1. (8 pts) Consider the following Boolean function, which is given as a minterm list.

\[ F(x, y, z) = m_1 + m_2 + m_7 \]

a) Give the canonical sum-of-products Boolean expression for function F.

\[ F(x, y, z) = x'y'z + x'yz' + xyz \]

b) Give the truth table for function F.

<table>
<thead>
<tr>
<th>x</th>
<th>y</th>
<th>z</th>
<th>F</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</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>0</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>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

c) Draw a two-level circuit diagram of function F using only gates from the set \{NOT, AND, OR\}.

A correct response would be a circuit diagram which has three NOT gates to generate the complement of each of the three variables, three AND gates to generate the three minterms, and one OR gate to bring together the outputs of the AND gates.
2. (16 pts) Consider the following component-level diagram of a combinational circuit which compares two unsigned two-bit values.

```
+------+
A-->|      |-->NE
B-->|      |
C-->|      |
D-->|      |
+------+
```

When the two-bit value AB is not equal to the two-bit value CD, the "NE" output signal is asserted (the logical value one). Otherwise, the "NE" output signal is deasserted (the logical value zero). For example, the two-bit value 10 is not equal to the two-bit value 01.

a) Describe the functionality of this circuit using a truth table.

<table>
<thead>
<tr>
<th>A B C D</th>
<th>NE</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0 0</td>
<td>0</td>
</tr>
<tr>
<td>0 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 0</td>
<td>1</td>
</tr>
<tr>
<td>0 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 0 1</td>
<td>0</td>
</tr>
<tr>
<td>0 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>0 1 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 0 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 0 1 0</td>
<td>0</td>
</tr>
<tr>
<td>1 0 1 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1 0 1</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 0</td>
<td>1</td>
</tr>
<tr>
<td>1 1 1 1</td>
<td>0</td>
</tr>
</tbody>
</table>

b) Give the Karnaugh map for this circuit.

```
NE    | C’D’ | C’D | CD  | CD’ |
-------|------|-----|-----|-----|
A’B’   | 0    | 1   | 1   | 1   |
A’B    | 1    | 0   | 1   | 1   |
AB     | 1    | 1   | 0   | 1   |
AB’    | 1    | 1   | 1   | 0   |
```

Note: the Karnaugh map will include circles to show the four 4-groups in the optimal expression.

c) Give the minimized sum-of-products Boolean expression for this circuit.

\[ NE = AC' + A'C + BD' + B'D \]
3. (16 pts) Consider a circuit which functions as a specialized counter: it begins at two and counts up to six (by ones), then starts over. That is, the circuit counts through the sequence <2, 3, 4, 5, 6, 2, 3, 4, 5, 6, ....>.

The circuit uses D flip-flops to store the current value, and combinational logic to increment the current value. Assume that the circuit is initialized asynchronously (the initialization is independent of the combinational logic and can be ignored for this problem).

a) Give the truth table for the combinational component of this counter.

<table>
<thead>
<tr>
<th>Curr a b c</th>
<th>Next A B C</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0 0</td>
<td>x x x</td>
</tr>
<tr>
<td>0 0 1</td>
<td>x x x</td>
</tr>
<tr>
<td>0 1 0</td>
<td>0 1 1</td>
</tr>
<tr>
<td>0 1 1</td>
<td>1 0 0</td>
</tr>
<tr>
<td>1 0 0</td>
<td>1 0 1</td>
</tr>
<tr>
<td>1 0 1</td>
<td>1 1 0</td>
</tr>
<tr>
<td>1 1 0</td>
<td>0 1 0</td>
</tr>
<tr>
<td>1 1 1</td>
<td>x x x</td>
</tr>
</tbody>
</table>

b) Give the Karnaugh maps for the combinational component of this counter.

