DEMORGAN'S THEOREMS

DeMorgan, a mathematician who knew Boole, proposed two theorems that are an important part of Boolean algebra. In practical terms. DeMorgan's theorems provide mathematical verification of the equivalency of the NAND and negative-OR gates and the equivalency of the NOR and negative-AND gates, which were discussed in part 3.

One of DeMorgan's theorems is stated as follows:

The complement of a product of variables is equal to the sum of the complements of the variables,

Stated another way,

The complement of two or more ANDed variables is equivalent to the OR of the complements of the individual variables.

The formula for expressing this theorem for two variables is

$$\overline{XY} = \overline{X} + \overline{Y}$$

DeMorgan's second theorem is stated as follows:

The complement of a sum of variables is equal to the product of the complements of the variables.

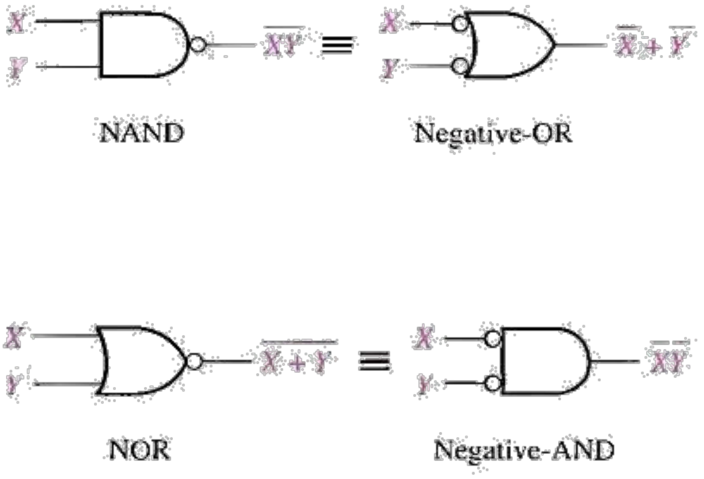
Stated another way,

The complement of two or more ORed variables is equivalent to the AND of the complements of the individual variables,

The formula for expressing this theorem for two variables is

$$\overline{X + Y} = \overline{X}\,\overline{Y}$$

Fig.(4-15) shows the gate equivalencies and truth tables for the two equations above.



Inputs		Output	
X	Y	$\overline{XY}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0

Inputs		Output	
X	Y	$\overline{X+Y}$	$\overline{X}\,\overline{Y}$
0	0	1	1
0	1	0	0
1	0	0	0
1	1	0	0

Fig.(4-15) Gate equivalencies and the corresponding truth tables that illustrate DeMorgan's theorems.

As stated, DeMorgan's theorems also apply to expressions in which there are more than two variables. The following examples illustrate the application of DeMorgan's theorems to 3-variable and 4-variable expressions.

Example

Apply DeMorgan's theorems to the expressions $\overline{XYZ}$ and $\overline{X} + \overline{Y} + \overline{z}$.

$$\overline{XYZ} = \overline{X} + \overline{Y} + \overline{Z}$$

$$\overline{X + y + Z} = \overline{X}\,\overline{Y}\,\overline{Z}$$

Example

Apply DeMorgan's theorems to the expressions WXYZ and W + X + y + z.

$$\overline{WXYZ} = \overline{W} + \overline{X} + \overline{y} + \overline{Z}$$

$$\overline{W + X + y + Z} = \overline{W}\,\overline{X}\,\overline{Y}\,\overline{Z}$$

Applying DeMorgan's Theorems

The following procedure illustrates the application of DeMorgan's theorems and Boolean algebra to the specific expression

$$\overline{\overline{A + B\overline{C}} + \overline{D(E + \overline{F})}}$$

Step 1. Identify the terms to which you can apply DeMorgan's theorems, and think of each term as a single variable. Let $\overline{A + B\overline{C}} = X$ and $\overline{D(E + \overline{F})} = Y$.

