

Álgebra de Boole y funciones lógicas

Área Tecnología Electrónica

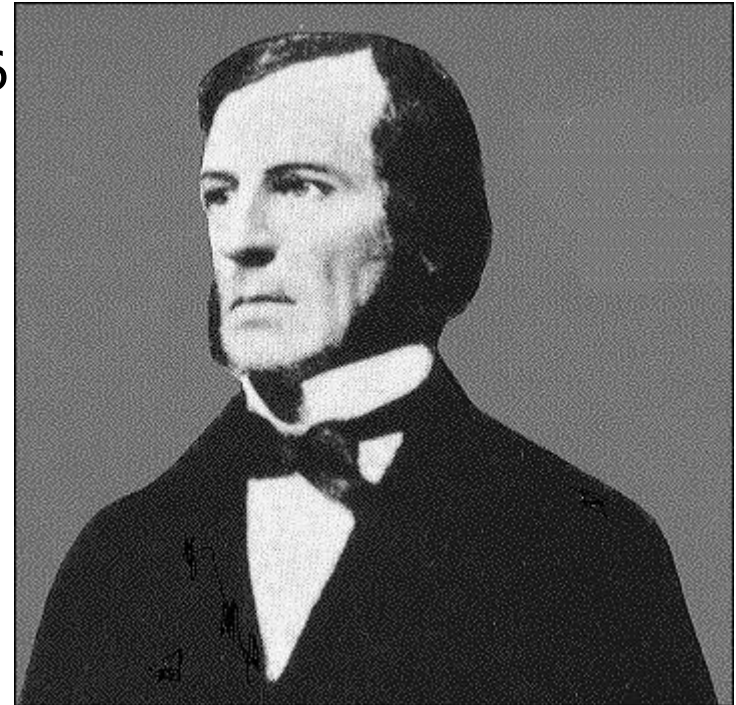


- 1. Introducción**
- 2. Definición de álgebra de Boole**
- 3. Teoremas y propiedades del álgebra de Boole**
- 4. Formas canónicas o normales de las funciones lógicas**
- 5. Implementación de funciones lógicas: puertas lógicas**
- 6. Simplificación de funciones lógicas**



George Boole

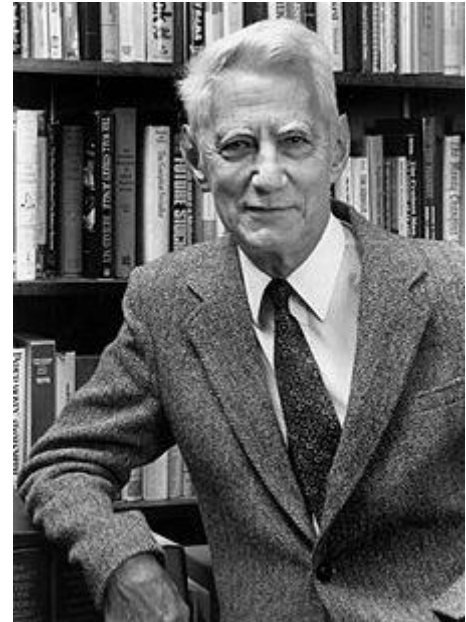
- Nació en 1815 en Lincolnshire, Inglaterra
- Profesor de colegio desde la edad de 16
- Filósofo y matemático
- Profesor en el Queens College en Cork, Irlanda





Claude E. Shannon

- Nació en Michigan en 1916
- Ingeniero Eléctrico y Matemático (Michigan '36)
- Estudios de grado en el M.I.T.
- En 1937, su Proyecto Fin de Carrera "A Symbolic Analysis of Relay and Switching Circuits" se convirtió en uno de los trabajos más importantes del siglo
- Trabajó en los Bell Labs, donde contribuyó en diferentes campos.





Introducción

- ⇒ El álgebra de Boole fue creada por el matemático británico George Boole (“An Investigation of the Laws of Thought”, 1854)
- ⇒ Constituye un formalismo matemático sencillo para representar el conocimiento y realizar cálculos.
- ⇒ Inicialmente se planteó como un formalismo más para realizar cálculos en Lógica Proposicional.
- ⇒ En 1939, Claude E. Shannon publicó su tesis de master (“A Symbolic Analysis of Relay and Switching Circuits”), en la que estableció la relación existente entre el álgebra de Boole y el estudio de los circuitos electrónicos.
- ⇒ **El Álgebra de Boole son las matemáticas de los circuitos digitales**



Introducción

⇒ Elementos del álgebra de Boole:

- Valores: verdadero (V, 1) y falso (F, 0).
 - Lógica binaria o bivaluada.
- Constantes: valores fijos (0,1).
- Variables: elementos cuyo valor puede cambiar.
 - Se designan por letras (a veces con subíndice): A , B_i , x_j .

⇒ Operaciones en el álgebra de Boole

- Son reglas de combinación de elementos que permiten hacer cálculos.
- Se representan mediante operadores.
- Operaciones básicas:
 - Adición o unión: $A+B$, $A \vee B$
 - Producto o intersección: $A \cdot B$, $A \wedge B$
 - Complementación o inversión: \overline{A} , A' , $\sim A$, $\neg A$



- ➔ Expresiones en el álgebra de Boole (formas booleanas, expresiones lógicas o expresiones de conmutación): son combinaciones de constantes, variables y operadores, incluyendo quizá paréntesis.

- ➔ Funciones en el álgebra de Boole (funciones lógicas o funciones de conmutación): son expresiones sin constantes (salvo que la función sea siempre cierta o siempre falsa).

- ➔ **Tablas de verdad**: representan los valores adoptados por las funciones lógicas de forma extensiva.
 - Tienen una columna por cada variable, más una adicional para el valor de la función.
 - Tienen una fila por cada posible combinación de valores de las variables.



Introducción

⇒ **Literal**: es una variable suelta, afirmada o negada.

⇒ **Término producto**: es una expresión booleana compuesta por un único literal o por un producto de literales.

- **Minitérmino (minterm)**: es un término producto que contiene todas las variables de la función, algunas de ellas pueden estar afirmadas y otras negadas.

⇒ **Término suma**: es una expresión booleana compuesta por un único literal o por una suma de literales.

- **Maxitérmino (maxterm)**: es un término suma que contiene todas las variables de la función, algunas de ellas pueden estar afirmadas y otras negadas

⇒ **Suma de productos (SOP, SdP)**: es una expresión booleana compuesta por un único término producto o por una suma de términos producto.

⇒ **Producto de sumas (POS, PdS)**: es una expresión booleana compuesta por un único término suma o por un producto de términos suma.


2. Definición del Álgebra de Boole

Un álgebra de Boole bivaluada es un conjunto B que cumple que:

1. $\forall a \in B, a = 0 \text{ ó } a = 1.$

2. Todo elemento tiene un complementario (función NOT).


➤ **NOT**: negación lógica o complementación.



a	$f(a) = \bar{a}$
0	1
1	0

3. La operación producto lógico (“.”, AND) que se define como:


➤ **AND**: producto lógico, intersección o conjunción.



a	b	$f(a,b) = a \cdot b$
0	0	0
0	1	0
1	0	0
1	1	1

4. La operación suma lógica (“+”, OR) se define como:

➤ **OR**: suma lógica, unión o disyunción.



a	b	$f(a,b) = a + b$
0	0	0
0	1	1
1	0	1
1	1	1

5. La operación AND tiene precedencia sobre la OR.

Operaciones del Álgebra de Boole

⇒ Otras operaciones usuales

- XOR o EOR (suma lógica exclusiva o diferencia simétrica)
- NOR (suma lógica complementada)
- NAND (producto lógico complementado)
- XNOR (suma lógica exclusiva complementada o equivalencia).

XOR

a	b	$f(a,b)=a\oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

NOR

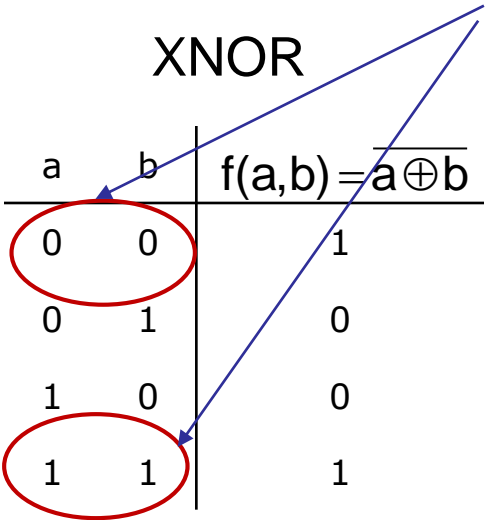
a	b	$f(a,b)=\overline{a+b}$
0	0	1
0	1	0
1	0	0
1	1	0

NAND

a	b	$f(a,b)=\overline{a\cdot b}$
0	0	1
0	1	1
1	0	1
1	1	0

XNOR

a	b	$f(a,b)=\overline{a\oplus b}$
0	0	1
0	1	0
1	0	0
1	1	1





Otra definición formal del Álgebra de Boole

⇒ Un conjunto B dotado de dos operaciones algebraicas “+” y “.” es un álgebra de Boole si y sólo si se verifican los postulados de Huntington:

1. Las operaciones + y • son conmutativas: $a+b = b+a$; $a \cdot b = b \cdot a$, $\forall a, b \in B$.

2. $\exists 0$ y $1 \in B$ tal que: $a+0 = 0+a = a$; $a \cdot 1 = 1 \cdot a = a$ $\forall a \in B$.

3. Cada operación es distributiva respecto de la otra. Es decir, que $\forall a, b \in B$ se cumple que:

$$a+(b \cdot c) = (a+b) \cdot (a+c)$$

$$a \cdot (b+c) = a \cdot b + a \cdot c$$

4. $\forall a \in B \exists$ su complementario $a' \in B$ tal que:

$$a + a' = 1; a \cdot a' = 0.$$



3. Teoremas y propiedades del Álgebra de Boole

⇒ **Principio de dualidad**: dado un teorema del álgebra de Boole, existe otro teorema que se obtiene sustituyendo:

- “+” por “.”
- “.” por “+”
- 0 por 1
- 1 por 0

El nuevo teorema así obtenido se denomina **teorema dual**.

