Uleam
UNIVERSIDAD LAICA
ELOY ALFARO DE MANABÍ

FACCI
FACULTAD DE CIENCIAS INFORMÁTICAS

ÁLGEBRA DE BOOLE Y SIMPLIFICACIÓN LÓGICA

CONTENIDO

- Introducción
- Operaciones y expresiones booleanas
- Leyes y reglas del álgebra de Boole
- Teoremas de DeMorgan
- Análisis booleano de los circuitos lógicos
- Simplificación mediante el álgebra de Boole
- Formas estándar de las expresiones booleanas
- Expresiones booleanas y tablas de verdad
- Mapas de Karnaugh

Introducción

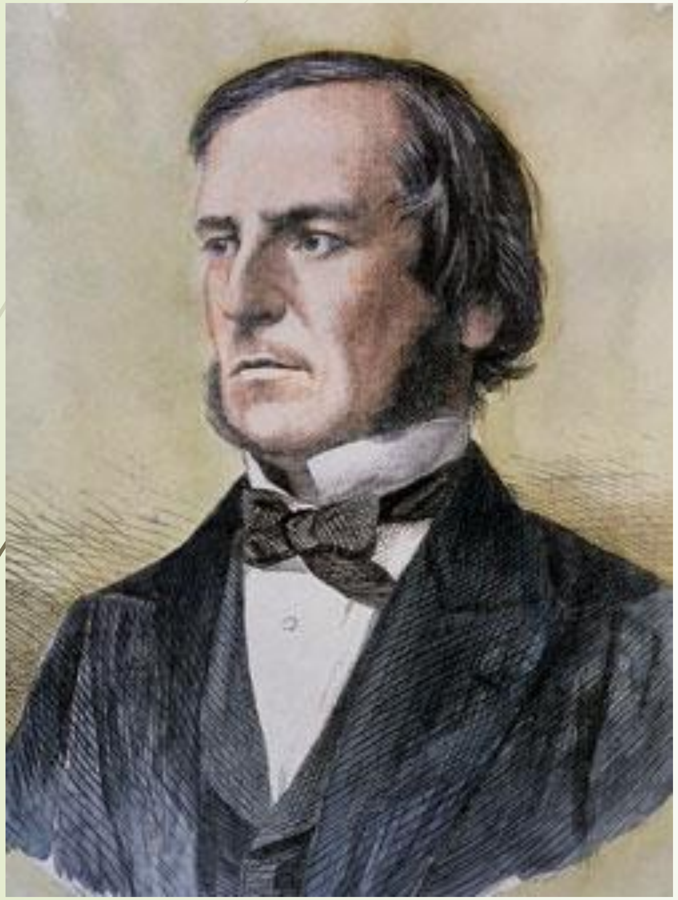


Fig. 1.- George Boole (1815 – 1864)

El álgebra de Boole son las matemáticas de los sistemas digitales, que permite fácilmente representar, analizar y diseñar circuitos digitales. Sus principios teóricos fueron desarrollados por el matemático inglés George Boole en su obra "Análisis matemático de la lógica" publicada en 1847. Es indispensable tener unos conocimientos básicos del álgebra booleana para estudiar y analizar los circuitos lógicos. En la clase anterior, se han presentado las operaciones y expresiones booleanas para las puertas NOT, AND, OR, NAND y NOR.

Cuando se diseña un circuito por métodos booleanos, el primer paso consiste generalmente en obtener su tabla de verdad de acuerdo con las condiciones de entrada y salida. A partir de esta tabla se deriva entonces en una *ecuación booleana* que se simplifica y conduce al circuito lógico deseado.

El circuito obtenido por este método es el óptimo porque requiere de un número mínimo de compuertas para su realización. Esto reduce el costo, el tamaño físico y el consumo de potencia del mismo y mejora su confiabilidad y velocidad. Todas estas consideraciones son importantes cuando diseñan circuitos digitales.

OPERACIONES Y EXPRESIONES BOOLEANAS

El álgebra booleana difiere en gran medida del álgebra ordinaria (*no encontraremos raíces cuadradas, logaritmos, progresiones geométricas, etc.*), ya que a las ***constantes*** y ***variables booleanas*** sólo se les permite tener dos valores posibles: 0 y 1.

Las variables booleanas se utilizan a menudo para representar el nivel de voltaje presente en un alambre o en las terminales de entrada/salida de un circuito. Por ejemplo en cierto sistema digital el valor booleano 0 podría asignarse a cualquier voltaje en el intervalo de 0 a 0.8 V, mientras que el valor booleano 1 podría asignarse a cualquier voltaje entre 2 y 5 V.

En la siguiente tabla muestra los términos más comunes para asignar un nivel lógico.

0 lógico	1 lógico
Falso	Verdadero
Apagado	Encendido
Bajo	Alto
No	Sí
Interruptor abierto	Interruptor cerrado

Suma booleana

La **suma booleana** es equivalente a la operación OR y a continuación se muestran sus reglas básicas junto con su relación con la puerta OR:

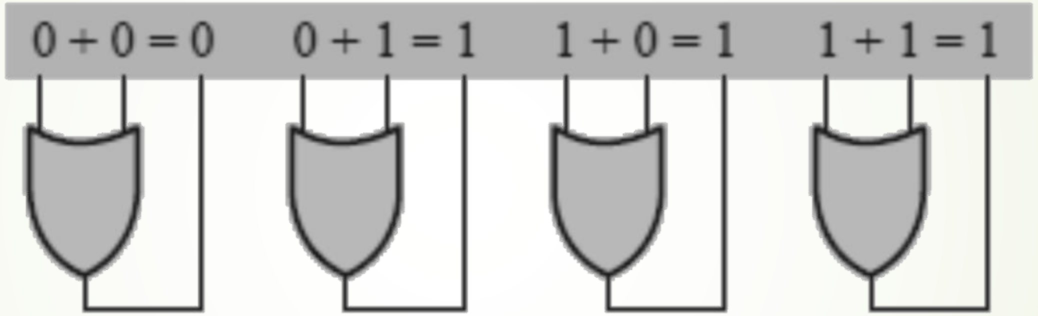


Figura 3.- La puerta OR es un sumador booleano

- ✓ En el álgebra de Boole, un ***término suma*** es una suma de literales.
- ✓ En los circuitos lógicos, un término suma se obtiene mediante una operación OR, sin que exista ninguna operación AND en la expresión.
- ✓ Un término suma es igual a 1 cuando uno o más de los literales del término es 1.
- ✓ Un término suma es igual a 0 sólo si cada uno de los literales son iguales a 0.

