SYNOPSYS®

WHITE PAPER

Accelerating Coverage Closure with AI-Based Verification Space Optimization

Authors

Taruna Reddy
Staff Product Marketing Manager

Will Chen
Principal Application Engineer

Badri Gopalan
Synopsys Scientist

Introduction

Coverage is at the heart of all modern semiconductor verification. There is no maxim more fundamental to this process than "if you haven't exercised it, you haven't verified it." Although covering a particular aspect of a chip design does not guarantee that all bugs are found—bug effect propagation and checker quality are also key factors—it is certainly true that bugs cannot possibly be triggered in logic that has not been exercised. Coverage is often regarded as a proxy for finding bugs and is therefore a key focus for verification using simulation-based testing.

Although hand-written tests sometimes have a limited role even today, most simulation tests use constrained-random stimulus. There is no obvious correlation between these tests and the parts of the design verified, so coverage is critical to establishing this link. Although verification is never truly done, coverage metrics provide important guidance in determining when enough iterations of tests have been run and the design is considered well enough verified to be taped out.

Even with the automation benefits of constrained-random testing, convergence to the coverage goals is often slow, with diminishing returns as simulation continues. There is still significant manual effort involved. The semiconductor industry clearly needs a better, even more automated way to "shift left" coverage closure and improve the final coverage results.

Intelligent Coverage Optimization (ICO) using artificial intelligence and machine learning (AI/ML) was included in the Synopsys VCS simulator to address these challenges by enhancing stimuli diversity, exposing testbench bugs, and improving coverage. To complement ICO, Synopsys VCS recently introduced Synopsys Verification Space Optimization (VSO.ai) to target coverage more directly at the fine and coarse-grained levels, also leveraging AI/ML. Synopsys VSO.ai is the focus of this whitepaper.

Challenges in Coverage Convergence

The basic concepts behind coverage in simulation seem simple enough. The verification team chooses the structural code coverage metrics (line, expression, block, etc.) of interest and automatically adds them to the simulation test runs. The engineers generally define additional functional coverage points and groups that represent portions of the design that they want to be sure are exercised. They may also define cross-coverage to watch for specific combinations of coverage points.

The team then defines the constraints that ensure automatically generated stimulus stays within legal bounds, and then kicks off simulation runs. As each test iteration generates constrained-random stimulus conforming to the rules, the simulator collects metrics for all the forms of coverage included. The team monitors the results and may decide to run more tests or tweak the constraints to try to improve the coverage results. At some point, they decide that they have done the best that they can within the schedule and resource restrictions of the chip project, and they tape out.

That sounds fairly straightforward, but Figure 1 shows the challenges that arise when this process is used on real-world chip designs.

The first major challenge is coverage definition, since achieving blanket coverage for a design of any significant size or complexity is impossible.

