

# TEMA 2: ÁLGEBRA DE BOOLE

## Índice:

1. Introducción al Álgebra de Boole
2. Puertas Lógicas Digitales
3. Funciones Lógicas
4. Simplificación e Implementación de Funciones

# TEMA 2: ÁLGEBRA DE BOOLE

## Bibliografía:

- ❑ T.L.Floyd. Fundamentos de Sistemas Digitales.
  - Cap. 3: Puertas Lógicas
  - Cap. 4: Álgebra de Boole y Simplificación Lógica
  - Cap. 5: Lógica Combinacional
- ❑ C.Blanco. Fundamentos de Electrónica Digital.
  - Cap. 2: Álgebra de Boole y Funciones Lógicas
- ❑ J.M<sup>a</sup> Angulo y J. García. Sistemas Digitales y Tecnología de Computadores.
  - Cap 3. Álgebra de Boole
- ❑ E. Mandado. Sistemas Electrónicos Digitales.
  - Cap 2. Álgebra de Boole
  - Cap. 3 Sistemas Combinacionales

# 1. Introducción al Álgebra de Boole. Definición. Axiomas

Es un conjunto de elementos que pueden tomar dos valores perfectamente diferenciados (que representaremos con **0** y **1**) relacionados por los operadores **+** (suma lógica) y **·** (producto lógico), que cumplen los siguientes axiomas:

- Ambas operaciones son conmutativas:

$$a + b = b + a$$

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

- Existen dos elementos neutros, uno por operación:

$$0 + a = a$$

$$a \cdot 1 = a$$

- Cada operación es distributiva respecto a la otra:

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

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

- Para todo elemento  $a$  existe un elemento complementario  $\bar{a}$ , que cumple:

$$a + \bar{a} = 1$$

$$a \cdot \bar{a} = 0$$

## Leyes y Teoremas

**Principio de dualidad:** Cada identidad deducida de los anteriores axiomas permanece válida si las operaciones de suma lógica y producto lógico y los elementos 0 y 1 se intercambian entre si.

**Idempotencia:**

$$\left. \begin{array}{l} a + a = a \\ a \cdot a = a \end{array} \right\} \forall a \in B$$

$$\left. \begin{array}{l} a + 1 = 1 \\ a \cdot 0 = 0 \end{array} \right\} \forall a \in B$$

Estas dos leyes, junto con el postulado que establece la existencia del elemento neutro, definen la suma y el producto lógico:

$a + b$	S
$0 + 0$	0
$0 + 1$	1
$1 + 0$	1
$1 + 1$	1

$a \cdot b$	P
$0 \cdot 0$	0
$0 \cdot 1$	0
$1 \cdot 0$	0
$1 \cdot 1$	1

**Ley de Absorción:**

$$\left. \begin{array}{l} a + a \cdot b = a \\ a \cdot (a + b) = a \end{array} \right\} \forall a, b \in B$$

**Involución:**

$$\overline{\overline{a}} = a \quad \forall a \in B$$

**Asociatividad:**

$$\left. \begin{array}{l} a + (b + c) = (a + b) + c \\ a \cdot (b \cdot c) = (a \cdot b) \cdot c \end{array} \right\} \forall a, b, c \in B$$

**Teorema del consenso:**

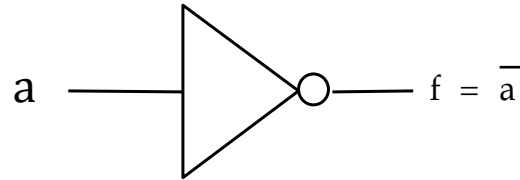
$$ab + \overline{a}c = ab + \overline{a}c + bc$$

**Teoremas de DeMorgan:**

$$\left. \begin{array}{l} \overline{a + b} = \overline{a} \cdot \overline{b} \\ \overline{a \cdot b} = \overline{a} + \overline{b} \end{array} \right\} \forall a, b \in B$$

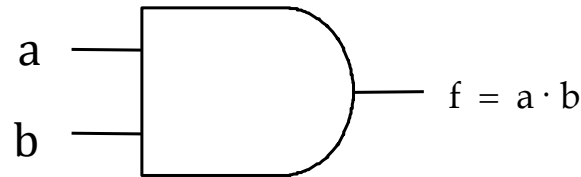
# Puertas Lógicas Digitales: Puertas Básicas

Inversor o Negador:



