Department of Electrical and Computer Engineering
University of Wisconsin - Madison

ECE/Comp Sci 352
Digital System Fundamentals
SOLUTIONS - Quiz # 2
March 8, 2001

Closed Book Examination
75 minutes

1) No calculators, hand-held or laptop computers, cellular phones or pagers allowed.
2) If a box is provided, the contents of the box will be graded as the final answer.
3) Show your work for consideration of partial credit.
4) If you write anything on the back of a page and want it considered in the grading you must write “See back of page (give page number)” on the problem page.
5) There is a sheet of identities at the end of the exam for your use.

<table>
<thead>
<tr>
<th>Problem</th>
<th>Points</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>22</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>16</td>
<td></td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>100</strong></td>
<td>****</td>
</tr>
</tbody>
</table>
1. (a) (8 points) For each of the circuits below, indicate whether the output of the circuit together with its input variables will have even or odd parity.

(b) (8 points) Fill in the truth table for the given circuit:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>C</th>
<th>T₁</th>
<th>T₂</th>
<th>T₃</th>
<th>T₄</th>
<th>F₁</th>
<th>F₂</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<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>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
2.  2a) (8 points) Implement the following functions using the decoder with inverted outputs and NAND gates and inverter shown below. No other logic may be used, only wires.

\[ F = \overline{A} B C + A \overline{B} C \]
\[ G = A B + A C = ABC + ABC + \overline{ABC} + \overline{ABC} \]

For the given variable order on diagram below, minterms of F are 6, 5, and of G are 3, 7, and 5. The solution with 6 and 3 interchanged was accepted too.

b) (8 points) A special encoder is to implement the following partial truth table. For all other input combinations, the encoded outputs \( C_3, C_2, C_1, C_0 \) are don’t cares. Give simplified expressions for \( C_3, C_2, C_1, \) and \( C_0. \)

<table>
<thead>
<tr>
<th>I1</th>
<th>I2</th>
<th>I3</th>
<th>I4</th>
<th>C3</th>
<th>C2</th>
<th>C1</th>
<th>C0</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

\[ C_3 = I_3 + I_4 \]
\[ C_2 = I_4 \]
\[ C_1 = I_2 + I_4 \]
\[ C_0 = I_1 + I_2 + I_4 \]
3. (a) (8 points) An 8-bit carry lookahead adder is given below.

(b) (6 points) A number has the 7-bit binary value 0101100. In the table below, give the 8-bit representation for positive and negative numbers using the representation specified.

<table>
<thead>
<tr>
<th>Operation</th>
<th>Representation</th>
</tr>
</thead>
<tbody>
<tr>
<td>+ A in Sign-Magnitude</td>
<td>0 0 1 0 1 1 0 0</td>
</tr>
<tr>
<td>– A in Sign-Magnitude</td>
<td>1 0 1 0 1 1 0 0</td>
</tr>
<tr>
<td>+ A in 1’s Complement</td>
<td>0 0 1 0 1 1 0 0</td>
</tr>
<tr>
<td>– A in 1’s Complement</td>
<td>1 1 0 1 0 0 1 1</td>
</tr>
<tr>
<td>+A in 2’s Complement</td>
<td>0 0 1 0 1 1 0 0</td>
</tr>
<tr>
<td>– A in 2’s Complement</td>
<td>1 1 0 1 0 1 0 0</td>
</tr>
</tbody>
</table>
4. (a) (12 points) A multiplexer and an inverter are shown below with the select input already defined. Implement the function \( F \) using the inverter and the multiplexer.

\[
F(A, B, C, D) = \overline{B \cdot C} + B \cdot D + A \cdot B \cdot \overline{C}
\]

b) (10 points) In a Xilinx field programmable gate array each cell consists of a structure that appears as inter-connected multiplexers. Programming is achieved by placing 0's and 1's stored in a memory on the data inputs to some of the multiplexers. A scaled-down version of a cell is shown below. Add the “select” input variables and the programming values (0's and 1's) on the data inputs on the four input multiplexers needed to implement the function. No other logic may be used! [Hint: expand the expression into the form:

\[
G = A \cdot B \cdot G_0(C, D, E, F) + \overline{A} \cdot B \cdot G_1(C, D, E, F) + A \cdot B \cdot G_2(C, D, E, F) + A \cdot B \cdot G_3(C, D, E, F).
\]

\[
G(A, B, C, D, E, F) = A \cdot B \cdot \overline{C} \cdot \overline{D} + A \cdot B \cdot \overline{C} \cdot D + B \cdot E \cdot \overline{F} + B \cdot E \cdot F + \overline{A} \cdot B \cdot \overline{C} + \overline{A} \cdot B \cdot E
\]

