Problem 1 (10 pts): Boolean algebra

1. Which of the following Boolean expressions are equivalent to \((a + \bar{b})(bc + \bar{a}c)\)? Refer to the Boolean algebra properties on the last page of the exam booklet if needed.

\[
\begin{align*}
(a + \bar{b})(\bar{b} + \bar{c})(\bar{a} + \bar{c}) \\
\bar{a}b + bc + \bar{a}c \\
\bar{a}b \overline{(b + \bar{c}) + \bar{a}c} \\
\bar{a}b(b + c)(\bar{a} + \bar{c})
\end{align*}
\]

2. Use proof by **perfect induction** to demonstrate that \(M_1 = \overline{m_1}\) for \(f(x, y)\).

3. The function \(f(w, x, y, z) = M_2 + M_{10}\), where \(M_2\) and \(M_{10}\) are maxterms. Which expression below is equivalent to \(f(w, x, y, z)\)?
Problem 2 (12 pts): Canonical forms

A 3-variable function \( g(x, y, z) \) is defined by the following truth table:

\[
\begin{array}{ccc|c}
 x & y & z & g \\
 0 & 0 & 0 & 0 \\
 0 & 0 & 1 & 0 \\
 0 & 1 & 0 & 1 \\
 0 & 1 & 1 & 1 \\
 1 & 0 & 0 & 0 \\
 1 & 0 & 1 & 0 \\
 1 & 1 & 0 & 0 \\
 1 & 1 & 1 & 1 \\
\end{array}
\]

1. Using the table, write the **canonical SOP** representation for \( g \) using **literals**.

2. Using the table, write the **canonical SOP** representation for \( g \) using **minterm** notation \( m_i \).

3. Using the table, write the **canonical POS** representation for \( g \) using **literals**.

4. Using the table, write the **canonical POS** representation for \( g \) using **maxterm** notation \( \bar{M}_i \).

5. For function \( f(a, b, c, d) = a'b'd' + a'cd' \), write the corresponding **canonical SOP**.

6. For function \( g(a, b, c, d) = (c' + d) (a + b' + c' + d) \), write the corresponding **canonical POS**.
Problem 3 (14 pts): Function simplification

Consider a 4-variable Boolean function \( f(w, x, y, z) \) given by its K-map (drawn twice):

\[
\begin{array}{cccc|cccc}
wx & yz & \text{map 1} & yz & \text{map 2} \\
00 & 00 & 1 & 1 & 1 & 1 \\
01 & 1 & X & 0 & 1 & 1 \\
11 & 0 & 0 & X & 1 & 1 \\
10 & 1 & 1 & 0 & X & 1 \\
\end{array}
\]

1. List all non-essential prime implicants.

2. Give a minimal SOP expression for \( f(w, x, y, z) \) and show the corresponding loops on the left map.

3. Is your solution unique? If no, give another minimal solution.

4. Give a minimal POS expression for \( f(w, x, y, z) \) and show the corresponding loops on the right map.

5. Do your answers to Part 2 and 4 represent the same Boolean function? Justify your answer.
Problem 4 (10 pts): 2-level circuits

1. Implement the Boolean function \( g(a, b, c, d) = (a + c') + d \) as a two-level network using NOR gates only. Assume that inverted inputs are available. Draw the circuit.

2. Shown below is an implementation of a 9-input OR function. Re-implement it using NOR and NAND gates only. (Hint: Use alternative representation for NAND gate.)

3. Implement a 4:1 MUX using a 2:4 decoder and a 2-level AND-to-OR circuit.
Problem 5 (11 pts): Combinational logic design

Design network $N$ such that the $n$-bit arithmetic unit shown below can perform the following arithmetic and logic operations for inputs $A=a_{n-1} \ldots a_0$ and $B=b_{n-1} \ldots b_0$:

$$F = \begin{cases} 
\text{A PLUS B} & \text{when } k_1 k_0 = 00 \\
\text{B MINUS A} & \text{when } k_1 k_0 = 01 \\
\text{A XOR B} & \text{when } k_1 k_0 = 10 \\
\text{A OR B} & \text{when } k_1 k_0 = 11 
\end{cases}$$

1. Draws K-maps for outputs $p_i q_i$ from circuit $N$.

2. Write minimal SOP expressions for $p_i$ and $q_i$.

3. Write a simple Boolean expression for $c_0$ as a function of $k_1$ and $k_0$. 

$$c_0 = \ldots$$
Problem 6 (11 pts): Combinational logic design

Design a bit-slice circuit that checks whether an unsigned integer $A = a_{n-1}a_{n-2} \ldots a_1a_0$ is a power of 2, starting with the least significant bit.

