Exam #2: Monday, April 11th, 2011

- Closed book.
- **No calculators permitted.**
- Show all of your work. Use written English, where applicable. Always write neatly.
- A solution requiring physical units is incorrect if the units are omitted from the result.
- Indicate solution in provided space. If none is provided, underline, circle or box the result.

HONOR CODE PLEDGE:
"On my honor I have neither given nor received aid on this exam"

Your first and last name (printed): SOLUTIONS

Your Signature:

Failure to sign the pledge may result in no credit for the exam.

Score: ____________ [78]

Score: ____________ [100]
1. (15 pts) Numbers

Show your work for all parts.

(a) Convert the following 8-bit **Sign and Magnitude** binary numbers to decimal.

<table>
<thead>
<tr>
<th>Binary Number</th>
<th>Decimal Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110 1010</td>
<td>106</td>
</tr>
<tr>
<td>1110 0100</td>
<td>-100</td>
</tr>
</tbody>
</table>

(b) Convert the following 8-bit **One's Complement** binary numbers to decimal.

<table>
<thead>
<tr>
<th>Binary Number</th>
<th>Decimal Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110 1010</td>
<td>106</td>
</tr>
<tr>
<td>1110 0100</td>
<td>-27</td>
</tr>
</tbody>
</table>

(c) Convert the following 8-bit **Two's Complement** binary numbers to decimal.

<table>
<thead>
<tr>
<th>Binary Number</th>
<th>Decimal Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>0110 1010</td>
<td>106</td>
</tr>
<tr>
<td>1110 0100</td>
<td>-28</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
0110\ 1010 & = 1 \times 2^1 + 1 \times 2^3 + 1 \times 2^5 + 1 \times 2^6 = 106_{10} \\
1110\ 0100 \text{ with Sign & Magnitude} & = -(4 + 32 + 64) = -100 \\
1110\ 0100 \text{ with 1's Comp} & = -(0001\ 1011) = -(1 + 2 + 8 + 16) = -27.
\end{align*}
\]

(additional work space on next page)
1110 0100 in 2's Comp = - (0001 1011

\[ \frac{+1}{00011100} \]

= - (4 + 8 + 16) = -28

(d) Convert the following unsigned binary number to octal and hexadecimal.

<table>
<thead>
<tr>
<th>Unsigned Binary</th>
<th>Octal</th>
<th>Hexadecimal</th>
</tr>
</thead>
<tbody>
<tr>
<td>10011010111001</td>
<td>23271</td>
<td>26B9</td>
</tr>
</tbody>
</table>

octal:

\[ \underline{010} \underline{011} \underline{010} \underline{111} \underline{001} \]

\[ 2 \quad 3 \quad 2 \quad 7 \quad 1 \]

hexadecimal:

\[ \underline{001} \underline{0} \underline{0} \underline{1} \underline{1} \underline{0} \underline{1} \underline{1} \underline{1} \underline{0} \underline{0} \underline{1} \]

\[ 2 \quad 6 \quad B \quad 9 \]
(e) Using repeated division and repeated multiplication, convert the following decimal number to unsigned binary.

<table>
<thead>
<tr>
<th>Decimal Number</th>
<th>Unsigned Binary Number</th>
</tr>
</thead>
<tbody>
<tr>
<td>46.25</td>
<td>101110.001</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
46/2 &= 23, \ R = 0 \quad a_5 = 0 \\
23/2 &= 11, \ R = 1 \quad a_4 = 1 \\
11/2 &= 5, \ R = 1 \quad a_3 = 1 \\
5/2 &= 2, \ R = 1 \quad a_2 = 1 \\
2/2 &= 1, \ R = 0 \quad a_1 = 0 \\
1/2 &= 0, \ R = 1 \quad a_0 = 1 \\
0.25 \times 2 &= 0.50 \quad a_{-1} = 0 \\
0.50 \times 2 &= 1.0 \quad a_{-2} = 1
\end{align*}
\]
2. [9 pts] Binary Arithmetic

All arithmetic should be done in **binary**.

