The Future of Engineering Education
2024 Annual Conference & Exposition
Oregon Convention Center
Portland, OR . June 23 - 26, 2024
ASEE
Paper ID #42039

An Industrial Tool Based Graduate Class in ECE Design Verification Curriculum

Shruti Sharma, Portland State University

I am a part time Instructor at Portland State University of the Assertion Based Verification Course. I work full-time at Intel.

Prof. Xiaoyu Song, Portland State University
Mohamed Ghonim, Portland State University

A graduate Student at Portland State University, currently focused on validation and verification methodologies in my research.

Jin Zhang, Synopsys Inc.

An Industrial Tool-based Graduate Class in ECE Design Verification Curriculum

Shruti Sharma, Mohamed Ghonim, Xiaoyu Song, Jin Zhang

Introduction

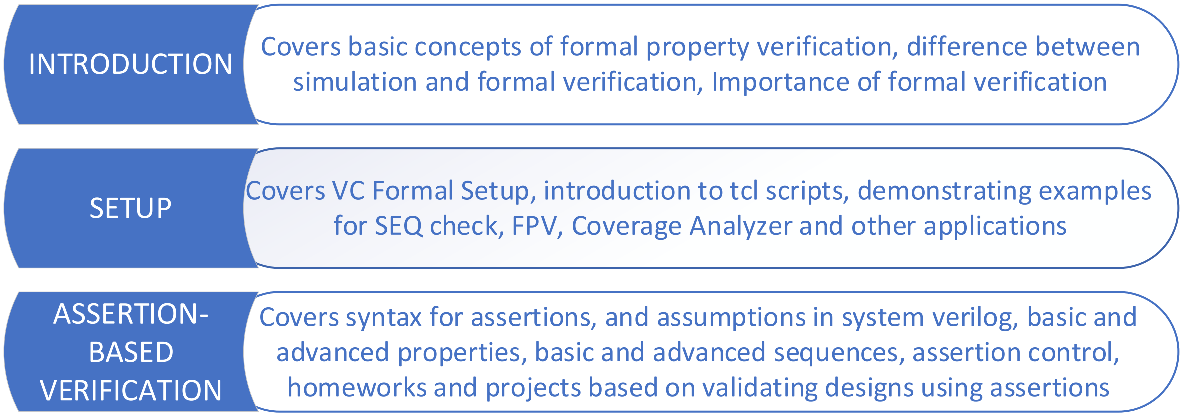
Verification of modern-day computing systems is becoming a bottleneck, taking up to 70 % of the time and effort in the design cycle [1]. Traditional approaches to design verification include simulation, where validation engineers create a test bench environment and develop test cases and checkers to monitor the behavior of the design. This presents several problems, including the long time taken to create the test bench infrastructure, long runtimes, and always posing the question of how many test cases would be enough to validate the design thoroughly. Longer runtimes for simulation are usually resolved by using techniques for simulation acceleration, like hardware emulation, which allows us to run complex and more thorough test cases to validate the design. However, there still exists a question of whether the test cases created were enough to validate the design, as simulation can help indicate the presence of bugs, but it does not guarantee the absence of bugs [2].

For the same reasons above, formal verification is becoming a widely accepted methodology for the verification of digital designs in the industry. Formal verification is a way of mathematically analyzing the design and providing proof of the correctness of the design for all states of execution, as opposed to validating the design for certain test cases. Formal verification methodology, however, is not intended to replace the simulation. For instance, formal verification can only work on relatively smaller design IPs. Therefore, formal verification is used along with the simulation to validate the digital designs fully. Design IPs are validated using formal verification by providing all sets of inputs possible, and later, multiple IPs are combined to validate the entire system. More details on simulation and formal validation-based approaches in hardware design verification are presented in [3].

Many writings have evolved that emphasize the introduction of formal verification in undergraduate and graduate programs. For instance, [4] and [6] put emphasis on the role of formal verification in software development while introducing the formal verification methodology in the graduate program. In [5], three projects are introduced for validating computer pipeline components using formal verification in a computer architecture course. [7] introduces formal verification in an HDL-based course for graduate-level programs.

