Portland State UNIVERSITY

DEPARTMENT OF ELECTRICAL AND COMPUTER ENGINEERING

SYNOPSYS VC FORMAL PROJECT PROPOSAL

ECE 412/413: CAPSTONE TEAM 7

MOHAMED GHONIM
AHLIAH NORDSTORM
DMYTRO PRYSTUPA
ALEXANDER KIM
CELINA WONG

VERSION: 1.3
REVISED: FEBRUARY 20, 2023

Table of Contents

Song VC Formal Installation and Documentation.pdf

1. Executive Summary

As Capstone team #7, we are working under the supervision of our sponsor, Dr. Song, to complete a project centered around installing, utilizing, and creating examples of Synopsys VC formal. Our main goal is to produce a comprehensive set of tutorial documentation that will be incorporated into PSU M.S. and Ph.D. classes in the future.

Synopsys is a highly respected electronic design automation (EDA) company with a focus on silicon design and verification for advanced chips. They are a renowned industry leader and the largest provider of EDA technology. The use of formal verification has become a necessity in critical applications such as airplane controller chips, as it leverages logic and mathematical axioms to prove the correctness of RTL designs. Synopsys' most powerful formal verification tool, VC Formal, comprises of 11 distinct applications, each of which functions as an individual software program.

Our team's focus will be on a variety of key components, including how to get started with VC Formal, the installation process, creating resources for new users such as tutorials, documents, and videos, and developing thorough examples for the different apps within VC Formal. We will emphasize creating both written tutorials and videos for different apps. Additionally, we will take into account that different users may have varying levels of familiarity with VC Formal, and adjust the tutorials accordingly to accommodate their varying needs.

2. Background

As digital logic designs, particularly RTL, become increasingly complex, there is a growing need for more effective methods of verifying such intricate systems both before and after synthesis. The use of static or simulation techniques is impractical for verifying such large and complex designs, with simulation alone requiring several hundred years to verify a modern commercial-sized microprocessor. Consequently, other techniques such as fuzz testing, a non-exhaustive simulation method using random inputs, are used to test these systems, which provides a reasonable level of confidence in the designs, but is still not entirely reliable or quick enough for the critical aspects of the designs.

This is where formal verification emerges as a critical method. Formal methods utilize fundamental logic and mathematical axioms to establish the correctness, or incorrectness, of the designs. Until around 2000, formal verification was a niche tool exclusively used to verify extremely complex designs, and only a small number of verification experts were capable of performing such work. Since 2015, formal verification has gained popularity and has become mainstream, with widespread use in the industry. Nevertheless, tech companies have difficulty hiring skilled and experienced engineers in this field, as academic emphasis on this subject of verification is insufficient.

Merely a few educational institutions within the United States provide instruction in this particular verification technique, typically employing academic tools that differ significantly from commercial tools. Our objective is to acquire knowledge, apply it, and record the application of one of the most potent commercial formal verification tools in the industry, namely "Synopsys VC Formal," in order to teach it within the engineering department of Portland State University.

At present, VC Formal training is exclusively provided within the industry, with no accessible training available within academia. This is a developing problem with state-of-the-art EDA tools. Although Synopsys offers a comprehensive user manual for VC Formal, consisting of over 1000 pages, as well as a few introductory videos, these resources are aimed at industry professionals and are not tailored to academia. It is unreasonable to expect students to read such a lengthy user manual to complete a class project. Therefore, we aim to provide concise yet comprehensive tutorials, including written and visual components, that are particularly useful within academic settings.

3. Project Overview

Our team will engage in a comprehensive research process to obtain a deep understanding of VC Formal and its various applications. After installing, learning, and using the tool, we will develop tutorial materials that will facilitate the training of future undergraduate and graduate students in using VC Formal and its applications.

To achieve this, we will produce two types of tutorials: concise yet detailed written tutorials and video tutorials. The written tutorials will provide step-by-step instructions on how to use the different verification apps within VC Formal. The materials will be concise but thorough, presenting users with clear and detailed guidance on how to navigate the different functions and features of the software.

On the other hand, the video tutorials will be designed to focus on the most critical and challenging aspects of VC Formal and its applications. Through these videos, learners will have an immersive and engaging learning experience, allowing them to follow along with the visual demonstrations of how to use the various functions of the tool. The videos will be an effective supplement to the written materials, providing learners with a hands-on approach to learning.

