Problem 1. CMOS Transistor Complements (22 points). In each of the parts below, only one half of the CMOS circuit is shown. Draw the other half (i.e., the complement) using p-type or n-type transistors as appropriate to complete the circuit.

Be sure to:
- use the correct symbol for each transistor (with a little circle for p-type, without for n-type)
- label the control input of each transistor
- draw connection to ground or power supply as appropriate, and
- draw right on top of the figure, where the missing portion is indicated by “?”

a) [3 points]
b) [4 points]

\[ \text{Diagram} \]

\[ Y \]

C) [3 points]

\[ \text{Diagram} \]

\[ Y \]
d) [4 points]

![Diagram](image1)

Y

A

B

C

D

E

?

---

e) [4 points]

![Diagram](image2)

Y

A

B

C

D

E

?
f) [4 points]
Problem 2. Complex CMOS Gates from Boolean Formulas (24 points)

For each of the parts, you are to draw a single CMOS gate that implements the given function, *using transistors*. That is, draw a single gate with complementary pull-up and pull-down networks (using p-type and n-type transistors, respectively). Be sure that the circuit you draw corresponds exactly to the expressions given, i.e., do not perform any simplification.

a) \( Y = \overline{AC} + B\overline{D} \)

   Hint: Since there is a bar over the entire formula, it would be easier to draw the bottom half of the gate first, then deduce the upper half as its complement. [8 points]
b) \( Y = (AB + C)(DE + F) \) [8 points]
c) \( Y = \overline{A} \overline{B} (\overline{E} + F) + \overline{C} + \overline{D} \)

Note: The entire formula here does not have a bar over it, but individual variables each have a bar; therefore, it would be easier to draw the upper half first. [8 points]
Problem 3. Truth Tables (14 points)

a) [4 points] Complete the truth table below showing multiplication of two 2-bit unsigned integers, \( A \) and \( B \), producing a 4-bit result \( P \). The 2-bit operands are represented as \( A_1A_0 \) and \( B_1B_0 \), respectively. The 4-bit result is represented as \( P_3P_2P_1P_0 \). Please enter your answer directly in the table below.

<p>| | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1A_0 )</td>
<td>( B_1B_0 )</td>
<td>( P_3 )</td>
<td>( P_2 )</td>
</tr>
<tr>
<td>0000</td>
<td>0000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0001</td>
<td>0001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0010</td>
<td>0010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0011</td>
<td>0011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0100</td>
<td>0100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0101</td>
<td>0101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0110</td>
<td>0110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0111</td>
<td>0111</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>1000</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1001</td>
<td>1001</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1010</td>
<td>1010</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1011</td>
<td>1011</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1100</td>
<td>1100</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1101</td>
<td>1101</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1110</td>
<td>1110</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1111</td>
<td>1111</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

b) [4 points] Suppose you wanted to compute the cube of a 2-bit number (i.e., \( A^3 \)). Complete the truth table below in which the input column contains the 2-bit input (\( A=A_1A_0 \)), and the output column is the 6-bit result (\( P=P_5P_4P_3P_2P_1P_0 \)).

<p>| | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>( A_1A_0 )</td>
<td>( P_3 )</td>
<td>( P_2 )</td>
<td>( P_1 )</td>
<td>( P_0 )</td>
<td></td>
</tr>
<tr>
<td>00</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>01</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

c) [6 points] For each of the 6 output bits from part (b), give the sum-of-products Boolean expression (circuit not needed):

\[
\begin{align*}
P_0 &= \\
P_1 &= \\
P_2 &= \\
P_3 &= \\
P_4 &= \\
P_5 &= 
\end{align*}
\]
Problem 4. Circuits to Truth Table (10 points)

For each of the parts, simply draw a truth table corresponding to the given circuit. If there are more than one outputs, draw them as separate columns in the same truth table.

a) [5 points]

![Circuit Diagram](image1)

From the truth table, do you recognize what this circuit does?

b) [5 points]

![Circuit Diagram](image2)

From the truth table, do you recognize what this circuit does?
Problem 5. Equations to Circuits (10 points)
For each of the parts, simply convert the given Boolean expression directly into a circuit diagram consisting of basic gates. Do not implement using transistors! Please do not perform any simplification/optimization. Simply replace each Boolean operation by the appropriate gate. You can assume you have all of these gates available (with any number of inputs): AND, OR, inverter, NAND, NOR, XOR, XNOR. Also, you may assume that for each input, its complement is also available (e.g., both $X$ and $\overline{X}$ are available). Note: The $\oplus$ and $\overline{\oplus}$ symbols represent XOR and XNOR operations, respectively.

a) [5 points] $F = (A \oplus B) + \overline{C} + (D \oplus E)\overline{F}$

b) [5 points] $F = \overline{XYZ} + (A + B)(C + D) \overline{\oplus} E$
Problem 6. Sum of Products (20 points)

For each of the parts, do the following in this order: (i) first draw a truth table, (ii) then give the sum-of-products Boolean expression for the output, and (iii) finally, draw a circuit diagram. You may only use inverters, AND gates and OR gates for your circuit. The AND and OR gates can have two or more than two inputs, as many as you need. You do not need to simplify/optimize your Boolean expression. You may assume that for each input, its complement is also available (e.g., both $A$ and $\overline{A}$ are available).

a) [10 points] Implement a function $F$ whose inputs are $A$, $B$ and $C$, such that the value of $F$ is the same as the majority of the inputs (i.e., at least two out of three inputs have that value).
b) [10 points] Implement a function $F$ whose inputs are $A$, $B$ and $C$, such that the value of $F$ is 1 if and only if an odd number of inputs (i.e., one or three inputs) is 1.