In this paper, we focus on designing a graduate-level course to introduce assertion-based formal verification of digital designs. The course structure is designed to introduce the application of formal methods using an industrial VC Formal verification tool. The course introduces the what, why, and where of formal verification through a demonstration of examples using VC Formal. Multiple homework assignments are designed to give students hands-on experience with the tool and familiarize them with formal property verification using assertions. A final project is introduced, which allows students to create a validation plan for a design of their choice and validate it using the applications of the VC Formal tool.

While designing the track for assertion-based verification, we designed the coursework to be easy to understand for students with no background in formal logic, no previous knowledge of formal property verification, and no background in the VC Formal tool. For that reason, the course is divided into three phases: Introduction, Setup, and Assertion-Based Verification. The introduction phase is designed to familiarize students with formal property verification concepts. The setup phase introduces students to VC Formal setup and demonstration using pre-made examples. The final phase is assertion-based verification, where students are taught concepts of assertions, assumptions, and coverpoints, and hands-on practice is provided using multiple homework projects and assignments. Figure 1 below gives an overview of three different phases in which the course is organized.



INTRODUCTION

Covers basic concepts of formal property verification, difference between simulation and formal verification, Importance of formal verification

SETUP

Covers VC Formal Setup, introduction to tcl scripts, demonstrating examples for SEQ check, FPV, Coverage Analyzer and other applications

ASSERTION-BASED VERIFICATION

Covers syntax for assertions, and assumptions in system verilog, basic and advanced properties, basic and advanced sequences, assertion control, homeworks and projects based on validating designs using assertions

Figure 1. There are three phases in which the course is organized: Introduction, Setup, and Assertion-Based Verification. For the successful execution of the course, the introduction should be covered first, and set up and assertion-based verification can be covered in parallel or one after the other.

The rest of the sections describe the course overview and future scope of the course.

Pedagogy for Teaching

The purpose of the course is to provide the students with the opportunity to get familiar with the practical tools used in industry settings, thus bridging the gap between academia and industry. The course focuses on the VC Formal tool from Synopsys, which is widely used in the industry for formal verification of digital designs. Through this course, we expect students to be proficient in formal property verification using assertions, that is, deciding what assertions and assumptions to write, how to write the assertions in System Verilog, and how to use the VC Formal tool to validate the assertions.

The course covers various VC Formal applications, including sequence equivalence check (SEQ), formal property verification (FPV), connectivity check (CC), automatically extracted property (AEP), coverage analyzer, and formal x-propagation verification (FXP). The FPV and SEQ applications are widely covered in the course as they are most relevant to assertion-based verification. Three homework assignments are created for hands-on practice on FPV and SEQ applications, and a final project is organized, allowing students to practice the concepts learned in the classroom. The homework assignments provide students with an architecture specification of a digital system and a System Verilog implementation of the design with intentional bugs implanted in the code. As part of the assignment, students are expected to use different VC Formal applications and write assertions and assumptions to find the bugs in the design. For the final project, students are expected to create a formal validation plan for a design of their choice and implement the plan using any of the applications in VC Formal.

The course is part of the design and verification track in the graduate curriculum at Portland State University and is designed to help students understand the aspects of assertion-based verification. This course does not include the concepts related to simulation-based verification, emulation, and formal logic for system verification. The course is specifically designed to stress the importance of assertions and provide experiential knowledge for the application of formal verification using the VC Formal tool.

The course is divided into three main phases:

1) Introduction: We considered several things while deciding on the flow of executing the course. First, we wanted to ensure that anyone who has not previously taken a simulation or formal verification class can understand the concepts. Therefore, we decided to include some introductory lectures on comparing the simulation, emulation, and formal verification methodologies. We designed the examples from scratch to demonstrate the difference to students.
2) Setup: The second part of the course provides an introduction to the VC Formal. Some live examples are created and shared with students with step by step set up of the VC Formal tool. Various reference examples are created from scratch as part of this course to familiarize students with the VC Formal applications. A reference guide is formulated for students for each VC Formal application using the materials available on Synopsys Solvnet.
3) Assertion-Based Verification: After getting familiarity with the VC Formal tool, we decided to move on to concepts of assertion-based verification. Over the course of multiple lectures on assertion-based verification, we walked through many examples and provided students with three homework assignments where students were provided with a buggy code and digital design specification, and they had to write assumptions and assertions and run them through the VC Formal tool to find the bugs in the design.

