Chapter 11

A Complete SystemVerilog Testbench

This chapter applies the many concepts you have learned about SystemVerilog features to verify a design. The testbench creates constrained random stimulus, and gathers functional coverage. It is structured according to the guidelines from Chapter 8 so you can inject new behavior without modifying the lower-level blocks.

The design is an ATM switch that was shown in Sutherland [2006], who based his SystemVerilog description on an example from Janick Bergeron's Verification Guild. Sutherland took the original Verilog design and used SystemVerilog design features to create a switch that can be configured from 4x4 to 16x16. The testbench in the original example creates ATM cells using `$urandom`, overwrites certain fields with ID values, sends them through the device, then checks that the same values were received.

The entire example, with the testbench and ATM switch, is available for download at `http://chris.spear.net/systemverilog`. This chapter shows just the testbench code.

11.1 Design Blocks

The overall connection between the design and testbench, shown in Figure 11-1, follows the pattern shown in Chapter 4.