### Boolean Algebra Identities

1. \( X + 0 = X \)
2. \( X \cdot 1 = X \)
3. \( X + 1 = 1 \)
4. \( X \cdot 0 = 0 \)
5. \( X + X = X \)
6. \( X \cdot X = X \)
7. \( X + \overline{X} = 1 \)
8. \( X \cdot \overline{X} = 0 \)
9. \( \overline{X} = X \)
10. \( X + Y = Y + X \)
11. \( X \cdot Y = Y \cdot X \)
12. \( X + (Y + Z) = (X + Y) + Z \)
13. \( X(YZ) = (XY)Z \)
14. \( X(Y + Z) = XY + XZ \)
15. \( X + YZ = (X + Y)(X + Z) \)
16. \( \overline{X + Y} = \overline{X} \cdot \overline{Y} \)
17. \( \overline{X \cdot Y} = \overline{X} + \overline{Y} \)
18. \( X + XY = X \)
19. \( X(X + Y) = X \)
20. \( XY + X\overline{Y} = X \)
21. \( (X + Y)(X + \overline{Y}) = X \)
22. \( X + X\overline{Y} = X + Y \)
23. \( X(\overline{X} + Y) = XY \)
24. \( XY + \overline{X}Z + YZ = XY + \overline{X}Z \)
25. \( (X + Y)(\overline{X} + Z)(Y + Z) = (X + Y)(\overline{X} + Z) \)

### Problem List

<table>
<thead>
<tr>
<th>PROBLEM</th>
<th>POINTS</th>
<th>SCORE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>20</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>15</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>TOTAL</td>
<td>100</td>
<td></td>
</tr>
</tbody>
</table>

- **Closed Book Examination**
- **Duration:** 75 minutes
- **No calculator or hand-held, lap-top computers allowed.**
- **Double check your answers. Only completely correct answers will receive full credit. Partial credits will be given conservatively.**
- **Please write your name on all pages.**
1. (Signed arithmetic)

(a) Perform the following arithmetic operations. The numbers are in 6-bit signed-magnitude form. The answer must also be in 6-bit signed magnitude form.

i. (2 pts) 001010 + 010010 =
   First add the magnitudes (01010 + 10010) to get 11100. Since both numbers are positive (as indicated by 0 in their sign bits), the result is also positive. Therefore, the answer is sign magnitude form is 011100.

ii. (2 pts) 001010 − 010010 =
    First subtract the magnitudes (01010 − 10010) to get 11000 with a borrow into the most significant bit. Since there is a borrow, the answer should be negative and the magnitude needs to be corrected by subtracting it from 100000. Since 100000 − 11000 is 010000, the correct answer is 101000.

(b) Perform the following arithmetic operations using 6-bit one’s complement representation. The answer must also be in 6-bit one’s complement representation.

i. (2 pts) 001010 + 010010 =
   First add the magnitudes and the signs together (001010 + 010010) to get 011100. Since there was no carry out of the MSB, the answer need not be corrected. Therefore, the answer is ones complement representation 011100.

ii. (2 pts) 010010 − 001110 =
    To subtract 001110, we take its ones complement and add. Ones complement of 001110 is 110001. 010010 − 001110 = 0100010 + 110001 = 1000011. Since there is a carry out of MSB, we need to add it back to the LSB. Therefore the answer is 000100.

(c) (8 pts) What are the largest (positive) number and the smallest (negative) number that can be represented in 8-bit two’s complement form? State your answers in both binary and decimal representation.

The largest (positive) number is 0111 1111 (+127). The smallest (negative) number is 1000 0000 (−128).
(d) Perform the following arithmetic operations using 8-bit two's complement representation. Indicate if there is an overflow.

i. (2 pts) \(01100100 + 00100110 = \) 
   
   First add the magnitudes and the signs together \((01100100 + 00100110)\) to get 10001010. Since the carry into the MSB was 1 and the carry out of the MSB was 0, there is an overflow. We could have also deduced that there was an overflow because we added two positive numbers and got a negative number as an answer.

ii. (2 pts) \(01100100 - 00100110 = \)

   Instead of subtraction we take the twos complement of the subtrahend and add. That is, \(01100100 - 00100110 = 01100100 + 11011010 = 00111110\) after ignoring the carry out of the MSB. The correct answer is 00111110.
2. (Logic Analysis)

(a) (10 pts) Find a sum-of-product Boolean expression for the output $Z$.

\[
T_1 = \overline{W}X = W + \overline{X} \\
T_2 = \overline{X}Z = X + \overline{Z} \\
T_3 = \overline{W} + X + \overline{Z} = W \overline{X}Z \\
T_4 = W \overline{X}Z + Y = WY + XY + Y \overline{Z} \\
f = WX + XZ + WY + XY + Y \overline{Z}.
\]