**Show your work.**

Using **Two's Complement Arithmetic**, implement addition (A+B) and subtraction (A-B) of the two 8-bit Two's Complement binary numbers given below.

For each operation indicate if **overflow** occurred.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>A + B</th>
<th>A - B</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101 1101</td>
<td>1000 0110</td>
<td>1110 0011</td>
<td>1101 0111</td>
</tr>
</tbody>
</table>

Did overflow occur for the addition? (circle one)

Y \(\bigcirc\) N

Did overflow occur for the subtraction? (circle one)

Y \(\bigcirc\) N

\[
\begin{align*}
A - B &= A + (-B) \\
B &= 1000 0110 \\
- B &= 0111 1001 \\
&+ 1 \\
\hline
0111 1010
\end{align*}
\]

\[
\begin{align*}
0101 1101 \\
+ 0111 1010 \\
\hline
1101 0111
\end{align*}
\]
3. [10 pts] Multiplexer and Decoder Circuits

(a) Determine the logic function realized by the given multiplexer circuit.

Describe the function using a Boolean expression, Minterm expansion, or Maxterm expansion.

\[
F = \sum m(0, 1, 2, 3, 4, 5, 6, 7) = \prod m(2, 5, 6, 7, 12, 14, 15)
\]
(b) Determine the logic function realized by the given decoder circuit.

Describe the function using a Boolean expression, Minterm expansion, or Maxterm expansion.

\[ F = \chi m(5, 6, 7) = \overline{A}BC + ABC + A\overline{BC} = AB + AC = A(B + C) \]

\[ = (M_1 \cdot M_3 \cdot M_6) \]
\[ = (\overline{M_1} \cdot \overline{M_3} \cdot \overline{M_6}) \]

Remember, a \textbf{NAND-NAND} circuit is equivalent to an \textbf{AND-OR} circuit. Both are used to realize a \textbf{SOP} expression.

\[ F(A, B, C) = \chi m(1, 3, 6) = \overline{A}BC + ABC + A\overline{BC} = M_1 + M_3 + M_6 \]

using De Morgan's Theorem:

\[ = (M_1 + M_3 + M_6)' \]
\[ = (M_1' \cdot M_3' \cdot M_6') \]

\[ \text{NAND-NAND} \]
4. [15 pts] Multiple-level NAND-only Logic Circuit

Given the following logic function:
\[ F(A,B,C,D) = \Sigma m(2, 3, 4, 7, 8, 12) \]

Assume all variables and their complements are available in the design.

Design a 3-level logic circuit using 2-input and 3-input NAND gates only.

Follow the steps given in parts (a) – (c) below:

(a) Using a K-map derive a minimum SOP or minimum POS expression.

Hint: remember, only one of these expressions will properly lead to a NAND-only circuit.

\[ F(A,B,C,D) = \overline{A} \overline{B} C + \overline{A} C D + B \overline{C} D + A \overline{C} D \]
(b) Using Boolean Algebra, manipulate the Boolean expression derived in part (a). The resulting expression should represent a 3-level logic circuit that uses only 2-input and 3-input logic gates.

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

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

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

\[ \text{2-input OR} \quad \text{3-input AND} \quad \text{2-input OR} \quad \text{3-input AND} \quad 3 \text{ levels} \]

(c) Draw the 3-level NAND-only logic circuit that realizes the Boolean expression in part (b).
5. [6 pts] Programmable Logic Array (PLA)

Implement the following logic functions in the PLA given below.