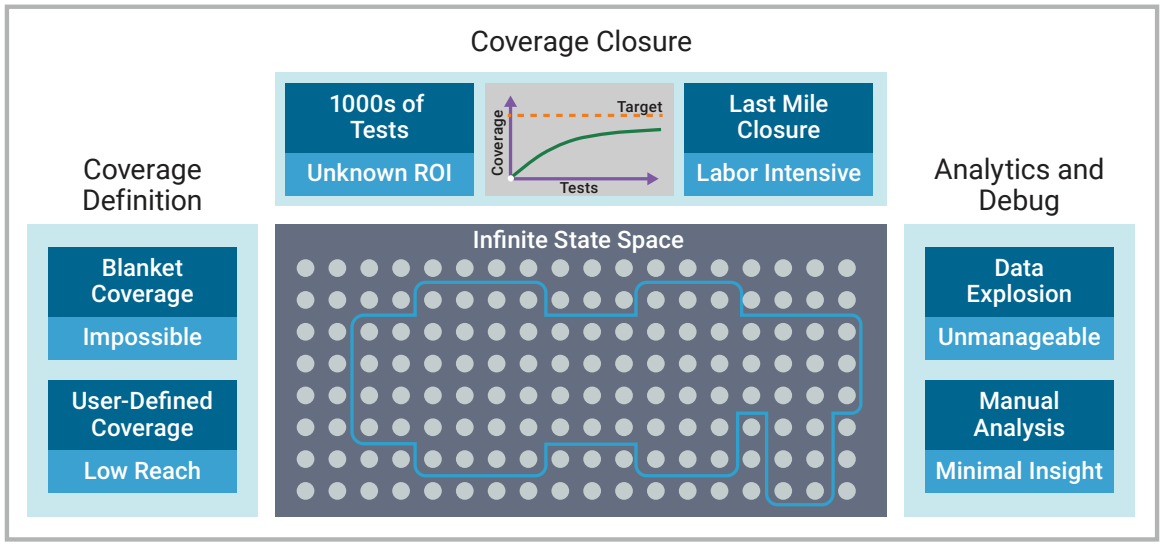

Figure 1: Challenges in the use of coverage in simulation

Since code coverage does not reflect the intended functionality of the design, user-defined coverage also plays an important role. However, this is a manual effort that generally spans only a limited percentage of the design's behavior. Whatever combination of structural and functional coverage is included in the simulation tests, trying to close that coverage and achieve metrics matching the verification goals is an even bigger challenge.

The graph in the top center of Figure 1 shows what happens on virtually every project. Initial test runs exercise many parts of the design and iterating constrained-random tests tends to improve coverage over time. A typical chip project runs many thousands of constrained-random simulation tests, and there is a great deal of repetitive activity in the design. Therefore, the rate of new coverage slows dramatically, and the return on investment (ROI) for each additional simulation run is reduced over time. The slowing ROI is the second major challenge in coverage closure,

At some point, the curve flattens out, and simulating indefinitely yields little or no improvement. The verification team must decide when enough coverage has been achieved to tape out. When this asymptotic convergence occurs, it is never at 100% of the coverage metrics. This is one reason why the project goals are set lower. Unfortunately, the flattening of the curve often occurs before the goals have been achieved. The verification team must try to figure out what is going on and improve the coverage as much as they can within the time and resources available. Since simply running more automated tests does not suffice, significant manual effort is required.

This “last mile” of the simulation closure process is the third major challenge, and it is hampered by two factors. First, the amount of data collected from the simulation runs is overwhelming. The second issue is trying to analyze this data and determine the root cause of a coverage hole: is it an illegal bin for this configuration or a true coverage hole? Manual analysis of massive data sets yields minimal insight. The only way to improve the situation is to define better coverage, run fewer tests, converge more quickly to higher results, and automate the analysis phase.

Introduction to Synopsys VSO.ai

AI and ML techniques are being brought to bear on hard problems in many industries, and chip development is no exception. Many electronic design automation (EDA) tools are starting to use AI/ML successfully to automate tasks, improve schedules, and optimize results. Synopsys is the clear EDA industry leader in the application of AI/ML to chip design, verification, implementation, and more. Specifically in the domain of simulation-based coverage closure, Synopsys VSO.ai is a unique and powerful solution.

There are many ways that Synopsys VSO.ai can help, and Figure 2 shows how it addresses the challenges illustrated in Figure 1. For the coverage definition challenge, Synopsys VSO.ai infers some types of coverage beyond traditional code coverage to complement user-specified coverage. ML can learn from experience and intelligently reuse coverage when appropriate. Even during a single project, learnings from earlier coverage results can help to improve coverage models.

