Algebra de Boole y simplificación de funciones lógicas

Contenido

Expresiones y operaciones Booleanas

- **Variable**: Símbolo que representa magnitudes lógicas. (0 ó 1). A
- **Complemento**: In<u>ve</u>rso de la **variable**. Se representa A ó A'
- **Literal**: Es una variable o el complemento de una variable.

Expresiones y operaciones Booleanas

- Suma booleana ≡ OR

  0 + 0 = 0

  0 + 1 = 1

  1 + 0 = 1

  1 + 1 = 1

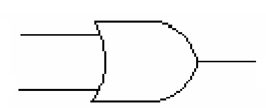
- Multiplicación booleana ≡ AND

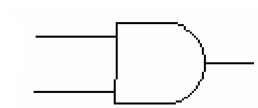
  0 • 0 = 0

  0 • 1 = 0

  1 • 0 = 0

  1 • 1 = 1





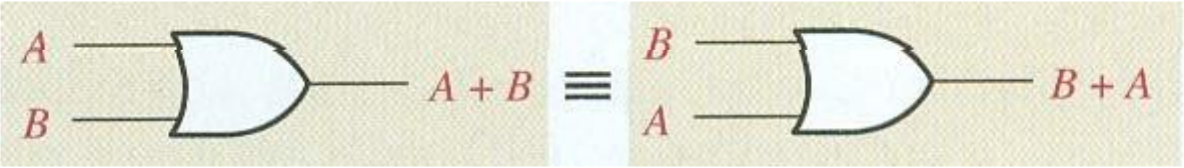
Propiedades del Algebra de Boole

- Conmutativa
- Asociativa
- Distributiva

Propiedades del Algebra de Boole

- Propiedad conmutativa de la suma:

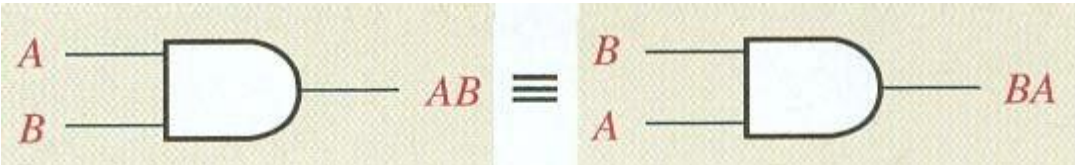
$A + B = B + A$



Propiedades del Algebra de Boole

- Propiedad conmutativa del producto:

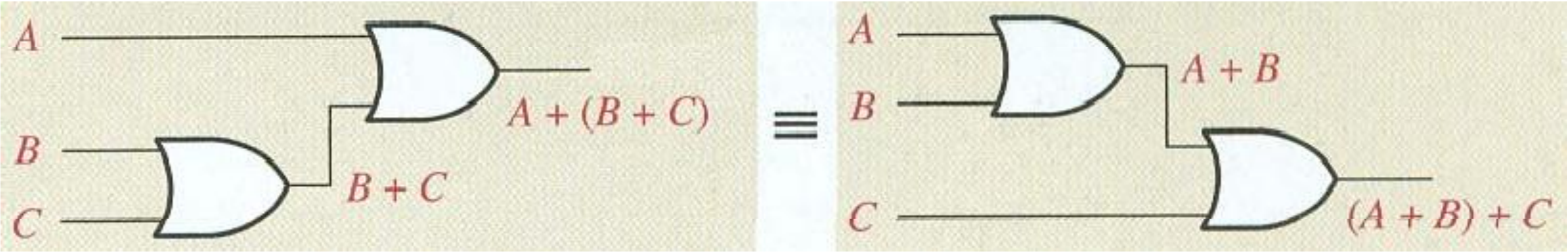
A • B = B • A



Propiedades del Algebra de Boole

- Asociativa de la suma:

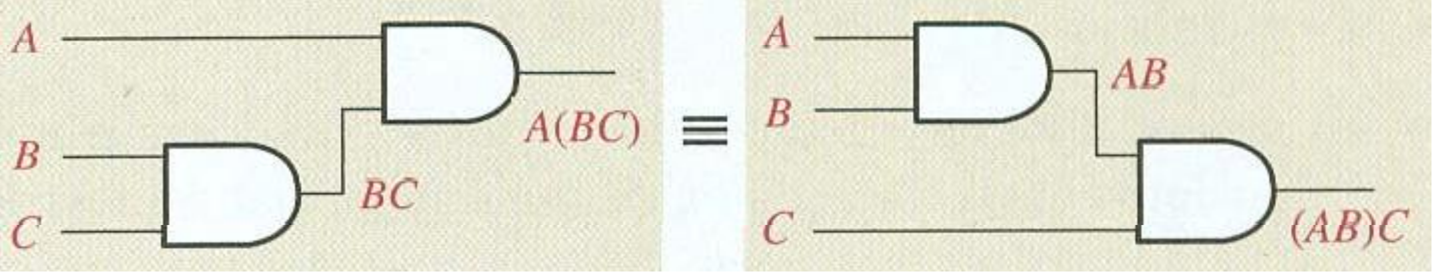
A + (B + C) = (A + B) + C



Propiedades del Algebra de Boole

- Asociativa del producto:

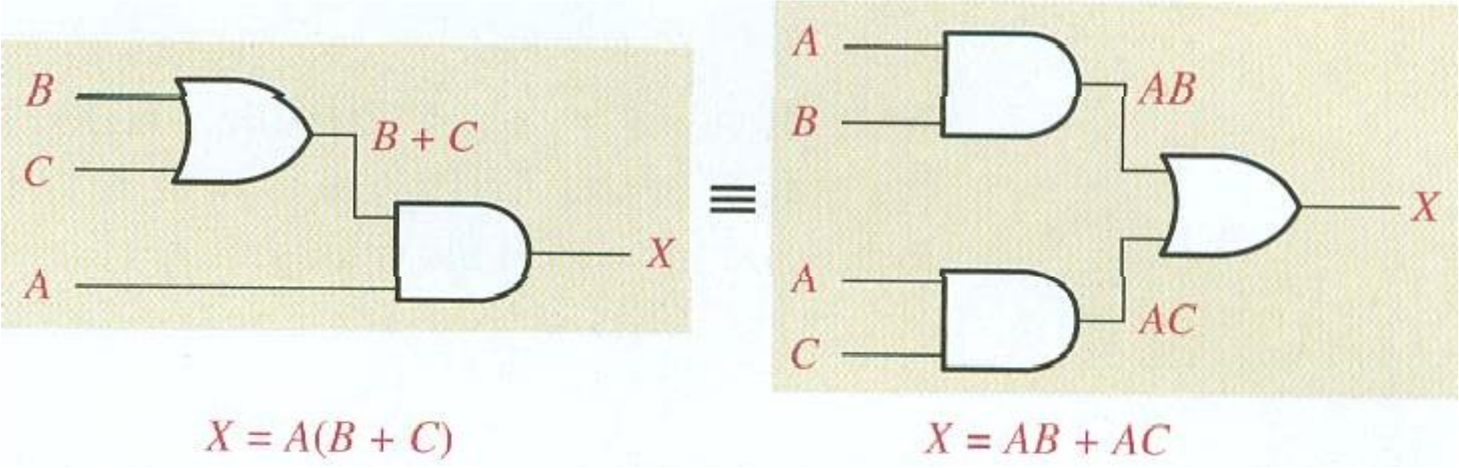
$$A \bullet (B \bullet C) = (A \bullet B) \bullet C$$



Propiedades del Algebra de Boole

- Distributiva:

