Portland State
UNIVERSITY

DEPARTMENT OF ELECTRICAL
AND COMPUTER ENGINEERING

SYNOPSYS VC FORMAL
FINAL REPORT

INDUSTRY SPONSOR AND ACADMIC ADVISOR

PROFESSOR XIAOYU SONG

ECE 413 CAPSTONE TEAM#7

MOHAMED GHONIM
AHLIAH NORDSTORM
DMYTRO PRYSTUPA
ALEXANDER KIM
CELINA WONG

VERSION:
REVISED:

Table of Contents

Executive Summary:

The Capstone Project: Formal Verification Using Synopsys VC Formal focuses on providing comprehensive tutorials for undergraduate and graduate students at Portland State University. Our objective is to equip students with practical knowledge and proficiency in formal verification techniques using Synopsys VC Formal, a powerful tool for silicon design and verification.

In collaboration with our sponsor, Dr. Song, we are working to develop user-friendly tutorial documentation for the installation, utilization, and creation of examples using VC Formal. This project addresses the need for accessible resources in formal verification within academic settings.

Synopsys, a renowned electronic design automation (EDA) company, has developed VC Formal as a leading tool for formal verification. By leveraging logical and mathematical principles, formal verification ensures the correctness of complex RTL designs, particularly in critical applications such as airplane controller chips. VC Formal consists of 11 distinct applications, each serving specific verification purposes.

Our project team focuses on various essential components, including introducing VC Formal to users, simplifying the installation process, and creating comprehensive tutorials and videos. The tutorials cater to users with varying levels of familiarity with VC Formal, providing step-by-step guidance and practical examples for each of the VC Formal apps.

Through this project, we aim to bridge the gap between academia and industry by empowering undergraduate and graduate students at Portland State University to apply formal verification techniques effectively. The tutorials and examples we provide will enhance students' understanding and proficiency in formal verification using Synopsys VC Formal, preparing them for future challenges in the field of silicon design and verification.

Background:

Digital logic designs, especially RTL (Register Transfer Level), have become increasingly complex, posing challenges for effective verification methods. Traditional techniques like simulation are impractical for verifying large and intricate designs due to time constraints. Fuzz testing, a non-exhaustive simulation method, provides some confidence but falls short in critical aspects of design verification.

To address these limitations, formal verification has emerged as a crucial approach. By leveraging logic and mathematical principles, formal methods can establish the correctness or incorrectness of designs. While formal verification was once a niche tool reserved for complex designs, it has gained mainstream popularity since 2015. However, the industry faces a shortage of skilled engineers in this field, and academic instruction in formal verification is limited.

Our project aims to bridge this gap by acquiring, applying, and documenting the use of Synopsys VC Formal, one of the industry's most powerful commercial formal verification tools. Our goal is to teach formal verification within the engineering department at Portland State University.

Currently, VC Formal training is primarily available in the industry, with limited accessibility in academia. Existing resources, such as Synopsys' extensive user manual of over 1000 pages and introductory videos, cater to industry professionals and lack academic focus. Expecting students to read such lengthy manuals for a class project is unrealistic. Therefore, we intend to develop concise yet comprehensive tutorials, incorporating written and visual components specifically tailored for academic settings.

By providing accessible and easy-to-understand tutorials, we aim to empower students at Portland State University to grasp the concepts and practical implementation of formal verification using Synopsys VC Formal. Our focus is on delivering practical knowledge that enhances their understanding of this verification technique, enabling them to tackle complex digital designs with confidence.

Product Design Specification (Requirements and Stakeholders)

Concept of Operation / User stories:

The primary objective of our project is to provide PSU students, specifically those enrolled in Dr. Song's ECE 582 and other M.S./Ph.D. classes, with comprehensive tutorial documentation on using VC Formal. The tutorials will guide students in understanding and completing coursework related to formal verification. Dr. Song will grant his students access to these tutorials, ensuring their availability for educational purposes. The tutorial documentation should have a long lifecycle, with updates made only if major changes occur in the VC Formal program. The success of our project will be measured by the ability of individuals with little to no knowledge of VC Formal to navigate and utilize our tutorial documentation effectively.

Stakeholders:

The stakeholders involved in our project include:

- PSU student body: They will benefit from gaining practical knowledge and experience in using VC Formal, enhancing their employability.
- Professor Xiaoyu Song: He can integrate VC Formal into PSU classes, providing students with an opportunity to apply various verification techniques using a state-of-the-art tool.
- PSU Electrical and Computer Engineering Department: By teaching this specialized tool, the department can improve its ranking and enhance students' career prospects. Faculty members will also have access to VC Formal tutorials for research and publications.
- Synopsys: As a key supporter, Synopsys will have its VC Formal tool taught to a significant number of graduate students each year. This will increase the pool of potential employees who are already familiar with Synopsys tools, making it more convenient for employers to deploy these tools in the industry. The increased popularity and usage of VC Formal in academia may lead to wider adoption in the industry, further expanding Synopsys's customer base.

