

Álgebra de Boole

El álgebra booleana es la teoría matemática que se aplica en la lógica combinatoria. Las variables booleanas son símbolos utilizados para representar magnitudes lógicas y pueden tener sólo dos valores posibles: 1 (valor alto) ó 0 (valor bajo).

Operaciones Booleanas y Compuertas Básicas

Las operaciones booleanas son posibles a través de los operadores binarios negación, suma y multiplicación, es decir que estos combinan dos o más variables para conformar funciones lógicas. Una compuerta es un circuito útil para realizar las operaciones anteriormente mencionadas.

Inversión o negación (complemento)

Esta operación se indica con una barra sobre la variable o por medio de un apóstrofe (comilla) en el lado superior derecho de la variable. El apóstrofe (') es un operador algebraico que invierte el valor de una variable, es decir, si X denota la señal de entrada de un inversor, entonces X' representa el complemento de tal señal.

Ejemplo

Sí $X = 0$ entonces $X' = 1$.

En la tabla de verdad 2.1.1. se muestra el resultado de la inversión lógica.

Ecuación	Entrada A	Salida B
$B=A'$	0	1
	1	0

Tabla 2.1.1. Tabla de verdad del inversor

El símbolo lógico de la negación booleana se representa en la figura 2.1.1.



Figura 2.1.1. Inversor.

Suma booleana

La representación matemática de una suma booleana de dos variables se hace por medio un signo más entre las dos variables.

Ejemplo

La suma booleana de las variables A y B se enuncia de la siguiente forma,

$$X = A + B$$

La suma booleana es 1 si alguna de las variables lógicas de la suma es 1 y es 0 cuando todas las variables son 0. Esta operación se asimila a la conexión paralela de contactos.

La tabla de verdad de la suma se muestra en la tabla 2.1.2.

Entrada A	Entrada B	Salida X
0	0	0
0	1	1
1	0	1
1	1	1

Tabla 2.1.2. Tabla de Verdad de la función OR

En circuitos digitales, el equivalente de la suma booleana es la operación OR y su símbolo lógico se representa en la figura 2.1.2.



Figura 2.1.2. Símbolo lógico para la compuerta OR.

Con la correspondiente ecuación $X = A + B$.

El inverso de la función OR es la función NOR. La tabla de verdad se muestra en la tabla 2.1.3.

Entrada A	Entrada B	Salida X
0	0	1
0	1	0
1	0	0
1	1	0

Tabla 2.1.3. Tabla de verdad de la función NOR

El símbolo lógico de la compuerta NOR se representa en la figura 2.1.3.



Figura 2.1.3. Símbolo lógico para la compuerta NOR

Con la correspondiente ecuación $X = (A+B)'$

La suma booleana difiere de la suma binaria cuando se suman dos unos. En la suma booleana no existe acarreo.

Multiplicación booleana

La representación matemática de una multiplicación booleana de dos variables se hace por medio un signo punto (\cdot) entre las dos variables.

La multiplicación booleana de las variables A y B se enuncia de la siguiente forma,

$$X = A \cdot B$$

La multiplicación booleana es 1 si todas las variables lógicas son 1, pero si alguna es 0, el resultado es 0. La multiplicación booleana se asimila a la conexión serie de contactos.

La tabla de verdad de la multiplicación booleana se muestra en la tabla 2.1.4.

Entrada A	Entrada B	Salida X
0	0	0
0	1	0
1	0	0
1	1	1

Tabla 2.1.4. Tabla de verdad de la función AND

En circuitos digitales, el equivalente de la multiplicación booleana es la operación AND y su símbolo se representa en la figura 2.1.4.



Figura 2.1.4. Símbolo lógico de la función AND

con la correspondiente ecuación $X = A \cdot B$

El inverso de la función AND es la función NAND. La tabla de verdad se muestra la tabla 2.1.5.

Entrada A	Entrada B	Salida X
0	0	1
0	1	1
1	0	1
1	1	0

Tabla 2.1.5. Tabla de verdad de la función NAND

El símbolo lógico de la compuerta NAND se representa en la figura 2.1.5.