Out of the three phases in the course, the second and third steps do not have to be performed in sequence. When we introduced the VC Formal and its applications, students were not familiar with the assertions. However, we had to use the assertions to provide a demonstration for the VC Formal. The expectation was that students would use the VC Formal later in their homework assignments after we started the assertion-based verification. After receiving the feedback from students, in the future, we may end up merging the second and third phases of the course and create separate lab assignments for the VC Formal, which need to be completed along with homework assignments as the course progresses.

Evaluation and Improvement

For the goal of improving the overall experience of the course in the future, we designed certain questions for the students to evaluate the course. The course was organized as one section, and was attended by 41 students. As a part of collecting the feedback, we sent out an electronic survey at the end of the course, which sought students' opinions in 4 major categories:

- Relevance to industry
- Impact on job interviews
- Efficiency of learning methods
  Confidence in using the VC Formal

We asked students to rate "relevance to industry" and "efficiency of learning methods" on a scale of 1-5. To understand the course's impact on job interviews, we asked students to provide feedback on whether the course was directly or indirectly helpful in job interviews or wasn't helpful. We also asked students to rate their confidence in using VC Formal applications on a scale of 1-10 to understand if there is a requirement to introduce more projects or re-evaluate existing projects. Figure 2 gives the summary of the student survey over the four categories mentioned in the paragraph.

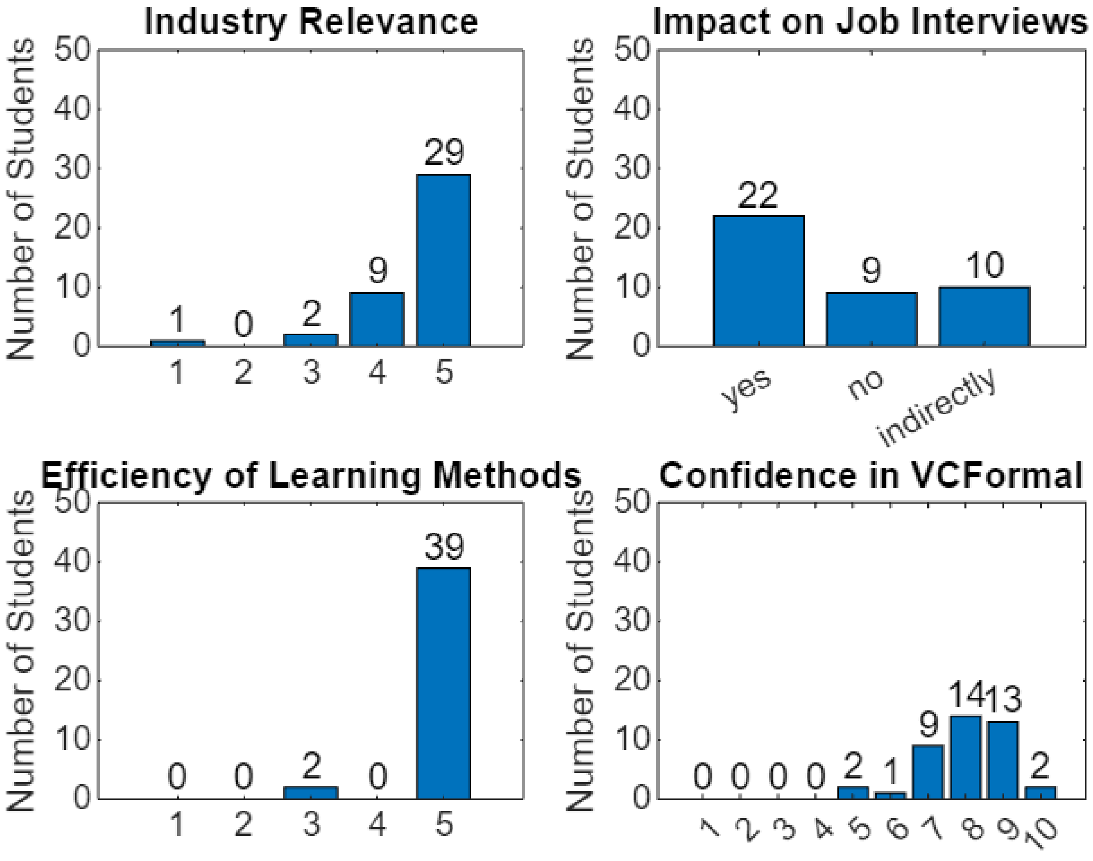


Figure 2. Course Evaluation by 41 students who attended the class. The course evaluation is done for 4 major points: 1) Course Relevance to Industry, 2) How helpful did students find the course to be in job interviews (Impact on job Interviews), 3) How efficient did

students find learning methods to be and 4) How confident do students feel about using VC Formal.