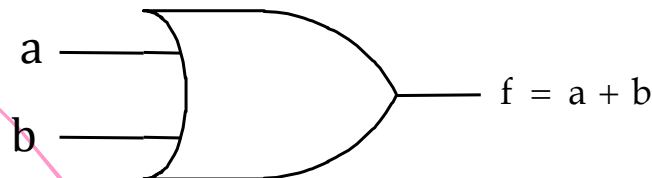
$a$	$f$
0	1
1	0

Puerta AND:



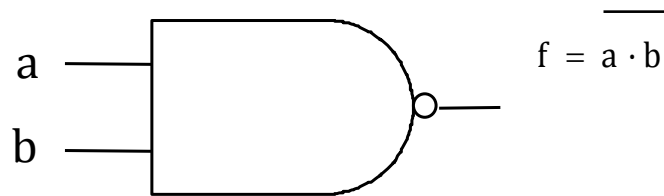
$a$	$b$	$f$
0	0	0
0	1	0
1	0	0
1	1	1

Puerta OR:



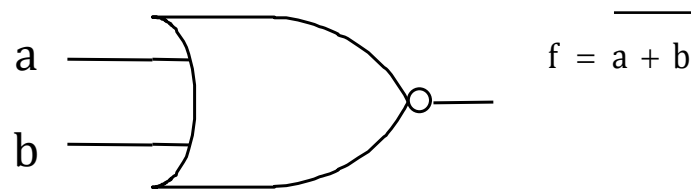
$a$	$b$	$f$
0	0	0
0	1	1
1	0	1
1	1	1

## Puerta NAND:



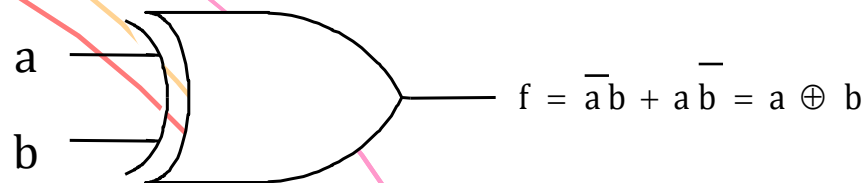
$a$	$b$	$f$
0	0	1
0	1	1
1	0	1
1	1	0

## Puerta NOR:



$a$	$b$	$f$
0	0	1
0	1	0
1	0	0
1	1	0

## Puerta OR Exclusiva (EXOR o XOR):



$a$	$b$	$f$
0	0	0
0	1	1
1	0	1
1	1	0

## Funciones Lógicas. Definición

Una función lógica es un conjunto de variables relacionadas entre si por las operaciones básicas definidas de suma lógica, producto lógico y negación:

$$f = f(a, b, c, \dots)$$

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

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

Toda función booleana se comporta como una variable del sistema.

Definimos un **término suma** como una suma de variables bien en su forma directa o complementada:

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

Si, por el contrario, dichas variables están relacionadas mediante productos lógicos diremos que se trata de un **término producto**:

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



## Funciones Lógicas. Representación estándar

Cuando relacionamos dos o más términos producto mediante la suma lógica, la expresión resultante diremos que queda expresada en forma de **Suma de Productos**:

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

Cuando relacionamos dos o más términos suma mediante el producto lógico, la expresión resultante estará expresada en forma de **Producto de Sumas**:

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

Se llama **término canónico** o **estándar** de una función lógica a todo producto o suma en la cual aparecen todas las variables que forman parte de la función, bien sea en su forma directa o inversa. En las funciones:

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

$ab\bar{c}$  y  $a\bar{b}\bar{c}$  son productos canónicos

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

$(a + \bar{b} + \bar{c})$  y  $(\bar{a} + \bar{b} + \bar{c})$  son sumas canónicas

Una función formada únicamente por términos canónicos diremos que es una función canónica o estándar.

Las expresiones en forma estándar pueden expresarse de forma mas sencilla a través de su equivalente numérico:

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

$$\overline{a}\overline{b}\overline{c} = 000_2 = 0_{10}$$

$$\overline{a}b\overline{c} = 010_2 = 2_{10}$$

$$a\overline{b}\overline{c} = 100_2 = 4_{10}$$

$$abc = 111_2 = 7_{10}$$

De forma similar:

$$f(a, b, c) = \overline{a}\overline{b}\overline{c} + \overline{a}b\overline{c} + a\overline{b}\overline{c} + abc = \sum_3 (0, 2, 4, 7)$$

$$f = (a + b + \overline{c} + d)(a + \overline{b} + \overline{c} + d)(a + b + c + \overline{d})(\overline{a} + \overline{b} + \overline{c} + \overline{d}) = \prod_4 (2, 6, 1, 15)$$