Ejemplo:

Si se cumple el siguiente teorema (propiedad conmutativa de la suma)

$$a+b = b+a$$

también se cumplirá su teorema dual (propiedad conmutativa del producto):

$$a \cdot b = b \cdot a$$



Teoremas y propiedades del Álgebra de Boole

Propiedad asociativa

Propiedad conmutativa

Propiedad distributiva

Elemento neutro

Teoremas de identidad

Teoremas de idempotencia

Teorema de involución

Teoremas de absorción

Teoremas del consenso

Leyes de De Morgan

$a+(b+c) = (a+b)+c = a+b+c$	$a \cdot (b \cdot c) = (a \cdot b) \cdot c = a \cdot b \cdot c$
$a+b = b+a$	$a \cdot b = b \cdot a$
$a+(b \cdot c) = (a+b) \cdot (a+c)$	$a \cdot (b+c) = a \cdot b + a \cdot c$
$0+a=a$	$1 \cdot a=a$
$1+a=1$	$0 \cdot a=0$
$a+a'=1$	$a \cdot a'=0$
$a+a=a$	$a \cdot a=a$
$(a')' = a$	
$a+a \cdot b = a$	$a \cdot (a+b) = a$
$a+a' \cdot b=a+b$	$a \cdot (a'+b)=a \cdot b$
$a \cdot b+a' \cdot c= a \cdot b+a' \cdot c+b \cdot c$	$(a+b) \cdot (a'+c)= (a+b) \cdot (a'+c) \cdot (b+c)$
$(a+b)' = a' \cdot b'$	$(a \cdot b)' = a'+b'$



Teoremas y propiedades del Álgebra de Boole

⇒ **Ley de De Morgan generalizada**: la inversa de una función se obtiene complementando todas las variables que aparecen en ella e intercambiando los operadores de suma y producto lógicos.

➤ **IMPORTANTE**: es preciso respetar las precedencias de la expresión booleana original.

Ejemplo:

$$f(a,b,c,d) = \bar{a} \cdot (b + c) + a \cdot \bar{c} + \bar{d}$$

$$\begin{aligned} \bar{f}(a,b,c,d) &= \overline{\bar{a} \cdot (b + c) + a \cdot \bar{c} + \bar{d}} = \left[\overline{\bar{a} \cdot (b + c)} \right] \cdot \left[\overline{a \cdot \bar{c}} \right] \cdot \bar{\bar{d}} = \\ &= \left[\bar{\bar{a}} + \overline{(b + c)} \right] \cdot \left[\bar{a} + \bar{\bar{c}} \right] \cdot d = (a + \bar{b} \cdot \bar{c}) \cdot (\bar{a} + c) \cdot d \end{aligned}$$



Teoremas y propiedades del Álgebra de Boole

⇒ **Teorema de expansión** (de descomposición de funciones):

$$f(a,b,c,\dots) = \bar{a} \cdot f(0,b,c,\dots) + a \cdot f(1,b,c,\dots)$$

$$f(a,b,c,\dots) = [a + f(0,b,c,\dots)] \cdot [\bar{a} + f(1,b,c,\dots)]$$

Ejemplo:

$$f(a,b,c,d) = \bar{a} \cdot (b + c) + a \cdot \bar{c} + \bar{d} \Rightarrow$$

$$f(a,b,c,d) = \bar{a} \cdot f(0,b,c,d) + a \cdot f(1,b,c,d) =$$

$$= \bar{a}[\bar{0} \cdot (b + c) + 0 \cdot \bar{c} + \bar{d}] + a \cdot [\bar{1} \cdot (b + c) + 1 \cdot \bar{c} + \bar{d}] =$$

$$= \bar{a} \cdot (b + c) + \bar{a} \cdot \bar{d} + a \cdot \bar{c} + a \cdot \bar{d} =$$

$$= \bar{a} \cdot (b + c) + a \cdot \bar{c} + \bar{d} \cdot (\bar{a} + a) =$$

$$= \bar{a} \cdot (b + c) + a \cdot \bar{c} + \bar{d}$$



4. Formas canónicas

⇒ Todas las expresiones booleanas, independientemente de su forma, pueden convertirse en cualquiera de las dos formas canónicas.

⇒ **Formas canónicas**, formas normales o formas estándares de una función booleana son expresiones booleanas de la función que verifican:

- **Primera forma canónica, primera forma normal o forma normal disyuntiva**: es una expresión de una función booleana compuesta por una suma de minitérminos.
- **Segunda forma canónica, segunda forma normal o forma normal conjuntiva**: es una expresión de una función booleana compuesta por un producto de maxitérminos.



Formas canónicas

⇒ **Minitérmino (minterm)**: término producto que contiene todas las variables de la función, algunas de las cuales pueden estar afirmadas y otras negadas.

- Ejemplo: $f(a,b,c)$

Sí son minitérminos:

$$\bar{a} \cdot \bar{b} \cdot \bar{c} \quad \bar{a} \cdot b \cdot \bar{c} \quad \bar{a} \cdot b \cdot c \quad a \cdot \bar{b} \cdot c \quad a \cdot b \cdot c$$

NO son minitérminos:

$$\bar{a} \cdot \bar{b} \quad \bar{b} \cdot c \quad \bar{a} \cdot c \quad a \cdot \bar{b} \quad a \cdot c$$

⇒ **Maxitérmino (maxterm)**: término suma que contiene todas las variables de la función, algunas de las cuales pueden estar afirmadas y otras negadas.

- Ejemplo: $f(a,b,c)$

Sí son maxitérminos:

$$\bar{a} + \bar{b} + \bar{c} \quad \bar{a} + b + \bar{c} \quad \bar{a} + b + c \quad a + \bar{b} + c \quad a + b + c$$

NO son maxitérminos:

$$\bar{a} + \bar{b} \quad \bar{b} + c \quad \bar{a} + c \quad a + \bar{b} \quad a + c$$



Primera forma canónica (1FC)

⇒ Los minitérminos se nombran con subíndices (m_i), donde i es un número obtenido tras pasar a base 10 el número binario formado al sustituir ordenadamente las variables afirmadas por 1 y las negadas por 0.

•Ejemplo: $f(a,b,c,d)$, minitérmino $\bar{a} \cdot b \cdot \bar{c} \cdot d = m_5$

⇒ Cada minitérmino está asociado a una fila de la tabla de verdad de la función lógica correspondiente.

⇒ **Primera forma canónica, primera forma normal o forma normal disyuntiva:** es una expresión de una función booleana compuesta por una suma de minitérminos.

⇒ La expresión en 1FC es única para cada función.

a	b	c	d	Minitérmino	m_i
0	0	0	0	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$	m_0
0	0	0	1	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d$	m_1
0	0	1	0	$\bar{a} \cdot \bar{b} \cdot c \cdot \bar{d}$	m_2
0	0	1	1	$\bar{a} \cdot \bar{b} \cdot c \cdot d$	m_3
0	1	0	0	$\bar{a} \cdot b \cdot \bar{c} \cdot \bar{d}$	m_4
0	1	0	1	$\bar{a} \cdot b \cdot \bar{c} \cdot d$	m_5
0	1	1	0	$\bar{a} \cdot b \cdot c \cdot \bar{d}$	m_6
0	1	1	1	$\bar{a} \cdot b \cdot c \cdot d$	m_7
1	0	0	0	$a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$	m_8
1	0	0	1	$a \cdot \bar{b} \cdot \bar{c} \cdot d$	m_9
1	0	1	0	$a \cdot \bar{b} \cdot c \cdot \bar{d}$	m_{10}
1	0	1	1	$a \cdot \bar{b} \cdot c \cdot d$	m_{11}
1	1	0	0	$a \cdot b \cdot \bar{c} \cdot \bar{d}$	m_{12}
1	1	0	1	$a \cdot b \cdot \bar{c} \cdot d$	m_{13}
1	1	1	0	$a \cdot b \cdot c \cdot \bar{d}$	m_{14}
1	1	1	1	$a \cdot b \cdot c \cdot d$	m_{15}



Primera forma canónica (1FC)

La expresión en 1FC de una función booleana es la suma de los minitérminos asociados a las filas que valen 1 en la tabla de verdad.

Ejemplo: $f(a,b,c) = \bar{a} \cdot (b + c) + a \cdot \bar{c}$

Calculando su tabla de verdad se obtiene lo siguiente:

a	b	c	a'	b+c	a' · (b+c)	c'	a · c'	f(i)
0	0	0	1	0	0	1	0	0
0	0	1	1	1	1	0	0	1
0	1	0	1	1	1	1	0	1
0	1	1	1	1	1	0	0	1
1	0	0	0	0	0	1	1	1
1	0	1	0	1	0	0	0	0
1	1	0	0	1	0	1	1	1
1	1	1	0	1	0	0	0	0

Entonces: $f(a,b,c) = m_1 + m_2 + m_3 + m_4 + m_6 = \sum_3 m(1,2,3,4,6)$



Segunda forma canónica (2FC)

⇒ Los maxitérminos se nombran con subíndices (M_i), donde i es un número obtenido tras pasar a base 10 el número binario formado al sustituir ordenadamente las variables afirmadas por 0 y las negadas por 1.

• Ejemplo: $f(a,b,c,d)$, maxitérmino $a + \bar{b} + c + \bar{d} = M_5$

⇒ Cada maxitérmino está asociado a una fila de la tabla de verdad de la función lógica correspondiente

⇒ **Segunda forma canónica, segunda forma normal o forma normal conjuntiva:** es una expresión de una función booleana compuesta por un producto de maxitérminos

⇒ La expresión en 2FC es única para cada función

a	b	c	d	Maxitérmino	M_i
0	0	0	0	$a + b + c + d$	M_0
0	0	0	1	$a + b + c + \bar{d}$	M_1
0	0	1	0	$a + b + \bar{c} + d$	M_2
0	0	1	1	$a + b + \bar{c} + \bar{d}$	M_3
0	1	0	0	$a + \bar{b} + c + d$	M_4
0	1	0	1	$a + \bar{b} + c + \bar{d}$	M_5
0	1	1	0	$a + \bar{b} + \bar{c} + d$	M_6
0	1	1	1	$a + \bar{b} + \bar{c} + \bar{d}$	M_7
1	0	0	0	$\bar{a} + b + c + d$	M_8
1	0	0	1	$\bar{a} + b + c + \bar{d}$	M_9
1	0	1	0	$\bar{a} + b + \bar{c} + d$	M_{10}
1	0	1	1	$\bar{a} + b + \bar{c} + \bar{d}$	M_{11}
1	1	0	0	$\bar{a} + \bar{b} + c + d$	M_{12}
1	1	0	1	$\bar{a} + \bar{b} + c + \bar{d}$	M_{13}
1	1	1	0	$\bar{a} + \bar{b} + \bar{c} + d$	M_{14}
1	1	1	1	$\bar{a} + \bar{b} + \bar{c} + \bar{d}$	M_{15}



Segunda forma canónica (2FC)

La expresión en 2FC de una función booleana es el producto de los maxitérminos asociados a las filas que valen 0 en la tabla de verdad.

Ejemplo: $f(a,b,c) = \bar{a} \cdot (b + c) + a \cdot \bar{c}$

a	b	c	f(i)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Tomando las filas que valen 0, tendremos $f'(a,b,c)$:

$$\begin{aligned}\bar{f}(a,b,c) &= m_0 \cdot \bar{f}(0) + m_1 \cdot \bar{f}(1) + m_2 \cdot \bar{f}(2) + m_3 \cdot \bar{f}(3) + m_4 \cdot \bar{f}(4) + m_5 \cdot \bar{f}(5) + m_6 \cdot \bar{f}(6) + m_7 \cdot \bar{f}(7) = \\ &= m_0 \cdot 1 + m_1 \cdot 0 + m_2 \cdot 0 + m_3 \cdot 0 + m_4 \cdot 0 + m_5 \cdot 1 + m_6 \cdot 0 + m_7 \cdot 1 = m_0 + m_5 + m_7\end{aligned}$$

Negando la expresión anterior obtendremos $f(a,b,c)$:

$$\bar{\bar{f}}(a,b,c) = f(a,b,c) = \overline{m_0 + m_5 + m_7} = \bar{m}_0 \cdot \bar{m}_5 \cdot \bar{m}_7 = M_0 \cdot M_5 \cdot M_7 \Rightarrow \mathbf{f(a,b,c) = \prod_3 M(0,5,7)}$$



Funciones definidas de forma incompleta

⇒ En algunos sistemas digitales reales, hay ciertas combinaciones de las variables de entrada que por no pueden producirse nunca.