Overall, the written and video tutorials will serve as essential resources for students to acquire a practical understanding of VC Formal and its different apps. Depending on the learners' preferences and learning styles, they can choose to use either or both of these resources to maximize their understanding and proficiency in using the tool.

4. Product Design Specification

4.1 Concept of Operation / User stories

- Will guide Dr. Song's ECE 582 (and other M.S/Ph.D. classes) students through VC Formal in order to understand and complete coursework
- Dr. Song will give access to these tutorials to his students
- The lifecycle should be fairly long, as long as Synopsys doesn't make any major updates to the VC Formal program
- This project is successful if anyone with little to no knowledge of VC Formal could walk through our tutorial documentation sets

4.2 Stakeholders

- The primary beneficiary of our project is the PSU student body who will gain practical knowledge and experience using one of the most advanced verification tools in the industry, increasing their employability.

- Additionally, Professor Xiaoyu Song can incorporate VC Formal into his PSU classes to provide students with the opportunity to use a state-of-the-art tool to put various verification techniques into practice.

- The PSU Electrical and Computer Engineering Department will become one of the few institutions in the country and around the world to teach this specialized tool, which could lead to an increase in school ranking and enhance students' career prospects.

- The PSU ECE faculty will also have access to tutorials on a sophisticated verification tool, which they can use for their research and publications.

- Synopsys is a key supporter to this project, as their VC Formal tool will be taught to a significant number of graduate students each year. This would provide employers with a large pool of potential employees who are already familiar with Synopsys tools, making it more convenient for them to deploy these tools in the industry. Additionally, the increased popularity and usage of VC Formal in academia could result in more companies adopting this technology, further increasing Synopsys's customer base.

4.3 Requirements

- Must
    - Thoroughly read and understand the basic functions of VC Formal and its “apps”. The apps here are listed in priority order:
        - The Formal Property Verification (FPV) App verifies properties in the design including user-defined and Assertion IP (AIP) properties.
        - The Formal Coverage Analyzer (FCA) App assists simulation-based verification coverage signoff and qualifies formal property verification with coverage signoff.
        - The Formal Connectivity Checking (CC) App checks if there is a structural and functional connection between the source and the destination.
        - The Automatically Extracted Properties (AEP) App automatically extracts properties from the design and verifies them.
        - The Sequential Equivalence Checking (SEQ) App compares two RTL designs and verifies that they are functionally equivalent.
        - The Formal Testbench Analyzer (FTA) checks if injected faults can be detected by the formal testbench thus measuring the quality of the formal testbench.
        - The Formal Register Verification (FRV) App automatically creates and formally verifies checks based on IP-XACT or RALF format for registers in the design.
        - The Formal Security Verification (FSV) App ensures that unexpected data propagation does not happen between secure and non-secure areas.
        - The X-Propagation Verification (FXP) App checks whether unknown signal values (X's) can propagate to dangerous points within the design.
        - The Data Path Validation (DPV) App uses HECTOR technology to verify data transformation blocks between untimed C/C++ and RTL models.
        - The Functional Safety (FuSa) App analyzes the controllability and observability of Z01X faults to calculate FMEDA (Failure Mode Effects and Diagnostic Analysis) metrics.

    - 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.
    - Thoroughly document sequential equivalence checking in the VC Formal (SEQ) app.

    - Make a thorough tutorial documenting LTL model-checking in VC Formal using SystemVerilog Assertions (SVA).

- Should
    - Develop one example per app, to work with VC Formal.
    - Make a step-by-step video on how to install and configure an operational environment for VC Formal on a local PC at PSU.

- May
    - Develop more than one example per app.
    - Make videos showing step-by-step how to configure and use some or all of the apps.
    - Develop a case study involving one or more RTL designs, and run formal verification on the same design using more than 2 VC apps.
    - Provide TCL templates for the apps that use TCL.

4.4 Specifications

Specifications are not needed as the requirements above are thoroughly described.

4.5 Deliverables

- Detailed documentation on how to install and configure VC Formal.
- A detailed overview of each VC Formal app, written in such a way that senior undergraduates/graduate students will understand it. Brief introductions to technologies will be required here. In essence, an onboarding document for each app explains what the app is for, and how it’s used.
- A detailed example in each VC Formal app that explains how to use it and what the expected outcome/results of the app are.