Requirements:

- *Must*:
    - Thoroughly understand the basic functions of VC Formal and its "apps" (in priority order).
    - Install and configure an operational environment for VC Formal on a local PC at PSU.
    - Document how to connect to PSU labs remotely using a VPN and a Linux client such as MobaXterm.
    - Explore the functionalities and usages of VC Formal.
    - Document sequential equivalence checking in the VC Formal (SEQ) app.
    - Create a thorough tutorial documenting LTL model-checking in VC Formal using SystemVerilog Assertions (SVA).

- Should:
    - Develop at least one example per app to work with VC Formal.
    - Create a step-by-step video demonstrating the installation and configuration of VC Formal on a local PC at PSU.
- May:
    - Develop multiple examples per app.
    - Create videos demonstrating the configuration and usage of some or all of the apps.
    - Conduct a case study involving one or more RTL designs, running formal verification using more than two VC apps.
    - Provide TCL templates for apps utilizing TCL.

Objectives and Deliverables:

The specific objective of this project is to provide comprehensive tutorial documentation on using Synopsys VC Formal, focusing on the various apps within the tool. The aim is to enable undergraduate and graduate students at Portland State University to gain practical knowledge and proficiency in formal verification techniques using VC Formal.

At the end of this project, our sponsor, Dr. Song, wanted to have a set of clear and accessible tutorials that guide students through the installation, configuration, and utilization of VC Formal. The tutorials would cover the prioritized apps, such as Formal Property Verification (FPV), Formal Coverage Analyzer (FCA), Formal Connectivity Checking (CC), and others, as well as specific topics like sequential equivalence checking and LTL model-checking.

While the project's main focus remained on Synopsys VC Formal, there was a change in the deliverables. Our sponsor requested additional tutorials using Cadence's formal verification tool/solution, JasperGold. Initially challenging, we took on the task and successfully learned and worked with JasperGold, delivering a tutorial on its usage as an added deliverable.

Overall, the project objectives remained consistent with a primary emphasis on providing tutorials for VC Formal. However, the expanded scope to include JasperGold tutorials allowed us to offer a more comprehensive learning experience for students, covering multiple formal verification tools.

Approach:

Given the unfamiliarity of the team with formal verification and VC Formal, the project initially appeared to be a significant challenge. The specialized nature of the topic and the expertise required to operate these tools and create tutorials made the project seem daunting. However, the team embraced the challenge and approached it methodically, taking one step at a time.

The first phase involved building a foundation of knowledge by researching formal verification, both at a non-technical level and a deeper technical level. Additionally, the team refreshed their understanding of the Linux environment, focusing on the installation and utilization of complex software tools. Valuable resources provided by Synopsys, including tutorials and documentation, were leveraged to gain a better understanding of VC Formal.

Throughout the project, the team maintained regular communication with their sponsor and academic advisor, Dr. Song. They sought clarification and guidance whenever questions arose or confusion arose during the process. Dr. Song's expertise and support were instrumental in overcoming challenges and addressing concerns.

To maximize productivity, the team decided to divide into smaller groups, with each group consisting of two students or one student coordinating the work. This approach allowed for parallel work on different tutorials while facilitating collaboration, mutual assistance in debugging, and issue resolution. The smaller groups proved to be an effective way of working on the project.

Despite the initial apprehensions and the complexity of the subject matter, the team's systematic approach, thorough research, and effective communication with their sponsor and advisor allowed them to tackle the project successfully. The division into smaller groups further enhanced productivity and collaboration, resulting in significant progress and achievement in creating tutorials for formal verification using VC Formal.

Design

Big picture:

Our design process for this Capstone Project: Formal Verification Using Synopsys VC Formal involved extensive learning, experimentation, and iterative refinement. Our primary focus was to gain proficiency in utilizing the various VC Formal apps and create tutorials that effectively explain their usage and offer practical examples.

To achieve this, we dedicated significant time to studying each app individually, understanding its functionalities, and exploring different techniques. We conducted thorough testing to ensure that our understanding of the apps was solid and accurate. Through this process, we built a strong foundation of knowledge and hands-on experience with VC Formal.

The creation of tutorials was a crucial aspect of our design. We wanted each tutorial to be independent and self-contained, allowing for easy sharing with students in relevant classes. This approach ensures that students can access tutorials specific to the app they are working with, without being overwhelmed by unnecessary information from other tutorials.

Throughout the project, we constantly revisited and refined our tutorials. As we gained more experience and knowledge, we identified areas where improvements and clarifications were necessary. For instance, we discovered techniques like tracing the driving signal in the waveform view, which prompted us to revisit earlier tutorials and incorporate these valuable insights.
By investing significant effort into learning, experimentation, and tutorial creation, our design provides a solid foundation for students to understand and utilize VC Formal. The tutorials are designed to be informative, user-friendly, and independent, offering practical examples that enhance the learning experience and promote effective utilization of VC Formal's apps.

Starting with the fundamentals

As we embarked on our project, we followed a structured approach by creating three fundamental tutorials before diving deep into VC Formal:

1. Tutorial 1: Installing Mobaxterm

- In this tutorial, we focused on installing Mobaxterm, a software necessary to access the remote PSU labs where VC Formal is installed.
- We provided a step-by-step guide, ensuring users could successfully install Mobaxterm and familiarize themselves with its functionalities.
- The tutorial aimed to establish a solid foundation for users to access the remote lab environment seamlessly.