⇒ En estos casos, la salida que pudiera producir el sistema ante dichas combinaciones de entrada es irrelevante, puesto que nunca se va a dar tal caso.

⇒ Las combinaciones imposibles de entrada se denominan **indiferencias, valores indiferentes, redundancias** o *don't care values*, y en la tabla de verdad se representan con el símbolo X.

⇒ Si aparece un símbolo X en una o varias filas de una tabla, nos daría exactamente igual sustituirla por un 1 ó por un 0.

Ejemplo: función que dice si un número en BCD es par.

$$f(a,b,c,d) = \sum_4 m(0,2,4,6,8) + X(10,11,12,13,14,15)$$

$$f(a,b,c,d) = \prod_4 M(1,3,5,7,9) \cdot X(10,11,12,13,14,15)$$

a	b	c	d	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	1
0	1	0	1	0
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
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



Formas Canónicas: resumen

⇒ Las formas canónicas se pueden extraer directamente de la tabla de verdad

⇒ Primera forma canónica (1FC): suma de minitérminos asociados a las filas con valor 1.

$$f(a,b,c,d) = \sum_4 m(0,3,4,5,10,11,14,15)$$

⇒ Segunda forma canónica (2FC): producto de maxitérminos asociados a las filas con valor 0.

$$f(a,b,c,d) = \prod_4 M(1,2,6,7,8,9,12,13)$$

⇒ Las formas canónicas son únicas para cada función: una función tiene una única expresión en 1FC y una única expresión en 2FC.

⇒ La 1FC y la 2FC de una función son equivalentes.



Formas Canónicas: resumen

⇒ Conversión de suma de productos a la 1FC:

Aplicar la propiedad $A + A' = 1$: Cada término producto no estándar se multiplica por un término formado por suma de la variable que falta y su complemento

Ejemplo: $f(A, B, C, D) = A\bar{B}C + \bar{A}\bar{B} + AB\bar{C}D$

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

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

$$f(A, B, C, D) = A\bar{B}CD + A\bar{B}C\bar{D} + \bar{A}\bar{B}CD + \bar{A}\bar{B}C\bar{D} + \bar{A}\bar{B}\bar{C}D + \bar{A}\bar{B}\bar{C}\bar{D} + AB\bar{C}D$$

⇒ Conversión producto de sumas a la 2FC:

Aplicar la propiedad $A \cdot A' = 0$: Se añade a cada término suma no estándar un término producto formado por la variable que falta y su complemento

Ejemplo: $f(A, B, C, D) = (A + \bar{B} + C) \cdot (\bar{A} + \bar{B}) \cdot (A + B + \bar{C} + D)$

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

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



5. Implementación de funciones lógicas: Puertas Lógicas

➔ **Puertas lógicas:** dispositivos electrónicos capaces de implementar operadores lógicos

➔ Para cada operación lógica (AND, OR, NOT, XOR, NAND, NOR, XNOR) existe la correspondiente puerta lógica que la materializa.

➔ Un circuito lógico se construye a partir de una expresión algebraica de la función lógica que queremos implementar, interconectando puertas lógicas básicas de acuerdo con dicha expresión.

- Una función dada puede representarse mediante múltiples expresiones algebraicas equivalentes.
- Una función dada puede materializarse con diferentes circuitos.
- Mientras más sencilla sea la expresión lógica utilizada, más sencillo será el circuito que materialice la función buscada.



Puertas Lógicas

La función lógica implementada por una puerta lógica depende de la interpretación (convenio) utilizada.

Los nombres dados a las puertas lógicas básicas coinciden con la función lógica que realizan interpretando los valores de sus entradas y salidas mediante el **convenio de lógica positiva**.

Estudiaremos las siguientes puertas lógicas básicas:

- ☐ Puerta **INVERSOR (NOT)**
 - ☐ Puerta **AND**
 - ☐ Puerta **OR**
 - ☐ Puerta **NAND**
 - ☐ Puerta **NOR**
 - ☐ Puerta **XOR**
 - ☐ Puerta **XNOR**
- } Puertas básicas
- } Puertas universales



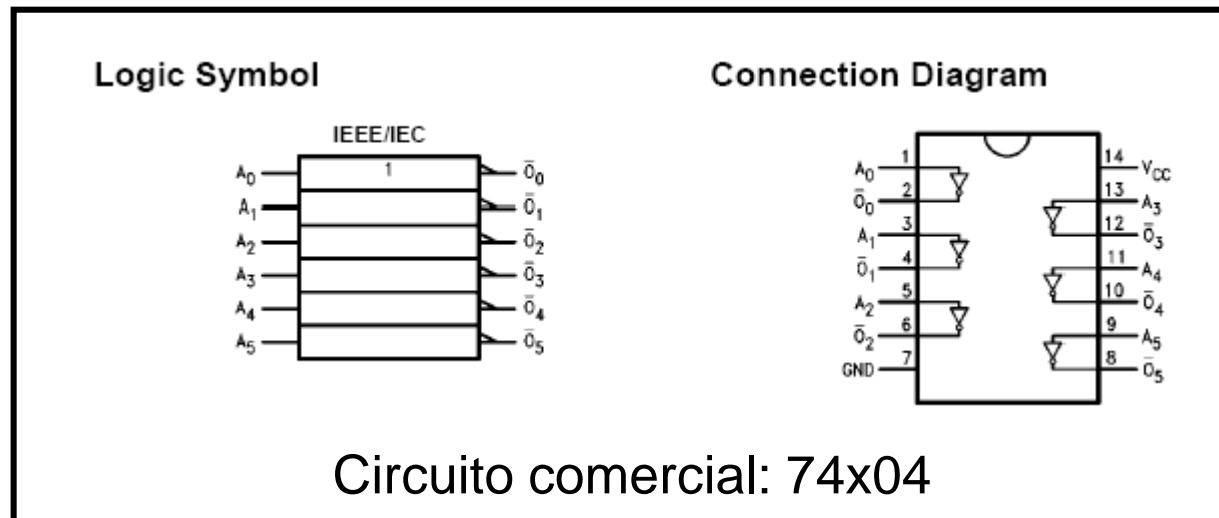
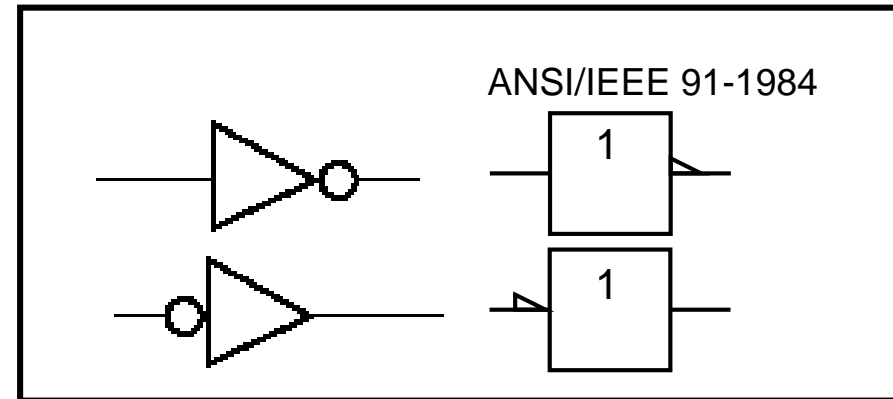
Puertas Lógicas: INVERSOR

➤ Realiza la operación lógica de **INVERSIÓN** o **COMPLEMENTACIÓN**: cambia un nivel lógico al nivel opuesto.

➤ Expresión lógica: $S = \overline{A}$

➤ Tabla de verdad:

A	S
0	1
1	0



➤ **Ejemplo de aplicación:** circuito que genera el complemento a 1 de un n^0 binario



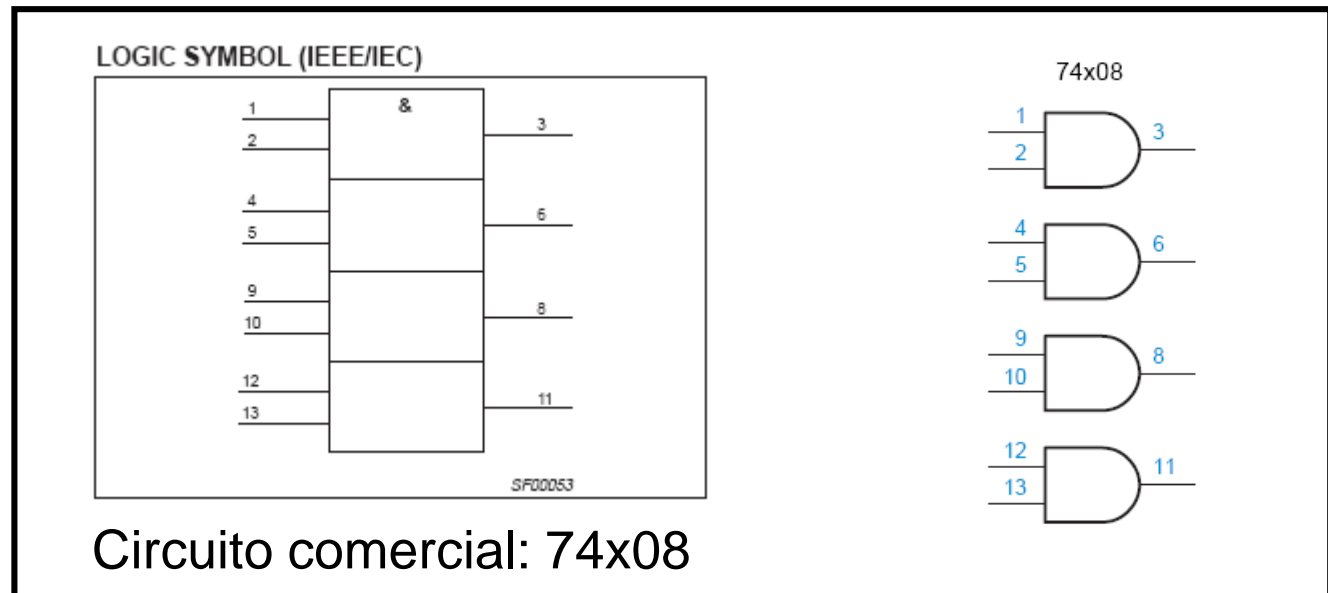
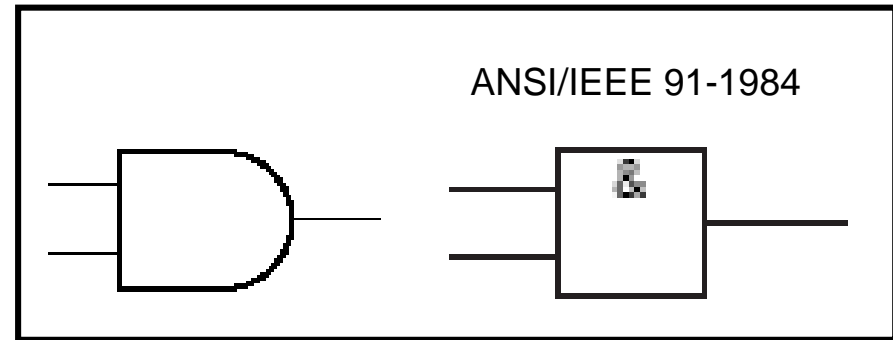
Puertas Lógicas: AND