Our team will provide comprehensive tutorial materials for Synopsys VC Formal that include written documents and tutorial videos to help undergraduate and graduate students easily learn and use the tool. The materials will be created and delivered in separate files and shared with Professor Song through a designated Google Drive folder for review and feedback.

The written materials will be in the form of PDFs and editable Word documents, with each file focusing on a specific VC Formal app or functionality. These files will provide step-by-step instructions on how to use the different features of VC Formal and will be concise yet comprehensive for easy understanding by students.

In addition to the written materials, we will also create tutorial videos demonstrating the different VC Formal apps and features. These videos will be uploaded to a designated Google Drive folder and shared with Professor Song, making them easily accessible to PSU students. The videos will also be uploaded to YouTube for easier access, although they will be unlisted and only accessible via a specific link due to copyright restrictions.

4.6 Initial Product Design

Our aim is to create introductory material for PSU's ECE students that will simplify Synopsys's VC Formal user guides and datasheets. Our goal is to create concise and effective tutorial documentations that will provide basic and advanced knowledge in using VC Formal, making it easier for students to continue building on their knowledge of the tool in the future. Our team will utilize our experience and research of both good and bad tutorial documentations to ensure that the users of our guides will be able to successfully navigate and utilize VC Formal.

- Hardware architecture

- As this is a documentation-focused project, there are no physical components to design block diagrams for, unlike other projects. However, we will provide sample templates that demonstrate how the various documents can be constructed. These templates will be designed to showcase the appropriate level of detail and organization required to effectively convey the material. By using these templates, future students will be able to create high-quality documentation that will assist them in gaining the knowledge and skills necessary to use VC Formal effectively.

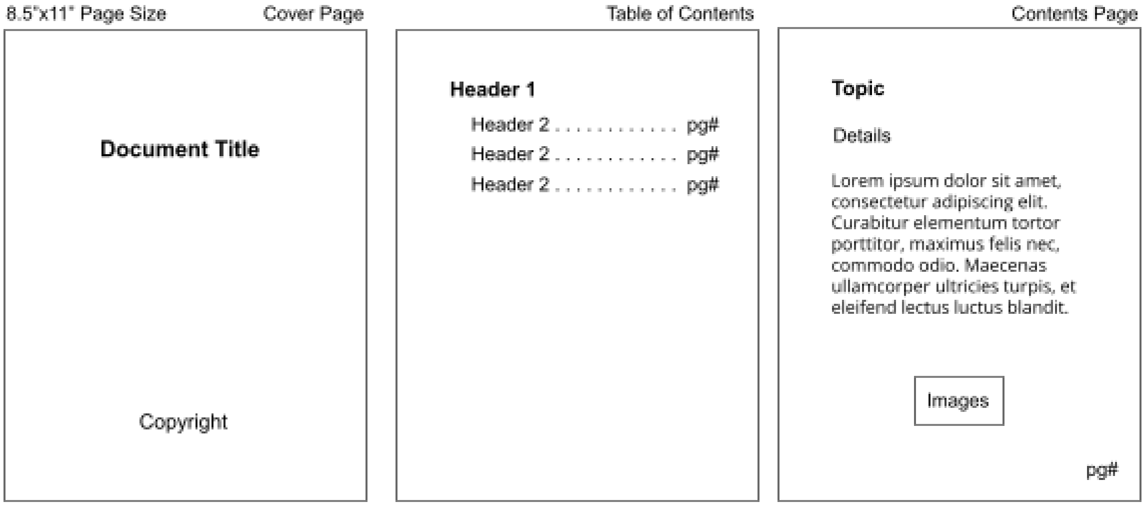
8.5"x11" Page Size
Cover Page
Document Title
Copyright
Table of Contents
Header 1
Header 2 . . . . pg#
Header 2 . . . . pg#
Header 2 . . . . pg#
Contents Page
Topic
Details
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur elementum tortor porttitor, maximus felis nec, commodo odio. Maecenas ullamcorper ultricies turpis, et eleifend lectus luctus blandit.
Images
pg#

Figure 1. Example outline template for tutorial documentations.

- Software architecture
  - Our work is focused on demonstrating how to use a software and we are not involved in the design or development of any software. Therefore, there won't be any software development or design diagrams included. However, we will provide some example diagrams to illustrate how the software is used.