Multiplicación booleana

La **multiplicación booleana** es equivalente a la operación AND y sus reglas básicas junto con sus relaciones con la puerta AND se ilustran a continuación:

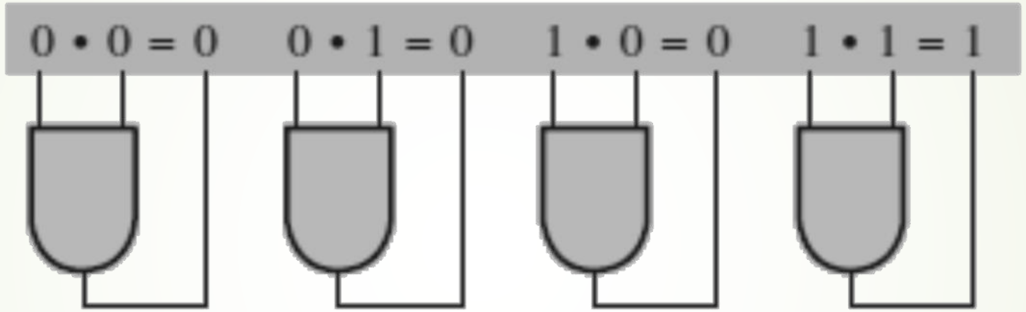


Figura 4.- *La puerta AND es un multiplicador booleano*

- ✓ En el álgebra de Boole, un ***término producto*** es un producto de literales.
- ✓ En los circuitos lógicos, un término producto se obtiene mediante una operación AND, sin que existe ninguna operación OR en la expresión.
- ✓ Un término producto es igual a 1 sólo si cada uno de los literales del término es 1.
- ✓ Un término producto es igual a 0 cuando uno o más de los literales son iguales a 0

LEYES Y REGLAS DEL ÁLGEBRA DE BOOLE

Al igual que en otras áreas de las matemáticas, existen en el álgebra de Boole una serie de reglas y leyes bien determinadas que tienen que seguirse para aplicarla correctamente. Las más importantes son las que se presentan a continuación

Ley conmutativa

Esta ley establece que el orden en que se aplica a las variables es indiferente. La Figura 5 ilustra la ley conmutativa aplicada a una puerta OR y una AND, en la que se puede ver que es indistinto a qué entrada asignemos cada una de las variables. (El símbolo ≡ significa "equivalente a".)

- ✓ *La ley conmutativa de la suma para dos variables es:*

$$A + B = B + A$$

- ✓ *La ley conmutativa de la multiplicación para dos variables es:*

$$AB = BA$$

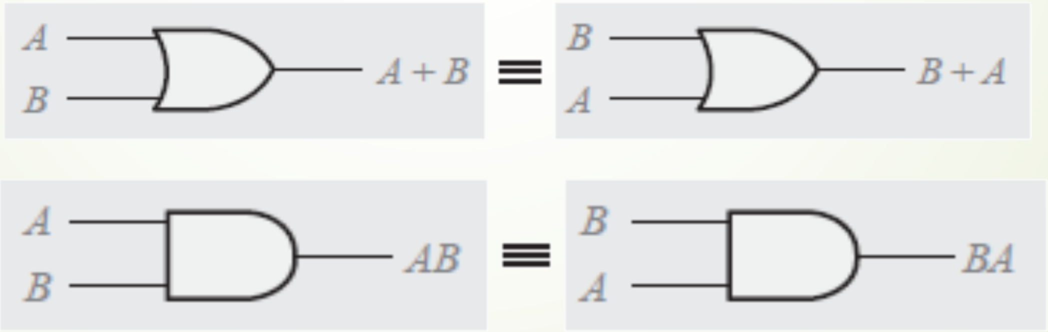


Figura 5.- *Aplicación de la ley conmutativa de la suma y la multiplicación*

Ley asociativa

Esta ley establece que cuando se aplica la operación OR a más de dos variables, el resultado es el mismo independientemente de la forma en que se agrupen las variables.

- ✓ *La ley asociativa de la suma para tres variables se escribe como sigue:*

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

- ✓ *La ley asociativa de la multiplicación para tres variables se escribe del siguiente modo:*

$$A(BC) = (AB)C$$

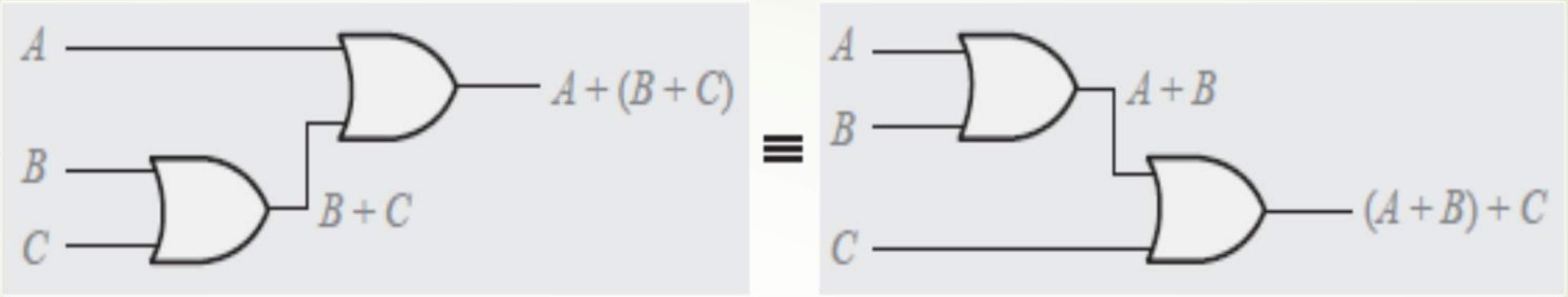


Fig. 6a.- *Aplicación de la ley asociativa de la suma*

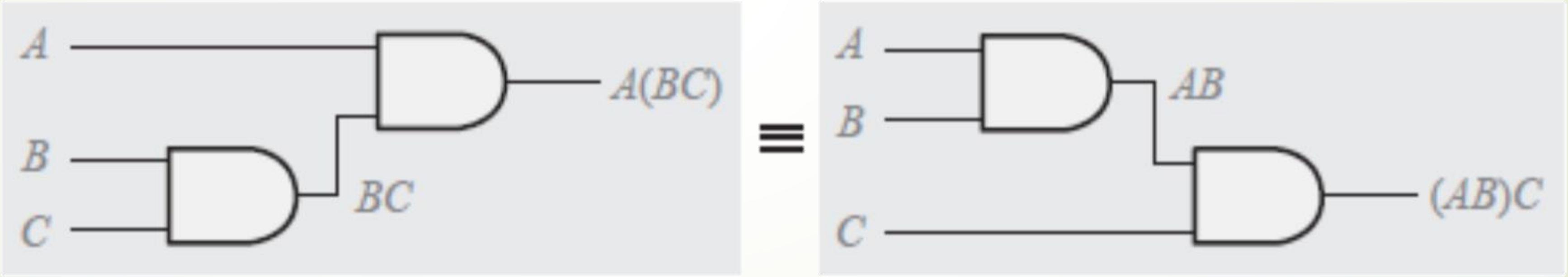


Fig. 6b.- *Aplicación de la ley asociativa de la multiplicación*

Ley distributiva

Esta ley establece que aplicar la operación OR a dos o más variables y luego aplicar la operación AND al resultado de esa operación y a otra variable aislada, es equivalente a aplicar la operación AND a la variable aislada con cada uno de los sumandos y luego realizar la operación OR con los productos resultantes. La ley distributiva expresa también el proceso de *sacar factor común* en el que la variable común A se saca como factor de los productos parciales, como por ejemplo, $AB + AC = A(B + C)$. La Figura 7 ilustra la ley distributiva mediante su implementación de puertas.