➤ Realiza la operación lógica de **MULTIPLICACIÓN LÓGICA**

➤ Expresión lógica: $S = A \cdot B$

➤ Tabla de verdad:

A	B	S
0	0	0
0	1	0
1	0	0
1	1	1



➤ **Ejemplo de aplicación:** La puerta AND como un dispositivo de **habilitación / deshabilitación**



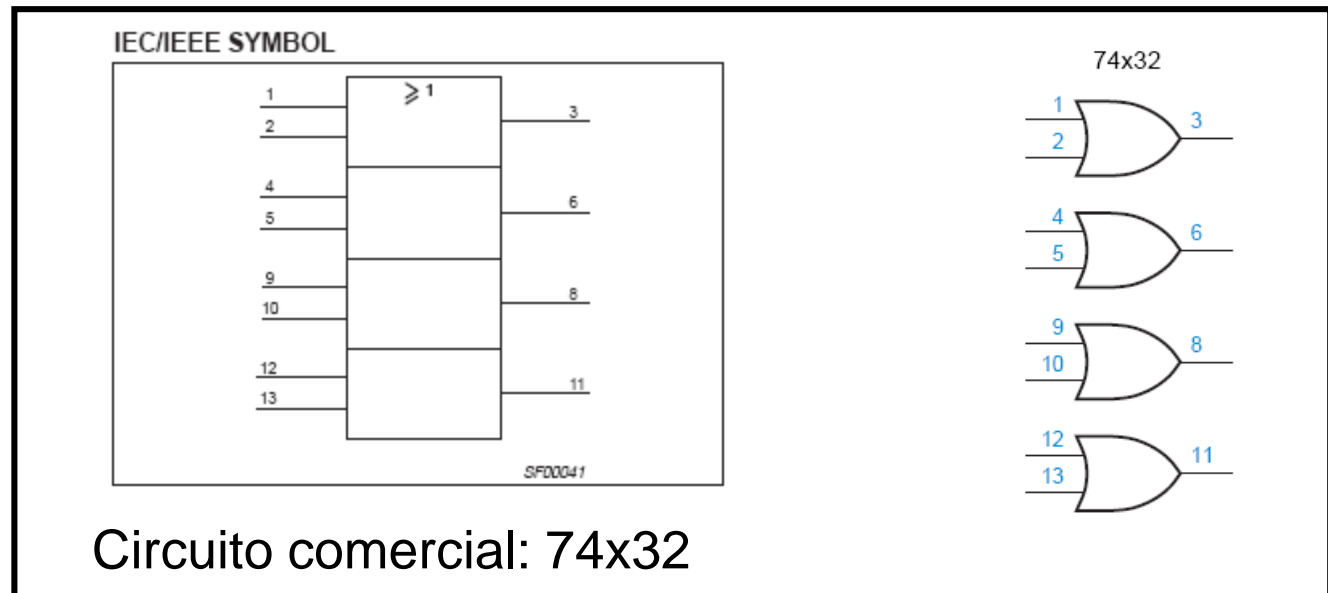
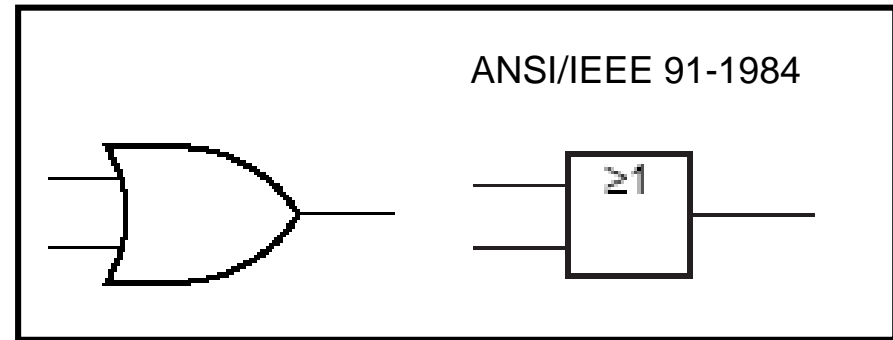
Puertas Lógicas: OR

➤ Realiza la operación lógica de **SUMA LÓGICA**

➤ Expresión lógica: $S = A + B$

➤ Tabla de verdad:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	1



➤ **Ejemplo de aplicación:** La puerta OR como un dispositivo de **habilitación / deshabilitación**



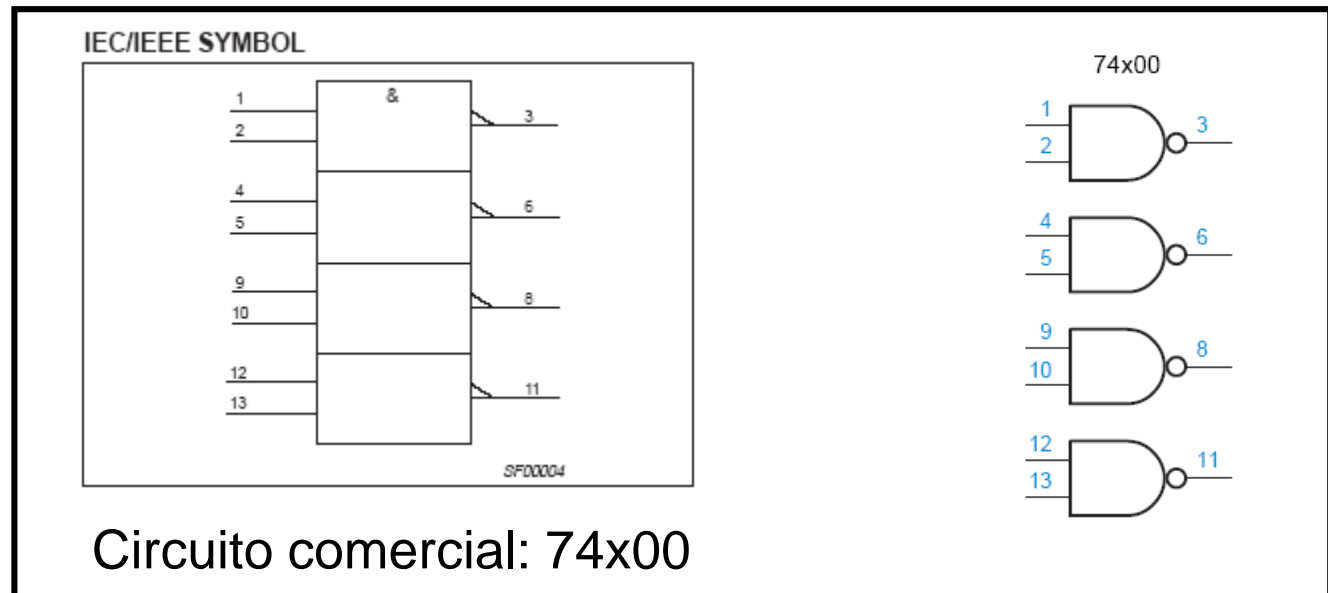
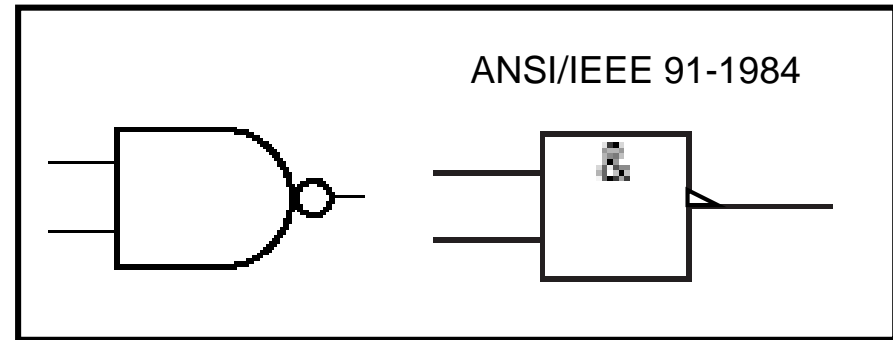
Puertas Lógicas: NAND

➤ Realiza la operación lógica de **NOT-AND** : una función AND con salida complementada

➤ Expresión lógica: $S = \overline{A \cdot B}$

➤ Tabla de verdad:

A	B	S
0	0	1
0	1	1
1	0	1
1	1	0



➤ **Puerta universal:** las puertas NAND pueden generar cualquiera de las puertas básicas NOT, AND, OR.



