

# Integrating Validation and Verification in the Digital Design Curriculum

Shrirang M. Yardi and Michael S. Hsiao  
Department of Electrical and Computer Engineering  
Virginia Tech, Blacksburg, VA 24061.  
*yardi@vt.edu, mhsiao@vt.edu*

**Abstract**—This paper presents a module-based approach on integrating verification terminology, concepts and examples into the current computer engineering design curriculum. The primary goal is to equip students with the necessary verification and validation skills which are either ignored or are dealt with in an ad-hoc manner in current design courses. By making these courses verification-centric and emphasizing design-for-verifiability, we can produce highly qualified designers to handle the ever-larger and complex designs of the future.

## I. INTRODUCTION

Continuing advances in semiconductor technology and system integration have resulted in the development of extremely large computing systems that are pervasive in various aspects of our lives. Consequently, designing of such systems is at the core of most of the digital design courses today. However, this approach does not accurately reflect the technical needs of the industry, especially in the area of design verification. According to the International Technology Roadmap for Semiconductors (ITRS) [1], two-thirds of the total effort in a typical design today is spent on verification. To meet this challenge, qualified next-generation designers are desperately needed. However, existing design courses expose students only to the design of these systems and do not include any verification concepts. Therefore, we believe that in addition to an increased focus on verification research, training students to see that verification is a critical step in the overall design is an essential parallel in shaping the design culture.

## II. DILEMMA OF THE CURRENT CURRICULUM

We first briefly review the current design courses at Virginia Tech. At Virginia Tech, the two main design courses are ECE3504—Digital Design Part I, and ECE4514—Digital Design Part II [2]. ECE3504 focuses on the following topics: combinational and sequential logic, logic minimization concepts, FSM design, asynchronous circuits, etc. and, the design of digital circuits using standard integrated circuits and PLDs (programmable logic devices). The follow-on course ECE4514 introduces students to a hardware description logic (e.g., Verilog) with an emphasis on system-level design concepts, RTL design description and commercial synthesis/simulation tools. The projects in both courses consist of design specifications, typically of increasing complexity as the course progresses. In both courses, students use simulation as the main workhorse for checking their design projects. No other forms of verification, such as coverage-directed, formal verification, or semi-formal verification are offered to the students to help enhance their debugging ability. These trial-and-error testing

habits result in students spending a disproportionate time on debugging designs, missing corner cases and failing to grasp the importance of a well-planned design-and-verification flow.

The lack of focus on verification concepts in these courses results in a serious deficiency in the students' understanding of the role of verification and its importance in the overall system design flow. Although there is a very active verification research community at Virginia Tech as well as other universities, very little of it is transferred to senior undergraduates that enter the industry. Further, the graduate-level courses that focus on verification generally target research goals, and thus they rarely have a design-and-verification focus in them.

In light of the above discussion, we believe that infusing sound verification methodologies into existing design courses is essential to introduce important verification and validation concepts to senior undergraduate students. Rather than teaching the core theory behind verification, our goal is to introduce fundamental concepts so that these designers appreciate and understand the importance of verification and learn to apply these concepts to reduce their debugging effort. This will prevent them from following ill-planned design flows and qualify them for the realities of the design industry.

## III. NEW COURSE DESIGN

This section describes our plan behind integrating verification concepts in the current design-centric curriculum. Our main goal is to do this in a non-intrusive manner, without heavy modification of the existing material covered by the instructors to teach these concepts. To this end, we are developing an open web-site with example projects, tutorials and sample lecture notes that can be downloaded and used by interested faculty. Figure 1 summarizes the structure of the proposed course.

### A. Course Structure

We have structured our course material into different modules such that each module can be integrated easily into the concepts introduced in traditional design courses. Each module consists of tutorials, lecture notes, and projects that provide the students both the background and hands-on experience with the relevant concepts. In the following, we provide details about each module.

**Module 1: Testbench Organization and Design:** Writing an effective testbench for a design is usually the first step towards creating a verification infrastructure. Since testbenches are not part of the actual design, students typically spend the least amount of time generating test cases. This module