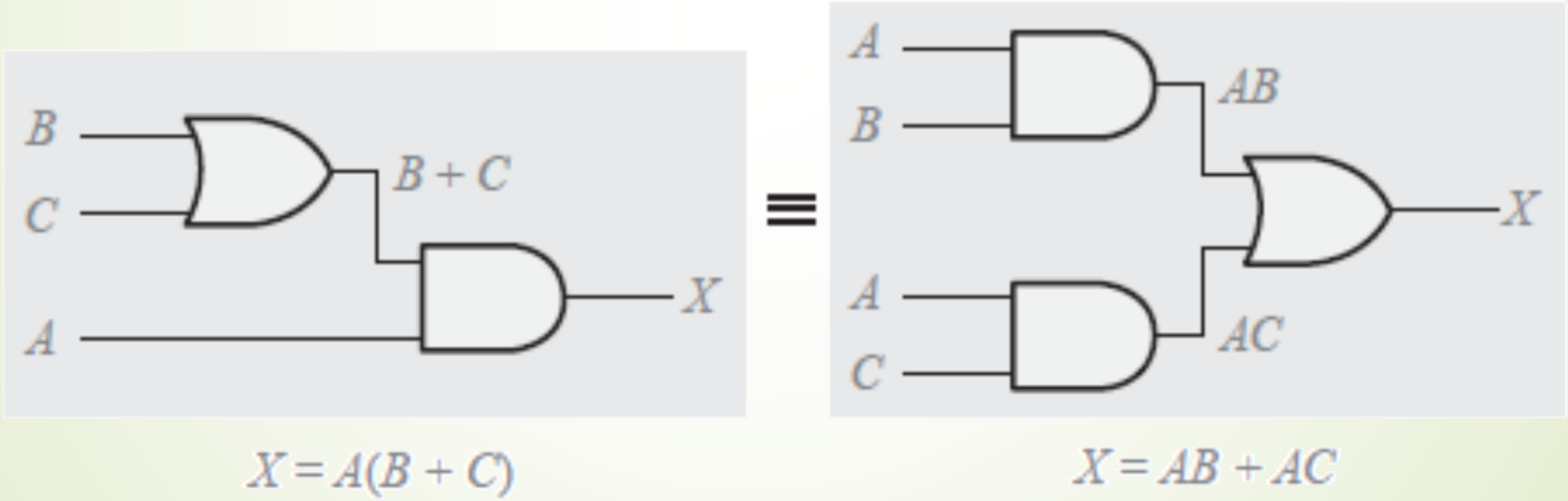


Fig. 7.- *Aplicación de la ley distributiva.*

Reglas del álgebra booleana

La Tabla 2 enumera las doce reglas básicas, muy útiles, para la manipulación y simplificación de **expresiones booleanas**. Las nueve primeras reglas las veremos en términos de su aplicación a las puertas lógicas. Las reglas 10 a 12 se obtendrán a partir de las reglas más sencillas y de las leyes anteriormente explicadas.

1. $A + 0 = A$	7. $A \cdot A = A$
2. $A + 1 = 1$	8. $A \cdot \overline{A} = 0$
3. $A \cdot 0 = 0$	9. $\overline{\overline{A}} = A$
4. $A \cdot 1 = A$	10. $A + AB = A$
5. $A + A = A$	11. $A + \overline{A}B = A + B$
6. $A + \overline{A} = 1$	12. $(A + B)(A + C) = A + BC$

A, B o C pueden representar una sola variable o una combinación de variables.

Tabla 2.- *Reglas básicas del Álgebra de Boole*

Regla 1. A + ***0*** = ***A*** Si aplicamos la operación OR a una variable cualquiera y a 0, el resultado es siempre igual a la variable. Si *A* es 1, la salida es igual a 1 y, por tanto, igual a *A*. Si *A* es 0, la salida es 0 e igualmente idéntica a *A*. Esta ley se ilustra en la Figura 8 en la que la entrada inferior está siempre a 0

$$X = A + 0 = A$$

Fig. 8.- *Regla 1, A + 0 = A.*

Regla 2. *A* + 1 = 1 Si se aplica la operación OR a una variable y a 1, el resultado es siempre igual a 1. Un 1 en una entrada de una puerta OR produce siempre un 1 en la salida, independientemente del valor de la otra entrada. Esta regla se ilustra en la Figura 9, en la que la entrada inferior está siempre a 1.

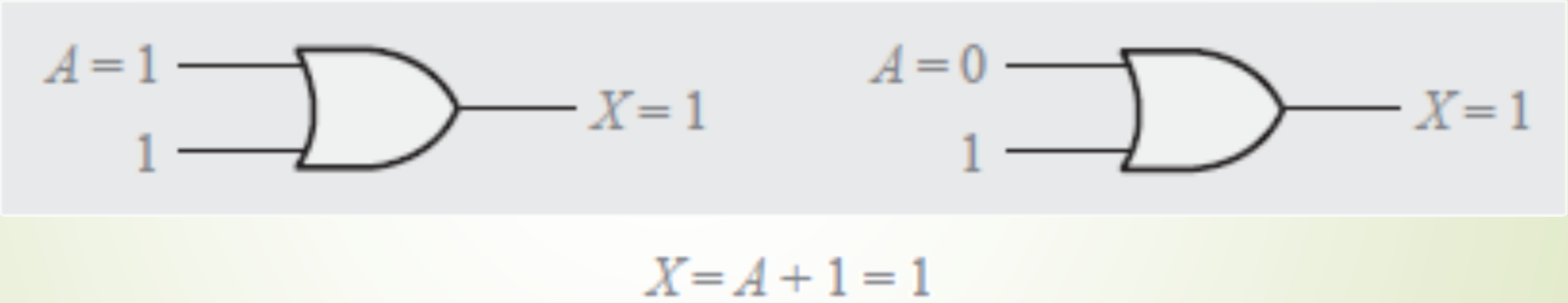
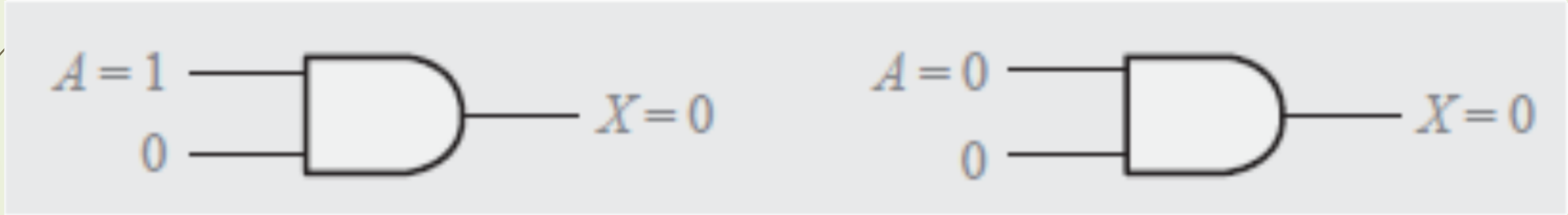