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

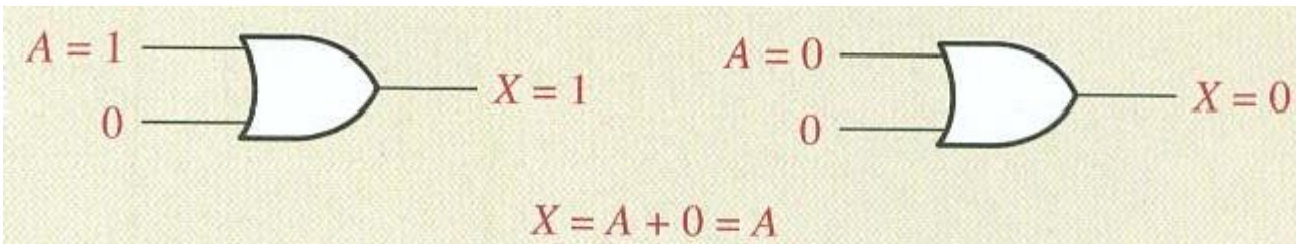


Reglas del Algebra de Boole

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

Reglas del Algebra de Boole

- Regla 1

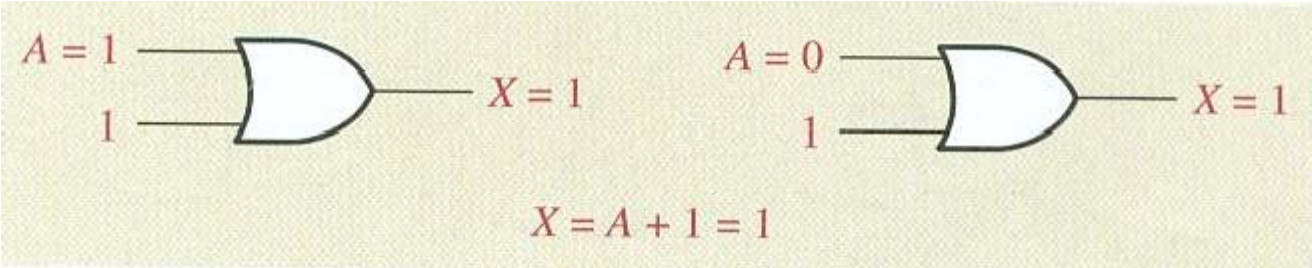


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Reglas del Algebra de Boole

- Regla 2

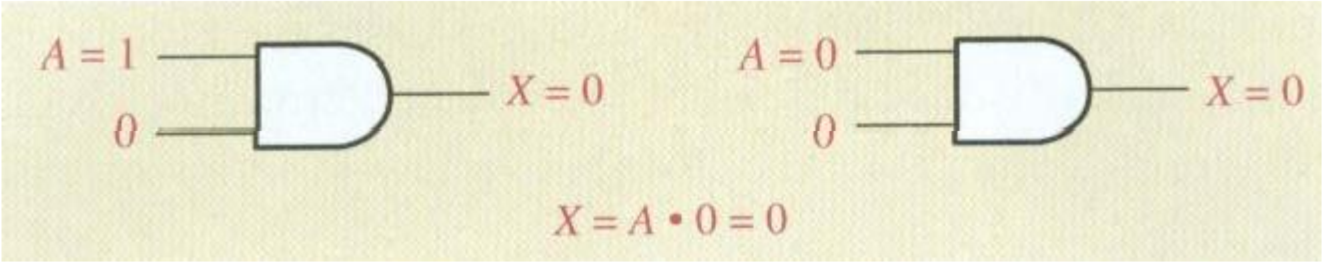


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Reglas del Algebra de Boole

- Regla 3

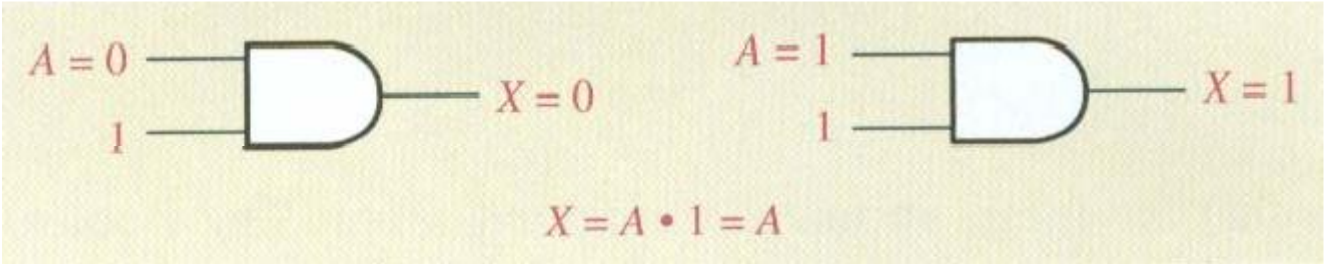


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Reglas del Algebra de Boole

- Regla 4

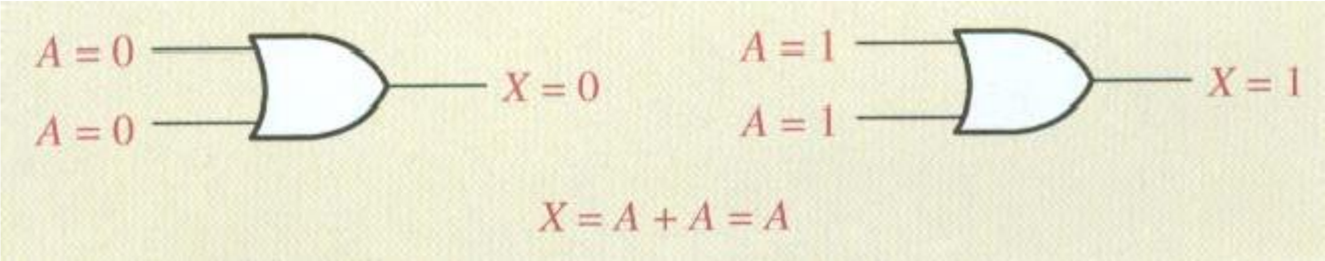


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Reglas del Algebra de Boole

- Regla 5

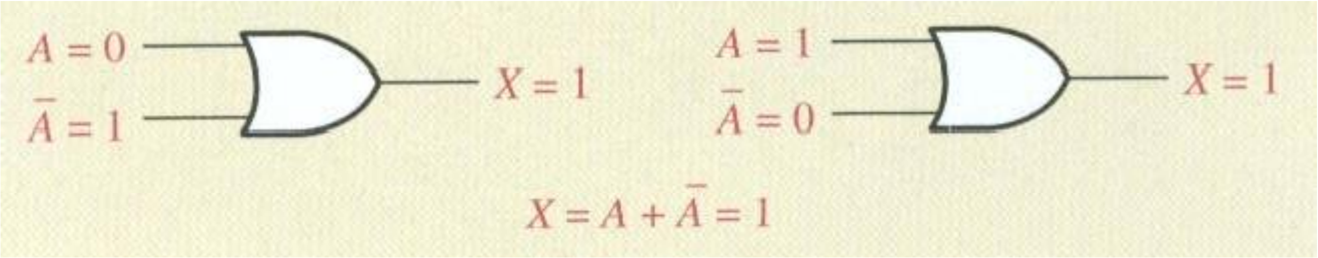


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Reglas del Algebra de Boole

- Regla 6

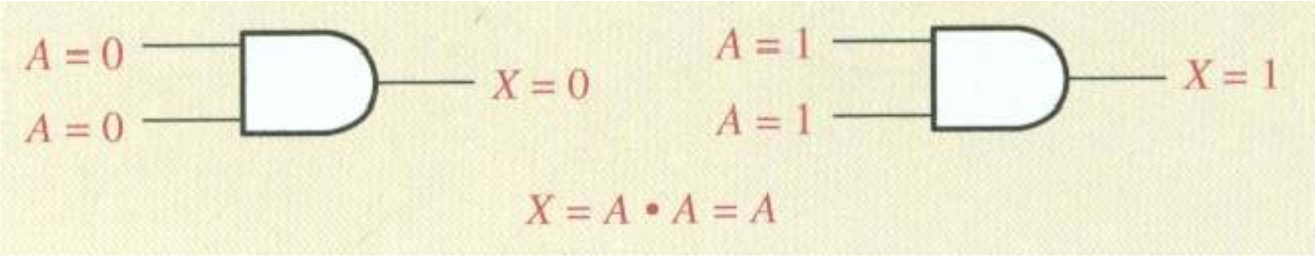


A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Reglas del Algebra de Boole