Step 2. Since $\overline{X + Y} = \overline{X}\,\overline{Y}$,

$$\overline{\overline{A + B\overline{C}} + \overline{D(E + \overline{F})}} = (\overline{\overline{A + B\overline{C}}})\,(\overline{\overline{D(E + \overline{F})}})$$

Step 3. Use rule 9 ($A = \overline{\overline{A}}$) to cancel the double bars over the left term (this is not part of DeMorgan's theorem).

$$(\overline{\overline{A + B\overline{C}}})\,(\overline{\overline{D(E + \overline{F})}}) = (A + B\overline{C})(\overline{D(E + \overline{F})})$$

Step 4. Applying DeMorgan's theorem to the second term,

$$(A + B\overline{C})(\overline{D(E + \overline{F})}) = (A + B\overline{C})(\overline{D} + \overline{(E + \overline{F})})$$

Step 5. Use rule 9 ($A = \overline{\overline{A}}$) to cancel the double bars over the $E + \overline{F}$ part of the term.

$$(A + B\overline{C})(\overline{D} + \overline{\overline{E + \overline{F}}}) = (A + B\overline{C})(\overline{D} + E + \overline{F})$$

Example

Apply DeMorgan's theorems to each of the following expressions:

(a) $\overline{(A + B + C)D}$ (b) $\overline{ABC + DEF}$ (c) $\overline{A\overline{B} + C\overline{D} + EF}$

Example

The Boolean expression for an exclusive-OR gate is $A\overline{B} + \overline{A}B$. With this as a starting point, use DeMorgan's theorems and any other rules or laws that are applicable to develop an expression for the exclusive-NOR gate.

BOOLEAN ANALYSIS OF LOGIC CIRCUITS

Boolean algebra provides a concise way to express the operation of a logic circuit formed by a combination of logic gates so that the output can be determined for various combinations of input values.

Boolean Expression for a Logic Circuit

To derive the Boolean expression for a given logic circuit, begin at the left-most inputs and work toward the final output, writing the expression for each gate. For the example circuit in Fig.(4-16), the Boolean expression is determined as follows:

The expression for the left-most AND gate with inputs C and D is CD. The output of the left-most AND gate is one of the inputs to the OR gate and B is the other input. Therefore, the expression for the OR gate is B + CD.

The output of the OR gate is one of the inputs to the right-most AND gate and A is the other input. Therefore, the expression for this AND gate is A(B + CD), which is the final output expression for the entire circuit.

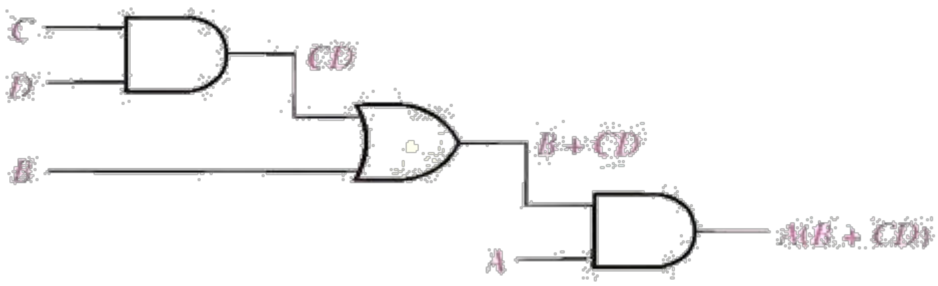


Fig.(4-16) A logic circuit showing the development of the Boolean expression for the output.

Constructing a Truth Table for a Logic Circuit

Once the Boolean expression for a given logic circuit has been determined, a truth table that shows the output for all possible values of the input variables can be developed. The procedure requires that you evaluate the Boolean expression for all possible combinations of values for the input variables. In the case of the circuit in Fig.(4-16), there are four input variables (A, B, C, and D) and therefore sixteen ($2^4 = 16$) combinations of values are possible.

Putting the Results in Truth Table format

The first step is to list the sixteen input variable combinations of 1s and 0s in a binary sequence as shown in Table 4-5. Next, place a 1 in the output column for each combination of input variables that was determined in the evaluation. Finally, place a 0 in the output column for all other combinations of input variables. These results are shown in the truth table in Table 4-5.

Table 4-5

INPUTS				OUTPUT
A	*B*	*C*	*D*	*A(B + CD)*
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
0	0	1	1	0
0	1	0	0	0
0	1	0	1	0
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

SIMPLIFICATION USING BOOLEAN ALGEBRA

A simplified Boolean expression uses the fewest gates possible to implement a given expression.