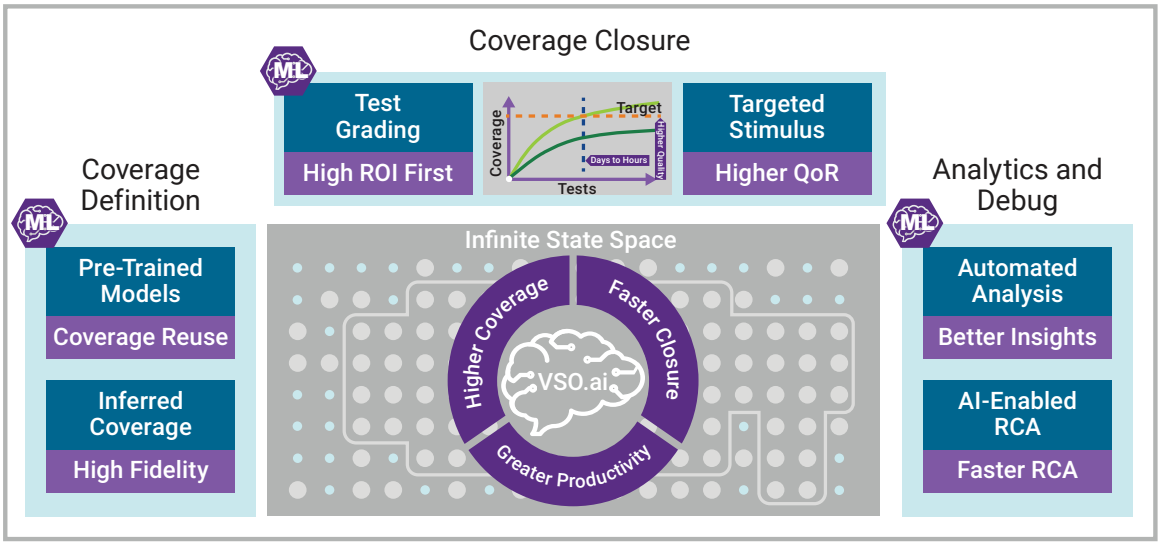

Figure 2: AI/ML improvements for coverage in simulation

The process of running better test simulations to tackle the declining ROI challenge is perhaps the most natural place for ML to help. As noted earlier, a lot of the tests are repetitive and yield little or no improvement in the coverage metrics. Manual regression optimization is fragile, since evolution of the design, coverage, and constraints over the course of the project may change the rankings many times. Trying to incorporate these changes by hand is impractical, but Synopsys VSO.ai works at the coarse-grained test level and provides automated, adaptive test optimization that learns as the results change. Running the tests with highest ROI first while eliminating redundant tests accelerates coverage closure and saves compute resources.

Synopsys VSO.ai also works at the fine-grained level within the simulator to improve the test quality of results (QoR) by adapting the constrained-random stimulus to better target unexercised coverage points. This not only accelerates coverage closure, but also drives convergence to a higher percentage value.

The last mile closure challenge is addressed by automated, AI-driven analysis of coverage results. Synopsys VSO.ai performs root cause analysis (RCA) to determine why specific coverage points are not being reached, for example due to a constraint conflict. If Synopsys VSO.ai can resolve the situation itself, it will, and otherwise it presents the verification engineers with actionable results such as identifying conflicting constraints.

Integration with Synopsys VCS

Effective deployment of AI/ML techniques in EDA solutions requires tight integration with traditional tools. Synopsys VSO.ai is no exception, and it achieves its unique advantages through a tight integration with Synopsys VCS. Figure 3 shows the traditional manual flow using Synopsys VCS to run tests and collect coverage.

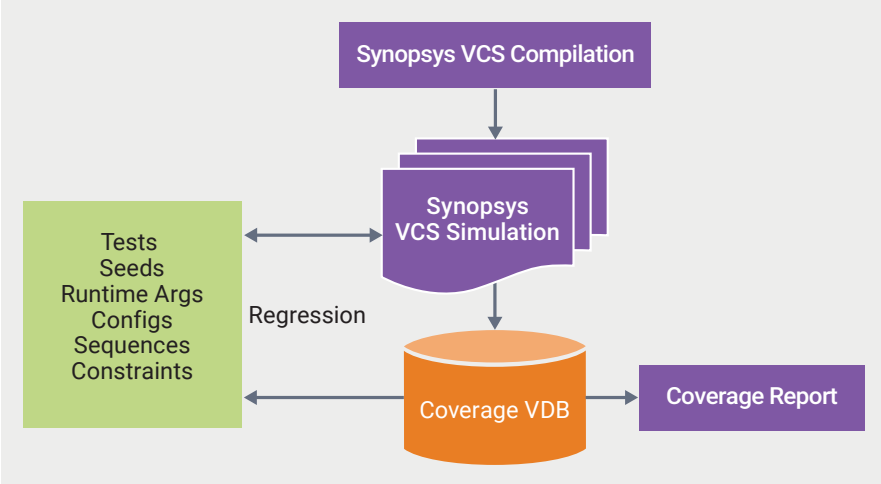

Figure 3: Manual simulation test flow

The design and verification environment (testbench) are compiled, and then the simulations are run. In addition to the constraints, there are several other types of information that may be specified by the verification team. These include various configuration switches and runtime arguments. As the tests run, coverage metrics are saved in a verification database (VDB). The results can be viewed in the form of a coverage report, and databases from multiple tests can be merged to generate a summary report.