Clearly indicate all interconnections, in both the AND array and the OR array, with an “X”.

\[
F_1(A,B,C) = \Sigma m(2, 5, 7) = ABC + \overline{A}BC + \overline{A}BC = \overline{A}BC + \overline{ABC} + AC
\]

\[
F_2(A,B,C) = \Pi M(1, 2, 3, 5, 6, 7) = \overline{A}B\overline{C} + A\overline{B}C + BC
\]

\[
F_3(A,B,C) = AB + A'B'C = ABC + ABC + \overline{A}BC + \overline{ABC} = \sum \Pi (1,3,6,7)
\]
6. [10 pts] Hazards

Given the following logic function:

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

(a) Derive the K-map for this logic function.

(b) Which type of static hazard will the corresponding logic circuit experience?

Circle one: 0-Hazard \[ \bigcirc \] 1-Hazard

Static 1-Hazards occur in AND-OR circuits (corresponding to SOP expression)

Static 0-Hazards occur in OR-AND circuits (corresponding to POS expression)
(c) Identify all of the static hazards in the corresponding logic circuit.

- Indicate the pair of cells in the K-map between which each static hazard exists.

For example, if a static hazard exists between cells 0 and 1 in the K-map, indicate:

\[ ABCD = 0000 \leftrightarrow ABCD = 0001 \]

\[
\begin{align*}
(12) & \quad 1100 \leftrightarrow 1101 \\
(14) & \quad 1110 \leftrightarrow 1111 \\
(3) & \quad 0011 \leftrightarrow 0111 \\
(2) & \quad 0010 \leftrightarrow 1010
\end{align*}
\]

\[ D : 0 \leftrightarrow 1 \]

\[ D : 0 \leftrightarrow 1 \]

\[ B : 0 \leftrightarrow 1 \]

\[ A : 0 \leftrightarrow 1 \]

(d) Design a hazard-free logic circuit

- Specify the Boolean expression that represents the hazard-free circuit.
- Do NOT draw the circuit diagram.

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

\[ \overline{B}C \overline{D} \]

eliminates three static 1-hazards.
7. [13 pts] Arithmetic Circuits

(a) Design a digital circuit to realize the following function:

$$F(X) = 2 \times X + 1$$

where \(X\) is a 4-bit unsigned binary number (i.e. \(X = X_3X_2X_1X_0\))

Your design may include any logic or arithmetic components that have been discussed in class.

Draw the circuit diagram.

Diagram:

$$F(x) = 2 \times X + 1$$

$$F(x) = 2 \times (x_3x_2x_1x_0) + 1$$

$$F(x) = (x_3x_2x_1x_0) + (x_3x_2x_1x_0) + 1$$

*note: with \(X = x_3x_2x_1x_0\) = 4-bit # (unsigned)

the maximum value = 1111 = 15

\(2 \times 15 + 1 = 30 + 1 = 31\)

Thus, 5 bits are required for the result \(F(x)\).
(b) Given the following 4-bit Adder/Subtractor

![Diagram of 4-bit Adder/Subtractor]

Design the logic circuit for the **Overflow** signal, using any combination of the following logic components:

- 2-input and 3-input AND and OR gates
- Inverters
- Multiplexers
- Decoders

**Hint:**

**Overflow = 1** when

- positive + positive results in negative
- negative + negative results in positive
- positive - negative results in negative
- negative - positive results in positive

1. Derive the Boolean expression for **Overflow** for **Addition**
2. Derive the Boolean expression for **Overflow** for **Subtraction**
3. Determine the required logic and control signals
4. Draw the logic circuit.

(additional work space on next page)
For addition:

$\text{Overflow} = A_3 \cdot B_3 \cdot \overline{S}_3 + \overline{A}_3 \cdot \overline{B}_3 \cdot S_3$

$\text{neg. + neg = pos.}$  $\text{pos + pos = neg.}$

For subtraction:

$\text{Overflow} = \overline{A}_3 \cdot B_3 \cdot S_3 + A_3 \cdot \overline{B}_3 \cdot \overline{S}_3$

$\text{pos - neg = neg}$  $\text{neg - pos = pos}$

Use a 2-to-1 multiplexer and the Add-Sub control signal to combine the two expressions above.