Example

Using Boolean algebra techniques, simplify this expression:

$$AB + A(B + C) + B(B + C)$$

Solution

Step 1: Apply the distributive law to the second and third terms in the expression, as follows:

$$AB + AB + AC + BB + BC$$

Step 2: Apply rule 7 (BB = B) to the fourth term.

$$AB + AB + AC + B + BC$$

Step 3: Apply rule 5 (AB + AB = AB) to the first two terms.

$$AB + AC + B + BC$$

Step 4: Apply rule 10 (B + BC = B) to the last two terms.

$AB + AC + B$

Step 5: Apply rule 10 ($AB + B = B$) to the first and third terms.

$B+AC$

At this point the expression is simplified as much as possible.

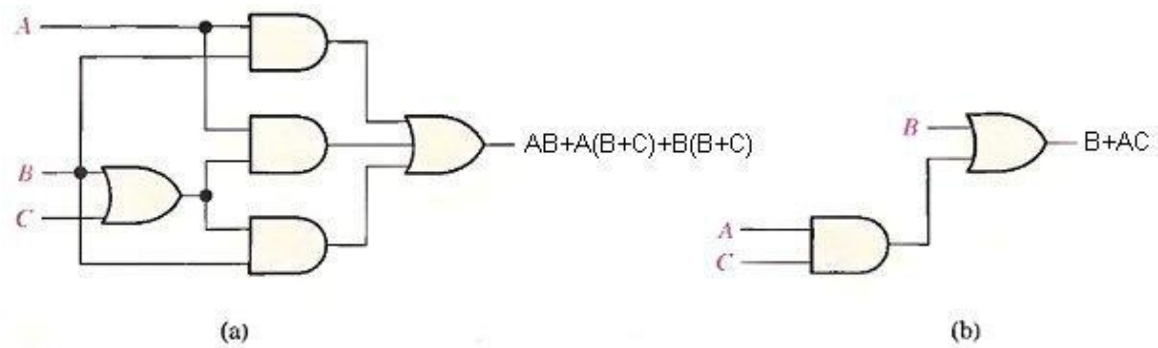


Fig.(4-17) Gate circuits for example above.

<u>Example</u>

Simplify the Boolean expressions:

1- $A\overline{B} + A(B + C) + B(B + C)$.

2- $[A\overline{B}(C + BD) + \overline{A}\,\overline{B}]C$

3- $\overline{A}BC + A\overline{B}\,\overline{C} + \overline{A}\,\overline{B}\,\overline{C} + AB\overline{C} + ABC$

Standard and Canonical Forms:

STANDARD FORMS OF BOOLEAN EXPRESSIONS

All Boolean expressions, regardless of their form, can be converted into either of two standard forms: the sum-of-products form or the product-of-sums form. Standardization makes the evaluation, simplification, and implementation of Boolean expressions much more systematic and easier.

The Sum-of-Products (SOP) Form

When two or more product terms are summed by Boolean addition, the resulting expression is a sum-of-products (SOP). Some examples are:

AB + ABC

ABC + CDE + BCD

AB + BCD + AC

Also, an SOP expression can contain a single-variable term, as in

A + ABC + BCD.

In an SOP expression a single overbar cannot extend over more than one variable.

Example

Convert each of the following Boolean expressions to SOP form:

(a) AB + B(CD + EF)

(b) (A + B)(B + C + D)

(c) (A + B) + C

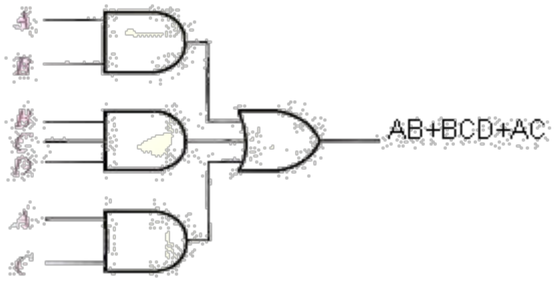


Fig.(4-18) Implementation of the SOP expression AB + BCD + AC.

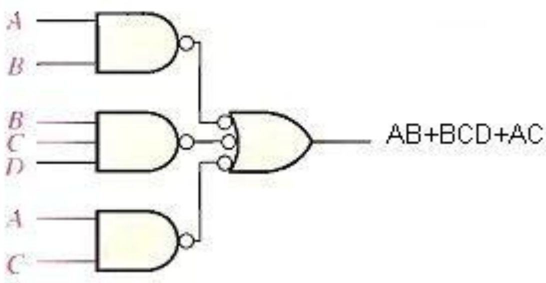


Fig.(4-19) This NAND/NAND implementation is equivalent
to the AND/OR in figure above.