## Equivalencia entre Formatos.

Sea la función expresada en forma de suma de productos:

$$f(a, b, c) = \bar{a}\bar{b}c + a\bar{b}\bar{c} + a\bar{b}c = \sum_3 (1, 4, 5)$$

La expresión negada de esta función estará compuesta por todos los elementos que no la cumplen:

$$\bar{f} = \sum_3 (0, 2, 3, 6, 7)$$

En forma algebraica:

$$\bar{f} = \bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + abc$$

Y puesto que

$$f = \bar{\bar{f}} = \overline{\bar{a}\bar{b}\bar{c} + \bar{a}b\bar{c} + \bar{a}bc + a\bar{b}\bar{c} + abc}$$

Y aplicando ahora los teoremas de DeMorgan:

$$\begin{aligned}
 f &= \overline{\overline{a} \overline{b} \overline{c} + \overline{a} b \overline{c} + \overline{a} b c + a b \overline{c} + a b c} = \\
 &= \overline{\overline{a} \overline{b} \overline{c}} \cdot \overline{\overline{a} b \overline{c}} \cdot \overline{\overline{a} b c} \cdot \overline{a b \overline{c}} \cdot \overline{a b c} = \\
 &= (a + b + c)(a + \overline{b} + c)(a + \overline{b} + \overline{c})(\overline{a} + \overline{b} + c)(\overline{a} + \overline{b} + \overline{c})
 \end{aligned}$$

Luego:

$$f = \sum_3 (1, 4, 5) = \prod_3 (0, 2, 3, 6, 7)$$

# Tabla de Verdad

Una tabla de verdad de una función lógica es una forma de representación de la misma, en la que se indica el valor (0 ó 1) que toma la función para cada una de las combinaciones posibles de las variables de las cuales depende, expresadas en forma de suma de productos.

$$f = \bar{a}\bar{b}c + \bar{a}b\bar{c} + a\bar{b}\bar{c} + abc$$

$$f = \sum_3 (1, 3, 4, 7)$$

$$f = \prod_3 (0, 2, 5, 6)$$

abc	f
000	0
001	1
010	0
011	1
100	1
101	0
110	0
111	1

# Simplificación mediante Álgebra de Boole

Se basa en la aplicación sistemática de los axiomas, teoremas y leyes ya comentadas. Partiendo de la expresión canónica, básicamente se apoya en la propiedad:

$$a b c + \dots + \bar{a} b c = b c + \dots$$

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

Ejemplos:

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

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

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

$$\begin{aligned} (a + b + c)(a + b + \bar{c})(a + \bar{b} + c) &= \\ = [(a + b) + c \bar{c}] [(a + c) + b \bar{b}] &= (a + b)(a + c) \end{aligned}$$

# Simplificación mediante Tablas (o mapas) de Karnaugh

Se basa en sistematizar la aplicación del método algebraico ya descrito mediante la construcción de tablas.

Estas tablas están constituidas por celdas a las que asignaremos una combinación de tal forma que cada una de ellas esta rodeada únicamente por otras en las que difiere en una sola variable.

De 3 variables:

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

De 4 variables:

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

De 5 Variables:

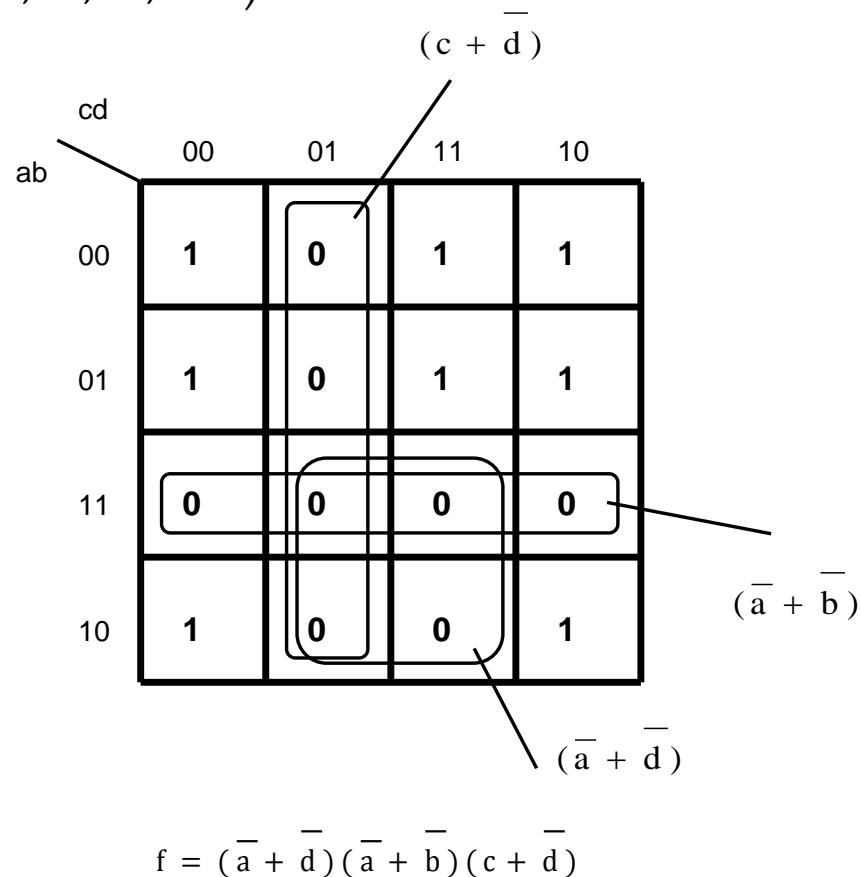
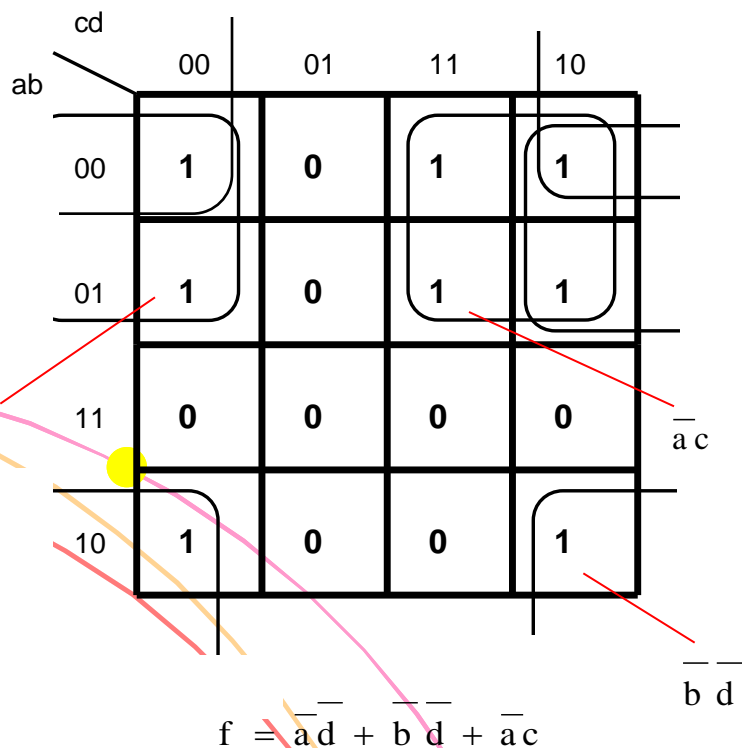
cde		000	001	011	010	110	111	101	100
ab	00	0	1	3	2	6	7	5	4
	01	8	9	11	10	14	15	13	12
	11	24	25	27	26	30	31	29	28
	10	16	17	19	18	22	23	21	20

Para completar las tablas procederemos como si se tratase de una tabla de verdad, colocando un 1 en las casillas de los términos que cumplen la función en forma de suma de productos y un 0 en las que no.



## Simplificación en forma de PoS y SoP

$$f = \sum_4 (0, 2, 3, 4, 6, 7, 8, 10)$$



$$f = \sum (0, 2, 4, 6, 9, 13, 16, 18, 20, 22, 26, 29, 30, 31)$$

		cde							
		000	001	011	010	110	111	101	100
ab	00	1	0	0	1	1	0	0	1
	01	0	1	0	0	0	0	1	0
	11	0	0	0	1	1	1	1	0
	10	1	0	0	1	1	0	0	1

$\bar{a}\bar{b}\bar{d}e$  (points to cell 011, 01)  
 $abce$  (points to cell 111, 11)  
 $\bar{b}\bar{e}$  (points to cell 000, 00)  
 $ad\bar{e}$  (points to cell 010, 00)

