Developing Infrastructure for an Advanced Application Specific Integrated Circuit (ASIC) Verification Course

Meeta Yadav[1], Ravi S Jenkal[1], Bob Oden[2], Pete LaFauchi[2], Paul D. Franzon[1]

[1.] Department of Electrical and Computer Engineering, North Carolina State University, Raleigh, North Carolina U.S.A.; <myadav|rsjenkal|paulf>@ncsu.edu

[2.] Mentor Graphics
Durham, North Carolina U.S.A.; <Bob_Oden|Pete_Lafuci>@mentor.com

ABSTRACT

The increased number of transistors per unit area, in keeping with Moore's Law, has brought with it a concomitant growth in the complexity of designs. This increase in design complexity has lead to an increased probability of errors. Also, the cost of these errors, if they go un-detected, increases by virtue of both design and manufacturing costs. Verifying the ASICs (Application Specific Integrated Circuits) before fabrication, which involves detecting errors in the design and validating the functional correctness can avoid the increased costs of the design. Verification has become an integral part of the design process; with 70% of design time spent on verification. The increase in emphasis by industry towards verification and the increasing push by researchers to bridge the verification and design gap has resulted in a need for students to have the skill set required to verify complex hardware designs and to work on verification research ideas. While there are a variety of teaching approaches, which emphasize different topics, and aspects of verification, we used a project-based teaching approach to teach the principles and techniques behind ASIC Verification. We present our two main strategic objectives in this paper. The first objective was to develop infrastructure to enable students to explore verification concepts and acquire skills through hands-on problem solving. The second objective was to develop a project-based teaching approach for verifying large complex systems.

1. INTRODUCTION

The ASIC Verification curriculum was developed keeping in mind the growing needs of industry and research for well qualified verification engineers. The curriculum was developed to give the students a systematic and methodical approach towards verification of complex digital systems and to develop an understanding towards the challenges of verification.

With the help of Mentor Graphics we developed a set of incremental laboratory exercises that were aimed to teach the students key concepts of layered testbenches, constrained random stimulus generation, functional coverage and assertions. To give the students hands on experience of verifying a complex digital system, a pipelined version of the LC3 processor was developed and errors were introduced in the design. The project objective was for the students to create a verification environment based on the principles mentioned above to detect bugs, analyze them and report coverage. The incremental approach of the labs and projects enabled the students to develop verification skill and also gain an understanding of challenges of verification.

2. INFRASTRUCTURE DEVELOPED

We developed Infrastructure in the form of laboratory and project exercises, to teach verification concepts of layered testbenches (reusable), constrained random stimulus generation, functional coverage and assertions.

A. Laboratory Exercises

We developed an incremental set of laboratory exercises that were aimed to utilize the capabilities of Mentor Graphics' QuestaSim tool. This tool is a part of the Questa Advanced Functional Verification Platform and is the latest tool in Mentor Graphics tools suite with constructs for mixed-language functional verification support. The laboratory exercises were structured such that they followed the delivery of the lecture materials and helped reinforce concepts with practical examples. A verification environment was developed to verify a basic Arithmetic and Logic Unit (ALU). The environment consisted of a driver, a generator, a scoreboard, a monitor and a checker as shown in Figure 1. For the lab exercises, parts of the environment were selectively taken out and students were walked through the steps of developing the missing part. The chronological order of laboratory exercises assigned to the students is as follows:

1. In the first exercise students had to develop a simple testbench with directed testing. This was done to provide the students an exposure to a SystemVerilog code structure and the usage of the Questa tool.
2. Incremental to the previous lab exercise, the second lab involved creation of a simple object oriented stimulus with multiple levels of transmit and receive structures with checking features. The structure was generated by stimulus packetization.
3. The third lab involved creation of a fully object oriented and layered (Generator, Base Scoreboard, Driver) input side to provide for controlled random stimulus generation. The test bench was broken up into different classes.
4. The fourth lab required creation of a fully object-oriented (receiver, checker) output side which, combined with the previous laboratory exercise, provides a complete example of fully-layered testing environment.
5. The fifth lab exercise involved addition of functional coverage points to the previous testbench to analyze the efficiency of the stimulus constraints. This also involved the visual analysis of coverage and its manipulation using Questa tool.
6. The final exercise required creation of simple assertions to provide an alternate means of error checking.

   By the end of the exercises the students were well positioned to tackle fairly complicated verification exercises on their own.

B. Project Exercises

We chose a project based approach to convey to students the complexity of verifying a complex digital systems, and the challenges associated with the task. A pipelined version of the LC3 microprocessor was used as the Design Under Test (DUT) and a verification environment as shown in Figure 1, was developed to give students hands on experience of verifying an ASIC. The main objective of the project was to enable the students to create a verification environment with the following features:

1. Reusable test bench (Object oriented and layered).
2. Exhaustive testing of different design features.
3. Constrained random stimulus generation.
4. Generating directed tests for corner cases.
5. Coverage .
6. Assertion based verification.