The Standard SOP Form

So far, you have seen SOP expressions in which some of the product terms do not contain all of the variables in the domain of the expression. For example, the expression ABC + ABD + ABCD has a domain made up of the variables A, B, C. and D. However, notice that the complete set of variables in the domain is not represented in the first two terms of the expression; that is, D or D is missing from the first term and C or C is missing from the second term.

A standard SOP expression is one in which all the variables in the domain appear in each product term in the expression. For example, ABCD + ABCD + ABCD is a standard SOP expression.

Converting Product Terms to Standard SOP:

Each product term in an SOP expression that does not contain all the variables in the domain can be expanded to standard SOP to include all variables in the domain and their complements. As stated in the following steps, a nonstandard SOP expression is converted into standard form using Boolean algebra rule 6 (A + A = 1) from Table 4-1: A variable added to its complement equals 1.

Step 1. Multiply each nonstandard product term by a term made up of the sum of a missing variable and its complement. This results in two product terms. As you know, you can multiply anything by 1 without changing its value.

Step 2. Repeat Step 1 until all resulting product terms contain all variables in the domain in either complemented or uncomplemented form. In converting a product term to standard form, the number of product terms is doubled for each missing variable.

Example

Convert the following Boolean expression into standard SOP form: ABC + AB + ABCD

Solution

The domain of this SOP expression A, B, C, D. Take one term at a time. The first term, ABC, is missing variable D or D, so multiply the first term by (D + D) as follows:

ABC = ABC(D + D) = ABCD + ABCD

In this case, two standard product terms are the result.

The second term, AB, is missing variables C or C and D or D, so first multiply the second term by C + C as follows:

AB = AB(C + C) = ABC + ABC

The two resulting terms are missing variable D or D, so multiply both terms by (D + D) as follows:

ABC(D + D) + ABC(D + D)

= A BCD + ABCD + ABCD + ABCD

In this case, four standard product terms are the result.

The third term, ABCD, is already in standard form. The complete standard SOP form of the original expression is as follows:

ABC + AB + ABCD = ABCD + ABCD + A BCD + ABCD + ABCD + ABCD + ABCD

The Product-of-Sums (POS) Form

A sum term was defined before as a term consisting of the sum (Boolean addition) of literals (variables or their complements). When two or more sum terms are multiplied, the resulting expression is a product-of-sums (POS). Some examples are

(A + B)(A + B + C)

(A + B + C)(C + D + E)(B + C + D)

(A + B)(A + B + C)(A + C)

A POS expression can contain a single-variable term, as in

A(A + B + C)(B + C + D).

In a POS expression, a single overbar cannot extend over more than one variable; however, more than one variable in a term can have an overbar. For example, a POS expression can have the term A + B + C but not A + B + C.

Implementation of a POS Expression simply requires ANDing the outputs of two or more OR gates. A sum term is produced by an OR operation and the product of two or more sum terms is produced by an AND operation. Fig.(4-

20) shows for the expression (A + B)(B + C + D)(A + C). The output X of the AND gate equals the POS expression.

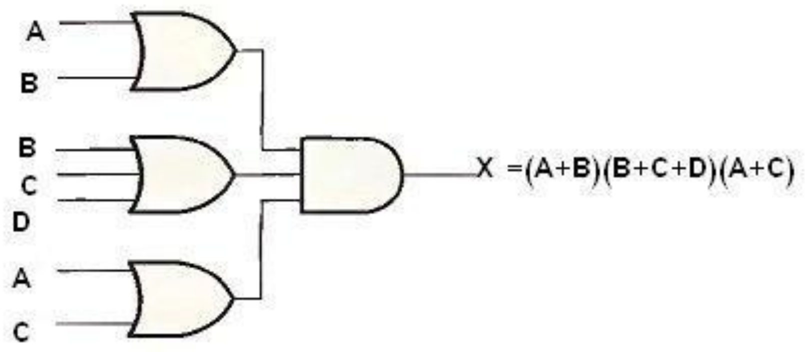


Fig.(4-20)

<u>The Standard POS Form</u>

So far, you have seen POS expressions in which some of the sum terms do not contain all of the variables in the domain of the expression. For example, the expression

(A + B + C) (A + B + D) (A + B + C + D)

has a domain made up of the variables A, B, C, and D. Notice that the complete set of variables in the domain is not represented in e first two terms of the expression; that is, D or D is missing from the first term and C or C is missing from the second term.