- Regla 7

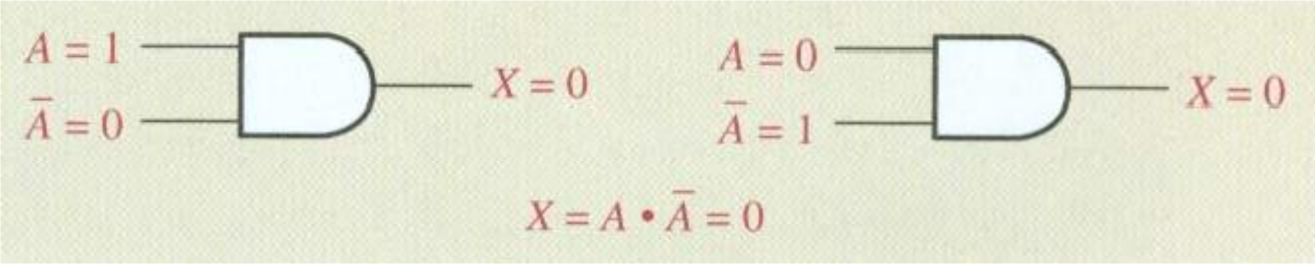


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Reglas del Algebra de Boole

- Regla 8

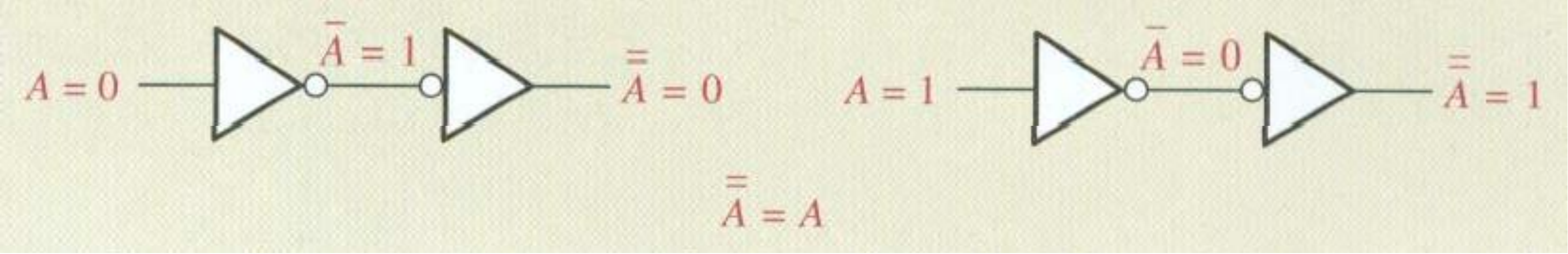


A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

Reglas del Algebra de Boole

- Regla 9

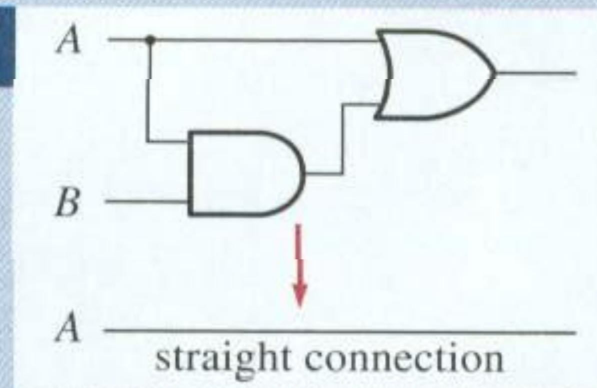


Reglas del Algebra de Boole

- Regla 10: A + AB = A

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

equal



A + AB = A (1+B) Ley distributiva

= A • 1 Regla 2: (1+B)=1

= A Regla 4: A •1=A

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

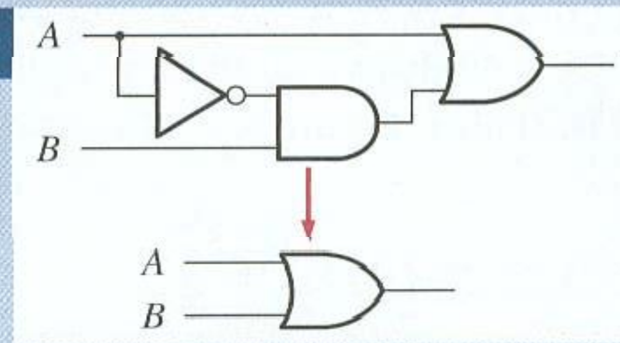
OR Truth Table

Reglas del Algebra de Boole

- Regla 11: $A + \overline{A}B = A + B$

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

equal



$$\begin{aligned}
A + \overline{A}B &= (A + AB) + \overline{A}B && \text{R10}: A = A + AB \\
&= (AA + AB) + \overline{A}B && \text{R7}: A = A.A \\
&= AA + AB + \overline{A}A + \overline{A}B && \text{R8}: \text{Sumar } A.\overline{A} = 0 \\
&= (A + \overline{A})(A + B) && \text{Factor común} \\
&= 1.(A + B) && \text{R6}: A + \overline{A} = 1 \\
&= A + B && \text{R4}: A.1 = A
\end{aligned}$$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

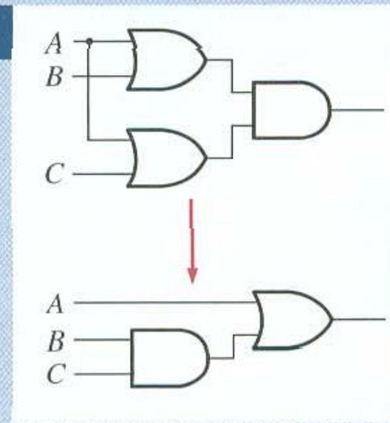
OR Truth Table

Reglas del Algebra de Boole

- Regla 12: (A + B)(A + C) = A + BC

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

equal



$$
\begin{aligned}
(A+B).(A+C) &= AA + AC + AB + BC && \text{distributiva} \\
&= A + AC + AB + BC && R7: A.A = A \\
&= A(1+C) + AB + BC && \text{factor común} \\
&= A.1 + AB + BC && R2: 1 + C = 1 \\
&= A(1+B) + BC && \text{factor común} \\
&= A.1 + BC && R2: 1 + B = 1 \\
&= A + BC && R4: A.1 = A
\end{aligned}
$$

A	B	X
0	0	0
0	1	0
1	0	0
1	1	1

AND Truth Table

A	B	X
0	0	0
0	1	1
1	0	1
1	1	1

OR Truth Table

Teoremas de DeMorgan

- Teorema 1

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

- Teorema 2

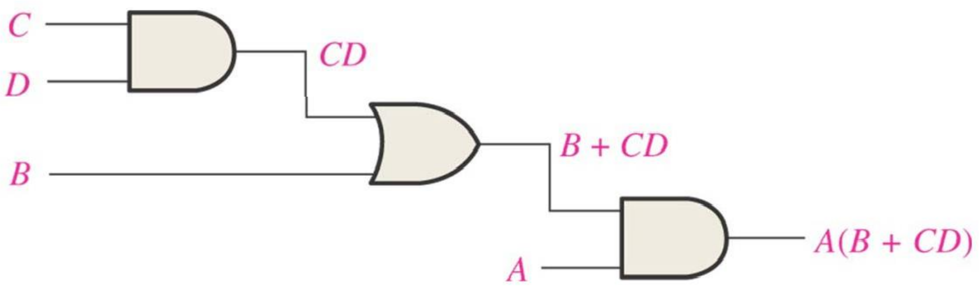
$$\overline{X+Y} = \overline{X}\,\overline{Y}$$

Recuerda:

**"Parte la barra,
cambia la operación"**

Analisis booleano de Circuitos

Expresion booleana y tabla de verdad de un circuito lógico

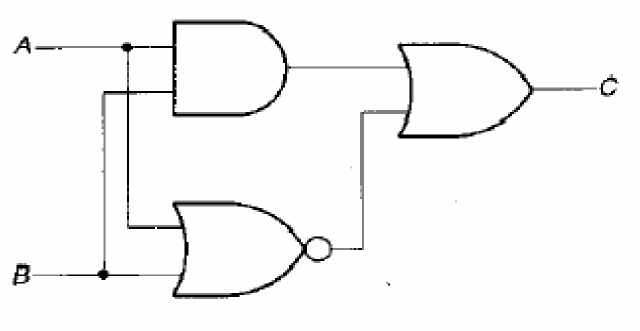


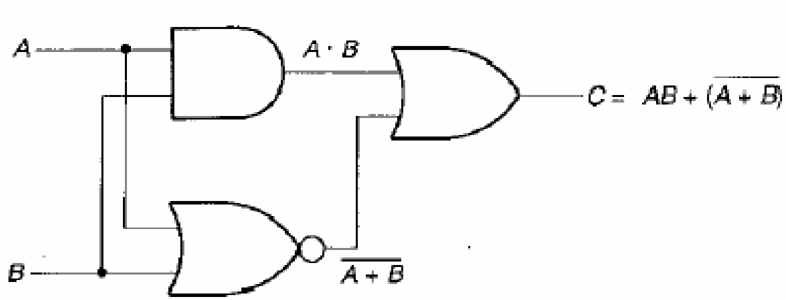
A	B	C	D	→ A(B+CD)
0	0	0	0	0
0	0	0	1	0
0	0	1	0	0
….				….
1	0	1	0	0
1	0	1	1	1
1	1	0	0	1
1	1	0	1	1
1	1	1	0	1
1	1	1	1	1

Ejemplo

<u>Ejemplo: Extracción de la expresión booleana de un sistema a partir de su diagrama lógico</u>

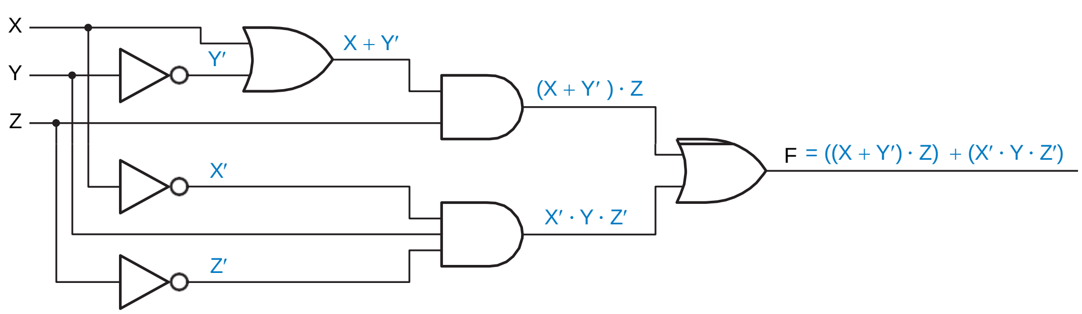
A partir del siguiente circuito lógico se nos pide que obtengamos su expresión booleana equivalente.





Ejemplo: Construcción de la Tabla de Verdad a partir de la expresión booleana

- Un circuito lógico puede describirse mediante una tabla de verdad.
- Evaluar la expresión booleana para todas las posibles combinaciones de valores de las variables de entrada

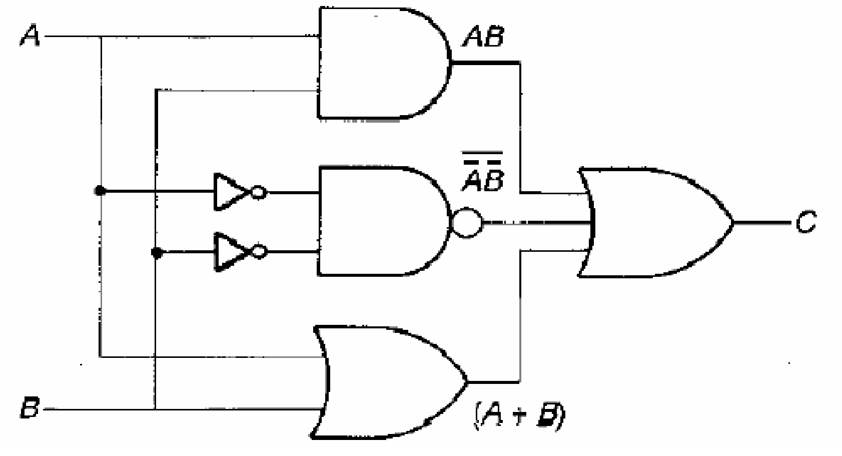


Row	X	Y	Z	F
0	0	0	0	0
1	0	0	1	1
2	0	1	0	1
3	0	1	1	0
4	1	0	0	0
5	1	0	1	1
6	1	1	0	0
7	1	1	1	1

Ejemplo

A partir de la siguiente expresión Booleana se nos pide que obtengamos su diagrama lógico equivalente.

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



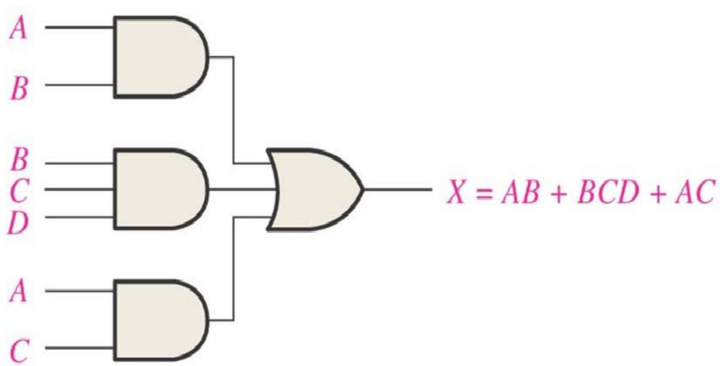
Formas estándar de las expresiones booleanas

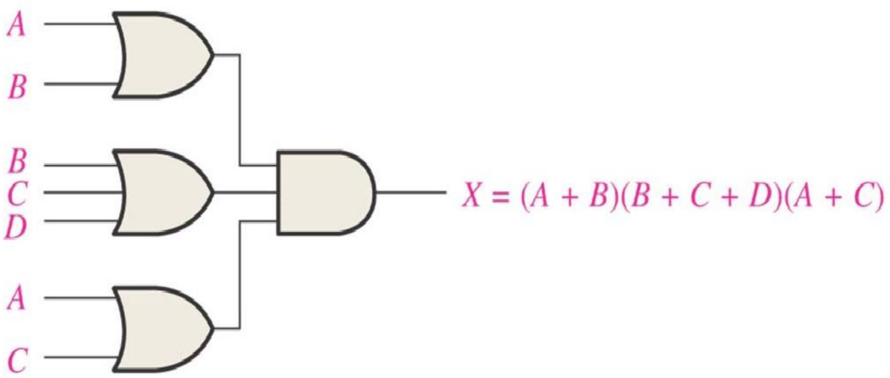
- Suma de productos (SOP)

  Ejemplo: X = AB + BCD + AC

Producto de sumas (POS)

Ejemplo: X = (A+B)(B+C+D)(A+C)





- Para cualquier expresión lógica existe una forma estándar SOP y POS **equivalente**

- Se denominan **formas canónica o estándar** a las SOP y POS en las que todas las variables aparecen en cada uno de los terminos:

Ejemplo:

$$\overline{A}\overline{B}C\overline{D} + A\overline{B}CD + AB\overline{C}\overline{D}$$

Conversión SOPs y POS - Tablas de Verdad

- Suma de Productos

A	B	C	X	Producto
0	0	0	0	
0	0	1	1	A’B’C
0	1	0	0	
0	1	1	0	
1	0	0	1	AB’C’
1	0	1	0	
1	1	0	0	
1	1	1	1	ABC

X = A’B’C + AB’C’ + ABC

- Producto de sumas

A	B	C	X	Suma
0	0	0	0	(A+B+C)
0	0	1	1	
0	1	0	0	(A+B’+C)
0	1	1	0	(A+B’+C’)
1	0	0	1	
1	0	1	0	(A’+B+C’)
1	1	0	0	(A’+B’+C)
1	1	1	1	

X = (A+B+C) (A+B’+C) (A+B’+C’)
(A’+B+C’) (A’+B’+C)

Forma estándar o canónica

- Cualquier función Booleana se puede expresar como suma de miniterminos (minterms) o como producto de maxiterminos (maxterms) y a estas formas se les dice que están en forma **estándar o canónica (el conjunto completo de variables del dominio está representado en cada término)**.