From the survey results and comments provided, we found that students who gave lower ratings to industry relevance have not had a chance to work with industry or are from different domains. Most students voted for the course to have a positive impact on the job interview experience. Students who voted for “indirect” pointed out that interviewers showed interest in assertion-based verification but didn’t ask for formal verification applications. Instead, assertion-based verification in simulation setup was a preferred methodology for job positions. In terms of the efficiency of learning methods, students pointed out that homework projects were the most helpful, followed by written lectures and tutorials. Students also highlighted that they preferred written and video lectures over the VC Formal and System Verilog Assertion user guide because of the concise information available in the notes. Students who gave a lower rating to “confidence in VC Formal” highlighted that they would want more VC Formal applications to be included in the course.

Future Work

Based on the feedback received through the student survey, we noted down several limitations of the course, and we strive to improve those in the future. First, although we found that the course was very helpful for the students in interviews, some students highlighted the need to include assertion-based validation in simulation in the course curriculum. We plan on adding dedicated lectures for assertions-in-simulation, and we also plan on adding assignments related to that. It will be of great interest to see if that improves the course's relevance to the industry. The second major feedback received from the survey asked to include more VC Formal applications in the course content. We plan on adding Data Path Validation using Hector, formal register verification, and security verification using VC Formal as part of the course in the future. Another feedback was related to assertion control functions and their applications. As the control functions are part of the simulation flow, we only briefly introduced them but did not have any dedicated lectures or practice material. We plan on adding these to the course curriculum in the future as part of assertions-in-simulation. We expect to see improvement in the survey for the “Industry Relevance,” “Impact on job interviews,” and “Confidence in VC Formal” after the updates in the course work.

Conclusion

Through this course, our goal is to familiarize the students with the latest verification technologies used in the industry. We started by creating a framework for the course such that students get the most experience with VC Formal. Subsequently, we decided on what theoretical aspects of formal verification and simulation methodologies we should include in the course. An efficient course flow allowed students to get hands-on experience with VC Formal. Industry-like assignments and homework designed for the course allowed students to learn how to use the VC Formal tool to debug digital designs efficiently. With a focus on assertion-based verification, we were able to expose students to the application of formal logic in real-life digital designs.

References

[1] Fusun Ozguner, Duane Marhefka, Joanne DeGroat, Bruce Wile, Jennifer Stofer and Lyle Hanrahan. Teaching future verification engineers: the forgotten side of logic design. In *Proceedings of the 38th Annual Design Automatic Conference,* pages 253-255, 2001.

[2] G. J. Pace and J. He. Formal reasoning with verilog HDL. In *Proceedings of the Workshop on Formal Techniques for Hardware and Hardware-like Systems*, Marstrand, Sweden, June 1998.

[3] William K Lam. Hardware design verification: simulation and formal method-based approaches. *Prentice Hall PTR,* 2005.

[4] Mansur Khazeev, Manuel Mazzara, Hamna Aslam and Daniel de Carvalho. Towards a broader acceptance of formal verification tools: Thie role of education. In *The Impact of the 4th Industrial Revolution on Engineering Education: Proceedings of the 22nd International Conference on Interactive Collaborative Learning (ICL2019)-Volume 2 22,* pages 188-200. Springer, 2020.

[5] Miroslav Velev. Integrating formal verification into an advanced computer architecture course. In *2003 Annual Conference,* 2003.

[6] Guido De Caso, Diego Garbervetsky and Daniel Gor. Integrated program verification tools in education. *Software:Practice and Experience,* 43(4):403-418, 2013.

[7] Mehran Massoumi and Assim Sagahyroon. ASIC verification: Integrating formal verification with hdl-based courses. *Computer Applications in Engineering Education,* 18(2):269-276, 2010.