Tabla 2.1.5. Símbolo lógico de la función NAND

Con la correspondiente ecuación $X = (A \cdot B)'$

Propiedades de las Operaciones Booleanas

Las operaciones booleanas están regidas por tres leyes similares a las del álgebra convencional. Estas incluyen las leyes conmutativas de la suma y la multiplicación y la ley distributiva.

Leyes conmutativas en dos variables

1. Ley conmutativa de la suma se enuncia como sigue

$$X + Y = Y + X$$

En aplicación a los circuitos digitales, podríamos decir que no importa el orden de conexión de las entradas a una compuerta OR.

2. Ley conmutativa de la multiplicación

$$X \cdot Y = Y \cdot X$$

En aplicación a los circuitos digitales, podríamos decir que no importa el orden de conexión de las entradas a una compuerta *AND*.

Leyes asociativas en tres variables

3. Ley asociativa de la adición, se escribe en forma algebraica de la siguiente forma

$$A + (B + C) = (A + B) + C$$

En la figura 2.1.6 se muestra la aplicación de la propiedad a las compuertas *OR*,

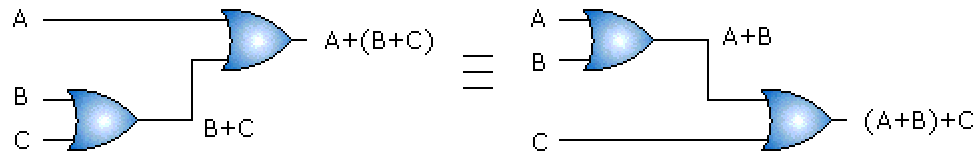


Figura 2.1.6. Ley asociativa de la adición

4. Ley asociativa de la multiplicación

$$A \cdot (B \cdot C) = (A \cdot B) \cdot C$$

En la figura 2.1.7 se muestra la aplicación de la propiedad a las compuertas *AND*,

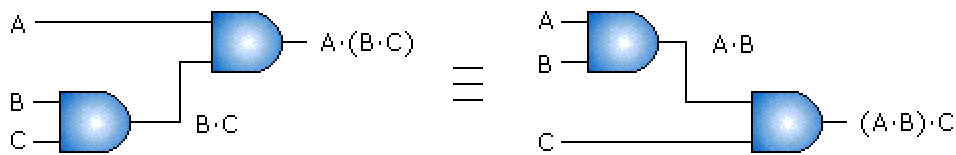


Figura 2.1.7. Ley asociativa de la multiplicación

Ley distributiva para tres variables

En el álgebra de Boole, la multiplicación lógica se distribuye sobre la suma lógica,

$$A \cdot (B + C) = A \cdot B + A \cdot C$$

En la figura 2.1.8 se muestra la aplicación de la propiedad a las compuertas *AND* y *OR*,

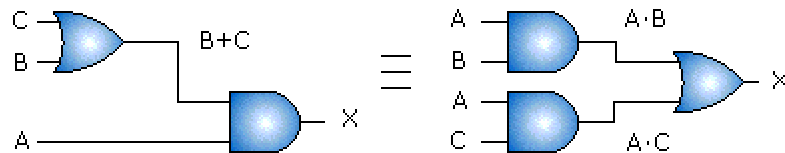


Figura 2.1.8. Ley distributiva para tres variables

Teoremas Booleanos

Los teoremas booleanos son enunciados siempre verdaderos, lo que permite la manipulación de expresiones algebraicas, facilitando el análisis ó síntesis de los circuitos digitales. Los teoremas booleanos son los siguientes:

1. $X + 0 = X$
2. $X + 1 = 1$
3. $X \cdot 0 = 0$
4. $X \cdot 1 = X$
5. $(X')' = X$
6. $X + X = X$
7. $X \cdot X = X$
8. $X + X' = 1$
9. $X \cdot X' = 0$
10. $X + XY = X$
11. $X + X' \cdot Y = X + Y$
12. $X \cdot Y + X \cdot Y' = X$ (Teorema de combinación)
13. $(X + Y)(X + Y') = X + X \cdot Y' + X \cdot Y = X$
14. $X \cdot Y + X \cdot Z + Y \cdot Z' = XZ + Y \cdot Z'$ (Consenso)

El teorema 12 se conoce como la ley distributiva para tres variables.