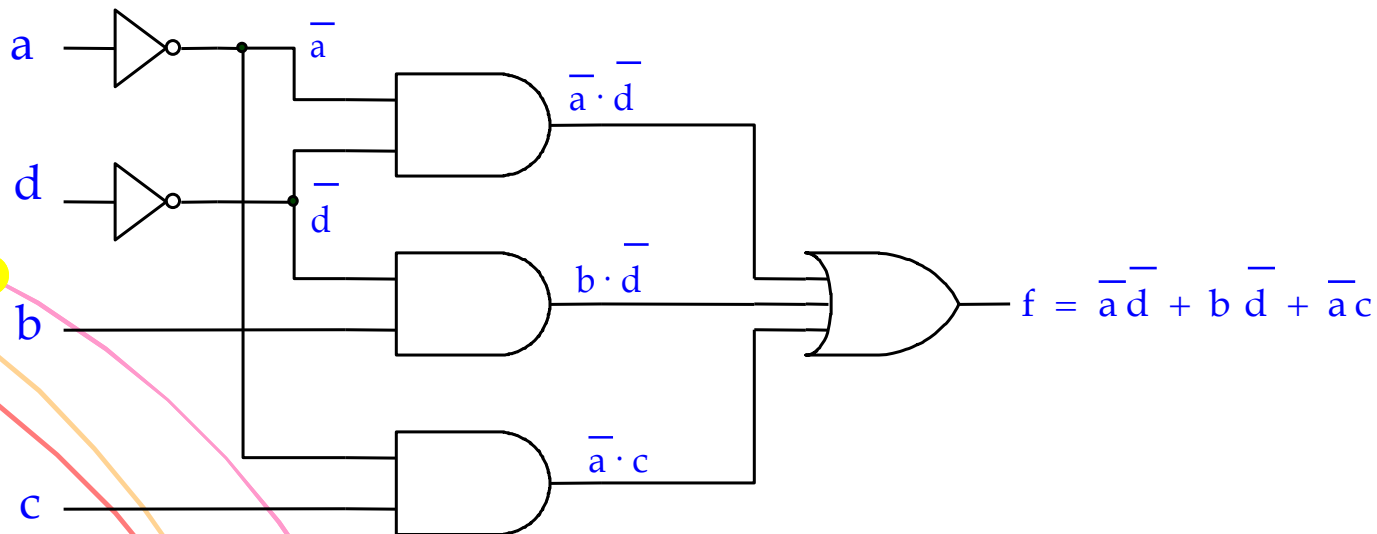
$$f = (\bar{a}\bar{b}\bar{d}e + abce + \bar{b}\bar{e} + ad\bar{e})$$

## Conjuntos Completos. Implementación

Un conjunto completo esta compuesto por un grupo de puertas mínimo necesario que permita implementar cualquier función:

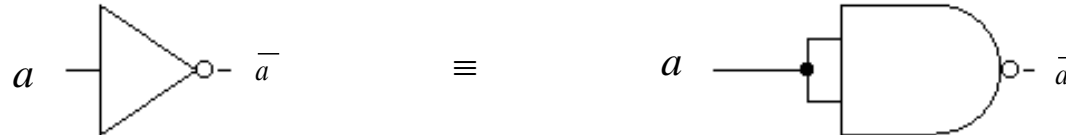
Ejemplos: Inversor, puerta AND y puerta OR. Puerta NAND. Puerta NOR

