Kevin Honeker
HW 6
3/13
--PROBLEM 2

```
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity prob2 is
    Port ( x1 : in STD_LOGIC;
           x2 : in STD_LOGIC;
          clk : in STD_LOGIC;
           z  : out STD_LOGIC;
           y  : out STD_LOGIC_VECTOR(1 downto 0));
end prob2;

architecture arch2 of prob2 is
type state_type is (sta, stb, stc);
signal ps, ns : state_type;
signal machine_output : std_logic_vector(1 downto 0);

begin
    process (clk)
    begin
        if clk'event and clk = '1' then
            ps <= ns;
         --  y <= machine_output;
         end if;
    end process;

    process (ps, x1, x2)
    begin
        if (ps = sta and x1 = '0') then
            z<='0';
            ns <= sta;
        elsif (ps = sta and x1 ='1') then
            z<= '0';
            ns <= stc;
        elsif (ps = stc and x2 = '0') then
                z<= '1';
                ns <= sta;
        elsif (ps = stc and x2 = '1') then
```

```
            z<= '0';
            ns<= stb;
        elsif (ps = stb and x2 = '0') then
            z<='0';
            ns<=stb;
        elsif (ps = stb and x2 = '0') then
            z<='1';
            ns<= sta;
         end if;
end process;
with ps select
   y <= "10" when sta,
        "01" when stc,
        "11" when stb,
        "10" when others;



end arch2;

--PROBLEM 4
library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity prob4 is
   Port ( x1 : in STD_LOGIC;
          x2 : in STD_LOGIC;
          z1 : out STD_LOGIC;
          z2 : out STD_LOGIC;
          clk : in STD_LOGIC;
          y : out STD_LOGIC_VECTOR (2 downto 0));
end prob4;

architecture arch4 of prob4 is

type state_type is (sta, stb, stc);
signal ps, ns : state_type;
--signal machine_output : std_logic_vector(1 downto 0);

begin
   process(clk)
   begin
```

```
    if clk'event and clk = '1' then
        ps <= ns;
      --  y <= machine_output;
     end if;
end process;

process(ps, x1, x2)
begin
    ps<=sta;
    z1<='0';

        case ps is
            when sta =>
                z1<='0';
                if x1 = '0' then
                    z2<='0';
                    ns<=stc;
                elsif x1 = '1' then
                    z2<='1';
                    ns<=stb;
                end if;

            when stb =>
                z1<='1';
                if x2 = '0' then
                    z2<='1';
                    ns<=stb;
                 elsif x2 = '1' then
                    z2<='0';
                    ns<=sta;
                 end if;
            when stc =>
                z1<='1';
                if x1 ='0' then
                    z2 <= '1';
                    ns<= sta;
                elsif x1 = '1' then
                    z2 <='1';
                    ns <= stb;
                end if;
            when others =>
                ns<=sta;
                z1<='0';
```

```
            end case;
        end process;

     with ps select
        y <= "001" when sta,
            "010" when stb,
            "100" when stc,
            "001" when others;
end arch4;

--PROBLEM 6

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;


entity prob6 is
    Port ( x : in STD_LOGIC;
           z1 : out STD_LOGIC;
           z2 : out STD_LOGIC;
           y : out STD_LOGIC_VECTOR (1 downto 0);
           clk : in STD_LOGIC);
end prob6;

architecture arch6 of prob6 is
type state_type is (st0, st1, st2, st3);
signal ps, ns : state_type;
begin

    process(clk)
    begin
        if clk'event and clk = '1' then
            ps <= ns;

         end if;
    end process;

    process(ps, x)
    begin
        case ps is
            when st0 =>
                z1<='1';
                if x = '1' then
```

```
                ns<=st0;
                z2<='0';
            elsif x='0' then
                ns<=st2;
                z2<='0';
            end if;
        when st2 =>
            z1<='1';
            if x = '1' then
                ns<=st0;
                z2<='0';
            elsif x='0' then
                ns<=st1;
                z2<='0';
            end if;
        when st1 =>
            z1<='0';
            if x='1' then
                ns<=st1;
                z2<='0';
            elsif x='0' then
                ns<=st3;
                z2<='0';
            end if;
        when st3 =>
            z1<='0';
            if x<='1' then
                ns<=st1;
                z2<='0';
            elsif x='0' then
                ns<=st0;
                z2<='1';
            end if;
        when others =>
                ns<=st0;
                z2<='0';
    end case;
  end process;
with ps select
  y <= "00" when st0,
    "01" when st1,
    "10" when st2,
    "11" when st3,
```

```
        "00" when others;
end arch6;

--PROBLEM 13

library IEEE;
use IEEE.STD_LOGIC_1164.ALL;



entity prob13 is
    Port ( x1 : in STD_LOGIC;
           x2, clk : in STD_LOGIC;
           y : out STD_LOGIC_VECTOR (2 downto 0);
           cs : out STD_LOGIC;
           rd : out STD_LOGIC);
end prob13;

architecture arch13 of prob13 is
type state_type is (sta, stb, stc);
signal ps, ns : state_type;
begin
 process(clk)
    begin
        if clk'event and clk = '1' then
            ps <= ns;

         end if;
    end process;
 process(ps, x1, x2)
    begin
    case ps is
        when sta =>
            if x1='1' then
                ns<=stc;
                cs<='1';
                rd<='0';
             elsif x1 = '0' then
                ns<=stb;
                cs<='0';
                rd<='1';
             end if;
         when stb=>
```

```
            ns<=stc;
            rd<='1';
            cs<='1';
          when stc =>
            if x2 ='1' then
               ns<=stc;
               cs<='0';
               rd<='1';
            elsif x2='0' then
               ns<=sta;
               cs<='0';
               rd<='0';
            end if;
          when others =>
            ns<= sta;
          end case;
      end process;
 with ps select
    y <= "001" when sta,
         "010" when stb,
         "100" when stc;
end arch13;
```