

## Chapter 9

# Functional Coverage

As designs become more complex, the only effective way to verify them effectively is with constrained-random testing (CRT). This approach elevates you above the tedium of writing individual directed tests, one for each feature in the design. However, if your testbench is taking a random walk through the space of all design states, how do you know if you have reached your destination? Even directed tests should be double checked with functional coverage. Over the life of a project, small changes in the DUT's timing or functionality can subtly alter the results from a directed test, so no longer verifies the same features. Whether you are using random or directed stimulus, you can gauge progress using coverage.

Functional coverage is a measure of which design features have been exercised by the tests. Start with the design specification and create a verification plan with a detailed list of what to test and how. For example, if your design connects to a bus, your tests need to exercise all the possible interactions between the design and bus, including relevant design states, delays, and error modes. The verification plan is a map to show you where to go. For more information on creating a verification plan, see Bergeron (2006).

In many complex systems, you may never achieve 100% coverage as schedules don't allow you to reach every possible corner case. After all, you didn't have time to write directed tests to get sufficient coverage, and even CRT is limited by the time it takes you to create and debug test cases, and analyze the results.

Figure 9-1 shows the feedback loop to analyze the coverage results and decide on which actions to take in order to converge on 100% coverage. Your first choice is to