10.1 VHDL for Control Units

The following shows the behavioral VHDL code for the PS/2 keyboard controller. The format follows exactly the VHDL code for Moore FSMs discussed in Section 7.6. Notice how closely the code for the FSM process follows the state diagram. Because of its simplicity, control units usually are synthesized this way rather than constructed manually. For this example, the output process does not generate output signals to control the keyboard; rather, it reads in data signals from the keyboard and packages the eight data bits as a byte for output.

```vhdl
LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.ALL;

ENTITY KeyboardCtrl IS
  PORT(
    Reset: IN STD_LOGIC;
    KeyboardClock: IN STD_LOGIC;
    KeyboardData: IN STD_LOGIC;
    keycode: OUT STD_LOGIC_VECTOR(7 DOWNTO 0));
END KeyboardCtrl;

ARCHITECTURE Behavioral OF KeyboardCtrl IS
  TYPE state_type IS (s_start, s_d0, s_d1, s_d2, s_d3, s_d4, s_d5, s_d6, s_d7, s_parity, s_stop);
  SIGNAL state: state_type;
BEGIN
  FSM: PROCESS(KeyboardClock, Reset)
  BEGIN
    IF (Reset = '1') THEN
      state <= s_start;
      -- this FSM is driven by the keyboard clock signal
    ELSIF (KeyboardClock'EVENT AND KeyboardClock = '1') THEN
      CASE state is
        WHEN s_start =&gt;
          state <= s_d0;
        WHEN s_d0 =&gt;
          state <= s_d1;
        WHEN s_d1 =&gt;
          state <= s_d2;
        WHEN s_d2 =&gt;
          state <= s_d3;
        WHEN s_d3 =&gt;
          state <= s_d4;
        WHEN s_d4 =&gt;
          state <= s_d5;
        WHEN s_d5 =&gt;
          state <= s_d6;
        WHEN s_d6 =&gt;
          state <= s_d7;
        WHEN s_d7 =&gt;
          state <= s_parity;
        WHEN s_parity =&gt;
          state <= s_stop;
        WHEN s_stop =&gt;
          state <= s_start;
      END CASE;
    END IF;
  END PROCESS;
END Behavioral;
```
VHDL code for the PS/2 keyboard controller.