1) In order to connect both the true and the false state, we need a logical expression also for $\overline{x}$.

$$X = A + B \cdot C$$

$$\Rightarrow \overline{X} = A + \overline{B} \cdot \overline{C}$$

$$= \overline{A} \cdot (\overline{B} \cdot \overline{C})$$

$$= \overline{A} \cdot (\overline{B} + \overline{C})$$

$$= \overline{A} \cdot (B + C)$$

Hence the Switch network can be drawn as follows:
true

false

-2-

true

false

A

B

C

X
2) Prove that
\[ A + \overline{B} \cdot C = (A + C) \cdot (A + \overline{B}) \]

<p>| | | | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>B</td>
<td>C</td>
<td>( \overline{B} )</td>
<td>( \overline{B} \cdot C )</td>
<td>( A + \overline{B} \cdot C )</td>
<td>( A + C )</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Identical

q.e.d.
b)

\[ A + \overline{B} \cdot C \]

\[ A + C \]

\[ A + \overline{B} \]

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

Identical q.e.d.
3) a) Prove that

\[(A \oplus B) \oplus C = A \oplus (B \oplus C)\]

<table>
<thead>
<tr>
<th></th>
<th>(A)</th>
<th>(B)</th>
<th>(C)</th>
<th>(A \oplus B)</th>
<th>left</th>
<th>(B \oplus C)</th>
<th>right</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</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>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

Identical
q.e.d.
b) Let:

\[ A \oplus B \oplus C := (A \oplus B) \oplus C = A \oplus (B \oplus C) \]

↑ definition

Prove that:

\[ (A \oplus B \oplus C) \oplus D = A \oplus (B \oplus C \oplus D) \]

\[ \begin{array}{cccc|c|c|c|c}
A & B & C & D & \text{Left} & \text{Right} \\
\hline
0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 1 & 0 & 1 \\
0 & 0 & 1 & 0 & 0 & 1 \\
0 & 0 & 1 & 1 & 0 & 0 \\
0 & 1 & 0 & 0 & 0 & 1 \\
0 & 1 & 0 & 1 & 0 & 0 \\
0 & 1 & 1 & 0 & 0 & 0 \\
0 & 1 & 1 & 1 & 0 & 0 \\
1 & 0 & 0 & 0 & 0 & 0 \\
1 & 0 & 0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 & 0 & 1 \\
1 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 0 & 0 & 0 & 0 \\
1 & 1 & 0 & 1 & 0 & 1 \\
1 & 1 & 1 & 0 & 0 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 \\
\end{array} \]

Identical \quad q.e.d.
B) Proof by induction:

Given the truth table of the XOR:

<table>
<thead>
<tr>
<th></th>
<th></th>
<th>A⊕B</th>
<th># of 1's</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>2</td>
</tr>
</tbody>
</table>

we notice that A⊕B can be interpreted as a parity bit. Whenever the sum of 1's of A and B is odd, A⊕B is 1 to make the total sum even. Whenever the sum of 1's of A and B is even, A⊕B is 0, keeping the total sum even.

Let us now look at (A⊕B)⊕C. If A⊕B is 0, then the number of 1's in A and B is even. If we add C=0, then (A⊕B)⊕C=0, and the total number of 1's remains
even, etc. All four combinations make the total number of 1's even. Hence \( A \oplus B \oplus C \) is still a parity bit.

This property remains for any number of elements, since the parity bit does not depend on either the number of terms or their sequence. \( \text{q.e.d.} \)

\[
\begin{array}{cccc}
A & B & C & D \\
0 & 0 & 0 & 0 \\
1 & 0 & 1 & 0 \\
1 & 0 & 1 & 0 \\
0 & 1 & 0 & 1 \\
1 & 0 & 1 & 0 \\
\end{array}
\]
4) Given the following circuit:

\[ -9- \]

a) Find the truth table:

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Cin</th>
<th>X</th>
<th>Y</th>
<th>Z</th>
<th>W</th>
<th>Sum</th>
<th>Cout</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Cout is not particularly useful!
\( b) \)

\[
\begin{align*}
\text{Sum} &= A \cdot \overline{B} + A \cdot \text{Cin} = A \cdot (\overline{B} + \text{Cin}) \\
\text{Cout} &= \phi
\end{align*}
\]

is a minimal realization using \( \text{AND, OR, NOT} \).
is an all-NAND realization.
is an all-NOR realization.
Going after the 1's, we find two big patterns with 8 elements each:

\[ T = D + \overline{B} \]
Going after the 0's, we find one pattern with 4 elements:

\[ F = B \cdot D \]

\[ \Rightarrow F = \overline{B \cdot D} \]

\[ = \overline{B} + D \]

same as before.