A	B	C	D	minterms
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

F=$\Sigma_{A,B,C}$ (1, 4, 7) = A'B'C + AB'C' + ABC

A	B	C	D	Maxiterms
0	0	0	0	A + B+ C
0	0	1	1	
0	1	0	0	A + B'+ C
0	1	1	0	A +B'+ C'
1	0	0	1	
1	0	1	0	A'+ B + C'
1	1	0	0	A'+ B' + C
1	1	1	1	

F= $\Pi_{A,B,C}$ (0, 2, 3, 5, 6) = (A+B+C)(A+B'+C)(A+B'+C')(A'+B+C')(A'+B'+C)

Forma canónica y normalizada

- **Se llama término canónico de una función lógica a todo producto o suma de literales en los cuales aparecen todas la variables en su forma directa o complementada.**
- **Los términos canónicos producto reciben el nombre de "minitérminos"**
- **Los términos canónicos suma reciben el nombre de "maxitérminos"**
- **Una función de BOOLE está en forma canónica cuando se expresa como suma de minitérminos o producto de maxotérminos.**
- **Dos funciones lógicas son equivalentes si, y solo si, sus formas canónicas son idénticas.**
- **La expresión algebraica en suma de productos o productos de sumas en la que no todos los términos son canónicos recibe el nombre de normalizada**

Ejemplos:

Forma normalizada

$$F_1(X, Y, Z) = XY + X'YZ'$$

$$F_2(X, Y, Z) = (X' + Y' + Z)(X + Y' + Z)(X + Y + Z)$$

Forma canónica

Forma canónica de la suma de productos

- La metodología empleada en la **transformación** de una suma de productos a su forma canónica se basa en la regla 6, que establece que una variable sumada con su complemento es siempre igual a 1; A + A' = 1. Los pasos son los siguientes:
  - 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 (regla 6).
  - 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 **ABC' + BC + A'** a su forma canónica.
    - 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 BC**
    - BC = BC ·(A+A') = ABC + A'BC
  - **Término A’**
    - A' = A'(C+C') = A'C+A'C' ; la expresión aún no tiene el formato canónico, entonces multiplicamos cada término por (B+B')
      A'C(B+B') +A'C'(B+B') = A'BC + A'B'C + A'BC' + A'B'C'

ABC' + BC + A' = ABC + A'BC + A'BC + A'B'C + A'BC' + A'B'C‘

Forma canónica del producto de sumas

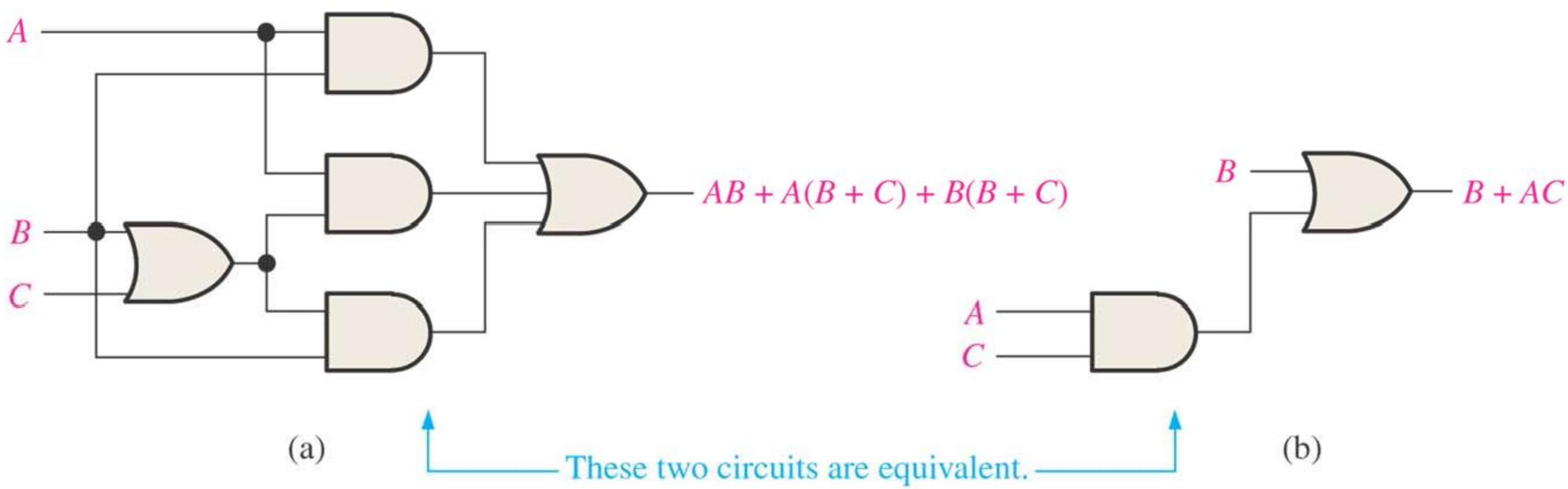
- La metodología empleada en la **transformaciones** de un producto de sumas a su forma canónica se basa en la regla 8, que establece que una variable multiplicada por su complemento es siempre igual a 0; AA' = 0. Los pasos son los siguientes:
  - Los términos suma que no contengan la(s) variable(s) del dominio, sumarlos un término formado por dicha variable y su complemento según regla 8.
  - Aplicar la regla 12: A + BC = (A+B)(A+C)
  - Repetir el paso 1 para todos los términos de la expresión que no contengan todas las variables (o sus complementos) del dominio.
- Ejemplo
  - Convertir la expresión booleana **(A+B’+C)(B’+C+D’)(A+B’+C+D’)** a su forma canónica.
  - **Término A+B’+C**
    - A+B’+C = A+B’+C+DD’ = (A+B’+C+D)(A+B’+C+D’)
  - **Término B’+C+D’**
    - B’+C+D’ = B’+C+D’+AA’ =(A+ B’+C+D’)(A’+ B’+C+D’)

(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’)

Simplificación mediante algebra de Boole

AB + A(B+C) + B(B+C)
AB + AB + AC + BB + BC
AB + AC + B + BC
AB + AC + B
B + AC

La simplificación consiste en implementar una función con el menor número de puertas posible



Mapas de Karnaugh

- Proporcionan un Método sistemático de minimización de **expresiones** booleanas
- Adecuadamente aplicado proporciona expresiones **mínimas** SOP o POS
- Es una forma de representación equivalente a la tabla de **verdad**
- Es la “receta” que emplearemos habitualmente

Método de trabajo Mapas de Karnaugh

- Proporciona un **método sistemático de simplificación** de sentencias booleanas generando expresiones mínimas (‘receta de simplificación’)