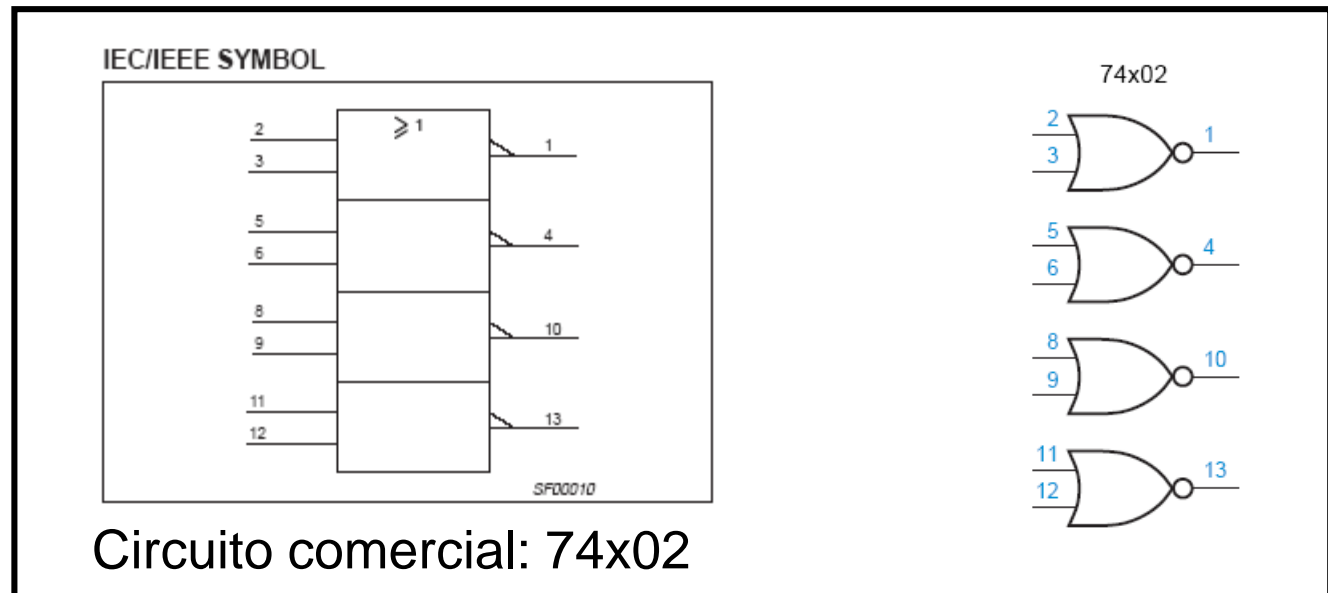
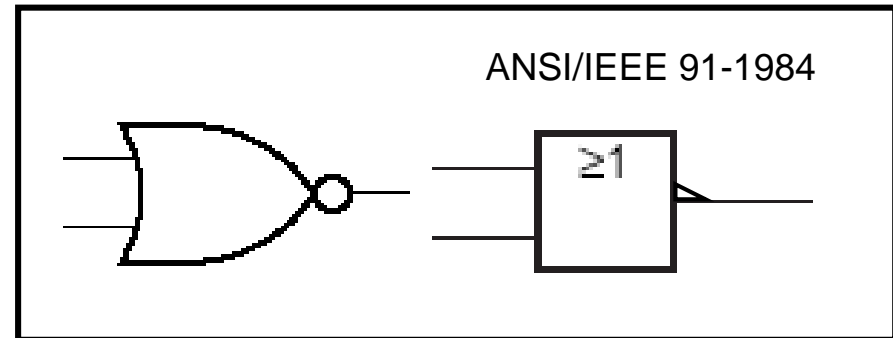
Puertas Lógicas: NOR

➤ Realiza la operación lógica de **NOT-OR** : una función OR con salida complementada.

➤ Expresión lógica: $S = \overline{A + B}$

➤ Tabla de verdad:

A	B	S
0	0	1
0	1	0
1	0	0
1	1	0



➤ **Puerta universal:** las puertas NOR pueden generar cualquiera de las puertas básicas NOT, AND, OR.



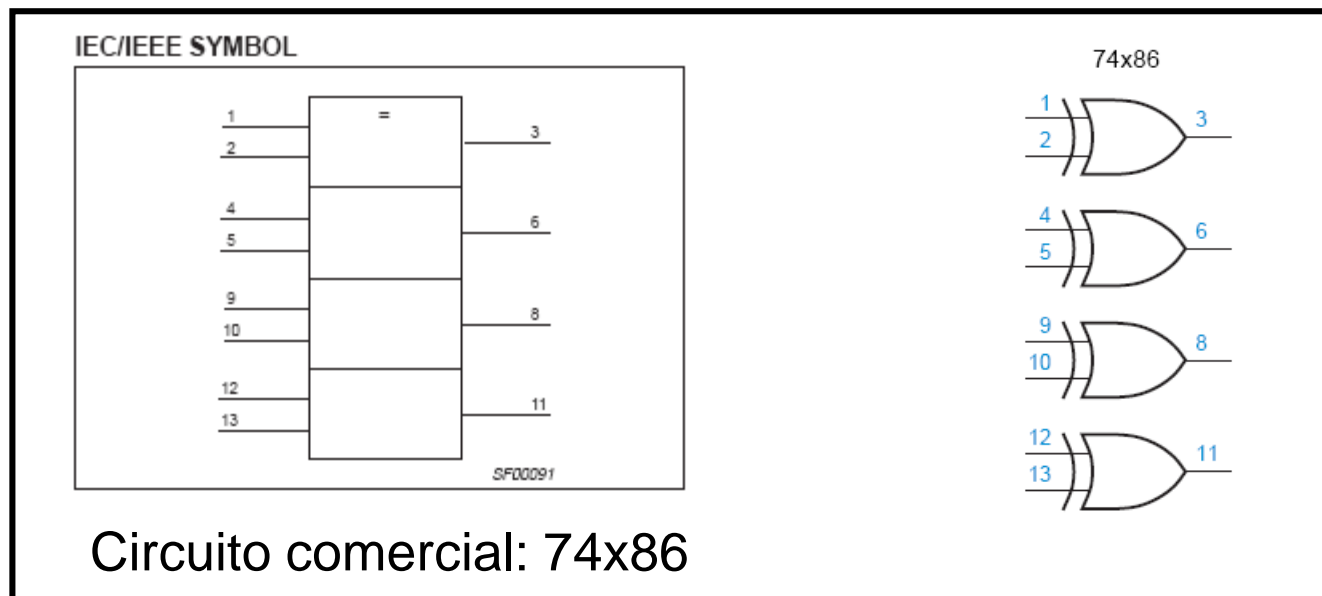
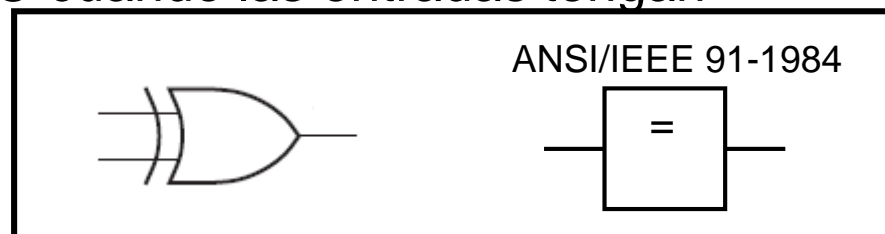
Puertas Lógicas: OR- EXCLUSIVA (XOR)

➤ La salida de una puerta OR-exclusiva se pone a nivel alto sólo cuando hay un n° impar de entradas a nivel alto. En el caso particular de una puerta con dos entradas, la salida estará a nivel ALTO cuando las entradas tengan niveles lógicos opuestos.

➤ Expresión lógica: $S = A \oplus B$

➤ Tabla de verdad:

A	B	S
0	0	0
0	1	1
1	0	1
1	1	0



➤ **Aplicaciones:** comparador, detectores de paridad, sumador



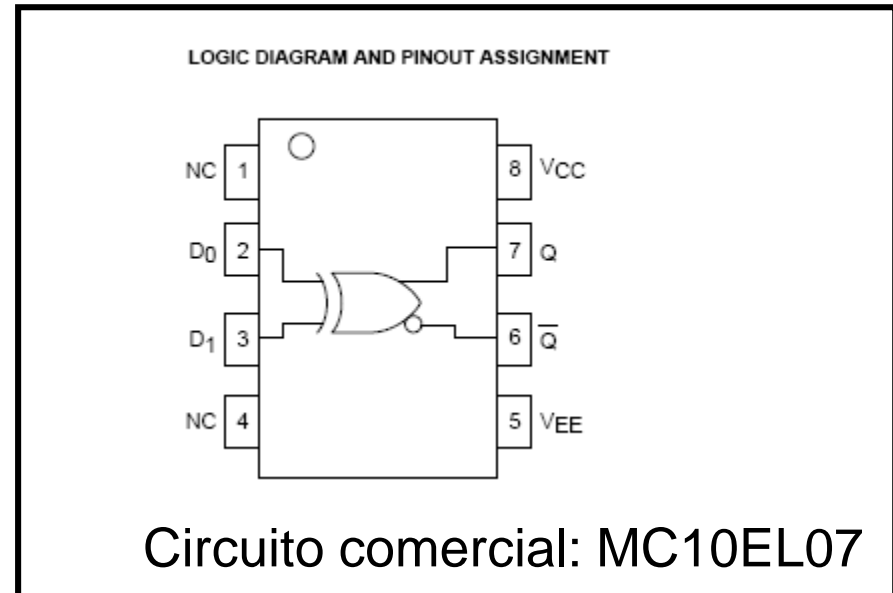
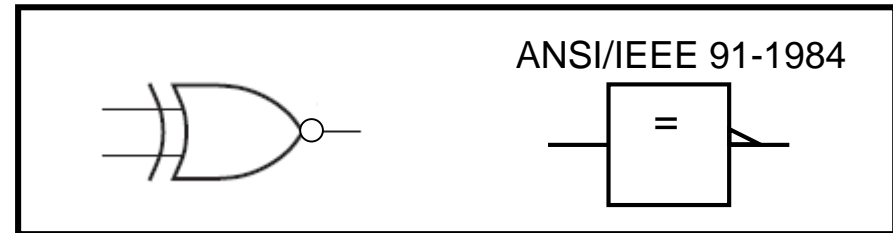
Puertas Lógicas: NOR- EXCLUSIVA (XNOR)

➤ Función OR-exclusiva con la salida complementada

➤ Expresión lógica: $S = \overline{A \oplus B}$

➤ Tabla de verdad:

