\( f = \overline{B}C + \overline{B}D + \overline{C} \)

\( f = CD + BC + \overline{BD} \)

FOLLOW UP 13
\[
\begin{array}{ccc}
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
1 & 1 & 1 \\
0 & 0 & 0 \\
1 & 1 & 0 \\
& & \\
A & B & C \\
0 & 1 & 1 \\
1 & 0 & 0 \\
& & \\
\end{array}
\]

\[
\begin{align*}
&\text{ABCD + ABCD + ABCE + ACDE + ABCE + ACDE + ACDE + ACDE + ABCD + ACDE + ACDE} \\
&\text{ABCE + ACDE + ACDE + ACDE + ACDE + ACDE + ACDE + ACDE + ACDE + ACDE} \\
&\text{ABCE} \\
&\text{F = ABD + ACD + ACD + ABD} \\
&\text{MINIMIZE F IN TRUE SOP FORM}
\end{align*}
\]
\[ \overline{BCD} + \overline{AC} + \overline{AD} + C = \overline{A} \]
Complexity.

In order to reduce false circuits, those "don't care" states can be used.

Input conditions:

The function will behave for those specific conditions.

Many times, we simply don't care how conditions are expressed or expected. In some applications, not all possible inputs

Don't care states.
At 3, use only NAND gates.

A number in \( \bullet \odot \) that is multiplied whenever the 4-bit input references "\(x\)" means a circuit that will output

\[ f \]
Inclusion in a group becomes "\( \subseteq \)"

and any \( X \) which is not a group is to change \( G \times X \) into

Note: The effect of including an \( X \) in

\[ F = A \cup C \cup A \cup B \cup C \]
NAND NAND NAND NAND NAND NAND

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

\[ f = \overline{A} \overline{B} \overline{C}D + \overline{A} \overline{B} \overline{C}D + \overline{A} \overline{B} \overline{C}D + \overline{A} \overline{B} \overline{C}D \]