Demostración teorema 12:

$$X \cdot Y + X \cdot Y' = X$$

Utilizando la [ley distributiva](#) para tres variables

$$X \cdot Y + X \cdot Y' = X \cdot (Y + Y')$$

Aplicando el teorema 8 se tiene,

$$X \cdot Y + X \cdot Y' = X \cdot 1$$

Dando como resultado,

$$X \cdot Y + X \cdot Y' = X$$

Esta expresión indica que la suma de dos productos canónicos adyacentes, es decir que difieren en una sola de las variables, se reduce al producto de los demás términos suprimiéndose dicha variable. El teorema 13 es otro caso del teorema de combinación. Los teoremas 12 y 13 se utilizarán en las lecciones siguientes de forma sistemática para sintetizar circuitos lógicos con los métodos de mapas de karnaugh y el algoritmo de Quine-McCluskey.

Teoremas de DeMorgan

Los teoremas de DeMorgan demuestran la equivalencia entre las puertas *NAND* y negativa - *OR*, y las puertas *NOR* y negativa - *AND*.

1. El complemento de la suma de variables es igual al producto de los complementos de las variables.

$$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$$

En el caso de dos variables se tiene,

$$(X + Y)' = X' \cdot Y'$$

El circuito equivalente a la ecuación anterior se muestra en la figura 2.1.9.

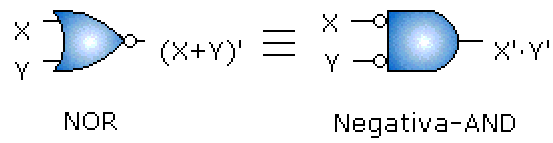


Figura 2.1.9. Símbolo lógico para la compuerta NOR.

Ejemplo

Obtener una compuerta OR utilizando compuertas NAND.

$$Y = (A + B) = [(A + B)']' = (A' \cdot B')'$$

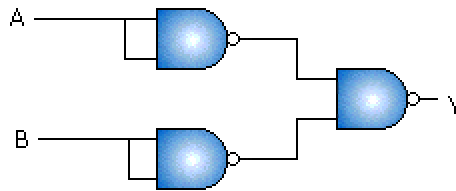


Figura 2.1.10. Compuerta OR utilizando compuertas NAND

2. El complemento del producto de variables es igual a la suma de los complementos de las variables.

$$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$$

En el caso de dos variables se tiene,

$$(X \cdot Y)' = X' + Y'$$

El circuito equivalente en dos variables a la ecuación se muestra en la figura 2.1.11.

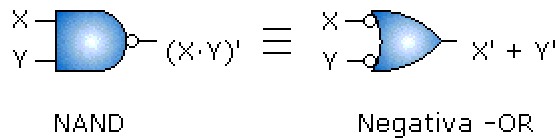


Figura 2.1.11. Símbolo lógico para la compuerta NOR.

Ejemplo

Obtener una compuerta AND utilizando compuertas NOR.

$$Y = A \cdot B = [(A \cdot B)']' = (A' + B)'$$

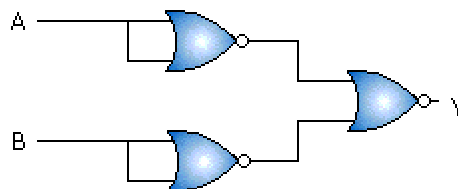


Figura 2.1.12. Circuito lógico para la compuerta AND

Simplificación de Expresiones Lógicas

El objetivo de la simplificación de expresiones lógicas es reducir la expresión al menor número posible de términos. Las expresiones lógicas se pueden simplificar utilizando los teoremas anteriores.