A	B	S
0	0	1
0	1	0
1	0	0
1	1	1

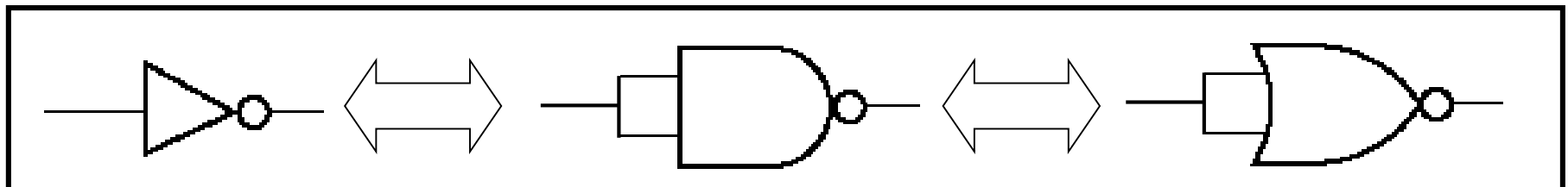
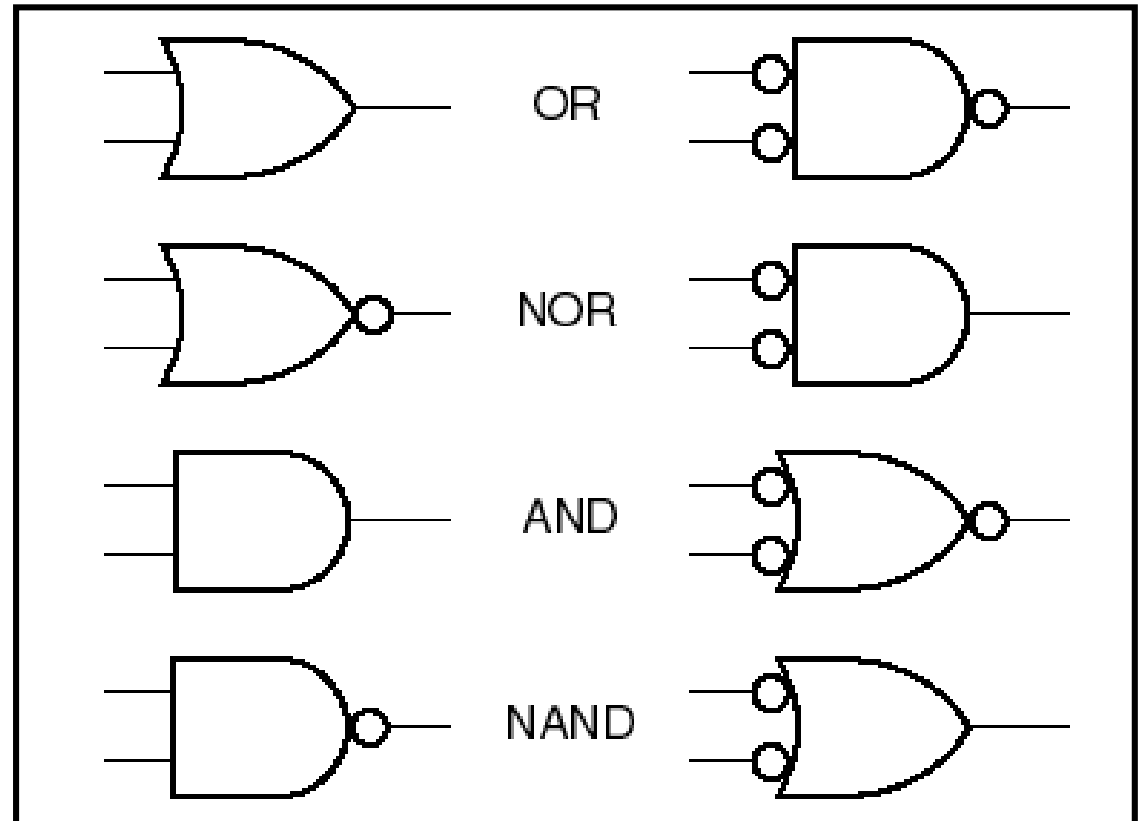


➤ **Aplicaciones:** comparador, detectores de paridad, sumador

➤ Aplicación de las leyes De Morgan a las puertas lógicas:

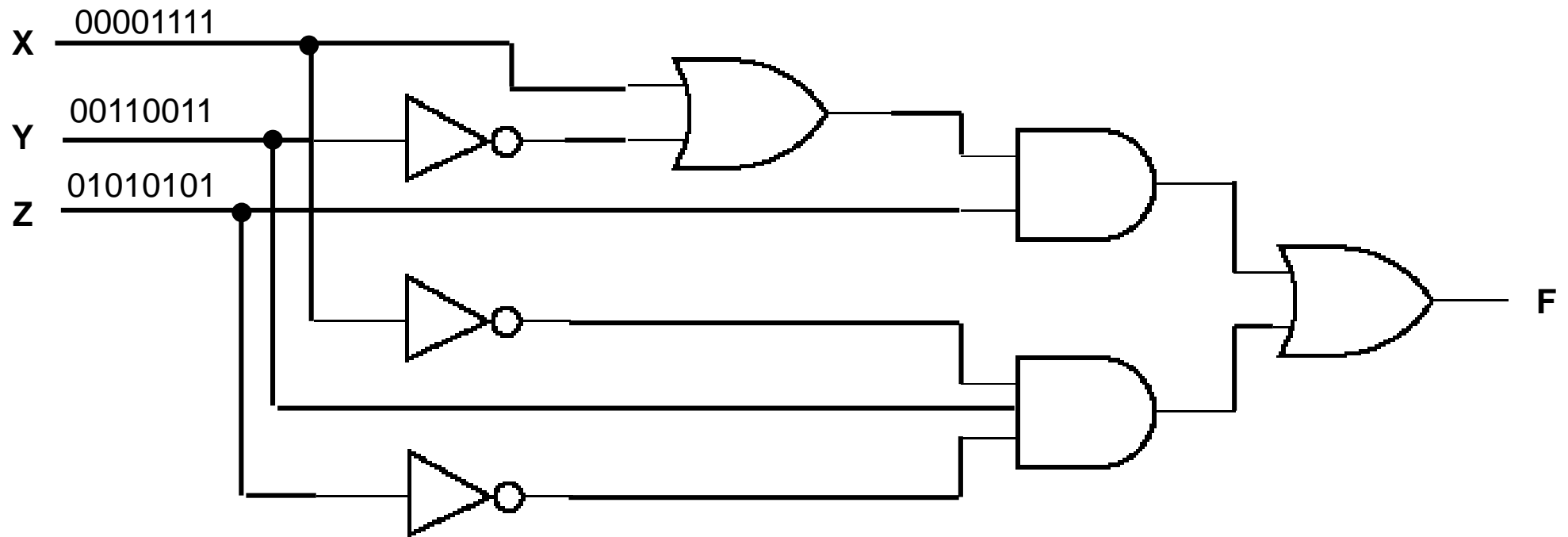
$$\overline{\overline{A + B}} = \overline{\overline{A}} \cdot \overline{\overline{B}}$$

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$





Análisis de circuitos combinacionales





6. Simplificaciones de funciones lógicas

⇒ Dado que existen múltiples circuitos para implementar una función lógica dada, lo mejor es utilizar el circuito más adecuado para cada situación.

⇒ Criterios posibles para manipular las expresiones lógicas:

- Obtener el circuito más barato reduciendo el número de términos.
- Obtener el circuito más rápido.
- Obtener el circuito formado por menos circuitos integrados de un tipo dado.
- Obtener un circuito sin valores transitorios no deseados (azares, *glitches*).

⇒ Simplificación: proceso que conduce a reducir el número de literales y términos de una función lógica.

- Existen métodos de simplificación automáticos para ser implementados en computadores (Quine-McCluskey es el más conocido).
- Se puede simplificar funciones mediante manipulaciones algebraicas (manual, costoso).
- Métodos gráficos: **Veitch-Karnaugh** (manual, sencillo para pocas variables).



Método de Veitch-Karnaugh

- ➔ Inventado por Veitch a principios de los años 50, y perfeccionado por Karnaugh.
- ➔ Se basa en construir unos diagramas adecuados para simplificar gráficamente.
- ➔ Diagrama (mapa, tabla) de Veitch-Karnaugh para una función de n variables: tabla rectangular de 2^n celdas, cada una de las cuales está asociada a una combinación de variables (y a una fila de la tabla de verdad).
 - En cada casilla hay un 1 ó un 0, dependiendo de la fila de la tabla de verdad asociada.
- ➔ Propiedad principal: cada casilla es adyacente a todas sus vecinas en horizontal y vertical, es decir, entre una casilla y su vecina sólo difiere el valor de una variable.
- ➔ Sólo son utilizables en la práctica los mapas para funciones de 2, 3, 4, 5 y 6 variables.



Método de Veitch-Karnaugh: 2 variables

⇒ El mapa tiene 4 casillas, cada una asociada a una combinación de los valores de las variables.

⇒ Cada casilla tiene 2 vecinas.

⇒ En cada casilla se ha añadido el nº de la fila de la tabla de verdad asociada a dicha casilla, así como la combinación de variables que la corresponde.

		b	
		0	1
a	0	$\bar{a} \cdot \bar{b}$ 0 $f(0)$	$\bar{a} \cdot b$ 1 $f(1)$
	1	$a \cdot \bar{b}$ 2 $f(2)$	$a \cdot b$ 3 $f(3)$

⇒ Vecindades:

- Casilla 0: 1 y 2.
- Casilla 1: 0 y 3.
- Casilla 2: 0 y 3.
- Casilla 3: 1 y 2.



Método de Veitch-Karnaugh: 2 variables

Ejemplo:

Tabla de verdad

a	b	f(i)
0	0	1
0	1	0
1	0	1
1	1	1

Función: $f(a,b)=m_0+m_2+m_3=M_1$

Mapa de V-K:

		b	
		0	1
a	0	1	0
	1	1	1



Método de Veitch-Karnaugh: 3 variables

⇒ El mapa tiene 8 casillas, y cada casilla tiene 3 vecinas.

bc a		bc			
		00	01	11	10
0	a	$\bar{a} \cdot \bar{b} \cdot \bar{c}$ 0 f(0)	$\bar{a} \cdot \bar{b} \cdot c$ 1 f(1)	$\bar{a} \cdot b \cdot c$ 3 f(3)	$\bar{a} \cdot b \cdot \bar{c}$ 2 f(2)
	1	$a \cdot \bar{b} \cdot \bar{c}$ 4 f(4)	$a \cdot \bar{b} \cdot c$ 5 f(5)	$a \cdot b \cdot c$ 7 f(7)	$a \cdot b \cdot \bar{c}$ 6 f(6)

⇒ Vecindades:

- Casilla 0: 1, 2 y 4.
- Casilla 1: 0, 3 y 5.
- Casilla 2: 0, 3 y 6.
- Casilla 3: 1, 2 y 7.
- Casilla 4: 0, 5 y 6.
- Casilla 5: 1, 4 y 7.
- Casilla 6: 2, 4 y 7.
- Casilla 7: 2, 4 y 7.



Método de Veitch-Karnaugh: 3 variables

Ejemplo:

Tabla de verdad

a	b	c	f(i)
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	1
1	1	1	0

Función: $f(a,b)=m_1+m_2+m_3+m_4+m_6=M_0 \cdot M_5 \cdot M_7$

Mapa de V-K:

		bc			
		00	01	11	10
a	0	0 ⁰ 0	1 ¹ 1	1 ³ 1	1 ² 1
	1	1 ⁴ 1	0 ⁵ 0	0 ⁷ 0	1 ⁶ 1



Método de Veitch-Karnaugh: 4 variables

➔ El mapa tiene 16 casillas, y cada una tiene 4 vecinas.

