VHDL synthesizer but not a VHDL simulator. For the designs of the complexity we have covered so far, little time is lost in not performing the source code simulation. Warp3 and other EDA tools that contain VHDL simulators allow you to simulate both the source code—to verify its functionality before you spend time to synthesize and fit the design to a device—and a post-layout VHDL model.

VHDL simulators typically allow interactive simulations, wherein you can apply vectors either by editing waveforms or using commands provided by the simulator. However, the real power behind VHDL simulation is the ability to write test benches. A test bench is a self-contained VHDL entity used for generating and applying vectors to another VHDL model, and verifying the accuracy of its outputs.

If you have access to Warp3 or another VHDL simulator, you may now wish to spend time becoming familiar with its command language. Simulating a few of the listings of this chapter would suffice. If you have access to Warp3 or another VHDL simulator and you wish to now focus on test benches rather than design techniques, descriptions, and their implementations in programmable logic, you should read Chapter 10, "Creating Test Benches" before proceeding with Chapter 5.

Problems

4.1. Translate the following code to an `if-then-else` statement:

```
transmit <= signal_a when state = idle else
            signal_b when state = incoming else
            signal_c when state = outgoing else
            signal_d;
```

4.2. Translate the following code to a `when-else` statement:

```
process (a, b, j, k)
   begin
      if a = '1' and b = '0' then
         step <= "0100";
      elsif a = '1' then
         step <= j;
      elsif b = '1' then
         step <= k;
      else
         step <= "----";
      end if;
   end process;
```