Name and Net ID:

- Be sure that your exam booklet has 10 pages.
- Write your name at the top of each page.
- This is a closed book exam.
- You are allowed TWO 8.5 \times 11\ " handwritten sheets of notes.
- We have provided a scratch sheet at the back.
- Absolutely no interaction between students is allowed.
- Show all of your work.
- Challenge questions are marked with ***.
- Don’t panic, and good luck!

“... for those obstinate questionings
of sense and outward things,”
from Recollections of Early Childhood*, 143-144

Problem 1  30 points

Problem 2  30 points

Problem 3  10 points

Problem 4  20 points

Problem 5  10 points

Total  100 points
Problem 1 (30 points): Boolean Logic and Gates

Part A (5 points): How many different minterms exist on five variables (for example, A, B, C, D, and E)?

Part B (5 points): Complete the following transistor-level circuit by drawing the bottom half of the circuit (from OUT to ground).

Part C (5 points): Write the function implemented by the circuit in Part B. In particular, express OUT in terms of A, B, C, and D. (Note: you need not write the function in SOP nor in POS form.)
**Part D** (5 points): Prove that the following two Boolean expressions are equivalent.

\[(\overline{A} \oplus B) + \overline{C} \overline{D} + A\overline{C} + A\overline{D} + BCD\]

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

*Hint: Do NOT manipulate algebraic expressions.* If you do, you must explain each step clearly to receive credit.

**Part E** (10 points): For each possible combination of the inputs $A$ and $B$ to the circuit shown below, write all stable states for outputs $Q$ and $P$ into the table below. If exactly two combinations of $Q$ and $P$ are stable for a given input combination, circle “bistable” for that input combination.

<table>
<thead>
<tr>
<th>$A$</th>
<th>$B$</th>
<th>$Q$</th>
<th>$P$</th>
<th>(circle only if appropriate)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td></td>
<td></td>
<td>bistable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
<td></td>
<td>bistable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
<td></td>
<td>bistable</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
<td></td>
<td>bistable</td>
</tr>
</tbody>
</table>
Problem 2 (30 points): Using K-maps

Parts A through C of this problem pertain to the same function.

Part A (5 points): Below are two copies of the same K-map for the function $F(A, B, C, D)$. Find a sum-of-products (SOP) expression with a minimum number of prime implicant terms for $F(A, B, C, D)$, and show the corresponding prime implicants on one of the K-maps. CROSS OUT the K-map that you DO NOT want graded.

$$F = \begin{array}{c|c|c|c|c} \hline CD & 00 & 01 & 11 & 10 \\ \hline 00 & 1 & 0 & x & 1 \\ 01 & 0 & 1 & 1 & x \\ 11 & 1 & 1 & 1 & 1 \\ 10 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Part B (5 points): Below are two MORE copies of the K-map for the SAME function $F(A, B, C, D)$ from Part A. Find a product-of-sums (POS) expression with a minimum number of prime implicant terms (factors) for $F(A, B, C, D)$, and show the corresponding implicants on one of the K-maps. CROSS OUT the K-map that you DO NOT want graded.

$$F = \begin{array}{c|c|c|c|c} \hline CD & 00 & 01 & 11 & 10 \\ \hline 00 & 1 & 0 & x & 1 \\ 01 & 0 & 1 & 1 & x \\ 11 & 1 & 1 & 1 & 1 \\ 10 & 0 & 0 & 0 & 1 \\ \hline \end{array}$$

Part C (5 points): Your answers to Part A and Part B should not represent the same Boolean logic function. Use specific aspects of this problem to illustrate and explain this difference.
Problem 2, continued:

Parts D through F pertain to the following problem.

You are asked to design a circuit that finds the 9’s complement of a decimal number. As you might guess, the 9’s complement is calculated by replacing each digit with the difference given by (9 - the digit).

Your circuit will accept as input a decimal digit represented as an unsigned number in 4 bits, \( B_3B_2B_1B_0 \). As output, you must produce a new decimal digit, \( F_3F_2F_1F_0 \), also represented as an unsigned value.

Finally, your circuit must produce the digit 0 for any input value outside of the range possible for a decimal digit.