Ejemplo

$$F = A \cdot B' \cdot C + A \cdot B' \cdot C'$$

$$F = A \cdot B' \cdot (C + C')$$

$$F = A \cdot B'$$

Ejemplo

$$F = (A'+B) \cdot (A+B)'$$

$$F = A \cdot A' + A' \cdot B' + A \cdot B + B \cdot B'$$

$$F = A' \cdot B' + A \cdot B$$

Ejemplo

$$F = [(A' + C) \cdot (B + D)]'$$

$$F = (A' + C)' + (B + D)'$$

$$F = A \cdot C' + B' \cdot D$$

Ejemplo

$$F = (X + Z') \cdot (Z + W \cdot Y)' + (V \cdot Z + W \cdot X') \cdot (Y + Z)'$$

$$F = (X + Z') \cdot [Z' \cdot (W' + Y')] + [(V \cdot Z + W \cdot X') \cdot (Y' \cdot Z')]$$

$$F = (X + Z') \cdot (Z' \cdot W' + Z' \cdot Y') + V \cdot Y' \cdot Z \cdot Z' + W \cdot X' \cdot Y' \cdot Z'$$

$$F = W' \cdot X \cdot Z' + X \cdot Y' \cdot Z' + Z' \cdot Z' \cdot W' + Z' \cdot Z' \cdot Y' + W \cdot X' \cdot Y' \cdot Z'$$

$$F = W' \cdot X \cdot Z' + X \cdot Y' \cdot Z' + W' \cdot Z' + Y' \cdot Z' + W \cdot X' \cdot Y' \cdot Z'$$

$$F = W' \cdot Z' \cdot (1 + X) + Y' \cdot Z' \cdot (1 + X) + W \cdot X' \cdot Y' \cdot Z'$$

$$F = W' \cdot Z' + Y' \cdot Z' + W \cdot X' \cdot Y' \cdot Z'$$

$$F = W' \cdot Z' + Y' \cdot Z' \cdot (1 + W \cdot X')$$

$$F = Z' \cdot (W' + Y')$$

Implementación de Funciones Lógicas mediante Compuertas.

La forma más fácil de encontrar la expresión de un circuito lógico consiste en comenzar con las entradas situadas más a la izquierda e ir avanzando hasta la salida de cada compuerta lógica, obteniendo la expresión para cada una de ellas. Al final del recorrido se debe tener la expresión para todo el circuito. La expresión resultante podemos simplificarla para obtener una más sencilla y así obtener un circuito más reducido.

Ejemplo

Encontrar la expresión para el circuito de la figura.

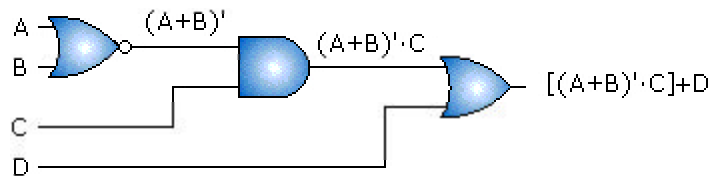


Figura 2.1.13. Símbolo lógico para la compuerta NOR.

1. La expresión de la compuerta NOR situada a la izquierda cuyas entradas son A y B es $(A+B)'$. Esta es la primera entrada de la compuerta AND situada a la derecha.
2. La expresión de la compuerta AND cuyas entradas son $(A+B)'$ y C es $(A+B)' \cdot C$.
3. La salida de la compuerta AND es la primera entrada de la compuerta OR del extremo derecho. Por lo tanto, la expresión de esta compuerta OR es $[(A+B)' \cdot C]+D$.

Síntesis de Diseño de Circuitos Combinatorios

Síntesis se entiende como la obtención de circuitos lógicos, a partir de una descripción inicial que utiliza el lenguaje convencional y luego es transferida a una tabla de verdad.

Una tabla de verdad es una representación básica de una función lógica, en la cual se listan las salidas del circuito lógico para las posibles combinaciones de entrada. Las combinaciones de entrada están ordenadas por renglones (líneas) y cada renglón contiene su salida respectiva. Por ejemplo, la tabla de verdad para una función lógica de 3 variables, tendrá 8 líneas para 8 combinaciones de entrada, conteniendo cada línea, su salida respectiva. En la tabla 2.2.1. se ilustra una función de 3 variables para el caso mencionado.