Fig. 9.- Regla 2. *A* + 1 = 1

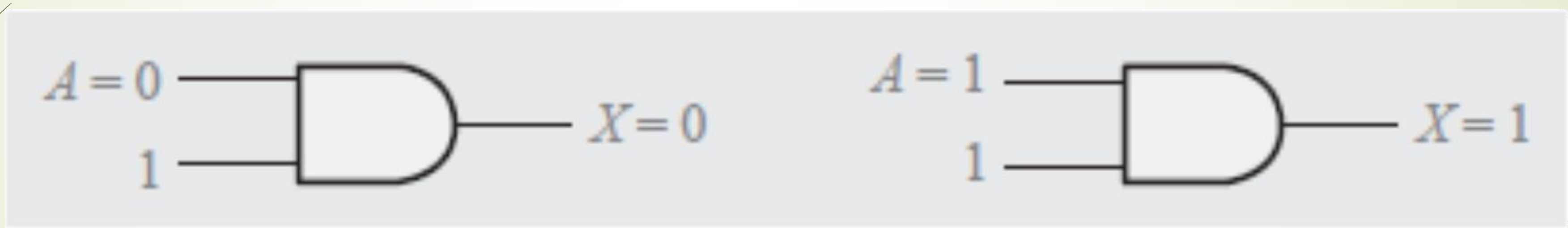
Regla 3. *A* · 0 = 0 Si se aplica la operación AND a una variable y a 0, el resultado es siempre igual a 0. Siempre que una de las entradas de una puerta AND sea 0, la salida siempre es 0, independientemente del valor de la otra entrada. Esta regla se ilustra en la Figura 10, en la que la entrada inferior está siempre a 0.



$$X = A \cdot 0 = 0$$

Fig. 10.- *Regla 3. A · 0 = 0*

Regla 4. *A* · 1 = *A* Si se aplica la operación AND a una variable y a 1, el resultado es siempre igual a la variable. Si la variable *A* es 0, la salida de la puerta AND será siempre 0, mientras que si *A* es 1, la salida será 1, dado que las dos entradas son 1. Esta regla se ilustra en la Figura 11, en la que la entrada inferior está siempre a 1.



$$X = A \cdot 1 = A$$

Fig. 11.- *Regla 4. A · 1 = A*

Regla 5. *A* + *A* = *A* Si se aplica la operación OR a una variable consigo misma, el resultado es siempre igual a la variable. Si *A* es 0, entonces 0 + 0 = 0, mientras que si *A* es 1, 1 + 1 =1. Esto se muestra en la Figura 12, en la que se aplica la misma variable a ambas entradas.

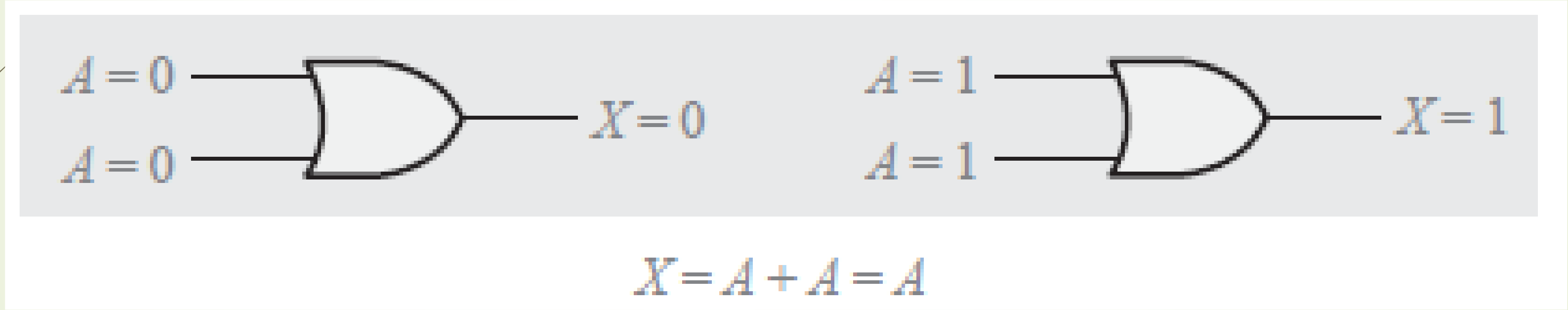


Fig. 12.- *Regla 5. A + A = A*

Regla 6. $\boldsymbol{A + \overline{A} = 1}$ Si se aplica la operación OR a una variable y a su complemento, el resultado es siempre igual a 1. si A es 0, entonces $0 + \overline{0} = 1$. Si A es 1, entonces $1 + \overline{1} = 1 + 0 = 1$. En la figura 13 podemos ver una puerta OR en la que sus entradas son una variable y su complemento.

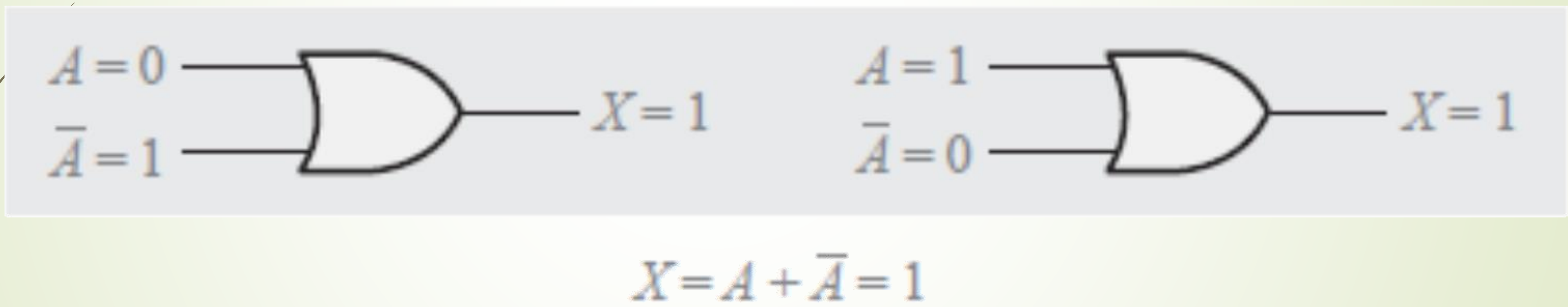


Fig. 13.- *Regla 6.* $A + \bar{A} = 1$

Regla 7. *A* · *A* = *A* Si se aplica la operación AND a una variable consigo misma, el resultado siempre es igual a la variable. Si A = 0, entonces 0 · 0 = 0, y si *A* = 1, entonces 1 · 1 = 1. Esta regla se ilustra en la Figura 14.