A standard POS expression is one in which all the variables in the domain appear in each sum term in the expression. For example,

(A + B + C + D)(A + B + C + D)(A + B + C + D)

is a standard POS expression. Any nonstandard POS expression (referred to simply as POS) can be converted to the standard form using Boolean algebra.

<u>Converting a Sum Term to Standard POS</u>

Each sum term in a POS expression that does not contain all the variables in the domain can be expanded to standard form to include all variables in the domain and their complements. As stated in the following steps, a nonstandard POS expression is converted into standard form using Boolean algebra rule 8 (A A = 0) from Table 4-1:

Step 1. Add to each nonstandard product term a term made up of the product of the missing variable and its complement. This results in two sum terms. As you know, you can add 0 to anything without changing its value.

Step 2. Apply rule 12 from Table 4-1: A + BC = (A + B)(A + C)

Step 3. Repeat Step 1 until all resulting sum terms contain all variables in the domain in either complemented or noncomplemented form.

Example

Convert the following Boolean expression into standard POS form: (A + B + C)(B + C + D)(A + B + C + D)

Solution

The domain of this POS expression is A, B, C, D. Take one term at a time. The first term, A + B + C, is missing variable D or D, so add DD and apply rule 12 as follows:

A + B + C = A + B + C + DD = (A + B + C + D)(A + B + C + D)

The second term, B + C + D, is missing variable A or A, so add AA and apply rule 12 as follows:

B + C + D = B + C + D + AA = (A + B + C + D)(A + B + C + D)

The third term, A + B + C + D, is already in standard form. The standard POS form of the original expression is as follows:

(A + B + C)(B + C + D)(A + B + C + D) = (A + B + C + D)(A + B + C + D) (A + B + C + D)(A + B + C + D) (A + B + C + D)

Examples:-

1. Identify each of the following expressions as SOP, standard SOP, POS, or standard POS:
   (a) $AB + \overline{A}BD + \overline{A}C\overline{D}$ (b) $(A + \overline{B} + C)(A + B + \overline{C})$
   (c) $\overline{A}BC + AB\overline{C}$ (d) $A(A + \overline{C})(A + B)$
2. Convert each SOP expression in Question 1 to standard form.
3. Convert each POS expression in Question 1 to standard form.

CANONICAL FORMS OF BOOLEAN EXPRESSIONS

n variables can be combined to form 2^n minterms.

Note that each maxterm is the complement of its corresponding minterm and vice versa.

Minterms and maxterms are related

- Any minterm m_i is the *complement* of the corresponding maxterm M_i

Minterm	Shorthand	Maxterm	Shorthand
x'y'z'	m_0	x + y + z	M_0
x'y'z	m_1	x + y + z'	M_1
x'yz'	m_2	x + y' + z	M_2
x'yz	m_3	x + y' + z'	M_3
xy'z'	m_4	x' + y + z	M_4
xy'z	m_5	x' + y + z'	M_5
xyz'	m_6	x' + y' + z	M_6
xyz	m_7	x' + y' + z'	M_7

- For example, $m_4' = M_4$ because (xy'z')' = x' + y + z

For example the function F

x	y	z	F
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

$F = \overline{x}\,\overline{y}\,z + x\,\overline{y}\,\overline{z} + x\,y\,z$

$F = m_1 + m_4 + m_7$

Any Boolean function can be expressed as a sum of minterms (sum of products **SOP**) or product of maxterms (product of sums **POS**).

$F = x\,y\,z + \overline{x}\,y\,z + x\,y\,z + x\,y\,z + x\,y\,\overline{z}$

The complement of F = F = F

$F = (x + y + z)\,(x + y + z)\,(x + y + \overline{z})\,(x + y + z)\,(x + y + z)$ $F = M_0\,M_2\,M_3\,M_5\,M_6$

Example

Express the Boolean function F = A + BC in a sum of minterms (SOP).

