Charles Owen

Embedded Systems
HW 6, Chapter 11

Table of Contents

Problem 2

```
-- Charles Owen
-- Embedded Systems
-- HW6
-- Problem 2


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity HW6 is
    Port(

    CLK     : in std_logic;
    X1, X2  : in std_logic;
    Y       : out std_logic_vector(1 downto 0);
    Z       : out std_logic

    );
end HW6;

architecture Behavioral of HW6 is

type state_type is (A, B, C);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type is "10 11 01";
signal PS, NS : state_type;

begin

sync_proc: process(CLK, NS)
           begin

           if (rising_edge(CLK)) then PS <= NS;
           end if;

           end process;

comb_proc: process(CLK, PS, X1, X2)
           begin

           Z <= '0';

           case PS is
           when A =>
            if(X1 = '0') then
            NS <= A;
            Z <= '0';
            elsif(X1='1') then
            NS <= C;
            Z <= '0';
            end if;
```

```
           when B =>
            if(X2 = '0') then
            NS <= A;
            Z <= '1';
            elsif(X2 = '1') then
            NS <= B;
            Z <= '0';
            end if;

           when C =>
            if(X2 = '0') then
            NS <= A;
            Z <= '0';
            elsif(X2 = '1') then
            NS <= B;
            Z <= '0';
            end if;

           when others =>
            Z <= '1'; NS <= A;

            end case;

           end process;

with PS select
    Y <= "10" when A,
         "11" when B,
         "01" when C,
         "10" when others;

end Behavioral;
```

Problem 4

```
-- Charles Owen
-- Embedded Systems
-- HW 6
-- Number 4


library IEEE;
use IEEE.STD_LOGIC_1164.ALL;
use IEEE.NUMERIC_STD.ALL;

entity number_4 is
    Port(

    CLK          : in std_logic;
    Z1, Z2       : out std_logic;
    X1, X2, INIT : in std_logic

    );
end number_4;

architecture Behavioral of number_4 is

type state_type is (A, B, C);
signal PS, NS   : state_type;

begin

sync_p  : process(CLK, NS, INIT)
          begin

          if(INIT = '1') then PS <= A; -- Treating INIT as a reset, because
not sure what to do.
          elsif(rising_edge(CLK)) then PS <= NS; -- sent email to inquire but
didn't hear back
          end if;

          end process;

comb_p  : process(PS, X1, X2)
          begin

          case (PS) is
          when A => Z1 <= '0';
           if (X1 = '1') then NS <= B; Z2 <= '1';
           else NS <= C; Z2 <= '0';
           end if;

          when B => Z1 <= '1';
           if (X1 = '1') then NS <= A; Z2 <= '0';
           else NS <= C; Z2 <= '1';
           end if;
```

```
          when C => Z1 <= '1';
           if (X1 = '1') then NS <= B; Z2 <= '1';
           else NS <= A; Z2 <= '1';
           end if;
          end case;

          end process;


end Behavioral;
```

Problem 6

```
-- Charles Owen
-- Ebmedded Systems
-- HW 6
-- Problem 6

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity number_6 is
    port(

    CLK     : in std_logic;
    X       : in std_logic;
    Z1, Z2  : out std_logic;
    Y       : out std_logic_vector(1 downto 0)

    );
end number_6;

architecture Behavioral of number_6 is

type state_type is (A, B, C, D);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type
    is "00 01 11 10";

signal NS, PS   : state_type;

begin

synch_p: process (CLK, NS)
         begin

         if(rising_edge(CLK)) then PS <= NS;

         end if;

         end process;

comb_p: process (X, PS)
        begin

        case PS is

        when A => Z1 <= '1';
         if(X = '1') then NS <= A; Z2 = '0';
         else NS <= D; Z2 <= '0';
         end if;

        when B => Z1 <= '0';
         if(X = '1') then NS <= B; Z2 = '0';
```

```
         else NS <= C; Z2 = '0';
         end if;

        when C => Z1 <= '0';
         if(X = '1') then NS <= B; Z2 = '0';
         else NS <= A; Z2 = '1';
         end if;

        when D => Z1 <= '1';
         if(X = '1') then NS <= A; Z2 = '0';
         else NS <= B; Z2 = '0';
         end if;

        end process;

with PS select

    Y <= "00" when A,
         "01" when B,
         "11" when C,
         "10" when D,
         "00" when others;


end Behavioral;
```

Problem 12

```
-- Charles Owen
-- Embedded Systems
-- HW 6
-- Problem 12

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity number_12 is
    Port(

    CLK         : in std_logic;
    X1, X2      : in std_logic;
    Z1, Z2      : out std_logic;
    Y           : out std_logic_vector(1 downto 0)

    );
end number_12;

architecture Behavioral of number_12 is

type state_type is (A, B, C);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type
    is "11 01 00"; -- in order A, B, C

signal PS, NS   : state_type;

begin

synch_p: process (CLK, NS)
         begin

         if (rising_edge(CLK)) then

         PS <= NS;

         end if;

         end process;

comb_p: process (PS, X1, X1)
        begin

        case (PS) is

        when A => Z2 <= '1';
         if (X1 = '0') then NS <= A; Z1 <= '0';
         else NS <= B; Z1 <= '1';
         end if;

        when B => Z2 <= '0';
```

```
         if (X2 = '0') then NS <= C; Z1 <= '1';
         else NS <= A; Z1 <= '0';
         end if;

        when C => Z2 <= '1';
         if (X2 = '1') then NS <= C; Z1 <= '1';
         else NS <= A; Z1 <= '0';
         end if;

        end case;

        end process;

 with PS select

 Y <= "11" when A,
      "01" when B,
      "00" when C,
      "11" when others;


end Behavioral;
```

Problem 13

```
-- Charles Owen
-- Embedded Systems
-- HW 6
-- Problem 13

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;

entity number_13 is
    Port (

    CLK         : in std_logic;
    X1, X2      : in std_logic;
    CS, RD      : out std_logic;
    Y           : out std_logic_vector(2 downto 0)

    );
end number_13;

architecture Behavioral of number_13 is

type state_type is (A, B, C);
attribute ENUM_ENCODING: STRING;
attribute ENUM_ENCODING of state_type: type
    is "001 010 100";

signal PS, NS   : state_type;

begin

synch_p: process (CLK, NS)
         begin

         if (rising_edge(CLK)) then

         PS <= NS;

         end if;

         end process;

comb_p: process (PS, X1, X2)
        begin

        case (PS) is

         when A =>
          if (X1 = '0') then NS <= B; CS <= '0'; RD <= '1';
          else NS <= C; CS <= '1';RD <= '0';
          end if;

         when B =>
```

```
          if (X1 = '0' or X1 = '1' or X2 = '0' or X2 = '1') then NS <= C; CS
<= '1'; RD <= '1';
          end if;

         when C =>
          if (X2 = '0') then NS <= A; CS <= '0'; RD <= '0';
          else NS <= C; CS <= '0'; RD <= '1';
          end if;

         end case;
        end process;

-- select simulated output below this line
 with PS select
   Y <= "001" when A,
        "010" when B,
        "100" when C,
        "001" when others;


end Behavioral;
```