(b) (10 pts) The figure below shows a four variable Boolean function implemented using a 8-to-1 multiplexer. Find the minterm expression for the output.

\[
g = ABCD + A\overline{B}CD + \overline{A}BCD + \overline{A}BCD + A\overline{B}CD + A\overline{B}CD + A\overline{B}CD + ABCD \\
g = \sum m(1, 2, 4, 5, 8, 10, 15). 
\]
3. (Logic Design)
   (a) (10 pts) Let \((X)_{10} = (X_1X_0)_2\) and \((Y)_{10} = (Y_1Y_0)_2\) be two 2-bit signed numbers in two's complement representation. Complete the truth table for a comparator whose output \(Z\) is 1 if and only if \(X > Y\).

\[
\begin{array}{c|cccc|c}
X_1 & X_0 & Y_1 & Y_0 & Z \\
0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 1 \\
0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 0 \\
0 & 1 & 1 & 1 & 1 \\
0 & 1 & 0 & 0 & 0 \\
0 & 1 & 0 & 1 & 0 \\
0 & 1 & 1 & 0 & 1 \\
0 & 1 & 1 & 1 & 0 \\
1 & 0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 \\
\end{array}
\]

Since the numbers are in twos complement representation \(00 = 0\), \(01 = 1\), \(10 = -2\) and \(11 = -1\).

(b) (10 pts) Let \(F(A, B, C, D) = \sum m(0, 3, 5, 6, 8, 12, 15)\). Implement \(F\) using two 3-to-8 decoders, an OR gate, and as many inverters as necessary. Assume that the decoders have active high outputs and active high enable, and that the OR gate has as many inputs as necessary in your design. Input variables are available in true form only.

![](image)

Different input to decoder connections must be accompanied by corresponding decoder to OR gate connections.
4. (Arithmetic Units)

(a) (8 pts) Your design library contains a 4-bit unsigned adder, AND, OR, XOR, and NOT gates. Using the components in this library, design a network that has two 4-bit inputs, $A$ and $B$, one control input, $Y$, and one 4-bit output, $S$, and performs the operation specified by $Y$ as shown in the table below. Neatly draw a block-logic diagram of your design; use a minimum number of gates. Inputs $A$ and $B$ bring signed numbers in two's complement representation to the circuit; its result $S$ is also in two's complement form. Don’t worry about overflow or other status signals.

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$A + B$</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>$A - B$</td>
<td></td>
</tr>
</tbody>
</table>

(b) (7 pts) Now extend your arithmetic unit design by introducing control input $X$ and logic to perform a third arithmetic operation. Again draw a block-logic diagram.

<table>
<thead>
<tr>
<th>$X$</th>
<th>$Y$</th>
<th>$S$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$A + B$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$A - B$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$B - A$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>Don’t care</td>
</tr>
</tbody>
</table>
5. (Addition/Subtraction)

(a) (7 pts) Design (ending with a neatly drawn logic diagram) a “half-subtractor,” i.e., a circuit with input signals \(X\) and \(Y\) and output signals \(D\) (difference) and \(B\) (borrow) which express \(X - Y\) in binary.

<table>
<thead>
<tr>
<th>(X)</th>
<th>(Y)</th>
<th>(B)</th>
<th>(D)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

(b) (8 pts) Consider a 4-bit carry lookahead adder. In this adder, write the logic equations for the following signals in terms of other generate, propagate and carry signals. Assume the overall input signals to the adder are \(A(3:0)\), \(B(3:0)\) and \(C_0\).

i. \(G_3 = A_3 \cdot B_3\)

ii. \(P_3 = A_3 \oplus B_3\)

iii. \(C_3 = G_2 + P_2G_1 + P_2P_1G_0 + P_2P_1P_0C_0\)

iv. \(S_3 = A_3 \oplus B_3 \oplus C_3\)
module myckt(D, Z, V);
  input [3:0] D;
  output [1:0] Z;
  output V;
  wire D0b, D1b, D2b;
  not g0(D0b, D[0]), g1(D1b, D[1]), g2(D2b, D[2]);
  or g3(V, D[0], D[1], D[2], D[3]);
  assign Z[0] = (D0b & D[1]) | (D0b & D1b & D2b & D[3]);
endmodule

The equation for $Z_1$ can be derived from the following truth table.

<table>
<thead>
<tr>
<th>$D_0$</th>
<th>$D_1$</th>
<th>$D_2$</th>
<th>$D_3$</th>
<th>$Z_1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x$</td>
<td>$x$</td>
<td>$x$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$x$</td>
<td>$x$</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>