2. Tutorial 2: Installing and Running Cisco VPN Client

- This tutorial addressed the installation and configuration of the VPN Client "Cisco VPN" to enable users to access the remote PSU lab securely.
- We provided detailed instructions, guiding users through the installation process and necessary configurations to establish a secure connection.
- The tutorial aimed to ensure users could connect to the remote lab environment without any complications.

3. Tutorial 3: Installing Synopsys VC Formal on Linux System

- In this tutorial, our focus was on installing Synopsys VC Formal on the Linux system used within Mobaxterm.
- We provided a comprehensive guide, walking users through the installation process and the setup of the environment for their first project.
- The tutorial aimed to equip users with the necessary tools and configurations to effectively work with VC Formal.

The tutorials laid the groundwork, ensuring users could access the remote lab environment, establish a secure connection, and have VC Formal installed and configured correctly. This step-by-step approach aimed to build users' confidence and prepare them for further exploration of VC Formal's functionalities and applications.

To provide users with a comprehensive understanding of VC Formal and its various applications, we developed tutorials for the VC Formal apps. Here is an overview of the VC Formal apps and a brief description of the corresponding tutorials:

1. **Formal Property Verification (FPV) App:**
- Tutorial: In this tutorial, we dive into the FPV App and guide users on how to verify properties in their designs. We cover both user-defined and Assertion IP (AIP) properties, providing detailed examples and explanations.

2. **Formal Coverage Analyzer (FCA) App:**
- Tutorial: Our tutorial focuses on the FCA App and its role in simulation-based verification coverage signoff. We demonstrate how to use FCA to qualify formal property verification with coverage signoff, ensuring comprehensive verification of the design.

3. **Formal Connectivity Checking (CC) App:**
- Tutorial: Here, we explore the CC App and provide step-by-step instructions on checking the structural and functional connectivity between source and destination. Users will learn how to ensure proper connectivity within their designs using this powerful app.

4. **Automatically Extracted Properties (AEP) App:**
- Tutorial: In this tutorial, we explain the AEP App and its functionality in automatically extracting properties from the design. Users will learn how to verify these properties efficiently using the AEP App.

5. **Sequential Equivalence Checking (SEQ) App:**
- Tutorial: Our tutorial on the SEQ App focuses on comparing two RTL designs to verify their functional equivalence. We provide detailed examples and discuss the significance of SEQ App in validating design modifications.

6. **Formal Testbench Analyzer (FTA) App:**
- Tutorial: Here, we guide users through the FTA App, which helps in assessing the quality of formal testbenches. The tutorial includes step-by-step instructions on using FTA and emphasizes the importance of fault detection analysis.

7. **Formal Register Verification (FRV) App:**
- Tutorial: Our tutorial for the FRV App showcases the creation and formal verification of checks for registers in the design. Users will gain insights into how FRV App can ensure the correctness of register implementations.

8. **Formal Security Verification (FSV) App:**

- Tutorial: This tutorial focuses on the FSV App, which is essential for preventing unintended data propagation between secure and non-secure areas. We explain how to use the FSV App effectively to enhance the security of the design.

9. **X-Propagation Verification (FXP) App:**

- Tutorial: Here, we delve into the FXP App and guide users on checking for the propagation of unknown signal values (X's) to dangerous points in the design. The tutorial includes practical examples and highlights the importance of X-propagation verification.

10. **Data Path Validation (DPV) App:**

- Tutorial: In this tutorial, we explore the DPV App, leveraging HECTOR technology to validate data transformation blocks between untimed C/C++ and RTL models. Users will learn how to effectively use DPV for data path verification.

11. **Functional Safety (FuSa) App:**

- Tutorial: Our tutorial on the FuSa App provides insights into analyzing the controllability and observability of Z01X faults, enabling users to calculate FMEDA metrics for functional safety assessment.

Through these tutorials, we aimed to cover each VC Formal app comprehensively, providing users with practical guidance and examples to understand their functionalities and leverage them effectively for formal verification.

12. **Cadence JasperGold Superlint - Ensuring Design Integrity through Static Linting**

- In this tutorial, we provide a comprehensive guide on using Cadence JasperGold Superlint, a powerful static linting tool for digital design verification. Learn how to identify potential coding errors, design rule violations, and improve design integrity. Our step-by-step instructions cover setting up the tool, configuring linting rules, running the process, and interpreting reports. Enhance the quality, reliability, and maintainability of your digital designs with Cadence JasperGold Superlint.

Test plan

Describe how you tested the system to prove that it does what you expected it to do. This might just reference an actual test plan in an appendix, or if the test plan is short enough, you could include it here.

Results

Concise but detailed results of your project. Deliverables achieved and not achieved. Success you had, and failures you had. Tests and measurements performed. How well did it meet stated requirements? Be quantitative when you can be, and present data appropriately (tables, graphs, etc.).
Some of this might be repeated from the Design section.