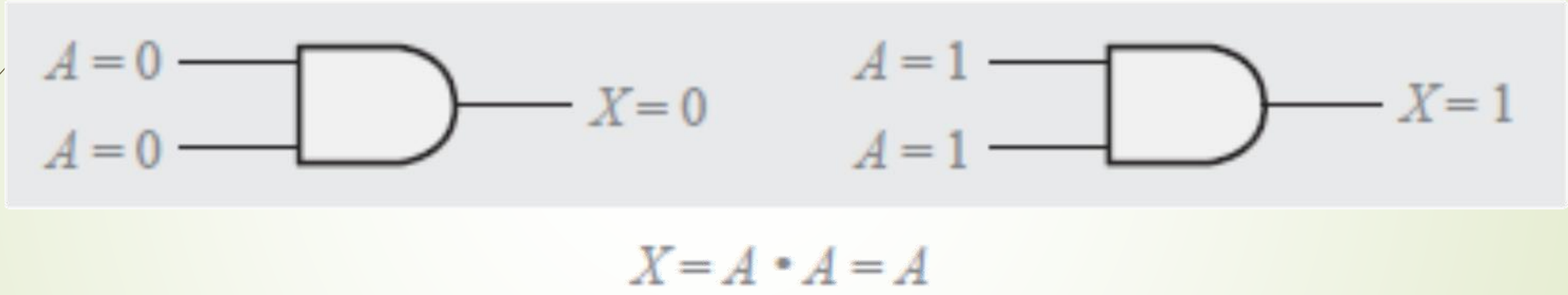
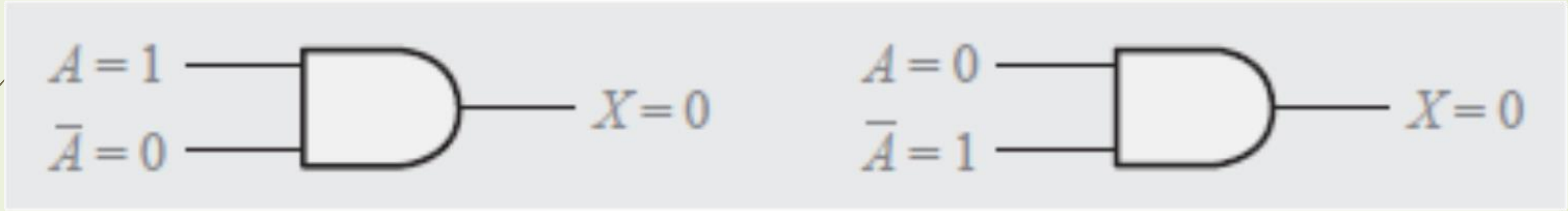


Fig. 14.- *Regla 7. A · A = A*

Regla 8. $\boldsymbol{A\,.\,\bar{A} = 0}$ Si se aplica la operación AND a una variable y a su complemento, el resultado es siempre igual a 0. Esta regla se basa en que siempre A o $\bar{A}$ será 0, y además en que cuando se aplica un 0 a una de las entradas de una puerta AND, la salida siempre es 0. Esta regla se ilustra en la Figura 15.



$$X = A \cdot \bar{A} = 0$$

Fig. 15.- *Regla 8.* $A\,.\,\bar{A} = 0$

Regla 9. $\bar{\bar{A}} = A$ El complemento del complemento de una variable es siempre la propia variable. El complemento de la variable A es $\bar{A}$ y el complemento de $\bar{A}$ será de nuevo A, que es la variable original. Esta regla se muestra en la figura 16 mediante el uso de dos inversores.

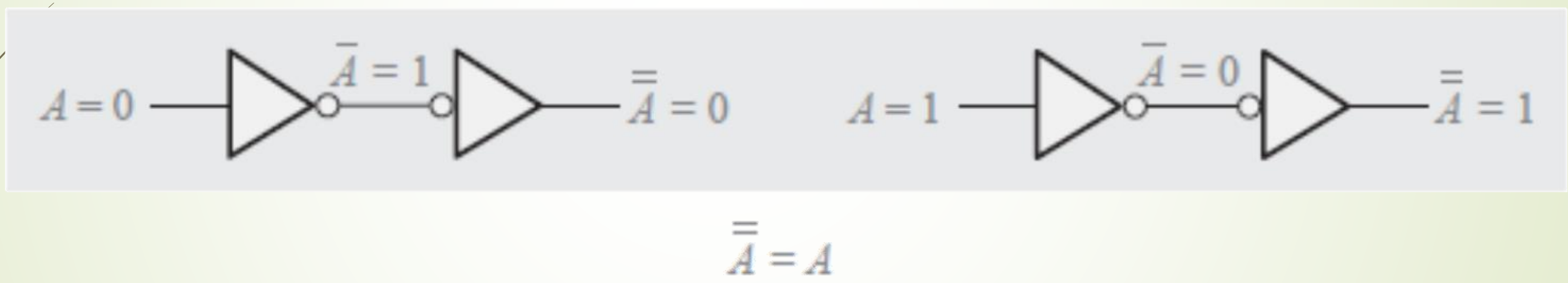


Fig. 16.- *Regla 9.* $\bar{\bar{A}} = A$

Regla 10. $\boldsymbol{A + AB = A}$ Esta regla se puede obtener aplicando la ley distributiva y las reglas 2 y 4, de la siguiente forma:

A + AB	= A(1 + B)	Sacar factor común (ley distributiva)
	= A . 1	Regla 2: (1+B) = 1
	= A	Regla 4: A . 1 = A

La demostración se muestra en la tabla 3, la cual incluye la tabla de verdad y la simplificación del circuito lógico resultante.

A	B	AB	$A+AB$
0	0	0	0
0	1	0	0
1	0	0	1
1	1	1	1

igual

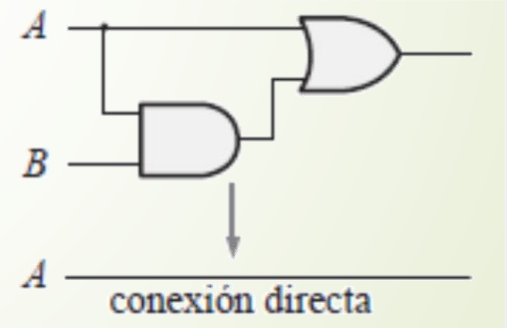


Tabla 3.- Regla 10: A + AB = A

Regla 11. $\boldsymbol{A + \bar{A}B = A + B}$ Esta regla puede demostrarse de la siguiente forma:

$A + \bar{A}B = (A + AB) + \bar{A}B$	Regla 10: A = A + AB
$= (AA + AB) + \bar{A}B$	Regla 7: A = AA
$= AA + AB + A\bar{A} + \bar{A}B$	Regla 8: sumar $A\bar{A} = 0$
$= (A + \bar{A})(\mathrm{A} + \mathrm{B})$	Sacar factor común
$= 1 . (A + B)$	Regla 6: $A + \bar{A} = 1$
$= A + B$	Regla 4: eliminar el 1

La demostración se muestra en la tabla 4, la cual incluye la tabla de verdad y la simplificación del circuito lógico resultante.

