

Chapter 6

Randomization

6.1 Introduction

As designs grow larger, it becomes more difficult to create a complete set of stimuli needed to check their functionality. You can write a directed testcase to check a certain set of features, but you cannot write enough directed testcases when the number of features keeps doubling on each project. Worse yet, the interactions between all these features are the source for the most devious bugs and are the least likely to be caught by going through a laundry list of features.

The solution is to create test cases automatically using constrained-random tests (CRT). A directed test finds the bugs you think are there, but a CRT finds bugs you never thought about, by using random stimulus. You restrict the test scenarios to those that are both valid and of interest by using constraints.

Creating the environment for a CRT takes more work than creating one for directed tests. A simple directed test just applies stimulus, and then you manually check the result. These results are captured as a golden log file and compared with future simulations to see whether the test passes or fails. A CRT requires an environment to predict the result, using a reference model, transfer function, or other techniques, plus functional coverage to measure the effectiveness of the stimulus. However, once this environment is in place, you can run hundreds of tests without having to hand-check the results, thereby improving your productivity. This trade-off of test-authoring time (your work) for CPU time (machine work) is what makes CRT so valuable.