Vecindades:

- Casilla 0: 1, 2, 4 y 8.
- Casilla 1: 0, 3, 5 y 9.
- Casilla 2: 0, 3, 6 y 10.
- Casilla 3: 1, 2, 7 y 11.
- Casilla 4: 0, 5, 6 y 12.
- Casilla 5: 1, 4, 7 y 13.
- Casilla 6: 2, 4, 7 y 14.
- Casilla 7: 2, 4, 7 y 15.
- Casilla 8: 0, 9, 10 y 12.
- Casilla 9: 1, 8, 11 y 13.
- Casilla 10: 2, 8, 11 y 14.
- Casilla 11: 3, 9, 10 y 15.
- Casilla 12: 4, 8, 13 y 14.
- Casilla 13: 5, 9, 12 y 15.
- Casilla 14: 6, 10, 12 y 15.
- Casilla 15: 7, 11, 13 y 14.

		cd			
ab		00	01	11	10
		$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$ 0	$\bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d$ 1	$\bar{a} \cdot \bar{b} \cdot c \cdot d$ 3	$\bar{a} \cdot \bar{b} \cdot c \cdot \bar{d}$ 2
00		f(0)	f(1)	f(3)	f(2)
01		$\bar{a} \cdot b \cdot \bar{c} \cdot \bar{d}$ 4	$\bar{a} \cdot b \cdot \bar{c} \cdot d$ 5	$\bar{a} \cdot b \cdot c \cdot d$ 7	$\bar{a} \cdot b \cdot c \cdot \bar{d}$ 6
		f(4)	f(5)	f(7)	f(6)
11		$a \cdot b \cdot \bar{c} \cdot \bar{d}$ 12	$a \cdot b \cdot \bar{c} \cdot d$ 13	$a \cdot b \cdot c \cdot d$ 15	$a \cdot b \cdot c \cdot \bar{d}$ 14
		f(12)	f(13)	f(15)	f(14)
10		$a \cdot \bar{b} \cdot \bar{c} \cdot \bar{d}$ 8	$a \cdot \bar{b} \cdot \bar{c} \cdot d$ 9	$a \cdot \bar{b} \cdot c \cdot d$ 11	$a \cdot \bar{b} \cdot c \cdot \bar{d}$ 10
		f(8)	f(9)	f(11)	f(10)



Método de Veitch-Karnaugh: 4 variables

a	b	c	d	f
0	0	0	0	1
0	0	0	1	0
0	0	1	0	0
0	0	1	1	1
0	1	0	0	1
0	1	0	1	1
0	1	1	0	0
0	1	1	1	0
1	0	0	0	0
1	0	0	1	0
1	0	1	0	1
1	0	1	1	1
1	1	0	0	0
1	1	0	1	0
1	1	1	0	1
1	1	1	1	1

$$\text{Función: } f(a,b,c,d) = \sum_4 m(0,3,4,5,10,11,14,15) = \prod_4 M(1,2,6,7,8,9,12,13)$$

		cd			
		00	01	11	10
ab	00	0 1	1 0	3 1	2 0
	01	4 1	5 1	7 0	6 0
	11	12 0	13 0	15 1	14 1
	10	8 0	9 0	11 1	10 1



Método de Veitch-Karnaugh: 5 variables

Tiene 32 casillas, y se construye con 2 mapas superpuestos de 4 variables.

- Cada casilla tiene 5 vecinas: 4 en su plano más la de su misma posición en el otro plano (la casilla 0 es vecina de la 16, la 1 de la 17, etc).

de					
bc		00	01	11	10
00		f(0)	f(1)	f(3)	f(2)
01		f(4)	f(5)	f(7)	f(6)
11		f(12)	f(13)	f(15)	f(14)
10		f(8)	f(9)	f(11)	f(10)

$a = 0$

de					
bc		00	01	11	10
00		f(16)	f(17)	f(19)	f(18)
01		f(20)	f(21)	f(23)	f(22)
11		f(28)	f(29)	f(31)	f(30)
10		f(24)	f(25)	f(27)	f(26)

$a = 1$



Método de Veitch-Karnaugh: 6 variables

Tiene 64 casillas, y se construye con 4 mapas superpuestos de 4 variables.

- Cada casilla tiene 6 vecinas: las 4 de su plano, más las 2 que están en su misma posición pero en planos adyacentes.

cd \ ef					
		00	01	11	10
cd	00	f(0)	f(1)	f(3)	f(2)
	01	f(4)	f(5)	f(7)	f(6)
	11	f(12)	f(13)	f(15)	f(14)
	10	f(8)	f(9)	f(11)	f(10)

a = 0 b = 0

cd \ ef					
		00	01	11	10
cd	00	f(16)	f(17)	f(19)	f(18)
	01	f(20)	f(21)	f(23)	f(22)
	11	f(28)	f(29)	f(31)	f(30)
	10	f(24)	f(25)	f(27)	f(26)

a = 0 b = 1

cd \ ef					
		00	01	11	10
cd	00	f(32)	f(33)	f(35)	f(34)
	01	f(36)	f(37)	f(39)	f(38)
	11	f(44)	f(45)	f(47)	f(46)
	10	f(40)	f(41)	f(43)	f(42)

a = 1 b = 0

cd \ ef					
		00	01	11	10
cd	00	f(48)	f(49)	f(51)	f(50)
	01	f(52)	f(53)	f(55)	f(54)
	11	f(60)	f(61)	f(63)	f(62)
	10	f(56)	f(57)	f(59)	f(58)

a = 1 b = 1

➔ Planos adyacentes:

- El plano 00 es adyacente a los planos 01 y 10.
- El plano 01 es adyacente a los planos 00 y 11.
- El plano 11 es adyacente a los planos 01 y 10.
- El plano 10 es adyacente a los planos 11 y 00.

➔ Algunas vecindades:

- Casilla 0:1,2,4,8,16,32.
- Casilla 16:17,18,20,24,0,48.
- Casilla 55:51,53,54,63,23,39.
- Casilla 41:40,43,33,45,57,9.
- Etc.



Simplificación por V-K con minterminos

La propiedad que permite simplificar gráficamente consiste en que entre cada dos casillas vecinas (que comparten un lado, en horizontal o en vertical) sólo difiere el valor de una variable \Rightarrow **si dos casillas vecinas contienen un 1, agrupándolas se puede aplicar la propiedad distributiva para simplificar la expresión algebraica resultante eliminando la variable que cambia.**

Ejemplo: $f(a,b)=m_1+m_2+m_3+m_4+m_6$

bc \ a		00		01		11		10	
		0		1		3		2	
0		0	0	1	1	1	1	1	1
1		1	1	0	0	0	0	1	1

Agrupando m_1 con m_3

$$\begin{aligned} m_1 + m_3 &= \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot c = \\ &= \bar{a} \cdot c \cdot (\bar{b} + b) = \bar{a} \cdot c \end{aligned}$$

Agrupando m_2 con m_6

$$\begin{aligned} m_2 + m_6 &= \bar{a} \cdot b \cdot \bar{c} + a \cdot b \cdot \bar{c} = \\ &= b \cdot \bar{c} \cdot (\bar{a} + a) = b \cdot \bar{c} \end{aligned}$$

Agrupando m_4 con m_6

$$\begin{aligned} m_4 + m_6 &= a \cdot \bar{b} \cdot \bar{c} + a \cdot b \cdot \bar{c} = \\ &= a \cdot \bar{c} \cdot (\bar{b} + b) = a \cdot \bar{c} \end{aligned}$$

Por tanto: $f(a,b,c) = \bar{a} \cdot c + b \cdot \bar{c} + a \cdot \bar{c}$



Simplificación por V-K con minitérminos

Simplificación:

- **Agrupación de 1s** pertenecientes a celdas adyacentes. El objetivo es maximizar el tamaño de los grupos y minimizar el nº de grupos.
 - Un grupo puede contener 1, 2, 4, 8, 16 celdas (potencias de 2)
 - Cada celda de un grupo tiene que ser adyacente a una o más celdas del mismo grupo, pero no todas las celdas del grupo tiene que ser adyacentes entre sí.
 - Cada 1 del mapa debe estar incluido en al menos en un grupo.
- Determinación de la **operación producto mínima para cada grupo**. Dentro de cada grupo, para obtener la expresión se eliminan las variables que cambian.
- Cuando se han obtenido todos los términos mínimos se suman para obtener la expresión **suma de productos mínima**.

Implementación

- Una posible implementación se realiza con puertas NAND, una por cada término producto y otra por la suma.



Simplificación por V-K con minitérminos

⇒ Si cuatro casillas vecinas dos a dos formando una línea o un rectángulo contienen todas el valor 1, aplicando la propiedad distributiva eliminamos las dos variables que cambian.

Ejemplo: $f(a,b) = m_0 + m_1 + m_2 + m_3 + m_6 + m_7$

bc					
a		00	01	11	10
		0	1	3	2
0	1	1	1	1	
1	0	0	1	1	1

Agrupando m_0, m_1, m_2 y m_3

$$\begin{aligned} m_0 + m_1 + m_2 + m_3 &= \\ &= \bar{a} \cdot \bar{b} \cdot \bar{c} + \bar{a} \cdot \bar{b} \cdot c + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c = \\ &= \bar{a} \cdot [\bar{b} \cdot (\bar{c} + c) + b \cdot (\bar{c} + c)] = \bar{a} \end{aligned}$$

Agrupando m_6, m_7, m_2 y m_3

$$\begin{aligned} m_6 + m_7 + m_2 + m_3 &= \\ &= a \cdot b \cdot \bar{c} + a \cdot b \cdot c + \bar{a} \cdot b \cdot \bar{c} + \bar{a} \cdot b \cdot c = \\ &= b \cdot [a \cdot (\bar{c} + c) + \bar{a} \cdot (\bar{c} + c)] = b \end{aligned}$$

Por tanto:

$$f(a,b,c) = \bar{a} + b$$



Simplificación por V-K con minitérminos

⇒ Si ocho casillas vecinas dos a dos formando un rectángulo contienen todas el valor 1, aplicando la propiedad distributiva eliminamos las tres variables que cambian.

Ejemplo: $f(a,b) = m_0 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 + m_{10} + m_{11} + m_{14} + m_{15}$

cd \ ab	00	01	11	10
00	1	1	1	1
01	1	1	1	1
11	0	0	1	1
10	0	0	1	1

Agrupando $m_0, m_1, m_2, m_3, m_4, m_5, m_6$ y m_7

$$\begin{aligned} m_0 + m_1 + m_2 + m_3 + m_4 + m_5 + m_6 + m_7 &= \\ &= \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot \bar{c} \cdot d + \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot c \cdot d + \\ &+ \bar{a} \cdot b \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot b \cdot \bar{c} \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot d = \\ &= \bar{a} \cdot [\bar{b} \cdot (\bar{c} \cdot (\bar{d} + d) + c \cdot (\bar{d} + d)) + b \cdot (\bar{c} \cdot (\bar{d} + d) + c \cdot (\bar{d} + d))] = \dots = \bar{a} \end{aligned}$$

Agrupando $m_{10}, m_{11}, m_2, m_3, m_{14}, m_{15}, m_6$ y m_7

$$\begin{aligned} m_2 + m_3 + m_6 + m_7 + m_{10} + m_{11} + m_{14} + m_{15} &= \\ &= \bar{a} \cdot \bar{b} \cdot c \cdot \bar{d} + \bar{a} \cdot \bar{b} \cdot c \cdot d + \bar{a} \cdot b \cdot c \cdot \bar{d} + \bar{a} \cdot b \cdot c \cdot d + \\ &+ a \cdot \bar{b} \cdot c \cdot \bar{d} + a \cdot \bar{b} \cdot c \cdot d + a \cdot b \cdot c \cdot \bar{d} + a \cdot b \cdot c \cdot d = \\ &= c \cdot [\bar{a} \cdot (\bar{b} \cdot (\bar{d} + d) + b \cdot (\bar{d} + d)) + a \cdot (\bar{b} \cdot (\bar{d} + d) + b \cdot (\bar{d} + d))] = \dots = c \end{aligned}$$

Por tanto: $f(a,b,c) = \bar{a} + c$



Simplificación por V-K con minitérminos

Subcubo: agrupación de 2^r casillas vecinas dos a dos con el mismo valor que forman una línea, un rectángulo o un paralelepípedo.