A	B	$\bar{A}B$	$A+\bar{A}B$	$A+B$
0	0	0	0	0
0	1	1	1	1
1	0	0	1	1
1	1	0	1	1

igual

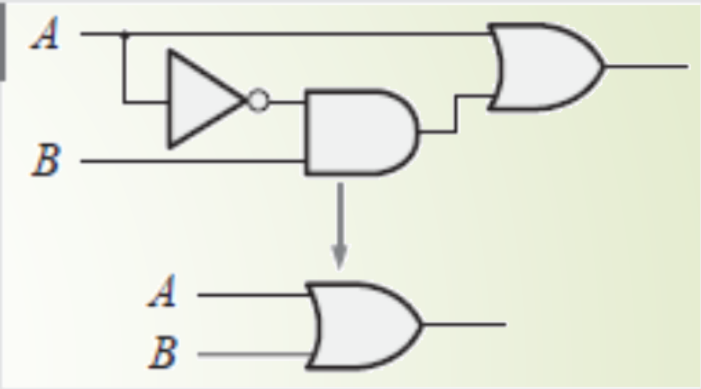


Tabla 4.- $A + \bar{A}B = A + B$

Regla 12. $(\boldsymbol{A}+\boldsymbol{B})(\boldsymbol{A}+\boldsymbol{C})=\boldsymbol{A}+\boldsymbol{BC}$ Esta regla puede demostrarse de la siguiente forma:

$(\boldsymbol{A}+\boldsymbol{B})(\boldsymbol{A}+\boldsymbol{C})=AA+AC+AB+BC$	Ley distributiva
$=A+AC+AB+BC$	Regla 7: A = AA
$=A(1+C)+\ AB+BC$	Sacar factor común
$=A.1+AB+BC$	Regla 2: 1+C=1
$=A(1+B)+BC$	Sacar factor común
$=A.1+BC$	Regla 2: 1+B=1
$=A+BC$	Regla 4: A.1=A

La demostración se muestra en la tabla 5, la cual incluye la tabla de verdad y la simplificación del circuito lógico resultante.

A	B	C	$A+B$	$A+C$	$(A+B)(A+C)$	BC	$A+BC$
0	0	0	0	0	0	0	0
0	0	1	0	1	0	0	0
0	1	0	1	0	0	0	0
0	1	1	1	1	1	1	1
1	0	0	1	1	1	0	1
1	0	1	1	1	1	0	1
1	1	0	1	1	1	0	1
1	1	1	1	1	1	1	1

igual

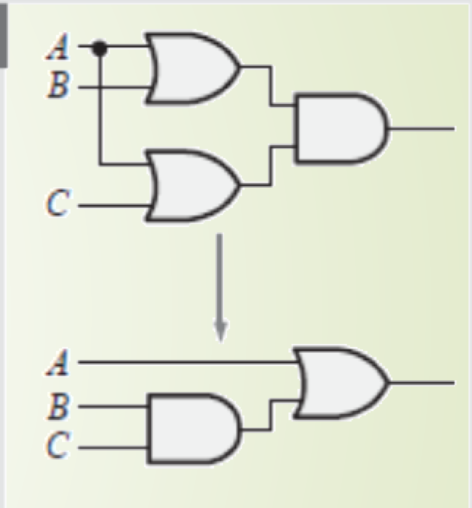


Tabla 5.- $(A+B)(A+C) = A+BC$

TEOREMAS DE DeMORGAN

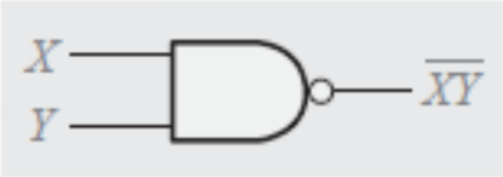
DeMorgan, matemático que conoció a Boole, propuso dos teoremas que constituyen una parte muy importante del álgebra de Boole. En términos prácticos, los teoremas de DeMorgan proporcionan una verificación matemática de la equivalencia entre las puertas NAND y negativa-OR, y las puertas NOR y negativa-AND

1. *El complemento de un producto de variables es igual a la suma de los complementos de las variables.*

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

2. *El complemento de una suma de variables es igual al producto de los complementos de las variables.*

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

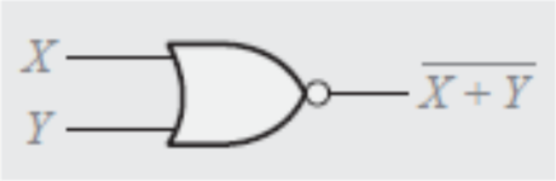


NAND

≡

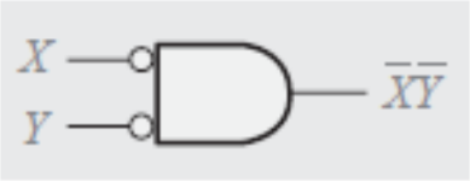
Negativa-OR

Entradas		Salida	
X	Y	$\overline{XY}$	$\overline{X}+\overline{Y}$
0	0	1	1
0	1	1	1
1	0	1	1
1	1	0	0



NOR

≡



Negativa-AND

Entradas		Salida	
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

Equivalencias de las puertas lógicas y tablas de verdad que ilustran los teoremas de DeMorgan. Observe la igualdad entre las dos columnas de salida de cada tabla. Esto demuestra que las puertas equivalentes realizan la misma función lógica

ANÁLISIS BOOLEANO DE LOS CIRCUITOS LÓGICOS

El álgebra de Boole proporciona una manera concisa de expresar el funcionamiento de un circuito lógico formado por una combinación de puertas lógicas, de tal forma que la salida puede determinarse por la combinación de los valores de entrada

Expresión booleana de un circuito lógico

Para obtener la expresión booleana de un determinado circuito lógico, la manera de proceder consiste en comenzar con las entradas situadas más a la izquierda e ir avanzando hasta las líneas de salida, escribiendo la expresión para cada puerta. Para el circuito ejemplo de la Figura 17, su expresión booleana se determina de la siguiente manera