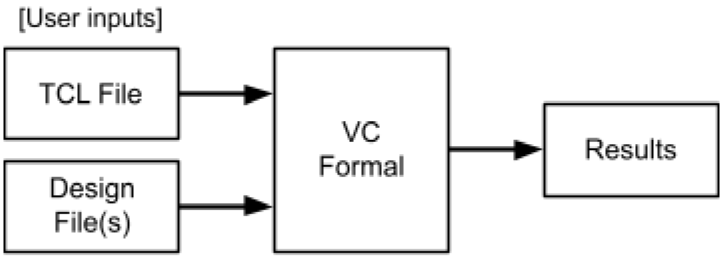

Figure 2. Simple block diagram for software architecture utilization.

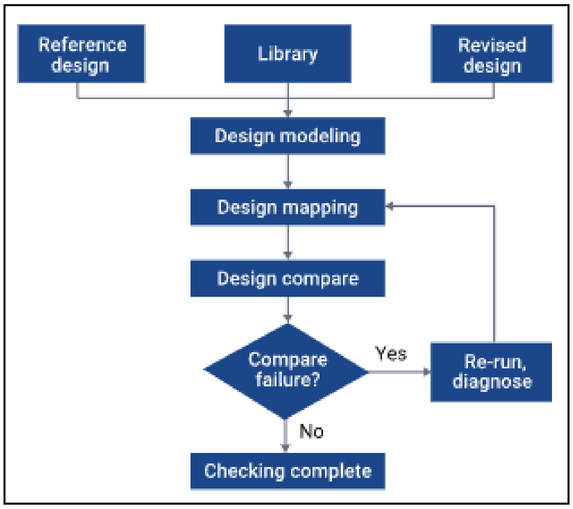

Figure 3. Equivalence checking block diagram.

Source: https://www.synopsys.com/glossary/what-is-equivalence-checking.html

- User interface/experience
  - Our product requires users to access the VC Formal application to be able to follow our tutorials.
  - Our product will be accessed through PDF files and linked videos to YouTube. The professor will upload the documents onto Canvas and students can view it on the browser or download them.

- Other considerations
    - Will Synopsys keep the documents secure?
    - Synopsys allowing a certain # of licenses to PSU?
- Back-up plans
    - We support ECE 582 with their learning journey of VC Formal.

4.7 Verification Plans

Our capstone team will ensure that the tutorials and demonstration documents for VC Formal meet the requirements and specifications set by our team and the sponsor. Members of our team will be responsible for the verification process. In the winter term of 2023, at least one of the VC Formal applications will be used in a Formal Verification of Hardware and Software Systems course at Portland State University. The students enrolled in this course will be the test subjects of our tutorials and demonstrations. Our team will verify that each student can use our tutorials to operate VC Formal and utilize all of its features and functions successfully. At the conclusion of the course, the students will be asked to provide feedback on the tutorials. The capstone group will utilize this feedback to verify whether the "must" requirements are being met.

Important points to look for:

- Document presentation, formatting, grammar
- For parts that require more explanation for better understanding and learning - is it thoroughly demonstrated?

Testing strategies:

- □ Was the remote user able to install the VPN and Linux client (MobaXterm) on their machine?
- □ Was the user able to access the remote lab using the installed VPN and Linux client?
- □ Was the user (remote or at PSU) able to successfully install VC Formal?
- □ Was the user able to set up the VNCserver and connect to it?
- □ Was the user able to invoke the VC Formal GUI using the Linux terminal?
- □ Was the user able to modify the TCL template for the SEQ app and compare the specification and implementation designs correctly?
- □ Was the user able to make sense of the results?
- □ Was the user able to use the other documented apps as well?
- □ Did the user find any ambiguities in the tutorial document? Is anything poorly documented and needs to be fixed, or just polished?

Fail - Student gets stuck in the middle of the document/tutorial.
Pass - Students are able to successfully make a remote connection to the PSU Linux system computers, install VC Formal, and use the different VC Formal Applications.

As part of our verification plans, we will solicit general feedback from the students in the Formal Verification of Hardware and Software Systems class who will be utilizing our VC Formal tutorials and demonstrations. We will ask open-ended questions such as "What aspects of the tutorials were great?" "What aspects were difficult?" "What was unclear or hard to understand initially?" and so on. The purpose of this feedback is to identify areas of improvement in the tutorial and demonstration documents, as well as to gather insights on how to better meet the needs and expectations of the users.