Solution

The term A is missing two variables because the domain of F is (A, B, C)

A = A(B + B) = AB + AB because B + B = 1

$\bar{B}C$ missing A, so

$\bar{B}C(A + \bar{A}) = A\bar{B}C + \bar{A}\bar{B}C$

$AB(C + \bar{C}) = ABC + AB\bar{C}$

$A\bar{B}(C + \bar{C}) = A\bar{B}C + A\bar{B}\bar{C}$

$F = ABC + AB\bar{C} + \underline{A\bar{B}C} + A\bar{B}\bar{C} + \underline{A\bar{B}C} + \bar{A}\bar{B}C$

Because A + A = A

$F = ABC + AB\bar{C} + A\bar{B}C + A\bar{B}\bar{C} + \bar{A}\bar{B}C$

$F = m_7 + m_6 + m_5 + m_4 + m_1$

In short notation

$F(A, B, C) = \sum(1, 4, 5, 6, 7)$

$\bar{F}(A, B, C) = \sum(0, 2, 3)$

The complement of a function expressed as the sum of minterms equal to the sum of minterms missing from the original function.

Truth table for $F = A + \bar{B}C$

	A	**B**	**C**	$\bar{\mathbf{B}}$	$\bar{\mathbf{B}}\mathbf{C}$	**F**
0	0	0	0	1	0	0
1	0	0	1	1	1	1
2	0	1	0	0	0	0
3	0	1	1	0	0	0
4	1	0	0	1	0	1
5	1	0	1	1	1	1
6	1	1	0	0	0	1
7	1	1	1	0	0	1

Example

Express F = xy + xz in a product of maxterms form.

Solution

F = xy + xz = (xy + x)(xy + z) = (x + x)(y + x)(x + z)(y + z)

remember x + x = 1

F = (y + x)(x + z)(y + z)

F = (x + y + zz)(x + yy + z)(xx + y + z)

F = (x + y + z)(x + y + z)(x + y + z)(x + y + z)(x + y +z)(x + y + z)

========== ------------------ ---------------- ==========

F = (x + y + z)(x + y + z)(x + y + z)(x + y + z)

F = M_4 M_5 M_0 M_2

F(x, y, z) = $\prod(0, 2, 4, 5)$

F(x, y, z) = $\prod(1, 3, 6, 7)$

The complement of a function expressed as the product of maxterms equal to the product of maxterms missing from the original function.

To convert from one canonical form to another, interchange the symbols $\sum$, $\prod$ and list those numbers missing from the original form.

F = M_4 M_5 M_0 M_2 = $m_1 + m_3 + m_6 + m_7$

F(x, y, z) = $\prod(0, 2, 4, 5) = \sum(1, 3, 6, 7)$

Example

Develop a truth table for the standard SOP expression ABC + ABC + ABC.

INPUTS			OUTPUT	
A	*B*	*C*	*X*	PRODUCT TERM
0	0	0	0	
0	0	1	1	$\bar{A}\bar{B}C$
0	1	0	0	
0	1	1	0	
1	0	0	1	$A\bar{B}\bar{C}$
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

Converting POS Expressions to Truth Table Format

Reca11 that a POS expression is equal to 0 only if at least one of the sum terms is equal to 0. To construct a truth table from a POS expression, list all the possible combinations of binary values of the variables just as was done for the SOP expression. Next, convert the POS expression to standard form if it is not already. Finally, place a 0 in the output column (X) for each binary value that makes the expression a 0 and place a 1 for all the remaining binary values. This procedure is illustrated in Example below:

Example

Determine the truth table for the following standard POS expression:

$$(A + B + C)(A + \bar{B} + C)(A + \bar{B} + \bar{C})(\bar{A} + B + \bar{C})(\bar{A} + \bar{B} + C)$$

Solution

There are three variables in the domain and the eight possible binary values are listed in the left three columns of. The binary values that make the sum terms in the expression equal to 0 are A+ B + C: 000; A + B + C: 010: A + B + C: 011; A + B + C: 10l; and A + B + C: 110. For each of these binary values, place a 0 in the output column as shown in the table. For each of the remaining binary combinations, place a 1 in the output column.

INPUTS			OUTPUT	
A	B	C	X	SUM TERM
0	0	0	0	$(A + B + C)$
0	0	1	1	
0	1	0	0	$(A + \overline{B} + C)$
0	1	1	0	$(A + \overline{B} + \overline{C})$
1	0	0	1	
1	0	1	0	$(\overline{A} + B + \overline{C})$
1	1	0	0	$(\overline{A} + \overline{B} + C)$
1	1	1	1	