$$f = \overline{a} \overline{d} + b \overline{d} + \overline{a} c$$

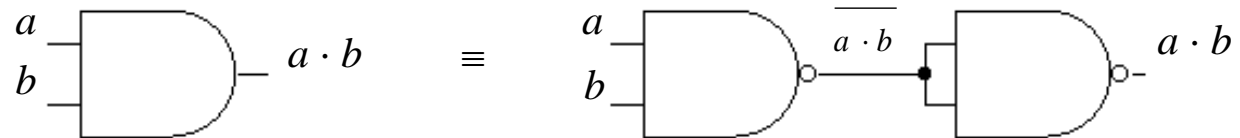


# Puerta NAND como Elemento Universal

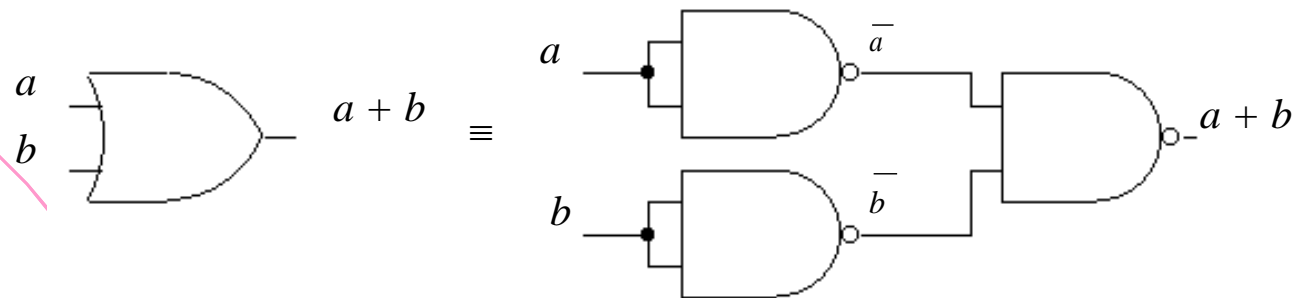
Inversor:



Puerta AND:



Puerta OR:

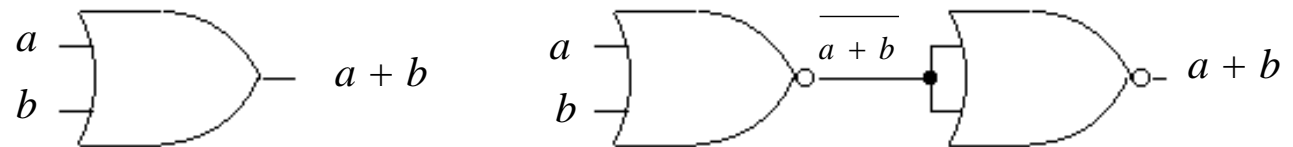


# Puerta NOR como Elemento Universal

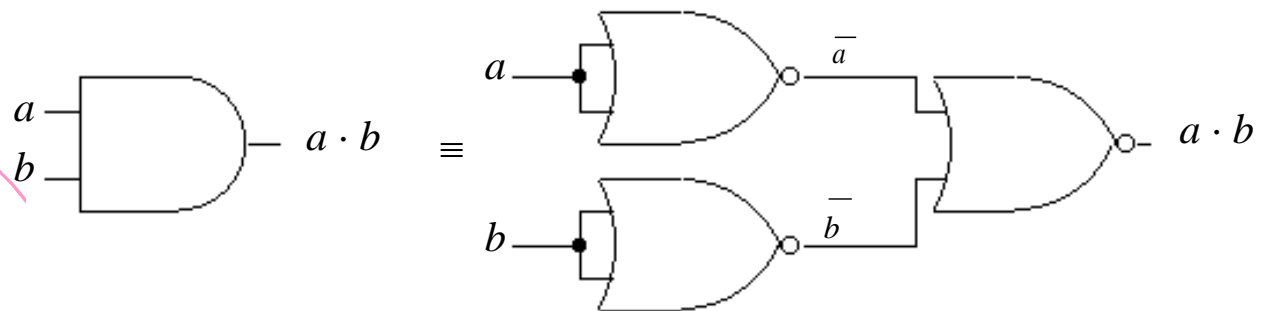
Inversor:



Puerta OR:

