```vhdl
library ieee;
use ieee.std_logic_1164.all;
use work.all;

entity state_machine3 is
   port(x0, x1: in std_logic;
      clock : in std_logic;
      reset: in std_logic;
      y0, y1: out std_logic);
end state_machine3;

architecture behave of state_machine3 is

   type states is (st1, st2, st3, st4);
   signal present_state : states;
   signal next_state: states;

begin

   clkd: process(clock, reset)
   begin
      if (reset = '1') then
         present_state <= st1;
      elsif (clock'event and clock = '1') then
         present_state <= next_state;
      end if;
   end process clkd;

   state_trans: process(present_state, x0, x1)
   begin
      case present_state is
         when st1 =>
            if (x0 = x1) then
               next_state <= st2;
               y0 <= x0; y1 <= x1;
            elsif (x0 = '0') then
               next_state <= st1;
               y0 <= '0'; y1 <= '1';
            else
               next_state <= st1;
               y0 <= '1'; y1 <= '1';
            end if;
         when st2 =>
            if (x0='0' and x1='1') then
               next_state <= st3;
               y0 <= '0'; y1 <='0';
            else
               next_state <= st4;
               y0 <= '1'; y1 <= '0';
            end if ;
         when st3 =>
```

```
                if (x0='0' and x1='0') then
                   next_state <= st3;
                   y0<='0'; y1<='0';
                elsif (x0='1' and x1='0') then
                   next_state <= st4;
                   y0<='0'; y1<='1';
                else
                   next_state <= st4;
                   y0<='0'; y1<='0';
                end if ;
             when st4 =>
                if (x0='0' and x1='1') then
                   next_state <= st4;
                   y0<='1'; y1<='1';
                elsif (x0='1' and x1='0') then
                   next_state <= st1;
                   y0<='1'; y1<='1';
                else
                   next_state <= st2;
                   y0<='0'; y1<='1';
                end if ;
          end case;
       end process state_trans;
    end behave;
```