In contrast, Figure 4 shows how this flow is enhanced with the use of Synopsys VSO.ai to automate several key steps. The coverage inference step occurs during compilation, complementing structural and user-specified coverage with automatically generated coverage. At a fine-grained level, the constraint solver within Synopsys VCS is directed by coverage, so it can more precisely generate new tests to hit unreached coverage points. The fact that the solver is deep within the simulator is one reason why tight integration is critical.

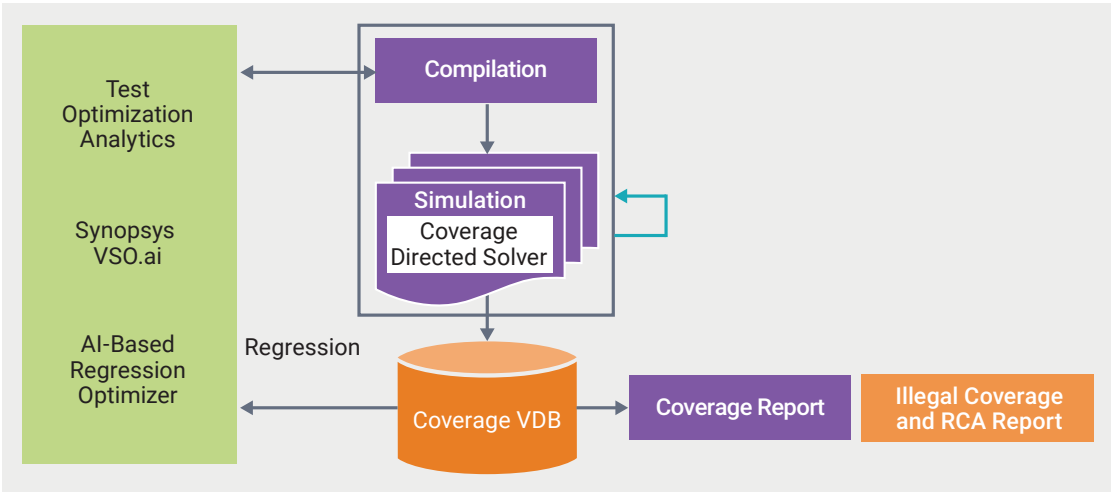

Figure 4: Simulation test flow with Synopsys VSO.ai

Manual test ordering and juggling based on test rankings are replaced by a fully automated flow. At a coarse-grained level, Synopsys VSO.ai organizes regressions for maximum ROI, using a minimal set of tests to save runtime and compute resources. It also has control of the Synopsys VCS settings and switches to optimize each test run. Likewise, manual analysis of coverage results is supplemented by RCA to identify coverage that may be unreachable and report root causes to resolve these situations.

If no higher coverage is required, then VSO.ai yields equivalent coverage results in far less time. In many cases, Synopsys VSO.ai uses ML techniques to improve coverage over time, as shown in Figure 5. Every run provides new information to improve constraint solving and to optimize the regression tests. Unlike manual approaches, this flow is fully adaptive as results change over the course of the chip project. ML can use the history from the current regression run, previous regression runs on the project, and even simulations run on similar previous projects.

