Homework Assignment Four

Due: Wednesday, March 16, 2005 at the beginning of class

Problem I  Problem 5.16, Brown/Vranesic textbook, page 311

5.16. (a) LIBRARY ieee;
    USE ieee.std_logic_1164.all;

    ENTITY row0 IS
        PORT ( q0, q1, cin, mk, mkp1 : IN STD_LOGIC;
            s, cout : OUT STD_LOGIC );
    END row0;

    ARCHITECTURE LogicFunc OF row0 IS
        SIGNAL a0, a1 : STD_LOGIC;
    BEGIN
        a0 <= q0 AND mkp1;
        a1 <= q1 AND mk;
        s <= cin XOR a0 XOR a1;
        cout <= (cin AND a0) OR (cin AND a1) OR (a0 AND a1);
    END LogicFunc;

(b) LIBRARY ieee;
    USE ieee.std_logic_1164.all;

    ENTITY row1 IS
        PORT ( q1, cin, mk, BitPpi : IN STD_LOGIC;
            s, cout : OUT STD_LOGIC );
    END row1;

    ARCHITECTURE LogicFunc OF row1 IS
        SIGNAL a0 : STD_LOGIC;
    BEGIN
        a0 <= q1 AND mk;
        s <= cin XOR a0 XOR BitPpi;
        cout <= (cin AND a0) OR (cin AND BitPpi) OR (a0 AND BitPpi);
    END LogicFunc;
Problem II. Problem 5.17, Brown/Vranesic textbook, page 311

5.17. The code in Figure P5.2 represents a multiplier. It multiplies the lower two bits of Input by the upper two bits of Input, producing the four-bit Output. The style of code is poor, because it is not readily apparent what is being described.
### Problem III. Problem 6.4, Brown/Vranesic textbook, page 372

6.4.

<table>
<thead>
<tr>
<th>$w_1$</th>
<th>$w_2$</th>
<th>$w_3$</th>
<th>$f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\text{w}_1 & \rightarrow f \\
\text{w}_2 \bar{\text{w}}_3 & \\
\text{w}_2 + \bar{\text{w}}_3 & \\
\end{align*}
\]

![Logic diagram for Problem 6.4](image1)

### Problem IV. Problem 6.16, Brown/Vranesic textbook, page 374

6.16. Using Shannon’s expansion in terms of $w_3$ we have

\[
\begin{align*}
f & = \overline{w}_3 w_2 + w_3 (w_1 + \overline{w}_2) \\
& = \overline{w}_3 w_2 + w_3 (\overline{w}_2 + w_2 w_1)
\end{align*}
\]

The corresponding circuit is

![Logic diagram for Problem 6.16](image2)
Problem V.  Problem 6.18, Brown/Vranesic textbook, page 375

6.18. The code in Figure P6.2 is a 2-to-4 decoder with an enable input. It is not a good style for defining this decoder. The code is not easy to read. Moreover, the VHDL compiler often turns if statements into multiplexers, in which case the resulting decoder may have multiplexers controlled by the En signal on the output side.

Problem VI.  Problem 6.20, Brown/Vranesic textbook, page 375

6.20.  
LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY prob6.20 IS
    PORT ( w : IN STD_LOGIC_VECTOR(1 TO 3);
           f : OUT STD_LOGIC );
END prob6.20;

ARCHITECTURE Behavior OF prob6.20 IS
BEGIN
    WITH w SELECT
        f <= '0' WHEN "000",
            '0' WHEN "100",
            '0' WHEN "111",
            '1' WHEN OTHERS;
END Behavior;
Problem VII. Problem 6.26, Brown/Vranesic textbook, page 375

6.26.  LIBRARY ieee;
USE ieee.std_logic_1164.all;

ENTITY if2to4 IS
  PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
         En : IN STD_LOGIC;
         y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) );
END if2to4;

ARCHITECTURE Behavior OF if2to4 IS
BEGIN
  PROCESS ( En, w )
  BEGIN
    IF En = '0' THEN
      y <= "0000" ;
    ELSE
      IF w = "00" THEN
        y <= "0001" ;
      ELSIF w = "01" THEN
        y <= "0010" ;
      ELSIF w = "10" THEN
        y <= "0100" ;
      ELSE
        y <= "1000" ;
      END IF;
    END IF ;
  END PROCESS ;
END Behavior ;

LIBRARY ieee;
USE ieee.std_logic_1164.all;
PACKAGE if2to4_package IS
  COMPONENT if2to4
    PORT ( w : IN STD_LOGIC_VECTOR(1 DOWNTO 0);
           En : IN STD_LOGIC ;
           y : OUT STD_LOGIC_VECTOR(3 DOWNTO 0) );
  END COMPONENT ;
END if2to4_package ;
LIBRARY ieee;
USE ieee.std_logic_1164.all;
USE work.ihu2o4_package.all;

ENTITY h3to8 IS
  PORT ( w : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
         En : IN STD_LOGIC;
         y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
END h3to8;

ARCHITECTURE Structure OF h3to8 IS
  SIGNAL EnableTop, EnableBot : STD_LOGIC;
  BEGIN
    EnableTop <= w(2) AND En;
    EnableBot <= (NOT w(2)) AND En;

    Decoder1: if2to4 PORT MAP ( w( 1 DOWNTO 0 ), EnableBot, y( 3 DOWNTO 0 ) );
    Decoder2: if2to4 PORT MAP ( w( 1 DOWNTO 0 ), EnableTop, y( 7 DOWNTO 4 ) );
  END Structure;

LIBRARY ieee;
USE ieee.std_logic_1164.all;
PACKAGE h3to8_package IS
  COMPONENT h3to8
    PORT ( w : IN STD_LOGIC_VECTOR(2 DOWNTO 0);
           En : IN STD_LOGIC;
           y : OUT STD_LOGIC_VECTOR(7 DOWNTO 0) );
  END COMPONENT;
END h3to8_package;
Problem VIII. Problem 6.31, Brown/Vranesic textbook, page 376
Complete this problem both schematically (with muxes) and in VHDL.

6.31. Using an arrangement similar to Figure 6.56, the desired circuit can be specified by the following truth table:

<table>
<thead>
<tr>
<th>Left</th>
<th>Right</th>
<th>$h$</th>
<th>$y'_3$</th>
<th>$y'_2$</th>
<th>$y'_1$</th>
<th>$y'_0$</th>
<th>$k$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>$w'_3$</td>
<td>$w'_2$</td>
<td>$w'_1$</td>
<td>$w'_0$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>$w'_0$</td>
<td>$w'_3$</td>
<td>$w'_2$</td>
<td>$w'_1$</td>
<td>$w'_0$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$w'_3$</td>
<td>$w'_2$</td>
<td>$w'_1$</td>
<td>$w'_0$</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

Using multiplexers, this truth table may be realized as:

[Diagram showing a circuit with multiplexers and the values of $h$, $y'_3$, $y'_2$, $y'_1$, $y'_0$, and $k$.]