Write a VHDL module for the sequential machine using a ROM (as in Figure 17-22) and a straight binary assignment.

17.23 Repeat Problem 17.22 using equations as in Figure 17-19 and using a one-hot state assignment. (*Hint*: It may be easier to do the one-hot state assignment properly if you draw the state graph first.)

17.24 The following VHDL code is for a 2-to-1 MUX, but it contains mistakes. What are the mistakes?

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
entity mux2 is
  port (d0, d1 : in bit;
  sel : in Boolean;
  z : out bit);
end mux2;

architecture bvhr of mux2 is
  signal muxsel : integer range 0 to 1;
begin
 process(d0, d1, select)
 begin
  muxsel <= 0;
  if sel then muxsel <= muxsel + 1; end if;
  case muxsel is
   when 0 => z <= d0 after 2ns;
   when 1 => z <= d1 after 2ns;
  end case;
 end process;
end bvhr;
```

17.25 Give the state table implemented by the following VHDL code.

```
entity Problem17_25 is
  port(X, CLK: in bit;
  Z1, Z2: out bit);
end Problem17_25;

architecture Table of Problem17_25 is
  signal State, Nextstate: integer range 0 to 3 := 0;
begin
  process(State, X)       --Combinational Circuit
  begin
    case State is
    when 0 =>
```