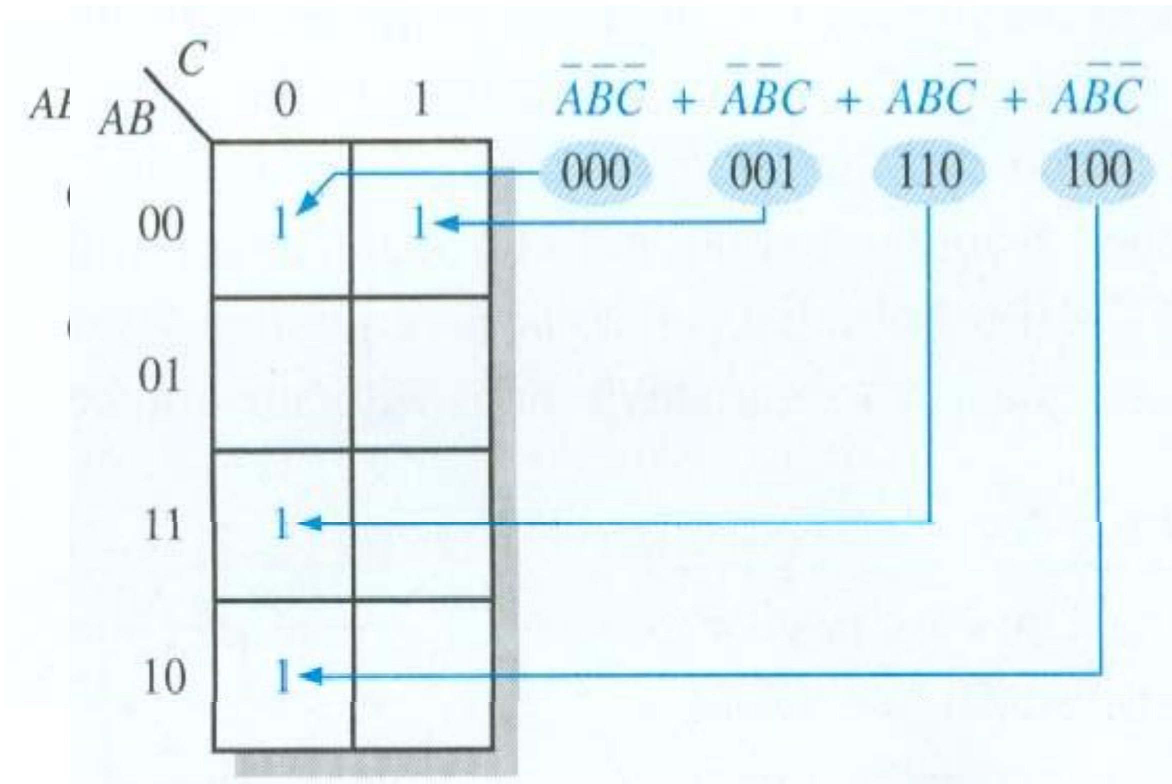
CARACTERÍSTICAS

- Útiles para expresiones de dos, tres, cuatro y cinco variables
  Es una matriz de **2^n celdas** en la que cada una representa un valor binario de las variables de entrada.
- El orden de los valores en filas y columnas es tal que **celdas adyacentes difieren únicamente en una varible**
- La simplificación de una determinada expresión consiste en agrupar adecuadamente las celdas
- Un número mayor de variables exige el uso de un método llamado Quine-McClusky

PASOS A SEGUIR

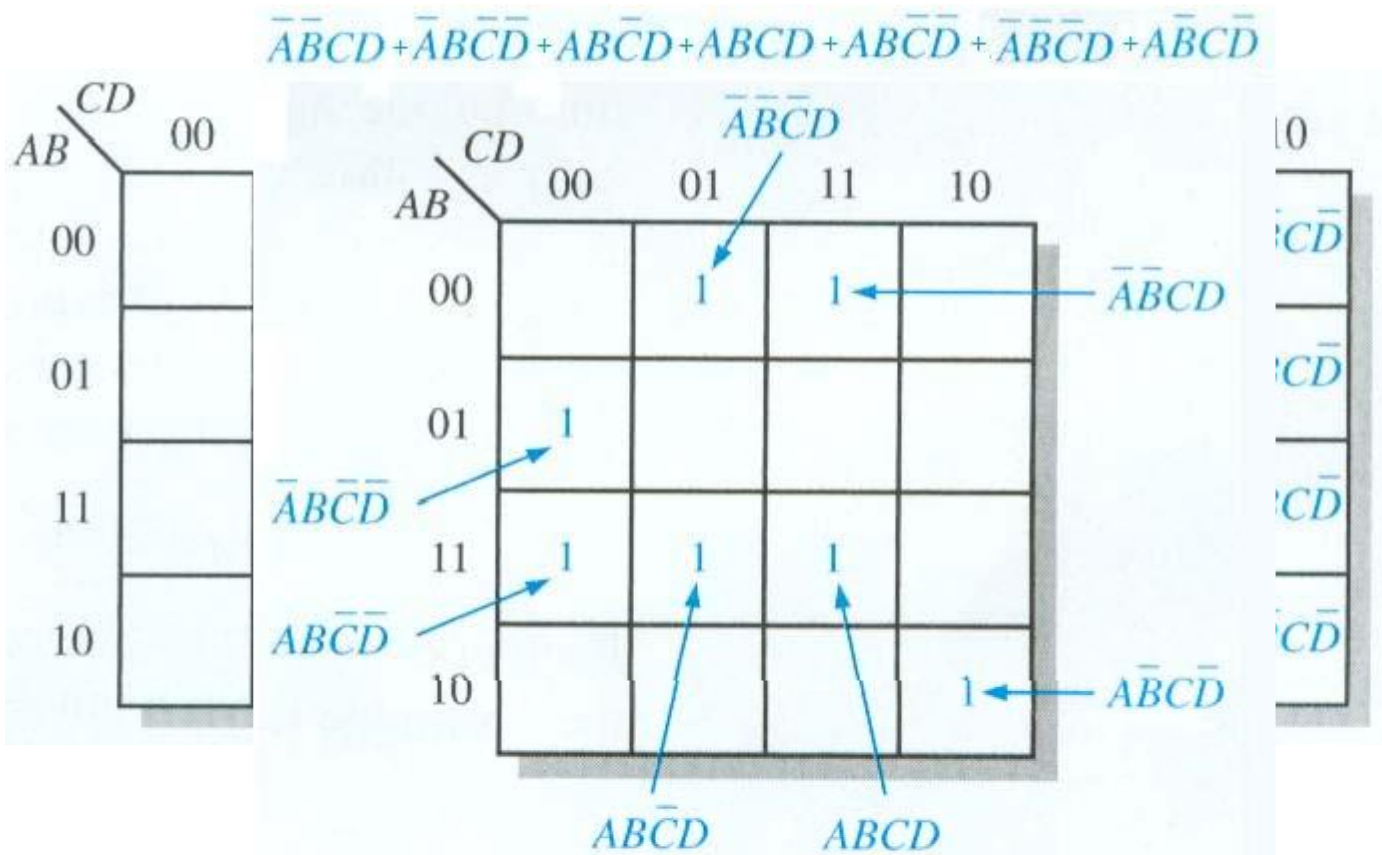
- Obtener la función lógica en **suma de productos canónica**
- Representar en el **mapa de Karnaugh** la función algebraica o tabla de verdad que se desee representar
- **Agrupar unos (*maximizar el tamaño de los grupos minimizando el número es estos*)**:
  - Un grupo tiene que contener 1, 2, 4, 8 o 16 celdas
  - Cada celda del grupo tiene que ser adyacente a una o mas celdas del grupo sin necesidad de que todas las celdas del grupo sean adyacentes entre sí.
  - Incluir siempre en cada grupo el mayor número posible de 1s
  - Cada 1 del mapa tiene que estar incluido en al menos un grupo. Los 1s que ya pertenezcan a un grupo pueden estar incluidos en otro, siempre que los grupos que se solapen contengan 1s no comunes.
- Simplificar:
  - **Eliminar variables que aparecen complementadas y sin complementar dentro del mismo grupo**