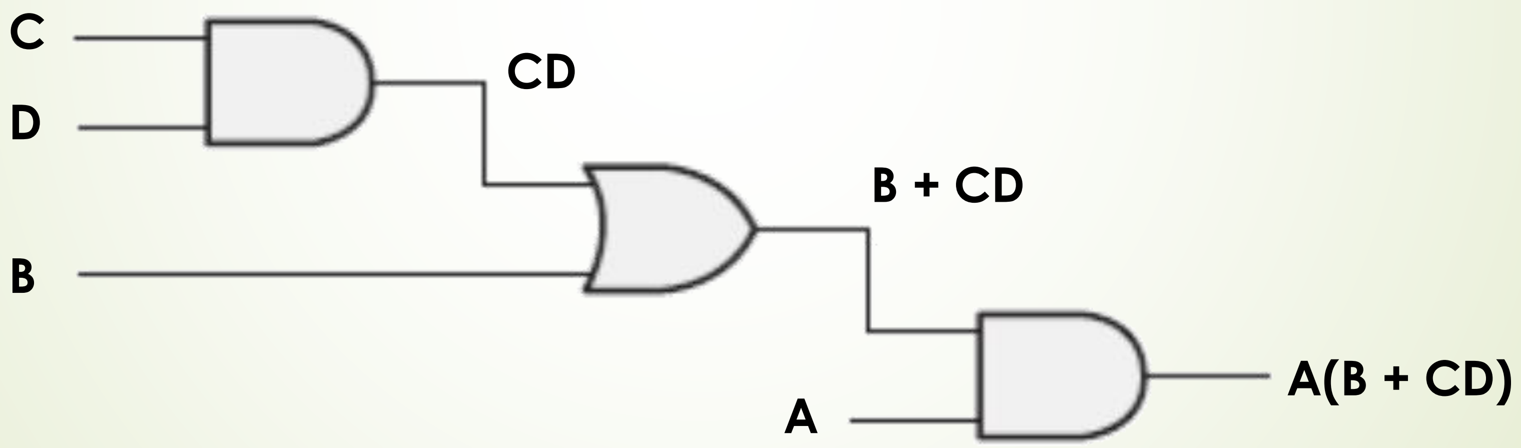


Fig. 17.- Circuito lógico que muestra el desarrollo de la expresión booleana para la salida

Construcción de una tabla de verdad para un circuito lógico

Una vez que se ha determinado la expresión booleana de un circuito dado, puede desarrollarse una tabla de verdad que represente la salida del circuito lógico para todos los valores posibles de las variables de entrada. El procedimiento requiere que se evalúe la expresión booleana para todas las posibles combinaciones de valores de las variables de entrada. En el caso del circuito de la Figura 17, existen cuatro variables de entrada (A, B, C y D) y, por tanto, hay dieciséis ($2^4 = 16$) posibles combinaciones de valores.

Evaluación de la expresión. Para evaluar la expresión $A(B + CD)$, en primer lugar hallamos los valores de las variables que hacen que la expresión sea igual a 1, utilizando las reglas de la suma y la multiplicación booleanas. En este caso, la expresión es igual a 1 sólo si $A = 1$ y $B + CD = 1$, ya que:

$$A(B + CD) = 1 \cdot 1 = 1$$

Entradas				Salida
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	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Tabla 6.- *Representación de los resultados en una tabla de verdad de la expresión booleana*

A(B + CD)

SIMPLIFICACIÓN MEDIANTE EL ÁLGEBRA DE BOOLE

Muchas veces, a la hora de aplicar el álgebra booleana, hay que reducir una expresión a su forma más simple o cambiarla a una forma más conveniente para conseguir una implementación más eficiente.

Simplificar la siguiente expresión utilizando técnicas del álgebra de Boole:

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

Respuesta: $B + AC$

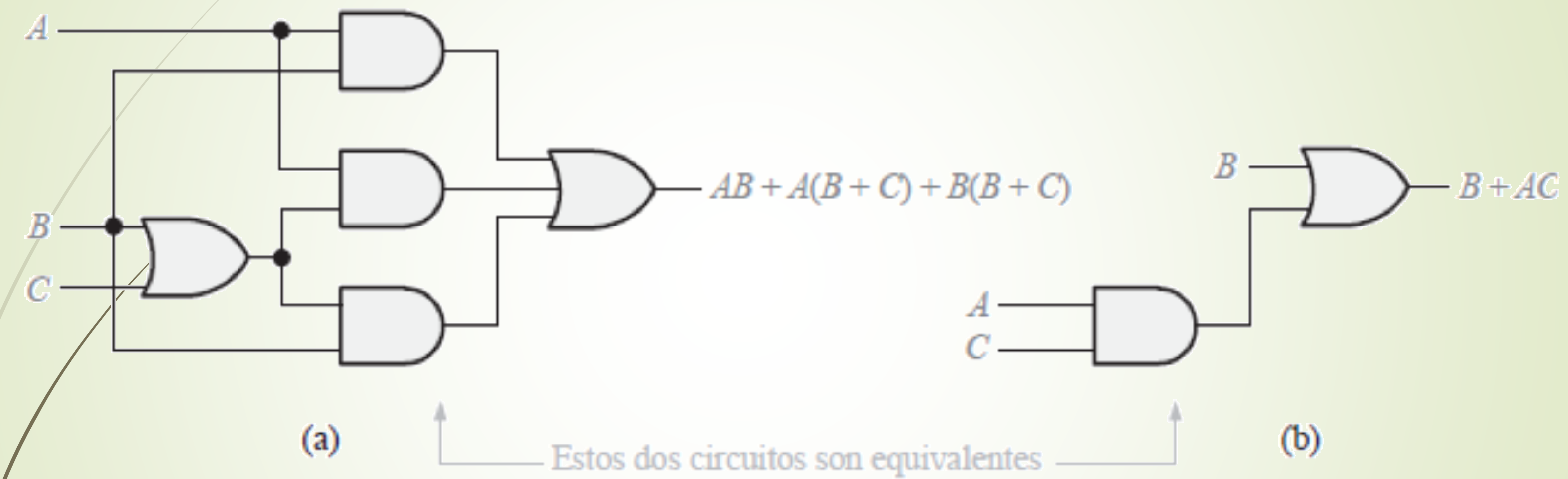


Fig. 18.- *Circuitos equivalentes del ejercicio anterior*

FORMAS ESTÁNDAR DE LAS EXPRESIONES BOOLEANAS

Todas las expresiones booleanas, independientemente de su forma, pueden convertirse en cualquiera de las dos formas estándar: ***suma de productos*** o ***producto de sumas***. La estandarización posibilita que la evaluación, simplificación e implementación de las expresiones booleanas sea mucho más sistemática y sencilla.