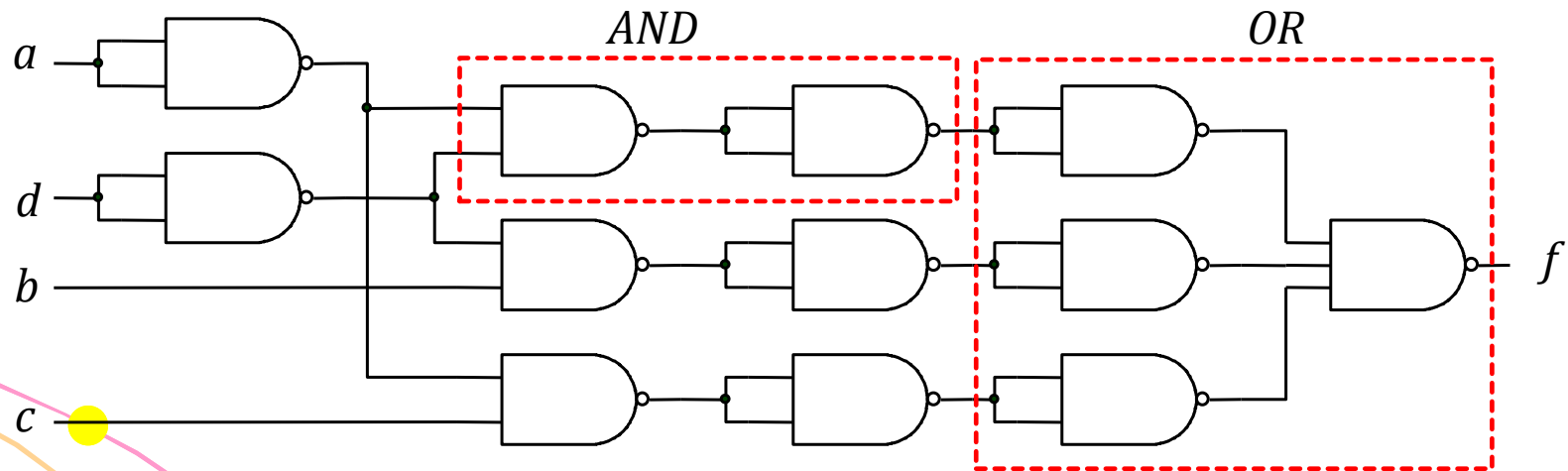


Puerta AND:



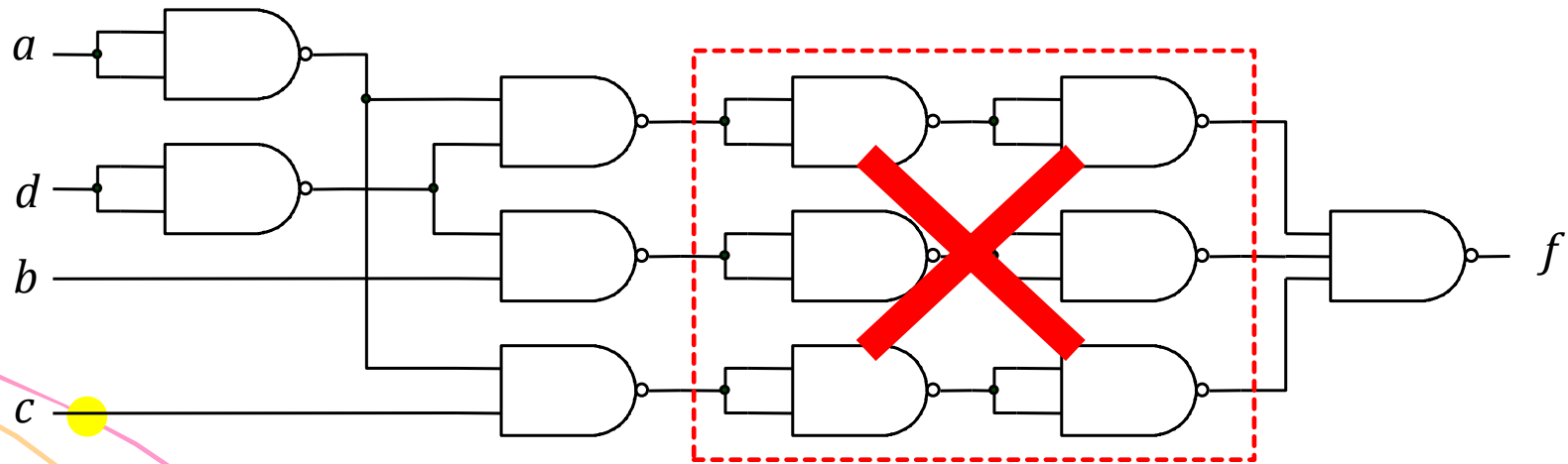
# Conjuntos completos. Implementación

$$f = \overline{a} \overline{d} + b \overline{d} + \overline{a} c$$



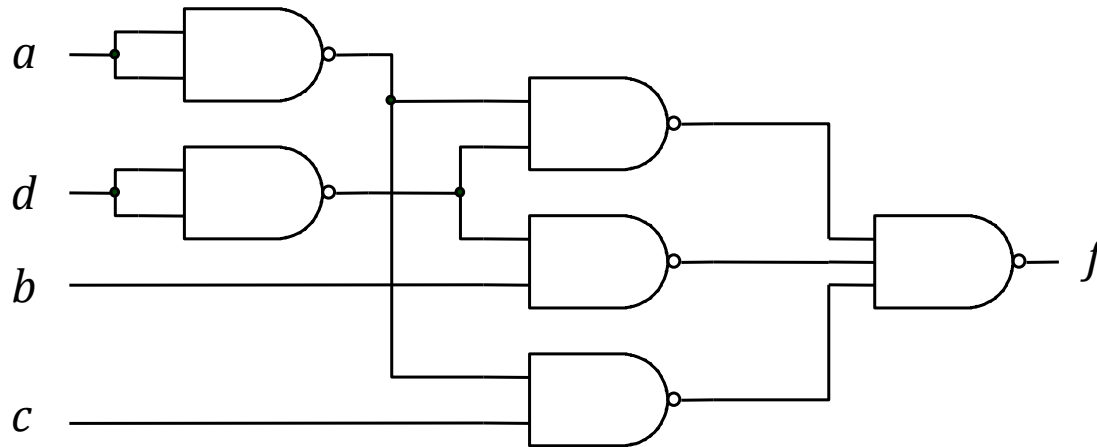
# Conjuntos completos. Implementación

$$f = \overline{a} \overline{d} + b \overline{d} + \overline{a} c$$



## Conjuntos completos. Implementación

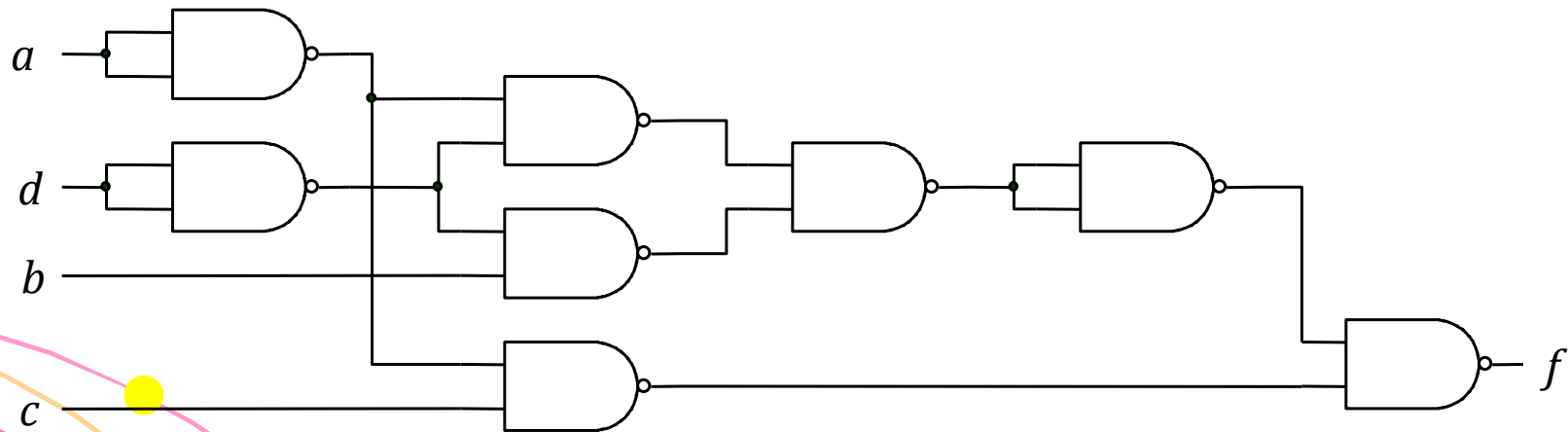
$$f = \overline{a} \overline{d} + b \overline{d} + \overline{a} c = \overline{\overline{\overline{a} \overline{d}} + \overline{b \overline{d}} + \overline{\overline{a} c}} = \overline{\overline{a} \overline{d}} \cdot \overline{b \overline{d}} \cdot \overline{\overline{a} c}$$





# Conjuntos completos. Implementación

$$f = \overline{a} \overline{d} + b \overline{d} + \overline{a} c = \overline{\overline{\overline{a} \overline{d} + b \overline{d} + \overline{a} c}} = \overline{\overline{\overline{a} \overline{d} + b \overline{d} + \overline{a} c}} = \overline{\overline{a} \overline{d} + b \overline{d} + \overline{a} c} = \overline{\overline{a} \overline{d} + b \overline{d} + \overline{a} c}$$



# Funciones Incompletas. Simplificación

Son aquellas que no tienen un valor definido para todas las posibles combinaciones de las variables de las que dependen.

abc	f
000	0
001	X
010	X
011	1
100	1
101	0
110	X
111	0

$$f = \sum_3 (3, 4) + \sum_{\emptyset} (1, 2, 6)$$

		bc			
		00	01	11	10
a	0	0	X	1	X
	1	1	0	0	X

$$f = \bar{a}b + a\bar{c}$$