION

With chip complexity constantly increasing, the difficulty, as well as the importance, of functional verification of new product designs is on the rise. The growing difficulty of verification has rendered inadequate the traditional training of new engineering graduates in this area. The Department of Electrical Engineering at The Ohio State University has developed a new course, "Functional Verification of Hardware Designs," to respond to the call from industry for skilled verification engineers. The course is unique in the strong collaboration with industry in its development. The benefits of this collaboration include the opportunity for study in a verification environment similar to that used by actual verification teams in industry, the tuning of the designs for the students to use for practicing verification techniques, and interaction with verification engineers and managers through guest lectures and verification reviews. Additionally, a student teaching assistant who aids the professor with the course has been trained in verification techniques during an internship with a verification team in industry. This practice should help to keep the course current with actual industry practices.

While the verification course offered by Ohio State will provide students with a start in the art of verification, students may wish to explore some topics in more depth through independent study. In particular, test benches for the course will be written in the C programming language. Students may wish to further explore other tools such as Verisity's Specman or Synopsys' VERA System Verifier. They may also wish to further evaluate the strengths of formal verification tools and code coverage metrics, and further learn to achieve the right balance between white box and black box testing.

The first offering of the course, in Spring Quarter 2001, has generated considerable student interest, with 40 students currently enrolled. Hopefully, this new course will prepare students to make an immediate contribution to the first-time success of design projects when they take their first jobs in industry.

6. REFERENCES

[1] Anderson, T. and Bhagat, R., "Tackling Functional Verification for Virtual Components," ISD Magazine, November 2000, pg. 26.

[2] Bergeron, J., Writing Testbenches: Functional Verification of HDL Models, Kluwer Academic Publishers, 2000.

[3] Rashinkar, P. And Singh, L., "New SoC Verification Techniques," Abstract for tutorial, IP/SOC 2001 Conference, March 19, 2001.