5. Project Management Plan

5.1 Timeline and Milestones

See Gantt Chart file attached along with this document.

The timeline shown in this Gantt Chart covers the period from the first week of the Winter term (January 9) until the ninth week of the Spring term (May 28), which is our estimated time for completing the project in time for the capstone showcase. The major milestones of this timeline include ECE 412 Deliverables, Productivity App Tutorial Set, High Value App Tutorial Set, Final Report Writing, and Create Final Poster.

The ECE 412 Deliverables milestone includes several smaller milestones such as weekly meetings, progress reports, team check-ins, and assignments to be uploaded to Canvas. The majority of direct capstone work is shown in the Productivity App and High Value App Tutorial Sets. We anticipate quicker turnarounds for the Productivity App sets compared to the High Value Apps, as they are expected to be easier to understand (1 week vs 2 weeks). However, they will have the same milestones: Familiarize Ourselves w/ App, Develop Example/s, Create Written Tutorial, Create Video Tutorial, and Update Sponsor w/ Progress.

We have allotted ourselves over a month to complete the Final Report and Final Poster, which will give us a comfortable cushion if any previous milestones take longer than expected. This timeline allows us to work in a structured and organized manner, ensuring that all the milestones and smaller tasks are completed on time to meet the project requirements.

5.2 Budget and Resources

In order to accomplish this capstone project, one of the essential resources required is access to the Synopsys VC Formal software licenses. Our industry sponsor must provide us access to and licenses for the Synopsys VC Formal software throughout the capstone timeline. In addition, the Maseeh College of Engineering and Computer Science servers' accessibility is necessary for the successful completion of this project. A remote session connecting us to the computer servers of Portland State University is crucial to access the Synopsys VC Formal licenses. We also rely on the Zoom communications platform for regular team meetings, which facilitates working together and completing the various required tasks within the capstone project.

5.3 Intellectual Property Discussion

In relation to intellectual property (IP), the following ownership arrangements apply:

- Synopsys owns the licenses, user manual, logos, and screenshots taken from VC Formal, as well as any confidential communications exchanged between Synopsys and the capstone team.

- Portland State University owns its logo, screenshots taken from its servers, and any systems or classes used in the project.

- Professor Song owns all the class materials used or referenced in the project, and any ECE 582/682 project document that the capstone team creates for use in his class.

- The written and video tutorials have a **50/50** shared ownership between PSU's ECE Department, meaning all current faculty advisors and instructors, and the members of this capstone project, as listed below:

    - Mohamed Ghonim
    - Ahliah Nordstrom
    - Dmytro Prystupa
    - Alex Kim
    - Celina Wong

  PSU has the right to use, modify, and change the content of those tutorials without the student's permission, and the students have the right to use and distribute the tutorials, while disclosing the IPs owned by Synopsys, PSU, and Professor Song.

It's important to note that these ownership arrangements are subject to any legal agreements or contracts that may be in place between the parties involved. The capstone team should also be aware of any relevant laws or regulations that apply to the use and sharing of intellectual property.

5.4 Team and Development Process

- **Mohamed Ghonim**
  Strengths to bring:
  - Coding in System Verilog, RTL Design and Linux
  - Video editing and creating tutorial videos.
  - Working on designing and developing examples to test in VC Formal
  - Reaching out to Synopsys Support and will continue to do so as needed.

- **Alex Kim**
  Strengths to bring: Documentation, communication, organization, conflict management, and collaboration skills. Microsoft Office Suite proficiency, problem solving and analytical thinking.

- **Celina Wong**
  Strengths to bring: Developing, formatting, and proofreading formal documents. Concise writing skills, ability to quickly learn various tools, and deep understanding of how to assist others.

- **Ahliah Nordstrom**
  Strengths to bring: Communicating with team, formatting documents, updating progress on collaboration sites
  Knowledge of: SystemVerilog coding and debugging.

- **Dmytro Prystupa**
  Strengths to bring: Documentation, Debugging, MacOS

Collaboration tools:

- Google Drive (preferred)
- Github
- Trello

Communication tools:

- Discord (preferred)
- Zoom

Appointed team leader/responsible for sponsor communication - **Mohamed Ghonim**