**Part D** (5 points): Fill in ONE OF the K-maps below for the function \( F_2 \). CROSS OUT the K-map that you DO NOT want graded.

\[
\begin{array}{cccc}
00 & 01 & 11 & 10 \\
\hline
00 & \Box & \Box & \Box \\
01 & \Box & \Box & \Box \\
11 & \Box & \Box & \Box \\
10 & \Box & \Box & \Box \\
\end{array}
\]

**Part E** (5 points): Write a Boolean expression for the function \( F_2 \) using a minimum number of terms.

\[
F_2 =
\]

**Part F** (5 points): Implement the function \( F_2 \) using either a two-level NAND or a two-level NOR approach (remember that inverters on the variables do not count as a level). For clarity, you may use multiple labels for each variable—for example, if two gates operate on a variable \( B_0 \), you may simply label an input to each gate as “\( B_0 \)” rather than trying to draw lines connecting the two inputs.
Problem 3 (10 points): C Program Analysis

Consider the following “mystery” C program.

```c
#include <stdio.h>
#define WORD_LENGTH 32
int main()
{
    unsigned int a, b;
    int value;
    unsigned int mask;
    int ii;
    unsigned int temp;

    scanf("%d", &a);
    scanf("%d", &b);
    value = 0;
    mask = 1;
    for ( ii = 0; ii < WORD_LENGTH; ii = ii + 1 ) {
        temp = (mask & (a ^ b)); /* ^ means exclusive-or in C */
        if ( temp != 0 ) {
            value = value + 1;
        }
        mask = mask << 1;
    }
    printf("The output value is %d\n", value);
    return 0;
}
```

Part A (4 points): Explain in one sentence the function of the program; that is, what does the value of “value” represent?

Part B (3 points): Assuming that a user enters 15 for `a` and 2 for `b`, write down EXACTLY the formatted text that will be printed on the terminal screen by the final `printf` statement in the program.

Part C (3 points): Assuming that a user enters 256 for `a` and 30 for `b`, write down EXACTLY the formatted text that will be printed on the terminal screen by the final `printf` statement in the program.
Problem 4 (20 points): Thermal Control

The table and diagram to the right show an $N$-bit 2’s complement comparator that you should use for this problem. The design is identical to that developed in Notes Set 2.4, except that the initialization inputs for the first bit slice are not shown (assume that they are set correctly).

<table>
<thead>
<tr>
<th>$Z_1$</th>
<th>$Z_0$</th>
<th>meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>$A = B$</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>$A &lt; B$</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>$A &gt; B$</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>not used</td>
</tr>
</tbody>
</table>

You are charged with the design of a temperature control unit for the new ECE building. Your system will be given three temperatures as 8-bit 2’s complement numbers and must produce 1-bit signals to indicate whether the heater should be turned on and whether the air conditioning should be turned on.

The temperatures are the current temperature $T$, the low temperature threshold $A$, and the high temperature threshold $B$. The heater $H$ should be turned on ($H = 1$) when $T$ falls below $A$. The air conditioning $C$ should be turned on ($C = 1$) when the temperature rises above $B$.

However, since humans are in charge of picking the threshold values, your system must protect against human error. In particular, if the rules just given say that you should turn on BOTH the heater AND the air conditioning, your system should instead turn on neither.

Part A (5 points): Draw a circuit that performs the comparisons needed for your thermal management system. The circuit should produce four bits of output. Call them $P$, $Q$, $R$, and $S$.

Part B (5 points): Based on your definitions for Part A, fill in ONE OF the K-maps below for the function $H$. CROSS OUT the K-map that you DO NOT want graded.

Part C (5 points): Write a Boolean expression for the function $H(P, Q, R, S)$ using a minimum number of terms.

$H =$
Problem 4, continued:

Copy your function from Part C here to make Part D easier.

\[ H = \]

Part D (5 points): Implement the function \( H(P, Q, R, S) \) using gates. For clarity, you may use multiple labels for each variable—for example, if two gates operate on a variable \( P \), you may simply label an input to each gate as \( "P" \) rather than trying to draw lines connecting the two inputs.
Problem 5 (10 points): Using Abstraction

Part A (5 points): You are designing a system that sends bits serially over a network link. The network that you are using requires that the number of consecutive 0s sent never exceeds three. In other words, if your system sends more than three 0 bits in a row, the network may fail. (In fact, 100 Mbps Ethernet uses a similar system.)

Let's say that you want to design a bit-sliced checker for words of data intended for transmission over the network.

Your checker must produce a 1 if an $N$-bit word contains any consecutive sequence of 0s of length more than three, or a 0 if no such sequence exists in the word.

How many bits must each bit slice pass to the next slice? Justify your answer.

Part B*** (5 points): A 2’s complement multiplier can be built from an unsigned multiplier by adding three extra blocks of logic. Two of these blocks are used to calculate the absolute value of the input operands, and the third is used to negate the product if exactly one of the input operands is negative.

Imagine that you adopt a similar approach to design a circuit that uses an $N$-bit unsigned multiplier to calculate $(-A \times B)$ for two $N$-bit 2’s complement numbers $A$ and $B$.

The unsigned multiplier produces $2N$ bits of output to ensure that any possible result can be expressed (overflow is impossible). How many bits are needed for your new design? (All $2N$ of them? More? Fewer?) Express your answer in terms of $N$, and justify your answer.