The chronological order of the project assigned to students was as follows:

1. In the first project the verification environment provided to the students contained a scoreboard, monitor (receiver) and a checker. The students were required to write a stimulus generator to exercise the design and detect the bugs. After detecting each bug the students reported the bugs with an analysis report. On correct identification and analysis a bug fix identification number was provided to the students that over-rode the bug in hardware.
2. In the second project the students had to build a verification environment based on the principles of layered test bench construction, constrained random stimulus generation and reusable testbench design. The students had to develop the generator, driver, scoreboard, monitor and checker and verify the DUT.
3. In the third project the students reused the verification environment created in the previous project and added assertions and functional coverage points to the design.

This combined set of skills gave the students the necessary and in-depth understanding of the required verification principles.

3. COVERED MATERIAL

We covered the several important aspects of verification during the duration of the course. The topics covered ways of creating a verification plan, several common verification methodologies [3], creation of constrained random stimulus generation [1], using interfaces, developing layered testbench design [1], performing functional coverage [4], checking for coverage convergence and developing assertions [5].

4. CONCLUSION

We have summarized the infrastructure developed and the project-based approach adopted by us to teach a course in ASIC Verification. The course equipped the students with the skills to verify large complex designs, gave students an insight into the challenges of verification and emphasized to them the importance of bridging the design and verification gap. The course also peaked research interest of students in this field. Figure 2 illustrates the feedback from the course during Fall of 2007 (65 students), 52% students were interested in pursuing research in the field of ASIC verification and 72% of students were interested in pursuing a career in the field of ASIC verification.

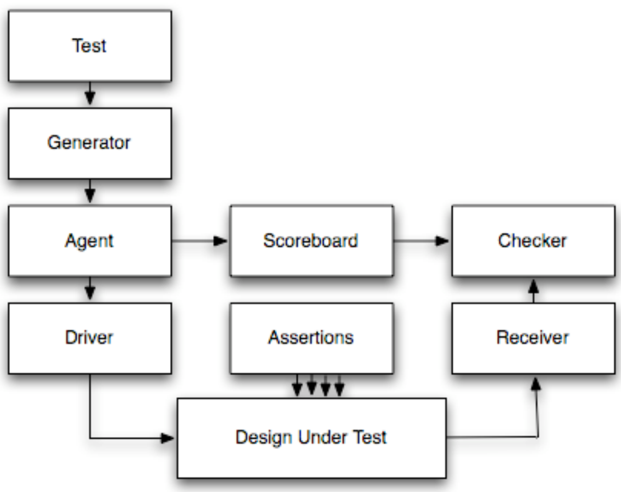

Figure 1: Verification environment with all the layers (Courtesy: Chris Spear. SystemVerilog for Verification: A Guide to Learning the Testbench Language Features [1])

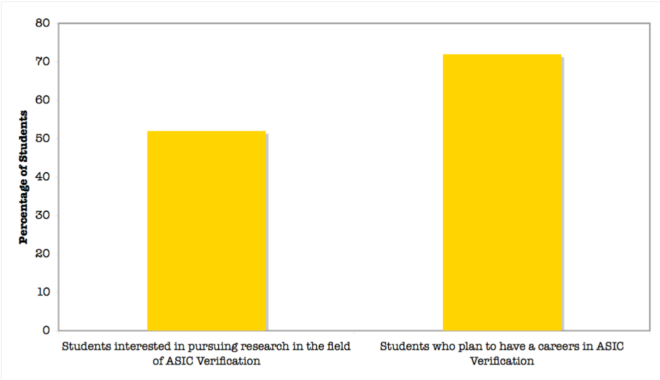

Figure 2: Feedback of students from Fall 2007. 52% students are interested in ASIC Verification research and 72% students are interested in pursuing a career in the field of verification.

5. ACKNOWLEDGEMENT

The authors would like to thank Mentor Graphics, especially Mr. Brad Potts and Mr. Ian Burgess, for their generous support and feedback for the development of this course.

REFERENCES

[1] Chris Spear, SystemVerilog for Verification: A Guide to Learning the Testbench Language Features Springer, 2006

[2] Stuart Sutherland, Simon Davidman, Peter Flake and P. Moorby, SystemVerilog for Design: A Guide to Using SystemVerilog for Hardware Design and Modeling Springer, 2006

[3] Janick Bergeron, Eduard Cerny, Alan Hunter, Andy Nightingale Verification Methodology Manual for SystemVerilog, Springer, 2005

[4] Accellera SystemVerilog 3.1a , Language Reference Manual Accelleras Extensions to Veril og

[5] Ben Cohen, S Venkataraman, Ajeetha Kumari, System Verrilog Assertions Handbook, VhdlCohen Publishing, 2006