1. (20 points)
   (a) (10 points) NAND, NOR gates
   Simplify the following Boolean function and then implement the simplified result using
   2-level NAND-NAND realization
   \[ F(x, y, z) = (x + y + z)(\overline{x} + \overline{y} + z) \]
   Assume complements of the input Boolean variables are available.
   **Answer:** \( F(x, y, z) = \overline{x}y + xy + z \)

   \[ \begin{array}{c}
   x \\
   y \\
   x \\
   y \\
   \end{array} \quad \begin{array}{c}
   \text{F(x,y,z)} \\
   \overline{z} \\
   \end{array} \]

   (b) (10 points) NOR gate implementation
   Convert the following logic schematic diagram into NOR-only realization. You may use
   only **two-input** NOR gates and inverters. Assume the complements of input Boolean
   variables are available.
   **Answer:** two possible answers, one without simplification, one after simplification.

   \[ \begin{array}{c}
   a \\
   \overline{b} \\
   c \\
   \overline{d} \\
   e \\
   \end{array} \quad \begin{array}{c}
   \text{F(a,b,c,d,e)} \\
   \end{array} \]

   \[ \begin{array}{c}
   a \\
   \overline{b} \\
   \overline{c} \\
   d \\
   e \\
   \end{array} \quad \begin{array}{c}
   \text{F(a,b,c,d,e)} \\
   \overline{b} \\
   \end{array} \]
2. (10 points) Tabular Method

Let:
\[ P(v,w,x,y,z) = \sum m(12,13,14,15,29,31) \]
\[ d(v,w,x,y,z) = \sum m(17,18). \]

Execute the Quine-McCloskey tabulation algorithm to find all the Prime Implicants of the function \( P(v,w,x,y,z) \), where the second summation is the don't care minterms. The algorithm has been started for you below (minterms are listed in increasing 1's count order). Complete the algorithm and CIRCLE the PRIME IMPlicants.

<table>
<thead>
<tr>
<th>( m_i )</th>
<th>Index Order</th>
<th>( \sqrt{\phantom{0}} )</th>
<th>1-Cubes</th>
<th>( \sqrt{\phantom{0}} )</th>
<th>2-Cubes</th>
<th>( \sqrt{\phantom{0}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>18(d)</td>
<td>10010</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>0110- (12,13)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>011-- (12,13,14,15)</td>
<td></td>
</tr>
<tr>
<td>17(d)</td>
<td>10001</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>011-0 (12,14)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>-11-1 (13,15,29,31)</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>01100</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>011-1 (13,15)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>01101</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>0111- (14,15)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>01110</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>-1101 (13,29)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>01111</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>-1111 (15,31)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
<tr>
<td>29</td>
<td>11101</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>111-1 (29,31)</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
<tr>
<td>31</td>
<td>11111</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td>( \sqrt{\phantom{0}} )</td>
<td></td>
</tr>
</tbody>
</table>

Note that there are no three-cubes and that duplicate two-cubes have been eliminated.

3. (10 points) Systematic Boolean Simplification

In the following table, the set of prime implicants (P.I.) corresponding to a 4-variable Boolean function \( F(a,b,c,d) \) are listed. Classify each P.I. into the categories of essential PI (EPI), less-than PI (LTP), or secondary essential PI (SEPI) in the covering table.

<table>
<thead>
<tr>
<th>PIs\Minterms</th>
<th>1</th>
<th>5</th>
<th>7</th>
<th>10</th>
<th>14</th>
<th>15</th>
<th>PI types</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \overline{a} \cdot \overline{c} \cdot d )</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>x</td>
<td>EPI</td>
<td></td>
<td></td>
</tr>
<tr>
<td>( a \cdot b \cdot c )</td>
<td>x</td>
<td>x</td>
<td>LTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( a \cdot \overline{c} \cdot \overline{d} )</td>
<td>x</td>
<td>x</td>
<td>EPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( \overline{a} \cdot b \cdot d )</td>
<td>x</td>
<td>x</td>
<td>LTP</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>( b \cdot c \cdot d )</td>
<td>x</td>
<td>x</td>
<td>SEPI</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
4. (15 points) Combinatorial circuit analysis
The logic diagram of a combinational logic circuit is given below:

Express the corresponding Boolean function in the product of Maxterm format:

Answer: \( f(a, b, c, d) = \overline{a} + b + c + d = \bigoplus M(15) \)

5. (15 Points) Decoder Implementations of Boolean Functions
The symbol and the function table for an active-low decoder is shown below:

<table>
<thead>
<tr>
<th>S0</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>S1</td>
<td>0</td>
<td>1</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>S1</th>
<th>S0</th>
<th>D3</th>
<th>D2</th>
<th>D1</th>
<th>D0</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Two such decoders constructed with NAND gates are connected to primary input variables A, B, C and D as shown in the diagram below.
(c) (3 points) Write a Boolean equation in the box below for the signal shown as "M" on the diagram in terms of the primary input signals $A, B, C$ and $D$.

$$M = \overline{(C \cdot D)} + \overline{D}$$

(d) (12 points) Use exactly one logic gate from the list of logic gates \{AND, NOT, OR, NAND, NOR\} to implement each the following two Boolean functions (i.e. one gate per Boolean function)

$$Q(A,B,C,D) = A + B + C + D = F + J$$
$$R(A,B,C,D) = A \cdot \overline{B} + C \cdot \overline{D} = H + L = (H \cdot L)$$

Draw the gate on the diagram above and connect the input to DECODER OUTPUTS (signals F,G,H,I,J,K,L, or M). Label the output of each of the logic gates.

6. (10 points) **Combinational circuit synthesis and four-variable K-Map**

A combinational circuit is to be designed according to the following specification: The inputs are $A_3A_2A_1A_0$, and the output is $Z$. The four-bit inputs represent a 4-bit binary number $A$. When $4 \leq A \leq 5$, or $11 \leq A \leq 14$, $Z = 0$. When, $1 \leq A \leq 2$, or $8 \leq A \leq 10$ $Z = 1$. Otherwise, the output is of no concern.

(a) (5 points) Find the corresponding K-map of $Z(A_3, A_2, A_1, A_0)$

**Answer:**

```
A_3A_2A_1A_0  00  01  11  10
00  X  1  X  1
01  0  0  X  X
11  0  0  X  0
10  1  1  0  1
```

(b) (5 points) Represent $Z(A_3, A_2, A_1, A_0)$ in sum of product standard form with minimum number of literals.

**Answer:** $Z(A_3, A_2, A_1, A_0) = \overline{A_2} \cdot \overline{A_1} + \overline{A_2} \cdot \overline{A_0}$

7. (10 points) **Multiplexer Logic Implementation**

Use the 8-to-1 multiplexer below to implement an exclusive or function for four bits. This is also known as the "odd" function. The function $\text{exor}(w,x,y,z)$ is to be:

$$\text{exor}(w,x,y,z) = w \oplus x \oplus y \oplus z$$

Draw the circuit by factoring out the variable “$w$" in the space below. You may use only NOT, OR, and AND gates.

**Answer:**
8. **(10 Points)** Carry Look-ahead Adders

Consider an adder with inputs $A = A_3 \ A_2 \ A_1 \ A_0$ and $B = B_3 \ B_2 \ B_1 \ B_0$, $C_0$ and outputs $S = S_3 \ S_2 \ S_1 \ S_0$ and $C_4$. Recall that for the stage $i$ in an adder, the propagate and generate functions are $P_i = A_i \oplus B_i$; $G_i = A_i \cdot B_i$; and the output sum $S_i$ and carry $C_{i+1}$ is defined as $S_i = P_i \oplus C_i$; and $C_{i+1} = G_i + P_i C_i$.

(a) (4 points) Write a Boolean expression in the SOP format for the carry, $C_4$ as a function of $P_i, G_i$, $i = 0, 1, 2, 3$, and $C_0$.

**Answer:**

$$C_4 = G_3 + P_3(G_2 + P_2(G_1 + P_1(G_0 + P_0 C_0)))$$

After simplification,

$$C_4 = G_3 + P_3G_2 + P_3P_2G_1 + P_3P_2P_1G_0 + P_3P_2P_1P_0C_0$$

(b) (6 points) Let $A = 1110$ and $C_0 = 1$, find Boolean expression in the SOP format for the carry, $C_4$ as a function of $B_3, B_2, B_1, and B_0 only$. Simplify the result to minimize the number of literals.

**Answer:** Note that $G_3 = B_3$, $P_3 = \overline{B_3}$, $G_2 = B_2$, $P_2 = \overline{B_2}$, $G_1 = B_1$, $P_1 = \overline{B_1}$, $G_0 = 0$, $P_0 = B_0$, $C_0 = 1$. Alternately, note that the only input which does NOT generate a carry is for $B = 0000$, thus $\overline{C_4} = (B_3 \cdot B_2 \cdot B_1 \cdot B_0)$. By DeMorgan's Rule

$$C_4 = B_3 + B_3B_2 + B_3B_2B_1 + B_3B_2B_1B_0 = B_3 + B_2 + B_1 + B_0$$