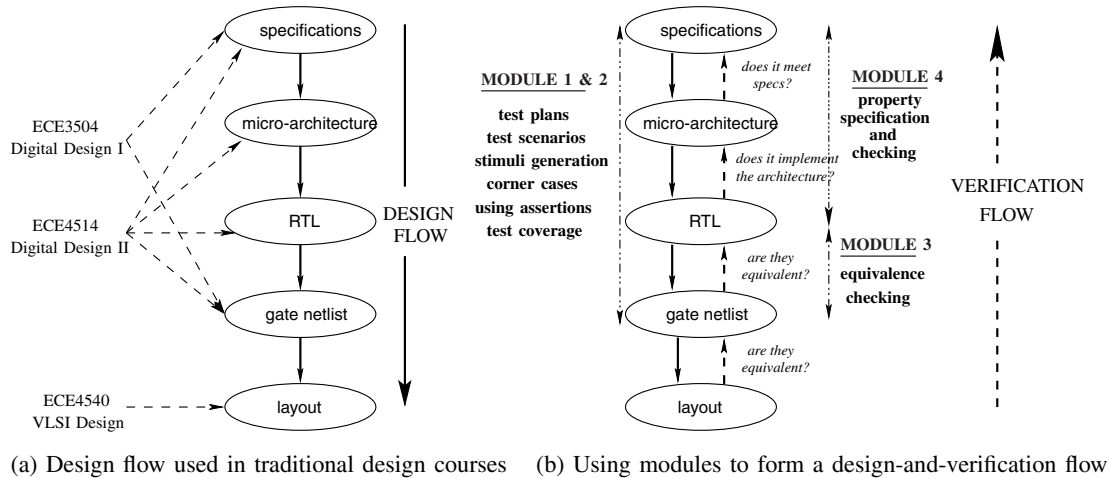


Fig. 1. Integrating verification into existing digital design courses

introduces the idea behind a “good” and “bad” testbench. In particular, it covers concepts like anatomy of a testbench, stimuli generation, test scenarios, consideration of corner cases and understanding/interpreting test results. The project for this module consists of two designs, a small circuit that can be tested exhaustively (for example, an 8-bit full adder) and a larger (possible hierarchical) design that cannot be tested exhaustively (for example, a 32-bit full adder). This module can be integrated into any (large or small) combinational and/or sequential design project.

**Module 2: Assertions and Coverage:** This module discusses how a design should be simulated over a *well-chosen subset of inputs* in a systematic manner and the idea of the “degree of confidence” in the test set. The concepts introduced include test plans, enhancing observability of bugs using assertions, assertion-based verification and using coverage to determine the gap between exhaustive and partial testing. This module can be integrated into any (reasonably large) sequential circuit design project. For design courses that introduce High-level Design Languages, this module can also be used to emphasize the use of assertions and design-for-verifiability.

**Module 3: Equivalence Checking:** This module acts as the first introduction to semi-formal and formal verification. It lays the concept of equivalence between a reference and test design, which is a critical step that makes aggressive design optimizations possible. Each optimized design needs to be equivalence checked against an earlier version of the design. This module can be integrated with any combinational and/or sequential design projects. Using such an equivalence flow, the instructor would be able to provide a “black-box” checking tool and can focus on teaching the high-level application of equivalence checking rather than the algorithms behind it.

**Module 4: Property Checking:** This module provides a more in-depth overview of formal verification and serves to introduce the concept of *active design testing* (as compared to passive simulation or equivalence checking). This module is appropriate for students who are familiar with advanced HDL concepts. It can be used in conjunction with fairly large, hierarchical design projects. In particular, this module outlines

what a property is, different kinds of properties, how they can be specified and why such active testing is important to large designs. For example, the designer should be able to verify if her design holds the property that an acknowledgment is always produced within 10 clock cycles after a request has been received, irrespective of any other interrupting signals that may arrive. In addition to just implementation of the specification, this module can be used to introduce the concept of *design modeling* using tools such as NuSMV or SPIN.

**Module 5: Introduction to Commercial Tools:** This final module gives the students an overview of the state-of-the-art commercial verification tools that are used in the industry. Examples of such tools include the Formality Equivalence Checker<sup>TM</sup>, Magellan RTL Formal Verification<sup>TM</sup> and Vera<sup>TM</sup> by Synopsys [3] and the Incisive Functional Verification Platform<sup>TM</sup> by Cadence [4]. This module serves to reinforce the concepts outlined in previous modules and shows how these are used in the industry today. A knowledge of how to operate these tools will serve to reduce the gap between design and verification when the students work on real designs.

#### IV. CONCLUSION

This paper addresses the infusion of verification and validation concepts into the current computer engineering curriculum that traditionally focuses only on design aspects. We presented a module-based approach that can be used to seamlessly integrate these concepts into existing design courses. We believe that the transfer of verification knowledge from the active research arenas to curricula must start now, instead of later, or the productivity gap between design and verification will only continue to widen at an ever-increasing pace.

#### REFERENCES

- [1] International Technology Roadmap for Semiconductors, 2005. Available at <http://www.itrs.net/>.
- [2] ECE Undergraduate Courses, Virginia Tech, 2007. Available at <http://www.ece.vt.edu/ugrad/courses.html/>.
- [3] Discovery Verification Platform, Synopsys, 2007. [www.synopsys.com/products/solutions/discovery\\_platform.html](http://www.synopsys.com/products/solutions/discovery_platform.html).
- [4] Incisive Functional Verification Platform, Cadence, 2007. [http://www.cadence.com/products/functional\\_ver/](http://www.cadence.com/products/functional_ver/).