Renglón o línea	A	B	C	Función de salida	Mintérmino	Maxtérmino
0	0	0	0	$F(0,0,0)$	$A' \cdot B' \cdot C'$	$A+B+C$
1	0	0	1	$F(0,0,1)$	$A' \cdot B' \cdot C$	$A+B+C'$
2	0	1	0	$F(0,1,0)$	$A' \cdot B \cdot C'$	$A+B'+C$
3	0	1	1	$F(0,1,1)$	$A' \cdot B \cdot C$	$A+B'+C'$
4	1	0	0	$F(1,0,0)$	$A \cdot B' \cdot C'$	$A'+B+C$
5	1	0	1	$F(1,0,1)$	$A \cdot B' \cdot C$	$A'+B+C'$
6	1	1	0	$F(1,1,0)$	$A \cdot B \cdot C'$	$A'+B'+C$
7	1	1	1	$F(1,1,1)$	$A \cdot B \cdot C$	$A'+B'+C'$

Tabla 2.2.1. Funciones de salida, maxtérminos y mintérminos

En general, la tabla de verdad para una función lógica de n variables tendrá 2^n líneas.

Métodos para Sintetizar Circuitos Lógicos

Los métodos para sintetizar circuitos lógicos requieren en primer lugar, la comprensión de algunos conceptos, entre ellos:

- Literal: Variable o el complemento de una variable.

Ejemplo: X' , Y' , X , Y .

- Dominio de una expresión booleana: Es el conjunto de variables contenido en una expresión booleana.

Ejemplo: Determine el dominio de la expresión $X' \cdot Y \cdot Z + X \cdot Y' \cdot Z \cdot W$.

El dominio es X, Y, Z, W .

- Término normal: Un producto o término suma en donde ninguna variable aparece repetida.

Ejemplo de término repetido: $X \cdot Y \cdot Y \cdot Z \cdot X' \cdot X' \cdot Y$

Ejemplo de término no repetido: $X' \cdot Y \cdot Z \cdot Z \cdot Y' \cdot X$

- Término producto: Un solo literal o el producto lógico (multiplicación booleana) de dos o más literales.

Ejemplo: X' , $X \cdot Y'$, $Z \cdot Y$, $X \cdot Y' \cdot Z$

Un término producto es 1 sólo para una combinación de valores de las variables.

Ejemplo: El término producto $X \cdot Y' \cdot Z$ es 1 sólo para $X=1$, $Y=0$ y $Z=1$ y es 0 para el resto de combinaciones. El valor en binario será 101 ó 5 en decimal.

- Término suma: Un solo literal o una suma lógica (suma booleana) de dos o más literales.

Ejemplo: X , $X + Y'$, $X' + Z'$, $X + Y + Z$, $X + Y' + Z'$

Un término suma es 1 cuando cualquier literal que lo compone es 1.

Ejemplo: El término $X + Y' + Z'$ es 0 para $X=0$ ó $Y=1$ ó $Z=1$ y es 1 para el resto de combinaciones. El valor en binario será 011 ó 3 en decimal.

- Suma de productos: Suma lógica de términos productos (Ver tabla 2.2.1).

Ejemplo: $X' + X \cdot Y' + Z \cdot Y + X \cdot Y' \cdot Z$

Forma estándar de la suma de productos

Una suma de productos no se encuentra en su forma estándar cuando alguno de los términos producto no contiene alguna de las variables del dominio de la expresión.

Ejemplo

$X' \cdot Y \cdot Z + X \cdot Y' \cdot Z \cdot W$. El dominio es X, Y, Z, W . El primer término producto no contiene el literal W ó W' .

Ejemplo

$X' \cdot Y \cdot Z' \cdot W + X \cdot Y' \cdot Z \cdot W$. En cada uno de los términos de la expresión aparecen todas las variables del dominio. Por lo tanto, la suma de productos está en su forma estándar.

- Producto de sumas: Producto lógico de términos suma (Ver tabla 2.2.1).

Ejemplo: $X \cdot (X+Y) \cdot (X'+Z) \cdot (X+Y+Z) \cdot (X+Y'+Z)$.

Forma estándar del producto de sumas

Un producto de sumas no se encuentra en su forma estándar cuando alguno de los términos suma no contiene alguna de las variables del dominio de la expresión.