Suma de productos

Cuando dos o más productos se suman mediante la adición booleana, la expresión resultante se denomina ***suma de productos*** (SOP, *Sum Of Products*).
Ejemplo:

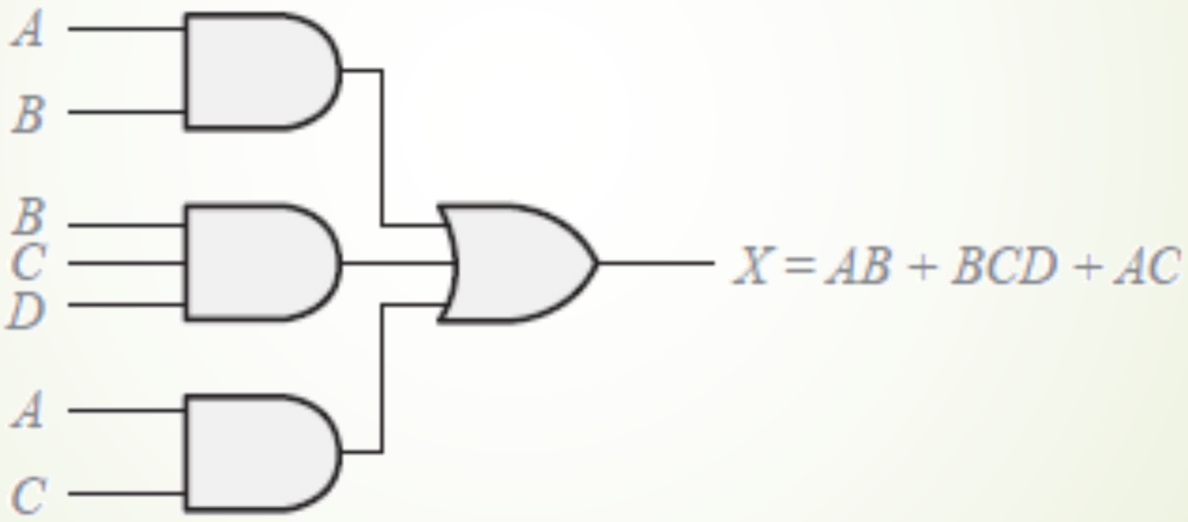


Fig. 19.- Implementación de la suma de productos AB + BCD + AC

Producto de sumas

Cuando dos o más términos suma se multiplican, la expresión resultante es un ***producto de sumas*** (POS, *Product Of Sums*).
Ejemplo:

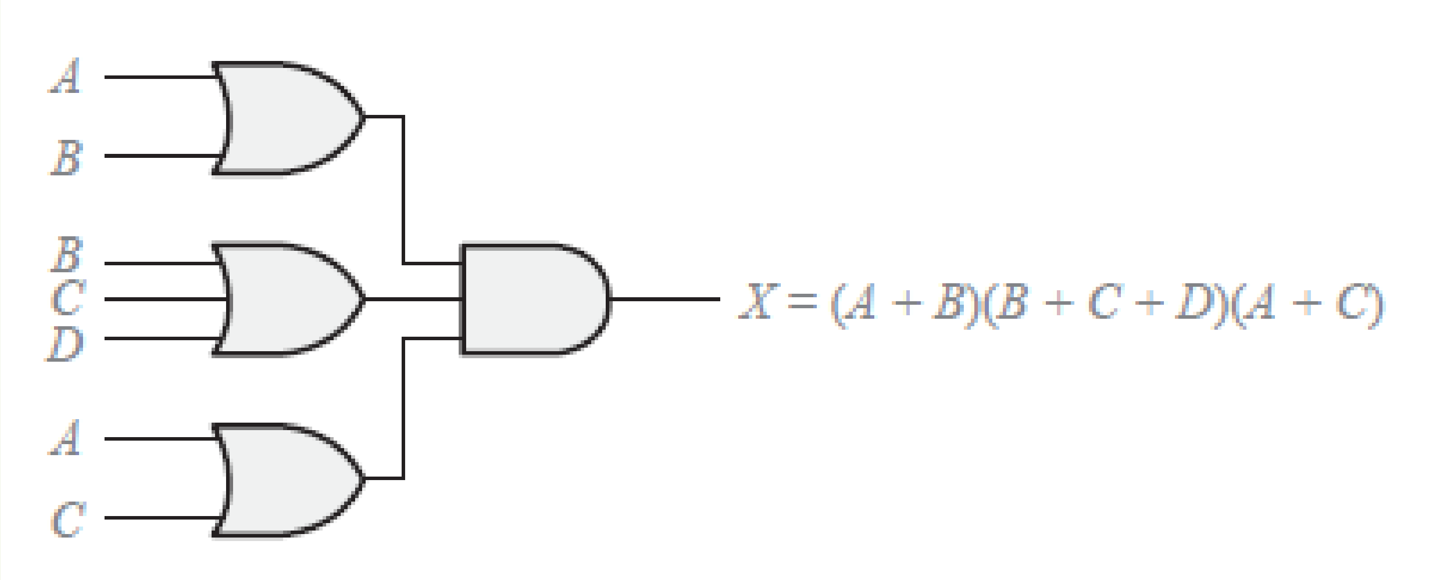


Fig. 20.- Implementación del producto de sumas (A + B)(B + C + D)(A + C)

EXPRESIONES BOOLEANAS Y TABLAS DE VERDAD

Todas las expresiones booleanas pueden convertirse fácilmente en tablas de verdad utilizando los valores binarios de cada término de la expresión. La tabla de verdad es una forma muy común, en un formato muy conciso, de expresar el funcionamiento lógico de un circuito. Además, las expresiones ***suma de productos*** y ***producto de sumas*** pueden determinarse mediante las tablas de verdad. Las tablas de verdad pueden encontrarse en las hojas de especificaciones y en otros textos relativos al funcionamiento de los circuitos y sistemas digitales.

Conversión de una suma de productos a tabla de verdad

Una tabla de verdad es sencillamente la lista de las posibles combinaciones de valores de las variables de entrada y sus correspondientes valores de salida (1 o 0). La conversión de una suma de productos se ilustra en el siguiente ejemplo:

- ✓ Desarrollar una tabla de verdad para la expresión suma de productos estándar
  $\bar{A}\bar{B}\bar{C} + A\bar{B}\bar{C} + ABC$

Entradas			Salida	
A	B	C	X	Término producto
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

Conversión de un producto de sumas a tablas de verdad

Un producto de sumas es igual a 0 sólo si y sólo si al menos uno de los términos suma es igual a 0. Para construir la tabla de verdad de un producto de sumas, basta con enumerar todas las posibles combinaciones de valores binarios de las variables del mismo modo que se hace para una suma de productos.

- ✓ Desarrollar una tabla de verdad para la expresión producto de sumas estándar siguiente:

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

Entradas			Salida	
A	**B**	**C**	**X**	**Término suma**
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	

MAPAS DE KARNAUGH

Un mapa de Karnaugh proporciona un método sistemático de simplificación de expresiones booleanas y, si se aplica adecuadamente, genera las expresiones suma de productos y producto de sumas más simples posibles, conocidas como expresiones mínimas.