Mapas de Karnaugh



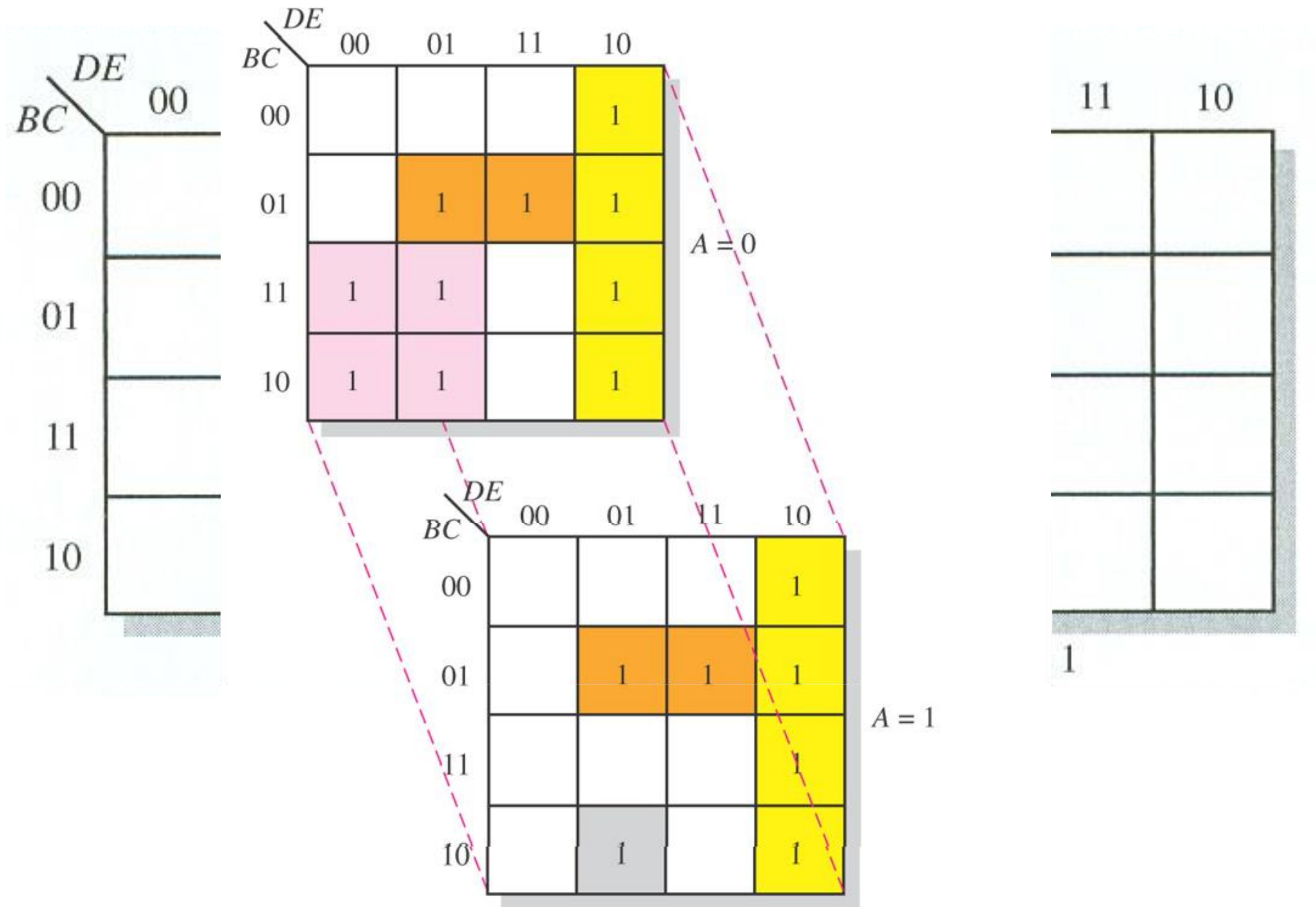
Ejemplo con 3 variables

Mapas de Karnaugh



Con 4 variables

Mapas de Karnaugh

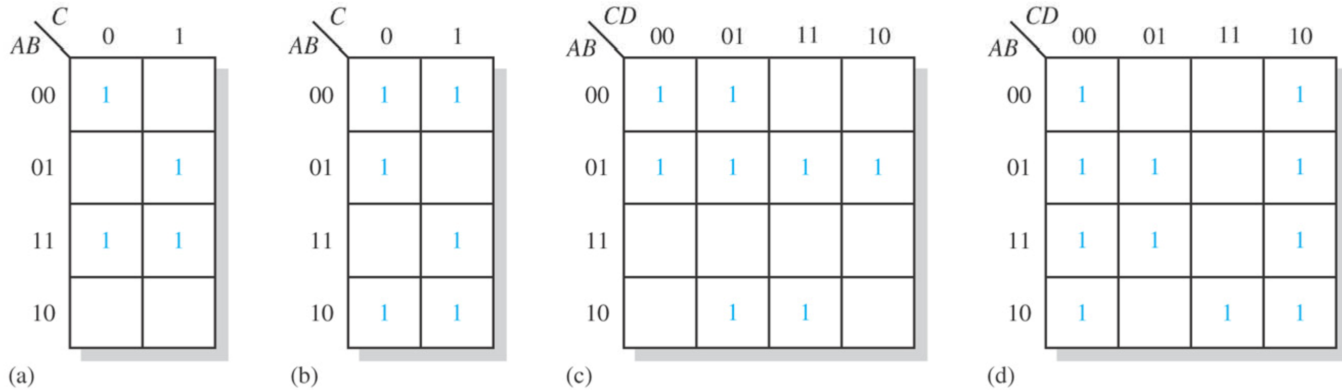


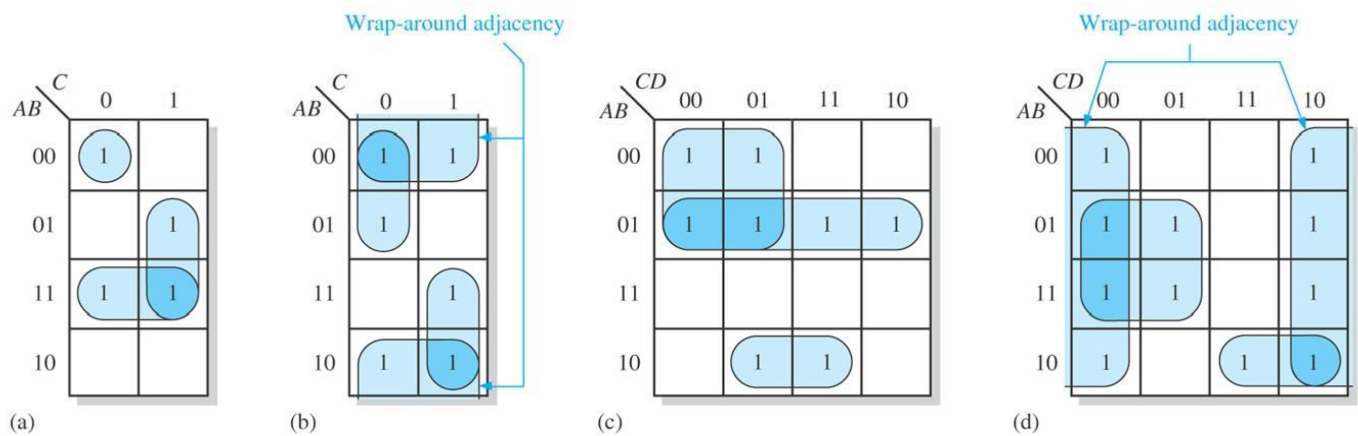
Mapas de Karnaugh para SOPs no estandares

A’	+ AB’	+ ABC’
000	100	110
001	101	
010		
011		

AB \ C	0	1
00	1	1
01	1	1
11	1	
10	1	1

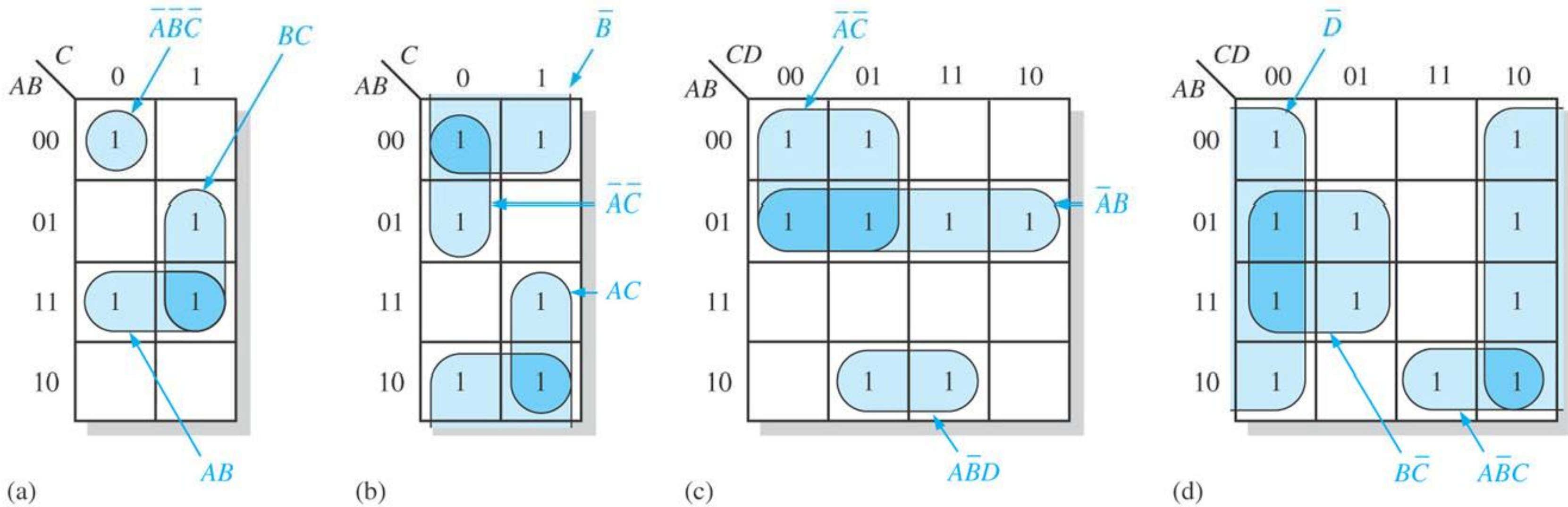
Simplificación de suma de productos mediante mapas de Karnaugh (I)