Ejemplo

$(X'+W+Z) \cdot (X'+Y'+Z+W) \cdot (X+Y)$. El dominio es X, Y, Z, W . El primer término suma no contiene el literal Y ó Y' . El tercer término suma no contiene los literales Z ó Z' y W ó W' .

Ejemplo

$(X' \cdot Y \cdot Z' \cdot W) \cdot (X \cdot Y' \cdot Z \cdot W)$. En cada uno de los términos de la expresión aparecen todas las variables del dominio. Por lo tanto, el producto de sumas está en su forma estándar.

- Mintérmino: Es un término de producto con n literales en el cual hay n variables. De n variables obtenemos 2^n mintérminos.

Ejemplo de mintérminos de 3 variables: $X' \cdot Y' \cdot Z', X' \cdot Y' \cdot Z, X' \cdot Y \cdot Z', X' \cdot Y \cdot Z, X \cdot Y' \cdot Z', X \cdot Y' \cdot Z, X \cdot Y \cdot Z', X \cdot Y \cdot Z$. (Ver tabla 2.2.1.).

- Maxtérmino: Es un término de suma con n literales en el cual hay n variables. De n variables obtenemos 2^n maxtérminos. (Ver tabla 2.2.1.).

Ejemplo de maxtérminos de 3 variables: $X+Y+Z, X+Y+Z', X+Y'+Z, X+Y'+Z', X'+Y+Z, X'+Y+Z', X'+Y'+Z, X'+Y'+Z'$. (Ver tabla 2.2.1.).

Los métodos existentes para sintetizar circuitos lógicos son:

- Suma de productos (SDP)
- Producto de sumas (PDS)
- Mapas de Karnaugh
- Algoritmo de Quine – McCluskey

Representación por Suma de Productos y Producto de Sumas

En la lección anterior vimos las definiciones básicas para comprender los métodos de síntesis de circuitos lógicos. En esta lección se explicarán los dos primeros de estos métodos para sintetizar circuitos lógicos.

Método de Suma de Productos (SDP) * * * * *

La suma de productos de una función lógica es la suma de los mintérminos correspondientes a las líneas de la tabla de verdad para las que la función produce una salida igual a 1. La función obtenida es la suma de productos.

Ejemplo

Obtener la suma de productos para la función lógica de la tabla 2.3.1.

Línea	A	B	C	Función de salida F ₁
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	1
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Tabla 2.3.1. Tabla de verdad para la función lógica F₁

La función puede ser expresada conformando un término mínimo por cada combinación de variables que producen un 1 en la función para luego obtener la suma de todos los términos. La función lógica para la tabla 2.3.1 se determina expresando las combinaciones 010, 100, 101 y 111 como $A \cdot B \cdot C'$, $A \cdot B' \cdot C'$, $A \cdot B' \cdot C$ y $A \cdot B \cdot C$.

$$F_1 = \sum_{A,B,C}(2,4,5,7) = A \cdot B \cdot C' + A \cdot B' \cdot C' + A \cdot B' \cdot C + A \cdot B \cdot C.$$

Cada mintérmino de la función anterior representa una compuerta AND de tres entradas y la implementación de la función es posible a través de la aplicación de la operación OR a las salidas de las cuatro compuertas AND. Por tanto, el número total de compuertas AND dependerá del total de mintérminos de la expresión. El circuito se muestra en la figura 2.3.1.

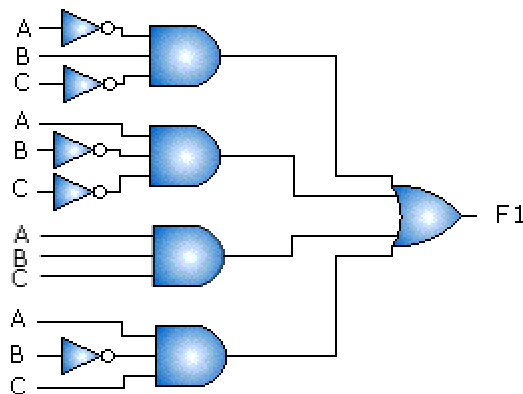


Figura 2.3.1. Circuito lógico para la función lógica F₁.