= \overline{A} \cdot B \cdot (C \cdot \overline{D} + \overline{C} \cdot D) + \overline{A} \cdot B \cdot (E \cdot \overline{F} + \overline{E} \cdot F) + A \cdot B \cdot (E \cdot \overline{F} + \overline{E} \cdot F) + \overline{A} \cdot B \cdot (C + E)
\]
5. a) (6 points) Perform the following computations on unsigned numbers. Clearly show your work for each computation given such as complement operations performed, carries or borrows out of the most significant bit (or most significant magnitude bit), etc. Don’t just give the final answer. Indicate in each case whether or not there is an overflow. If the result is negative, give the final answer as a sign (–) and a magnitude.

\[
\begin{array}{c}
111010011 \\
+11101110 \\
11000001
\end{array}
\quad \text{Overflow? Yes \ No}
\]

\[
\begin{array}{c}
111011111 \\
-111011110 \\
11110001 \Rightarrow (-)00001111
\end{array}
\quad \text{Overflow? Yes \ No}
\]

\[
\begin{array}{c}
11100001 \\
-11011110 \\
00000011
\end{array}
\quad \text{Overflow? Yes \ No}
\]

b) (10 points) Perform the following computations on signed numbers. The operands are given in the representation specified. Clearly show your work for each computation given such as complement operations performed, carries or borrows out of the most significant bit (or most significant magnitude bit), etc. Don’t just give the final answer. Note that for a complement representation, the answer are to be left in the representation. Indicate in each case whether or not there is an overflow.

**Sign-Magnitude**

\[
\begin{array}{c}
1 \\
10101011 \\
-11001011 \\
111000000 \Rightarrow 00100000
\end{array}
\quad \text{Overflow? Yes \ No}
\]

**2’s Complement**

\[
\begin{array}{c}
111010011 \\
+11101110 \\
11000001
\end{array}
\quad \text{Overflow? Yes \ No}
\]

**1’s Complement**

\[
\begin{array}{c}
1\text{100011001} \\
+1\text{100011001} \\
0\text{101001010}
\end{array}
\quad \text{Overflow? Yes \ No}
\]

\[
\begin{array}{c}
+1 \Rightarrow 0\text{10100111}
\end{array}
\]

**2’s Complement**

\[
\begin{array}{c}
1\text{0101010} \Rightarrow 1\text{0101010} \\
-1\text{0000000} \Rightarrow +1\text{0000000} \\
0\text{0101010}
\end{array}
\quad \text{Overflow? Yes \ No}
\]

**Sign-Magnitude**

\[
\begin{array}{c}
1\text{00011001} \\
+0\text{00011001} \\
1\text{00000000}
\end{array}
\quad \text{Overflow? Yes \ No}
\]
6. An engineer in an advanced extraterrestrial alien society is to design a digit adder for a code similar to our BCD. However, these aliens have six fingers on each hand instead of five. As a consequence, their digit arithmetic is base 12 with digits 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A and B. Using the two 4-bit binary adders given below, you are to design the logic in the outlined boxes to form a base 12 adder by answering each of the following questions:

a) (4 points) What value in binary must be added to the output from Z of the top adder to perform the correction for the base 12 addition?

Add value 0100

SPECIAL NOTE: If you can’t answer the above, contact an exam proctor and they will give you the value and deduct 4 points so that you can work the rest of the problem!

b) (8 points) Add minimized two-level logic for \( C_{\text{out}} \), the signal that indicates that a correction is to be made and a carry is to occur into the next base 12 digit, in the upper box in the diagram.

c) (4 points) Add the wires in the lower box in the diagram that apply the right constant value from part a) above or a 4-bit zero to the left port of the second adder.

\[ C = K + Z_3 Z_2 \]