- **El subcubo está compuesto por un número de casillas (r) que es potencia de 2** (1 casilla, 2 casillas, 4 casillas, 8 casillas, 16 casillas, etc).
- **Las casillas que componen el subcubo deben ser vecinas 2 a 2, y además estarán alineadas o formarán un rectángulo o un paralelepípedo.**
- **Todas las casillas del subcubo tendrán el mismo valor (algunas pueden tener el valor X si sirven para que el subcubo sea más grande).**

➡ En un mapa de V-K de n variables, un subcubo de 2^r casillas se simplifica así:

- Las r variables que cambian se eliminan.
- El producto de las $n-r$ variables restantes, que no cambian, constituyen la expresión simplificada del subcubo.

En un subcubo de 1 casilla no se elimina ninguna variable.

En un subcubo de 2 casillas se elimina 1 variable.

En un subcubo de 4 casillas se eliminan 2 variables.

En un subcubo de 8 casillas se eliminan 3 variables.



Simplificación por V-K con minitérminos

Para simplificar la función tenemos que incluir todas las casillas con valor 1 (ó X) en algún subcubo.

➡ El resultado final del proceso de simplificación de la función es una suma de productos (SOP, SdP) que depende de cómo escojamos los subcubos.

- Se obtiene una función más sencilla cogiendo el mínimo número de grupos posibles, y lo más grandes posibles.

➡ Procedimiento

1. Formar subcubos de 1 con las casillas sueltas (las que no se pueden agrupar con otras).
2. Formar subcubos de 2 con las casillas que sólo pueden formar subcubos de 2.
3. Formar subcubos de 4 con las casillas que queden y que puedan formar subcubos de 4 y no de 8.
4. Repetir 3 formando grupos de 8, 16, etc.
5. El proceso termina cuando todas las casillas a 1 están cogidas en algún subcubo. De cada subcubo sale un término de la expresión simplificada.



Simplificación por V-K con maxitérminos

Se realiza de forma parecida a como se hace con los minitérminos, con las siguientes diferencias:

- Los subcubos están formados por casillas con valor 0 ó X (en los minitérminos se cogían las casillas con 1 ó X).
- Al escribir el término simplificado, la complementación de las variables es la contraria.
 - Las variables que irían complementadas en minitérminos van sin complementar en maxitérminos y las variables que irían sin complementar en minitérminos van complementadas en maxitérminos.
- El término resultante de cada subcubo de casillas a 0 es una suma de literales (en minitérminos era un producto de literales).
- La función simplificada resultante es un producto de sumas (POS, PdS) (en minitérminos era una suma de productos).
- **Implementación:** puertas **NOR**



Simplificación por V-K

Ejemplo: simplificar por minitérminos y por maxitérminos la función $f(a,b,c,d)$ cuya tabla de verdad se indica a continuación.

Construimos el mapa de V-K:

cd \ ab	cd			
	00	01	11	10
00	X	0	0	1
01	0	X	0	1
11	1	0	1	0
10	1	0	0	1

a	b	c	d	f
0	0	0	0	X
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Paso 1: tomar los 1 que no puedan formar subcubos con otras casillas



Simplificación por V-K

Paso 1: tomar los 1 que no puedan formar subcubos con otras casillas.

cd		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = a \cdot b \cdot c \cdot d$$

Paso 2: tomar los 1 que sólo puedan formar subcubos de 2 casillas \Rightarrow



Simplificación por V-K

Paso 2: tomar los 1 que sólo puedan formar subcubos de 2 casillas.

cd ab		cd			
		00	01	11	10
00	X	0	0	1	
01	0	X	0	1	
11	1	0	1	0	
10	1	0	0	1	

$$S1 = a \cdot b \cdot c \cdot d$$

$$S2 = a \cdot \bar{c} \cdot \bar{d}$$

¿Repetir paso 2? \Rightarrow



Simplificación por V-K

Paso 2: tomar los 1 que sólo puedan formar subcubos de 2 casillas.

cd					
ab	cd	00	01	11	10
		00	01	11	10
00	00	X	0	0	1
01	00	0	X	0	1
11	00	1	0	1	0
10	00	1	0	0	1

$$S1 = a \cdot b \cdot c \cdot d$$

$$S2 = a \cdot \bar{c} \cdot \bar{d}$$

$$S3 = \bar{a} \cdot c \cdot \bar{d}$$

Paso 3: tomar los 1 que no estén tomados y puedan formar subcubos de 4 y no de 8



Simplificación por V-K

Paso 3: tomar los 1 que no estén tomados y puedan formar subcubos de 4 y no de 8.

cd		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = a \cdot b \cdot c \cdot d$$

$$S2 = a \cdot \bar{c} \cdot \bar{d}$$

$$S3 = \bar{a} \cdot c \cdot \bar{d}$$

$$S4 = \bar{b} \cdot \bar{d}$$

Paso 4: no ha lugar porque todos los 1 ya están cogidos en algún subcubo.

$$f_1(a,b,c,d) = a \cdot b \cdot c \cdot d + a \cdot \bar{c} \cdot \bar{d} + \bar{a} \cdot c \cdot \bar{d} + \bar{b} \cdot \bar{d}$$

⇒ Ejemplo: simplificar por minitérminos y por maxitérminos la función $f(a,b,c,d)$ cuya tabla de verdad se indica a continuación.

Construimos el mapa de V-K:

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

a	b	c	d	f
0	0	0	0	X
0	0	0	1	0
0	0	1	0	1
0	0	1	1	0
0	1	0	0	0
0	1	0	1	X
0	1	1	0	1
0	1	1	1	0
1	0	0	0	1
1	0	0	1	0
1	0	1	0	1
1	0	1	1	0
1	1	0	0	1
1	1	0	1	0
1	1	1	0	0
1	1	1	1	1

Paso 1: tomar los 0 que no puedan formar subcubos con otras casillas



Simplificación por V-K

Paso 1: tomar los 0 que no puedan formar subcubos con otras casillas.

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

Paso 2: tomar los 0 que sólo puedan formar subcubos de 2 casillas \Rightarrow



Simplificación por V-K

Paso 2: tomar los 0 que sólo puedan formar subcubos de 2 casillas \Rightarrow NO HAY NINGUNO.

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

Paso 3: tomar los 0 que no estén tomados y puedan formar subcubos de 4 y no de 8 \Rightarrow



Simplificación por V-K

Paso 3: tomar los 0 que no estén tomados y puedan formar subcubos de 4 y no de 8.

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

$$S2 = b + \bar{d}$$

¿Repetir paso 3? \Rightarrow



Simplificación por V-K: maxitérminos

Paso 3: tomar los 0 que no estén tomados y puedan formar subcubos de 4 y no de 8.

cd ab		cd			
		00	01	11	10
00	X	0	0	1	
01	0	X	0	1	
11	1	0	1	0	
10	1	0	0	1	

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

$$S2 = b + \bar{d}$$

$$S3 = a + \bar{d}$$

¿Repetir paso 3? \Rightarrow



Simplificación por V-K: maxitérminos

Paso 3: tomar los 0 que no estén tomados y puedan formar subcubos de 4 y no de 8.

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

$$S2 = b + \bar{d}$$

$$S3 = a + \bar{d}$$

$$S4 = a + c$$

¿Repetir paso 3? \Rightarrow



Simplificación por V-K: maxitérminos

Paso 3: tomar los 0 que no estén tomados y puedan formar subcubos de 4 y no de 8.

		cd			
		00	01	11	10
ab	00	X	0	0	1
	01	0	X	0	1
	11	1	0	1	0
	10	1	0	0	1

$$S1 = \bar{a} + \bar{b} + \bar{c} + d$$

$$S2 = b + \bar{d}$$

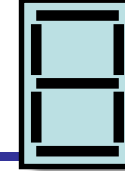
$$S3 = a + \bar{d}$$

$$S4 = a + c$$

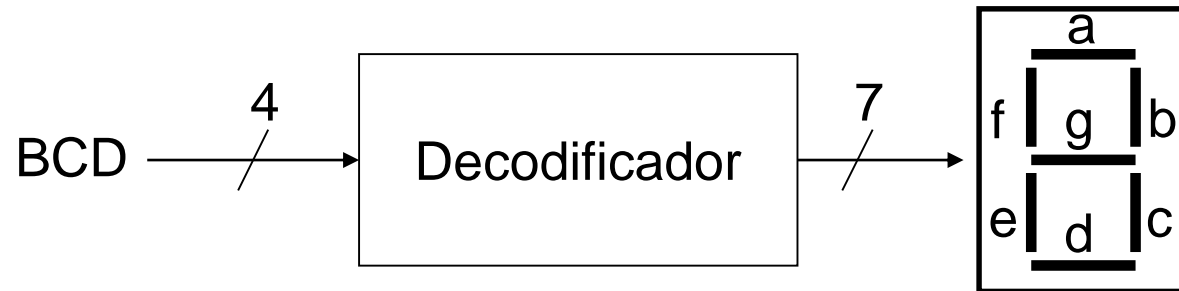
$$S5 = c + \bar{d}$$

Paso 4: no ha lugar porque todos los 0 ya están cogidos en algún subcubo.

$$f_2(a,b,c,d) = (\bar{a} + \bar{b} + \bar{c} + d) \cdot (b + \bar{d}) \cdot (a + \bar{d}) \cdot (a + c) \cdot (c + \bar{d})$$

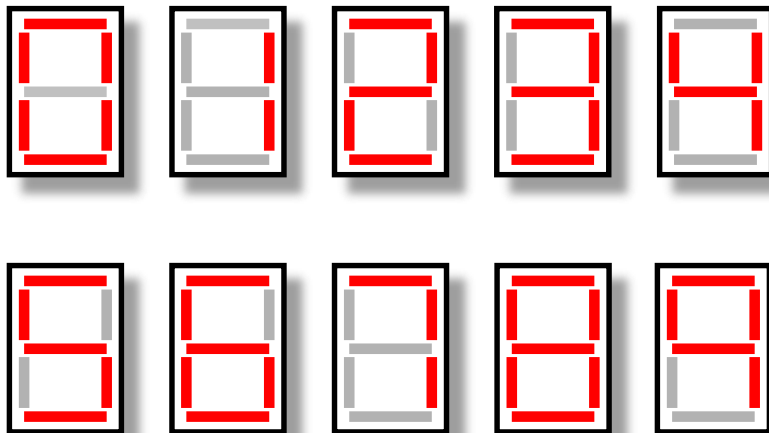


Displays de 7 segmentos



Ánodo común: nivel bajo de tensión activa el segmento

Cátodo común: nivel alto de tensión activa el segmento



Dígito	Segmentos
0	a,b,c,d,e,f
1	b,c
2	a,b,d,e,g
3	a,b,d,e,g
4	b,c,f,g
5	a,c,d,f,g
6	a,c,d,e,f,g
7	a,b,c
8	a,b,c,d,e,f,g
9	a,b,c,d,f,g