1. This problem has to do with the following minterm-form specification for a logic function:

\[ F(A, B, C, D) = \Sigma m(1, 3, 4, 6, 8, 11, 14) \]

a. (8 points) Draw the Truth Table for this function in the space below. You must list the 4 variables in the order given (A on the left and D on the right).

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

b. (8 points) Draw the Karnaugh Map (K-Map) for this function in the space below. You must list the values for the \(AB\) pair across the top and the values for the \(CD\) pair down the side.

```
   CD  00  01  11  10
  00  0  0  1  1  0  1
  01  0  1  1  1  0  1
  11  0  0  0  1  1  0
  10  0  1  1  1  0  1
```

2. (10 points) Prove the following Boolean identity using Boolean Algebra properties. Each step must use one and only one Boolean Algebra property from the provided reference sheet (but you may use the same property more than once in one step). Identify which property that you used for a given step with the number of the property. You may not use a Truth Table or K-map or Venn diagram to prove the identity.

\[ B(\overline{A} + C) = \overline{A}BC + \overline{A}BC + ABC \]

Work with LHS:

- Expression: \[ B(\overline{A} + C) \]
  - Property: LHS
  - Step 12a
- \[ \overline{A} + BC \]
  - Property: 10a
- \[ \overline{A}BC + \overline{A}BC + ABC \]
  - Property: 14a (twice)
- \[ \overline{A}BC + \overline{A}BC + ABC \]
  - Property: 7b

Result is RHS -- Done

Lots of other solutions.
3. This problem deals with the following K-Map for function $F(A,E,C)$. Each $d$ in the K-Map indicates a don’t-care output.

```
   00  01  11  10
C
0  0   1   0   0
1  d   1   1   0
```

In this problem, you will be working with a Product of Sums (PoS) synthesis for this function, as follows:

a. (3 points) Identify all of the PoS Prime Implicants (PIs) for this logic function, writing a logic expression for each one.

$$B_1 ( \overline{A} + C)$$

b. (3 points) Identify all of the PoS Essential Prime Implicants (EPIs) for this logic function, writing a logic expression for each one.

$$B_1 ( \overline{A} + C)$$

c. (3 points) Find the minimum-cost PoS synthesis for this function; write an expression for it.

$$EPI's \text{ form cover,}$$

$$F = B_1 ( \overline{A} + C)$$

d. (3 points) Find the cost (as we have defined it in this course) for your minimum-cost PoS synthesis in part (c).

$$\begin{array}{c}
\text{Gates: 2} \\
\text{Inputs: 2 + 2} \\
\text{Cost: 6}
\end{array}$$

e. (8 points) In the space below, draw (using your logic template) the logic network for your minimum-cost PoS synthesis in part (c), using only NOR gates; you should use only NOR gates even for generating the complement of an input (normally done by a NOT gate).
4. This problem deals with the following K-Map for function $G(A, B, C, D)$.

In this problem, you will be working with several different syntheses for this function, as follows:

a. (3 points) If you were to synthesize this function with a Canonical Sum of Products (CSoP) synthesis, what would be its cost? Note that you do not have to actually find such a synthesis.

$$\text{Cost}_{\text{CSoP}} = (i+2)m + 1$$

$$= (4+2)9 + 1 = 55$$

b. (3 points) If you were to synthesize this function with a Canonical Product of Sums (CPS) synthesis, what would be its cost? Note that you do not have to actually find such a synthesis.

$$\text{Cost}_{\text{CPS}} = (i+2)M + 1$$

$$= (4+2)7 + 1 = 43$$

The remaining parts deal with finding a minimum-cost Sum of Products (SoP) synthesis for this function.

c. (4 points) For a Sum of Products (SoP) synthesis, identify all of the Prime Implicants (PIs) for this logic function, writing a logic expression for each one.

$$\overline{B} \cdot D, B \cdot \overline{C}, \overline{A} \cdot C \cdot D, \overline{A} \cdot B \cdot D, B \cdot C \cdot D$$

d. (4 points) Identify all of the Essential Prime Implicants (EPIs) for this logic function, writing a logic expression for each one.

$$\overline{B} \cdot D, B \cdot \overline{C}, B \cdot C \cdot D$$

e. (6 points) Construct the minimum-cost SoP synthesis for this function. You must explain how you constructed your synthesis; that is, give the order in which you added product terms, and why you chose each product term. When you have constructed the synthesis, write an expression for it. You should not draw a logic diagram for your synthesis.