En una suma de productos se cumple la igualdad de la función al valor lógico 1 si al menos uno de sus términos productos es igual a 1.

Ejemplo

Obtener la suma de productos para la función lógica de la tabla 2.3.2.

A	B	F ₂
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 2.3.2. Tabla de verdad de la función F₂.

En la tabla de verdad existen dos condiciones para las cuales la salida es 1. Estas son las siguientes:

1. La primera se presenta cuando A es Bajo(0) y B es Alto(1). El resultado 1 de esta condición se puede expresar como el producto lógico:

$$A' \cdot B$$

2. La segunda condición se presenta cuando A es 1 y B es 0. Esta condición ocasiona un resultado 1, si el producto lógico es:

$$A \cdot B'$$

Como cualquiera de estas 2 condiciones hace que la salida sea 1, entonces la función lógica que los representa es la suma lógica de los productos anteriores:

$$F_2 = A' \cdot B + A \cdot B' = A \oplus B$$

La representación de la función anterior con compuertas OR y AND se muestra en la figura 2.3.2.

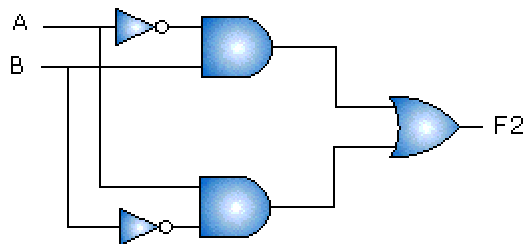


Figura 2.3.2. Función F₂ utilizando compuertas AND Y OR

Esta función corresponde a la función OR exclusiva, cuya compuerta se representa en la figura 2.3.3.



Figura 2.3.3. Símbolo lógico de la función OR - exclusiva.

Ejemplo

Obtener la función SDP para la función lógica de la tabla 2.3.3. Simplificar la función y dibujarla.

A	B	F ₃
0	0	1
0	1	0
1	0	0
1	1	1

Tabla 2.3.3. Tabla de verdad de la función F_3

Utilizando suma de productos para las líneas 1 y 4 de la tabla se obtiene,

$$F_3 = A' \cdot B' + A \cdot B, \text{ simplificando}$$

$$F_3 = (A+B)' + A \cdot B$$

$$F_3 = (A \oplus B)'$$

El circuito lógico de la función anterior se muestra en la figura 2.3.4.

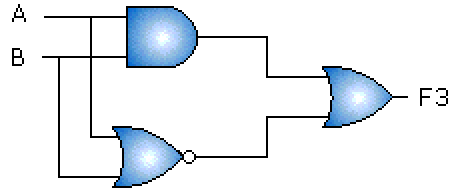


Figura 2.3.4. Función F_3 utilizando compuertas AND, NOR y OR.

El símbolo lógico de la compuerta NOR - Exclusiva se muestra en la figura 2.3.5.



Figura 2.3.5. Símbolo lógico de la función NOR - exclusiva

Conversión de una expresión lógica a formato de suma de productos

La metodología empleada en la transformación de una suma de productos a su forma estándar se basa en el teorema 6 (Ver lección 1 parte 2), que establece que una variable sumada con su complemento es siempre igual a 1; $A + A' = 1$. Los pasos son los siguientes:

1. Los términos producto que no contengan la(s) variable(s) del dominio, multiplicarlos por un término formado por dicha variable más el complemento de la misma (teorema 6).
2. Repetir el paso 1 para todos los términos de la expresión que no contengan todas las variables (o sus complementos) del dominio. Resolver los términos intervenidos.

Ejemplo

Convertir la expresión booleana $A \cdot B \cdot C' + B \cdot C + A'$ a su forma estándar.