The U cell should be designed so that the n-bit network shown above correctly checks if the unsigned integer number $A$ is a power of two. The final output $f$ should be 1 iff $A$ is a power of two. (*Hint: If a number is a power of two, how many 1s are in its binary representation?*)

1. List the possible 'answers' (or information) that your bit slice may need to communicate to the next bit slice (and receive from the previous bit slice). One such answer is already provided for you to get started.

<table>
<thead>
<tr>
<th>$p_i$</th>
<th>$q_i$</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>Have not seen any 1s so far</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

2. What are initial values for $p_0$ and $q_0$?

3. Draw K-maps for $p_{i+1}$ and $q_{i+1}$.

4. Give minimal POS expressions for $p_{i+1}$ and $q_{i+1}$.

5. Write a Boolean expression for the decision circuit $F$ that maps from the outputs of the most-significant-bit bit-slice $p_nq_n$ to the answer $f=1$ if $A$ is a power of two, or $f=0$ otherwise.

$f(p_n, q_n) = $
Problem 7 (16 pts): Sequential logic components

Part A. (10 pts) Shown below is the logic diagram of a gated T latch. It consists of 2 NOR gates and two AND gates and has 2 inputs: T and Clk.

1. Complete the next-state table for this circuit.

<table>
<thead>
<tr>
<th>Clk</th>
<th>T</th>
<th>Q+</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

2. Express next state Q+ as a function of Clk, T, and Q.

Answer:

3. Describe the behavior of the T latch using common everyday English.

Answer:

Part B. (6 pts) Complete the design of a 3-bit register that performs the operations listed in the table to the right. Parallel load inputs are labeled and indexed as P[]. Serial input is labeled as S[]. You may use inputs without drawing the wires, just write the appropriate labels at the MUX inputs.

<table>
<thead>
<tr>
<th>S[1] S[0]</th>
<th>Operation</th>
</tr>
</thead>
<tbody>
<tr>
<td>0 0</td>
<td>Parallel load</td>
</tr>
<tr>
<td>0 1</td>
<td>Circular shift left</td>
</tr>
<tr>
<td>1 0</td>
<td>Hold current value</td>
</tr>
<tr>
<td>1 1</td>
<td>Clear</td>
</tr>
</tbody>
</table>

---

**Solution:**

Clk | T   | Q+  |
-----|-----|-----|
0 | 0 | 0 |
0 | 1 | 1 |
1 | 0 | 1 |
1 | 1 | 0 |

Expression: Q+ = Clk * T + Clk * Q

Answer: To be written.

Behavior: To be written.

**Solution Diagram:**

```
<table>
<thead>
<tr>
<th>S[1]</th>
<th>S[0]</th>
<th>S[0]</th>
<th>S[0]</th>
</tr>
</thead>
<tbody>
<tr>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
</tr>
<tr>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
</tr>
<tr>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
<td>S[0]</td>
</tr>
</tbody>
</table>
```

---

8
Problem 8 (16 pts): Boolean algebra in C

1. Implement a program in C that prints canonical SOP representation for function
   \[ g(a, b, c) = a' (b \oplus c) + ac \] using minterm notation. The program is partially implemented,
you only need to complete some parts of it. Use bit-wise operators only.

```c
#include <stdio.h>

int main()
{
    unsigned int a, b, c;
    int g;
    int i;
    int notfirst = 0;

    printf("g(a, b, c) = ");
    for (a = 0; a <= 1;
    {            __; b = b + 1)
    {          
    for (   ; c <= 1; c = c + 1)
     {            ___; /* function g */
     g =
     if (g ==
     {
         if (notfirst) printf("+");
         i =
         printf("m%d", i);
         notfirst = 1;
     }        
    }
    }

    printf("\n");
    return 0;
}
```

2. Write down EXACTLY the formatted text that will be printed on the terminal screen by the program.