<table>
<thead>
<tr>
<th>Choice #</th>
<th>PI</th>
<th>Reason</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$B \cdot D$</td>
<td>EPI</td>
</tr>
<tr>
<td>2</td>
<td>$B \cdot \overline{C}$</td>
<td>EPI</td>
</tr>
<tr>
<td>3</td>
<td>$B \cdot C \cdot D$</td>
<td>EPI</td>
</tr>
<tr>
<td>4</td>
<td>$\overline{A} \cdot C \cdot D$</td>
<td>Includes remaining $f=1$ minterm and completes the cover.</td>
</tr>
</tbody>
</table>

$$G = B \cdot D + B \cdot \overline{C} + B \cdot C \cdot D + \overline{A} \cdot C \cdot D$$

The $G$ could be $\overline{A} \cdot B \cdot D$.
5. Two functions $P$ and $Q$ both have the same set of inputs ($e$, $g$, $h$, $j$). The K-Map of each function is given below.

\[
\begin{array}{cccc}
\text{b) eg} & 00 & 01 & 11 & 10 \\
00 & 1 & 0 & 1 & 0 \\
01 & 1 & 0 & 1 & 0 \\
11 & 1 & 1 & 1 & 0 \\
10 & 1 & 1 & 0 & 0 \\
\end{array}
\]

**Function P**

The minimum-cost Sum of Products (SoP) synthesis for each function is given below, along with the cost (as we have defined it in this course) of each individual function:

\[
P = \bar{e} \cdot \bar{g} + e \cdot g + (\text{either } \bar{e} \cdot h \cdot j \text{ or } g \cdot \bar{h} \cdot j) \quad \text{cost} = 14
\]

\[
Q = h \cdot j + \bar{e} \cdot h \quad \text{cost} = 9
\]

For this problem, you will focus on a jointly-minimized SoP synthesis for the 4-input ($e$, $g$, $h$, $j$), 2-output ($P$, $Q$) function. That is, you are trying to minimize the cost of the resulting 4-input, 2-output logic circuit.

a. (8 points) Give a logic expression for each output ($P$ and $Q$) that will result in a jointly-minimized Sum of Products (SoP) synthesis for the 4-input, 2-output function. Provide an explanation of your synthesis if you think it will be helpful.

If we choose $\bar{e} \cdot h \cdot j$ as the last term for $P$, we can re-use that term for $Q$ (even though it is not a PI for $Q$):

\[
P = \bar{e} \cdot \bar{g} + e \cdot g + \bar{e} \cdot h \cdot j
\]

\[
Q = h \cdot j + \bar{e} \cdot h \cdot j \quad \text{stand}
\]

b. (5 points) Draw the jointly-minimized, 4-input, 2-output logic circuit with AND/OR/NOT gates (using your logic template).

c. (3 points) What is the cost (as we have defined it in this course) of your jointly-minimized, 4-input, 2-output logic circuit?

\[
\text{Gates: 16} \\
\text{Inputs: 14} \\
\text{20 = Cost} \\
\text{(vs. 14+9=23 with no sharing)}
\]
6. (10 points) This problem requires you to convert a word description of a situation into a Boolean Algebra logic expression.

Students at a certain school are categorized as being either tall or short in height and either fast or slow in speed. To be eligible to be on the basketball team, a student must be either tall or fast. However, a student is ineligible (not eligible) to be on the team if the student has failed a class or if the student has failed a drug test.

Let $E$ be a binary variable that has value 1 if a student at this school is eligible to be on the basketball team and has value 0 if the student is not eligible. Using the rules given above, write a Boolean expression for the function $E$ in terms of $T$, $F$, $D$, and $C$. You do not need to identify each Boolean Algebra property that you use. Your expression for $E$ need not (indeed, should not) be in Canonical Sum of Products (CSOP) form.

You do need to clearly define each of the four “input” variables and what each variable’s binary values represent. Notice that I have done this for the binary “output” variable ($E$) that represents whether or not a student at this school is eligible to be on the basketball team.

There is more space for this problem on the next page.

Input variables:

$T = 1$ if Tall
$= 0$ if Short

$F = 1$ if Fast
$= 0$ if Slow.

$C = 1$ if failed a Class
$= 0$ if not.

$D = 1$ if failed Drug Test
$= 0$ if not.

Either C or D would make a student ineligible, so we must have $(C + D)$ in order to be eligible. Also, require $(T + F)$.

\[
E = (T + F) \overline{(C + D)} = (T + F) \overline{C} \overline{D} = T \overline{C} D + F \overline{C} D
\]

$E = T \overline{C} D + F \overline{C} D$

7. (8 points) This problem concerns the NMOS logic circuit shown below involving three NMOS transistors (A, B, and C).

Recalling the operation of NMOS transistors, fill in the value (Hi or Lo) for the output voltage $V_f$ for each combination of input voltage values given in the table below.

<table>
<thead>
<tr>
<th>$V_A$</th>
<th>$V_B$</th>
<th>$V_C$</th>
<th>$V_f$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lo</td>
<td>Lo</td>
<td>Lo</td>
<td>Hi</td>
</tr>
<tr>
<td>Lo</td>
<td>Lo</td>
<td>Hi</td>
<td>Hi</td>
</tr>
<tr>
<td>Lo</td>
<td>Hi</td>
<td>Lo</td>
<td>Hi</td>
</tr>
<tr>
<td>Hi</td>
<td>Lo</td>
<td>Lo</td>
<td>Lo</td>
</tr>
<tr>
<td>Hi</td>
<td>Hi</td>
<td>Lo</td>
<td>Lo</td>
</tr>
<tr>
<td>Hi</td>
<td>Hi</td>
<td>Hi</td>
<td>Hi</td>
</tr>
</tbody>
</table>