El dominio de la expresión es el conjunto de variables A , B y C . Se observa la falta de formato estándar para el segundo y tercer término producto. Sobre ellos se aplicará el procedimiento, para luego volver a agrupar toda la expresión:

Término $B \cdot C$

$$B \cdot C = B \cdot C \cdot (A+A') = A \cdot B \cdot C + A' \cdot B \cdot C$$

Término A

$A' = A' \cdot (C+C') = A' \cdot C + A' \cdot C'$; la expresión aún no tiene el formato estándar, entonces multiplicamos cada término por $(B+B')$
 $A' \cdot C \cdot (B+B') + A' \cdot C' \cdot (B+B') = A' \cdot B \cdot C + A' \cdot B' \cdot C + A' \cdot B \cdot C' + A' \cdot B' \cdot C'$

La expresión en su formato estándar es:

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

Método de producto de sumas (PDS)

El producto de sumas de una función lógica es la multiplicación de los maxtérminos correspondientes a las líneas de la tabla de verdad para las que la función produce una salida igual a 0. La función obtenida es el producto de sumas.

Ejemplo

Obtener el producto de sumas para la función lógica de la tabla 2.3.4.

Renglón o línea	A	B	C	Función de salida F_4
0	0	0	0	0
1	0	0	1	0
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	1
7	1	1	1	1

Tabla 2.3.4. Tabla de verdad para la función lógica F_4

La función puede ser expresada conformando un término máximo para cada combinación de variables que producen un 0 en la función y luego obtener el producto de todos los términos. La función lógica para la tabla 2.3.4 se determina expresando las combinaciones 000, 001, 011 y 110 como $(A+B+C)$, $(A+B+C')$, $(A+B'+C')$ y $(A'+B+C)$. La función lógica es la siguiente:

$$F_4 = \sum_{A,B,C} (0,1,3,4) = (A+B+C) \cdot (A+B+C') \cdot (A+B'+C') \cdot (A'+B+C)$$

Cada maxtérmino de la función anterior representa una compuerta OR de tres entradas y la implementación de la función es posible a través de la aplicación de la operación AND a las salidas de las cuatro compuertas AND. Por tanto, el número total de compuertas AND dependerá del total de mintérminos de la expresión. El circuito se muestra en la figura 2.3.6.

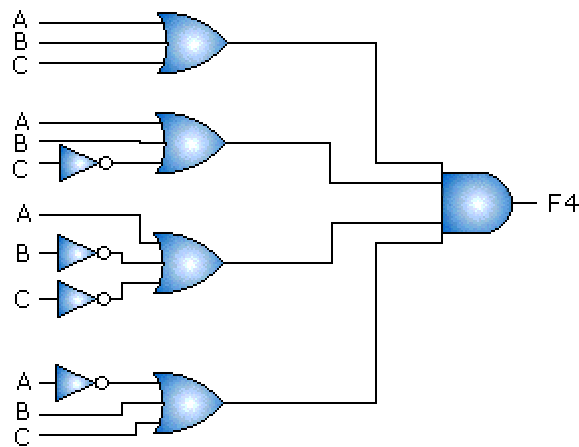


Figura 2.3.6. Circuito lógico para la función lógica F_4

Un producto de sumas es igual a 0 si al menos uno de los términos suma es igual a 0.

Ejemplo

Obtener el producto de sumas para la función lógica de la tabla 2.3.5.

A	B	F_5
0	0	0
0	1	1
1	0	1
1	1	0

Tabla 2.3.5. Tabla de verdad de la función OR - exclusiva

Considere el complemento de la función de Boole F_5 . Este puede obtenerse de la tabla 2.3.5. formando un término mínimo por cada combinación que produce un cero y luego haciendo la suma de los términos. El complemento de F_5 se expresa así:

$$F_5' = A' \cdot B' + A \cdot B$$

La expresión F_5 se obtiene la negar F_5' :

$$F_5 = (F_5')' = (A' \cdot B' + A \cdot B)' = (A' \cdot B')' \cdot (A \cdot B)' = [(A')' + (B)'] \cdot (A + B)' = (A + B) \cdot (A' + B')$$

Si cualquiera de los términos del PDS es cero, la función es cero.

*****Nota:** De los 2 métodos anteriores, se pueden escoger algunos criterios para aplicar un método u otro, siendo estos los siguientes:

- Si en la última columna de la tabla de verdad, o sea en la columna que indica los resultados, si predominan los ceros es más conveniente utilizar la suma de productos.
- Si en la columna que indica los resultados, predominan los unos, es más conveniente utilizar el método del producto de sumas.