Abstract

As formal verification engineers, the authors always face challenges to accurately access the current status of test benches. Many questions need to be answered at certain stages of a project. E.g., do we need more assertions? Did we over-constrain inputs that caused the drop of an important design scenario? Are proof bounds for bounded proofs good enough to catch potential design bugs? For the properties that are fully proven, do they cover the design logic that were intended to cover? We cannot get answers to these four most-asked questions without extracting information from formal engines, which is not feasible for general users. However, like coverage metrics from simulation-based verification, formal verification coverage models can be defined and used as metrics to measure formal verification progress and completeness. Some academic research on formal verification coverage and commercial formal verification tools are starting to support some coverage usages in the past two years. However, none of them clearly specified what engineers would really need and provided a good way to present formal coverage results in a standard way.

In this paper, the authors will introduce formal verification coverage models and their usages by real-life examples. The four most-asked questions finally have reasonable and acceptable answers supported by metrics.

Coverage Rises Up to the Challenge

- Objective metrics of a formal verification test bench
- Coverage Closure methodology same as simulation based verification closure

Coverage Model to check whether measure whether enough assertions have been written to cover the design spaces that we intend to check

Stimuli Coverage : Did I over-constrain my inputs?"

Coverage Model focuses on how formal tools drive inputs to reach RTL design space under the current input constraints

Input Stimuli Coverage

- \( \text{COV}(B) = \left( \begin{array}[]{c} \text{dev} \times (\text{sts} - 1) - \text{cnt} \times \text{cnt} - \text{cnt} \\
\text{dev} \times \text{cnt} \times (\text{cnt} - 1) - \text{cnt} \times \text{cnt} \times \text{cnt} \end{array} \right) \)

- \( \text{dev} \): one of \( \text{nts} \) assertions of the bench
- \( \text{sts} \): total number of stimuli
- \( \text{cnt} \): total number of coverage targets

Coverage Model focuses on how formal tools drive inputs to reach RTL design space under the current input constraints

Formal Core Coverage : Do proofs cover the design logic that were intended to cover?"

Coverage Model analyzes if there are design bugs outside the formal proof core but still within the COI of the assertion

Conclusion

Formal verification without coverage closure is the same as doing simulation without coverage closure. With the increasing usage of formal verification for circuit design, we expect these formal verification coverage models will become standard models for formal tools and are used by formal verification sign-off process.