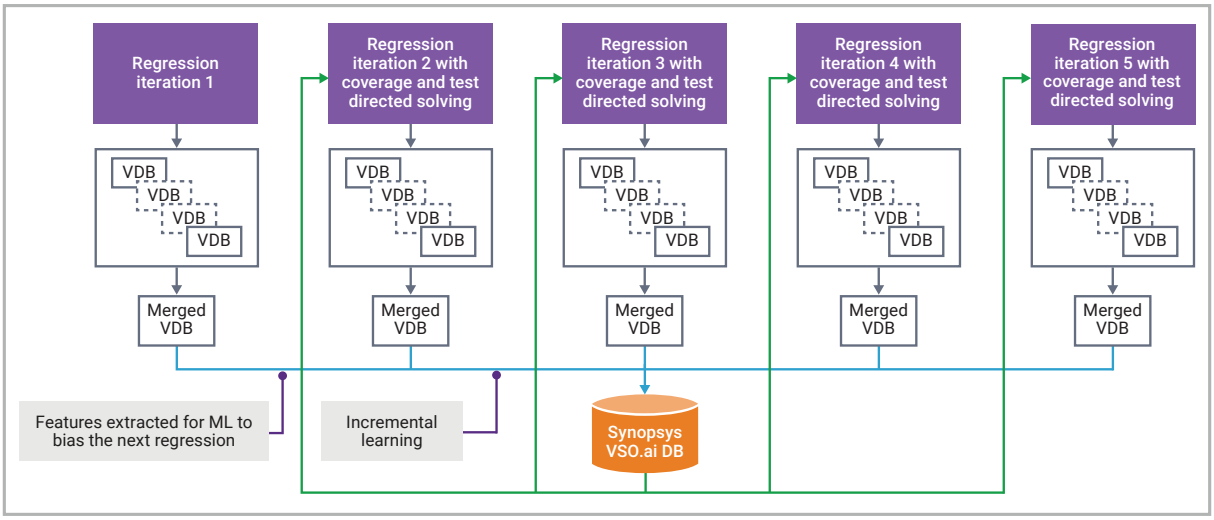

Figure 5: How Synopsys VSO.ai uses ML to improve regressions

Synopsys VSO.ai is also valuable in the development stage when running a subset of the full regression, with or without new changes, to ensure verification quality as measured by user or tool defined metrics. The benefit in this case is faster TTR to quickly get to the same or similar level of coverage.

Synopsys VSO.ai Results

Synopsys VSO.ai optimizes simulation regressions to achieve the same coverage in less time or to improve coverage as needed. For example, one customer achieved at least a 2X reduction in number of regression tests across four IP blocks, as presented at a recent Synopsys Users Group (SNUG) event. In another example, shown in Figure 6, VSO.ai was able to reduce the number of tests to achieve 100% functional coverage by 3X on the OpenTitan HMAC intellectual property (IP) block.

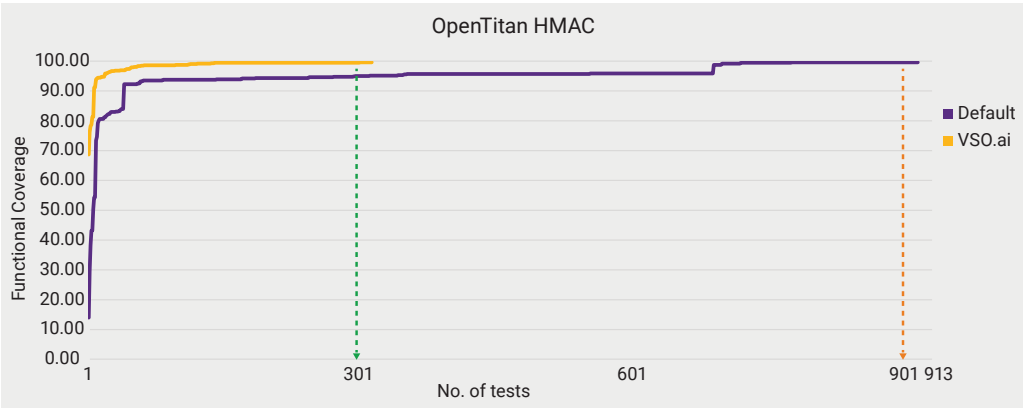

Figure 6: Synopsys VSO.ai results for OpenTitan HMAC

Experience with a wide variety of both IP blocks and full system-on-chip (SoC) designs has shown impressive results. Time to achieve coverage goals is reduced 1.5-10X and coverage results are improved by as much as 10% due to regression coverage optimization and coverage-directed solving. Typically, 5-20% of the coverage bins are identified as potentially unreachable due to RCA. The value of Synopsys VCO.ai is clear and undeniable.

Conclusion

Coverage is still an incomplete proxy for finding bugs, and there is no silver bullet for addressing the challenge of coverage closure. As in many other areas of EDA, AI/ML technology brings relief. Synopsys VCS and VSO.ai combine for the industry's most innovative and most complete solution for accelerating regressions and coverage closure.

05/15/23.CS1077133425-VSO.ai-WP.

Pub: March 2023