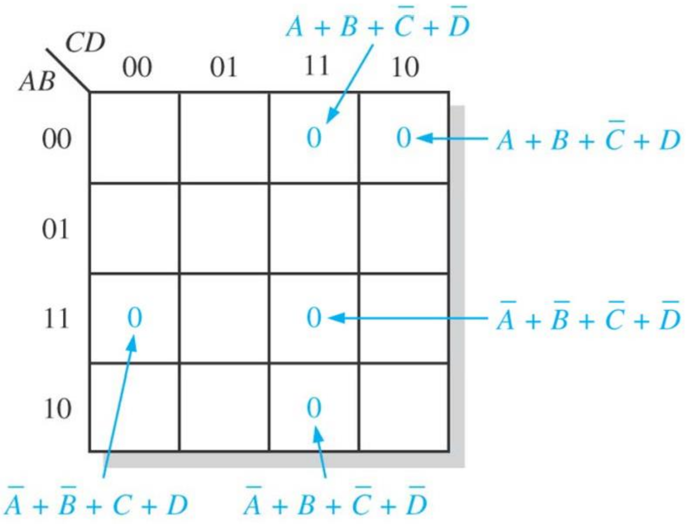
Simplificación de suma de productos mediante mapas de Karnaugh (II)

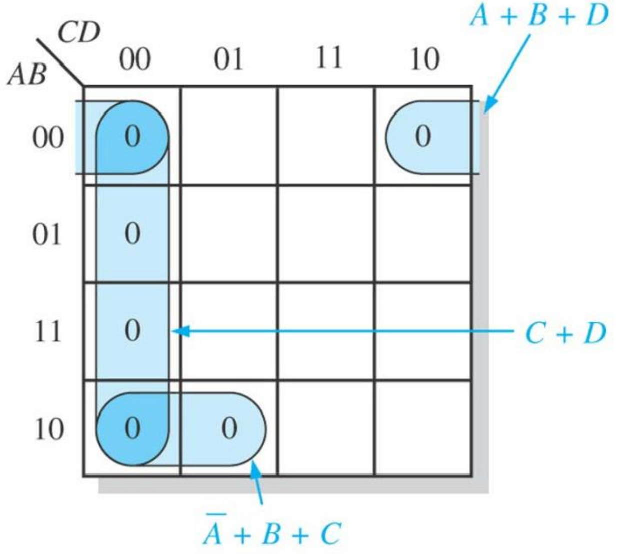
- Cada grupo da lugar a un termino
- En el término no aparecen las variables que en la tabla aparecen **complementadas** y no complementadas



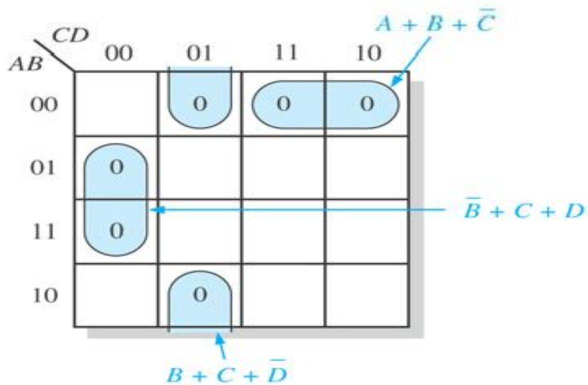
a) AB + BC + A’B’C’
b) B’ + A’C’ + AC
c) A’B + A’C’ + AB’D
d) D’ + AB’C +BC’

Simplificación de producto de sumas mediante mapas de Karnaugh (I)

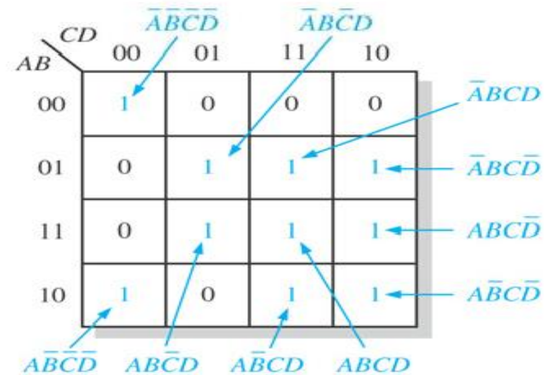




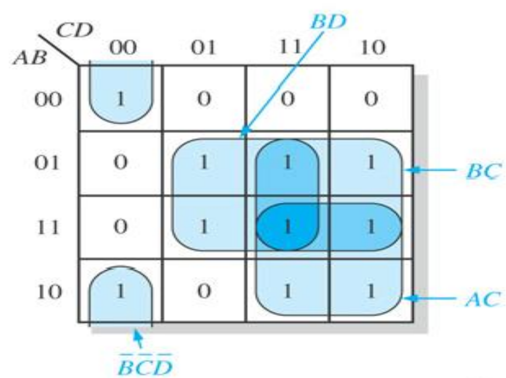
Simplificación de producto de sumas mediante mapas de Karnaugh (II)



(a) Minimum POS: $(A + B + C)(\overline{B} + \overline{C} + D)(B + C + \overline{D})$



(b) Standard SOP:
$\overline{A}\overline{B}\overline{C}\overline{D} + \overline{A}B\overline{C}D + \overline{A}BCD + \overline{A}BC\overline{D} + ABC\overline{D} + A\overline{B}C\overline{D} +$
$A\overline{B}\overline{C}\overline{D} + AB\overline{C}D + A\overline{B}CD + ABCD$



(c) Minimum SOP: $AC + BC + BD + \overline{B}\overline{C}\overline{D}$

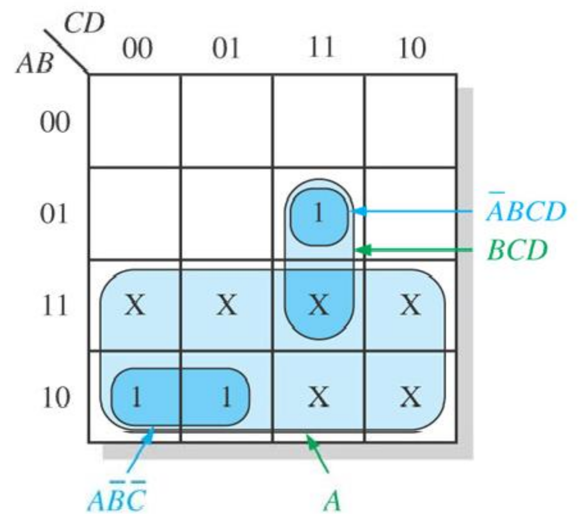
Conversión entre SOPs y POSs mediante el mapa de Karnaugh

Simplificación de suma de productos mediante mapas de Karnaugh con condiciones "indiferentes"

Inputs $ABCD$	Output Y
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	1
1 0 0 0	1
1 0 0 1	1
1 0 1 0	X
1 0 1 1	X
1 1 0 0	X
1 1 0 1	X
1 1 1 0	X
1 1 1 1	X

Don't cares

(a) Truth table



(b) Without "don't cares" $Y = A\bar{B}\bar{C} + \bar{A}BCD$
With "don't cares" $Y = A + BCD$