

Scaling Formal Unreachability Analysis to SoC Designs with VC Formal FCA Auto-Scale

Author

Pragati Shukla

Applications Engineering,
Staff Engineer

Overview

Achieving simulation coverage closure is an important verification signoff effort. While direct and random simulation can usually reach 80% of the coverage goals in a reasonable amount of time, it is the last 20% of the uncovered goals that require a lot of time and effort.

When there are uncovered goals (also referred to as coverage holes), designers and verification engineers need to spend time to distinguish whether they are reachable (real holes) or unreachable (false holes). Real holes will require additional tests to be added to regression while false holes could indicate design issues or can be excluded from coverage goals. For large SoC designs with hundreds of millions of coverage goals, manually reviewing even a small subset of these uncovered goals takes considerable amount time. Distinguishing between real and false holes requires deep design knowledge. As a result, this iterative coverage closure process can often take months of effort.

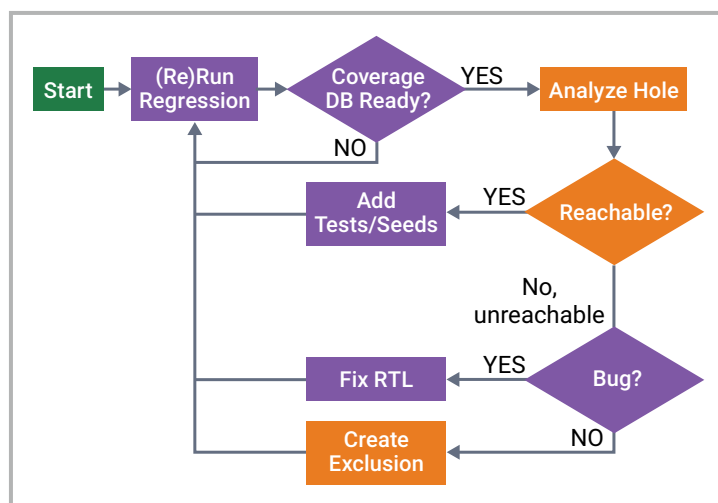


Figure 1: Typical coverage closure process

Formal unreachability analysis (UNR) is often deployed to formally analyze the design and report all the unreachable coverage goals through exhaustive formal analysis. This white paper reviews the challenges of a traditional UNR flow and discusses how Synopsys VC Formal™ FCA (Formal Coverage Analyzer), a powerful App in the VC Formal next-generation formal verification solution, can be used to automatically detect unreachable code (or dead code) in the RTL. When coupled with simulation VC Formal FCA can save significant manual effort in identifying unreachable code and accelerating coverage closure.

Challenges with the UNR Flow

While the traditional UNR flow is important for coverage closure and signoff, it can be challenging to performing UNR for large, complex, and often parameterized SoC designers. Here are a few examples:

- High memory requirement to create formal models for the SoC designs
- Large number of coverage goals created (multi-millions)
- The resulting long runtime and low convergence rate for formal unreachability analysis

The methodology to manually partition the SoC design into smaller pieces is not scalable because it is an iterative process to define partitions for every design resulting in maximum effort for limited re-use. In addition, this potentially risks losing vital data, like constants and parameters from the top module, which are crucial for block level UNR runs.

Given these challenges, there is a clear need for an automated solution to perform UNR for SoC designs that can provide better convergence and faster results.

Introducing the Auto-Scale Flow

Synopsys VC Formal FCA's auto-scale flow addresses these challenges through an automatic divide-and-conquer strategy. First, the auto-scale flow splits instrumented coverage goals into different verification tasks to be analyzed separately so users can configure each task. With the reduced coverage goals in each task, the size of the formal model is reduced, reducing memory requirements. Next, a UNR analysis is run concurrently on these tasks and an exclusion file is created for each one. With reduced model size and goals to be analyzed, formal performance is improved, and a higher convergence rate can be achieved. From there, the results are aggregated, and a combined exclusion file is created from each task. And lastly, optimization techniques such as automatic abstractions can be applied to further reduce the formal model complexity thus improving convergence.

To understand the true value of implementing VC Formal FCA auto-scale, consider the following customer example: User A had a design with 10 million coverage goals. Before implementing the auto-scale flow the user created 6 manual partitions and was able to achieve only 55% FSM coverage. But, when the user implemented the auto-scale flow without any manual partition, the FSM coverage reached 99%. The overall coverage score (considering other coverage metrics) jumped from 79% to 91%, saving weeks of time and effort for designers and verification engineers.

The following image provides a visual comparison of the steps taken in a traditional UNR flow versus the VC Formal FCA auto-scale flow:

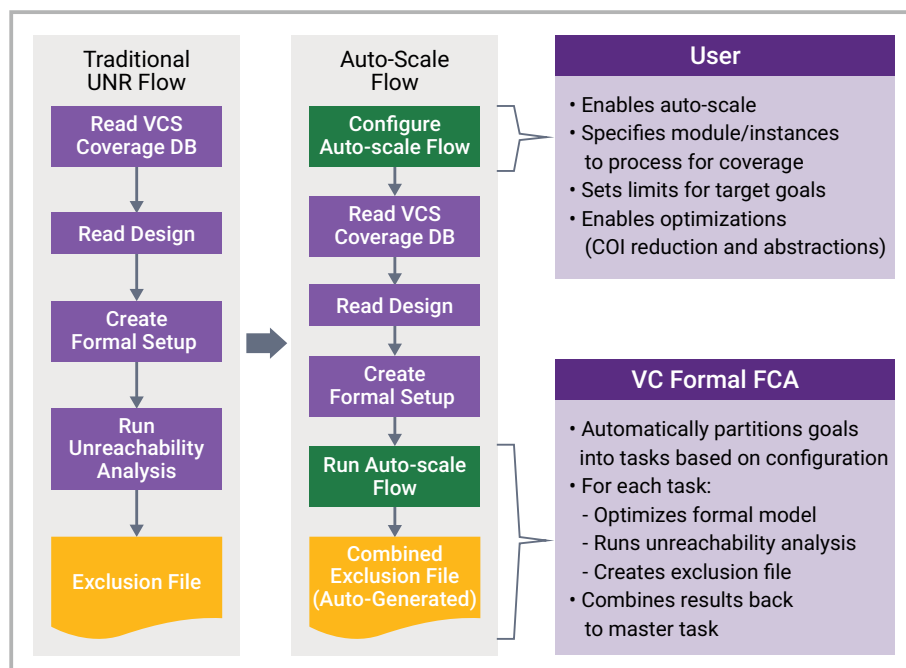


Figure 2: Comparing a traditional UNR flow to VC Formal FCA Auto-Scale flow

Native Integrations Enhance VC Formal FCA Capabilities

VC Formal FCA is natively integrated with Synopsys VCS® so it can extract uncovered goals from the VCS simulation coverage database seamlessly and immediately perform unreachability analysis. If there are any unreachable cover items, an exclusion file is automatically generated. In addition, the simulation coverage database along with the generated exclusion file can be loaded into the Synopsys Verdi® coverage viewer so users can analyze the exclusions and identify focus areas for improving coverage, saving time from manually identifying false holes.

Conclusion

Using the VC Formal FCA auto-scale flow enables formal unreachability analysis at the SoC level without using manual partitioning to improve coverage metrics while reducing user time and effort. The native integration of VC Formal FCA, VCS and Verdi makes it possible to achieve coverage closure at the SoC level, accelerating the verification signoff effort.