\[
\begin{array}{c|ccc|ccc}
A() & b'c' & b'c & bc & bc' \\
\hline
a' & x & x & 1 & 0 \\
\hline
a  & 1 & 1 & x & 0 \\
\hline
\end{array}
\]

\[
\begin{array}{c|ccc|ccc}
B() & b'c' & b'c & bc & bc' \\
\hline
a' & x & x & 0 & 1 \\
\hline
a  & 0 & 1 & x & 1 \\
\hline
\end{array}
\]

\[
\begin{array}{c|ccc|ccc}
C() & b'c' & b'c & bc & bc' \\
\hline
a' & x & x & 0 & 1 \\
\hline
a  & 1 & 0 & x & 0 \\
\hline
\end{array}
\]

c) Give the minimized sum-of-products Boolean expressions for the combinational component of this counter.

\[
A() = b' + c
\]

\[
B() = bc' + b'c \quad \text{(alternative: } bc' + ac\text{)}
\]

\[
C() = a'c' + b'c'
\]
4. (22 pts) Consider the following C program. In the space provided, give the hexadecimal value which will be displayed by each call to "printf".

#include <stdio.h>

int main()
{
    int aaa = 172;
    int bbb = -172;
    int ccc = 0x7c00009d;
    int ddd = 0xb60000a3;

    printf( "%08X\n", aaa );          // 000000AC
    printf( "%08X\n", bbb );          // FFFFFFF54
    printf( "%08X\n", "ccc ");        // 83FFFF62
    printf( "%08X\n", ccc | ddd );     // FE0000BF
    printf( "%08X\n", ccc ^ ddd );     // CA00003E
    printf( "%08X\n", ccc & ddd );     // 34000081
    printf( "%08X\n", ccc >> 8 );      // 007C0000
    printf( "%08X\n", ccc << 12 );     // 0009D000
    printf( "%08X\n", ddd >> 16 );     // FFFFB600
    printf( "%08X\n", ddd << 20 );     // 0A300000
}
5. (12 pts) Suppose that a machine uses eight bits to represent signed integers, uses twos complement arithmetic, and has the integer condition codes discussed at lecture. For each of the problems below, give the result (in binary) of the addition as it would be performed by this machine, and give the value of the indicated condition code bits after the operation is complete.

a) Sum of the eight-bit operands (given in binary).

\[
\begin{array}{c|c}
10110011 & \text{Sum: 01010000} \\
ADD 10011101 & N: 0 \quad Z: 0 \quad V: 1 \quad C: 1 \\
\end{array}
\]

b) Sum of the eight-bit operands (given in binary).

\[
\begin{array}{c|c}
00001000 & \text{Sum: 00000000} \\
ADD 11111000 & N: 0 \quad Z: 1 \quad V: 0 \quad C: 1 \\
\end{array}
\]

c) Sum of the eight-bit operands (given in binary).

\[
\begin{array}{c|c}
10100111 & \text{Sum: 10011101} \\
ADD 11110110 & N: 1 \quad Z: 0 \quad V: 0 \quad C: 1 \\
\end{array}
\]

6. (4 pts) Consider the following list of terms associated with IEEE floating point representation.

- Infinity
- NaN
- Zero
- Denormal
- Normalized

For each double-precision floating point number shown below (in hexadecimal), state the term from the list above which best describes that number.

\[
\begin{array}{c}
001f000000000000 \quad \text{Normalized} \\
ffff000000000000 \quad \text{NaN} \\
80000000ffffffffff \quad \text{Denormal} \\
7ff000000000000 \quad \text{Infinity}
\end{array}
\]
7. (12 pts) The following questions refer to IEEE floating point numbers. Show your work for possible partial credit.

a) Give the 64-bit double precision internal representation (in hexadecimal) of the decimal value given below.

-9.625  C023400000000000

b) Give the decimal value of the 32-bit single precision floating point number whose internal representation is given below (in hexadecimal).

3f400000  +0.75

c) Give the 32-bit single precision internal representation (in hexadecimal) of the decimal value given below.

+13.375  41560000