Formal Verification Bootcamp

This document gives an overview of the TVS 2-day introduction to effective use of writing assertions for formal verification. Although the course will not focus on any particular tool, a tool will be used to demonstrate and practise the concepts introduced during the day (those parts of the course are highlighted in yellow).

1 Introduction to SVA

1.1 Writing Basic System Verilog Assertions

- Introduction to the language
- The main combinatorial language constructs (syntax and semantics)
- Constraints and properties
- Examples of constraints
  - Reading and understanding
  - Writing from fresh
- Examples of properties
  - Reading and understanding
  - Writing from fresh

1.2 Proving a basic property

- Using a tool to prove a basic property on a real design (e.g. FIFO)

2 Advanced SVA

2.1 Writing Complex System Verilog Assertions

- The main sequential language constructs (syntax and semantics)
- Examples of sequential constraints
  - Reading and understanding
  - Writing from fresh
- Examples of sequential properties
  - Reading and understanding
  - Writing from fresh

2.2 Proving a complex property

- Using a tool to prove a complex property on a real design (e.g. FIFO)
- Interpreting the three possible outcomes
  - Proved, Failed, Unproven

3 Debugging a failing property

- Using a tool to debug a failing property

4 Formal metrics

- Measuring coverage and completeness
- Measuring coverage with a particular tool

5 Automated formal verification

- Looking at the "out-of-the-box apps" available with a particular tool (e.g. clock domain crossing, SoC connection checking, reset checks)
- Automated HDL checks (such as Synthesizability, Race conditions, Code reusability)

6 Full formal verification of a block

- Developing constraints for a complex design (e.g. FIFO with APB interface)
  - Over constraint
  - Under constraint
- Developing properties for the complex design
  - Determining a "full" set of properties
- Run the constraints and properties using the tool

7 Formal in the design flow

7.1 The AHAA model

- Understanding the various applications of formal verification
  - Bug avoidance
  - Bug hunting
  - Bug absence
  - Bug analysis
- Understanding when and where to apply formal during a project

7.2 Formal for designers

- Designer bring-up
- Writing assertions "on-the-fly" using a tool
- Where (which file) to write the assertions?

7.3 Formal reuse

- Reuse of assertions between dynamic and static verifications
- Running simulations with our formal assertions added
- Reuse in assume/guarantee relationships
- Assertions as VIP

7.4 Formal in context

- Formal as part of a verification plan
- Combining formal results with other activities for a signoff decision