

TEMA 5.2

FUNCIONES LÓGICAS

CUD

TEMA 5
SISTEMAS DIGITALES
FUNDAMENTOS DE
ELECTRÓNICA



Centro Universitario
de la Defensa Zaragoza

17 de febrero de 2015

TEMA 5.2 – FUNCIONES LÓGICAS

- Puertas lógicas
- Simplificación de funciones lógicas

TEMA 5.2 – FUNCIONES LÓGICAS

- Puertas lógicas
- Simplificación de funciones lógicas

PUERTAS LÓGICAS

- Son circuitos electrónicos que realizan las funciones básicas del Álgebra de Boole.
- Para cada puerta utilizaremos un símbolo
- Listado:
 - AND
 - OR
 - NOT
 - NAND
 - NOR
 - XOR
 - XNOR

PUERTA AND

Símbolo

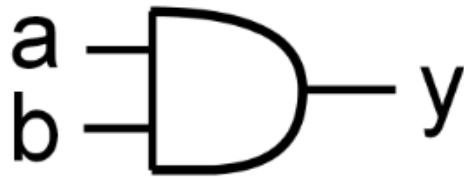


Tabla verdad

a	b	y
0	0	0
0	1	0
1	0	0
1	1	1

Expresión booleana

$$a \cdot b = y$$

Leyes del álgebra booleana

$$a \cdot 0 = 0$$

$$a \cdot 1 = a$$

$$a \cdot a = a$$

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

PUERTA OR

Símbolo

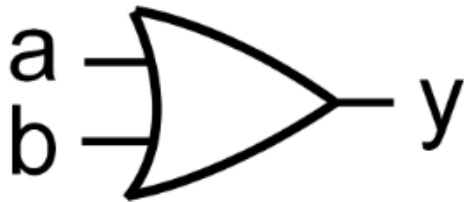


Tabla verdad

a	b	y
0	0	0
0	1	1
1	0	1
1	1	1

Expresión booleana

$$a+b = y$$

Leyes del álgebra booleana

$$a+0 = a$$

$$a+1 = 1$$

$$a+a = a$$

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

PUERTA NOT

Símbolo



Expresión booleana

$$y = \bar{a}$$

Tabla verdad

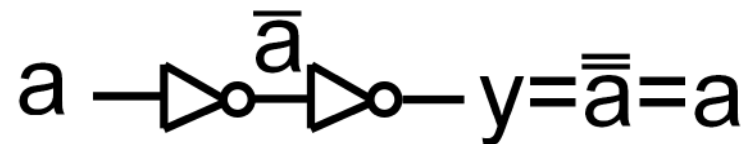
a	y
0	1
1	0

Leyes del álgebra booleana

$$\text{Si } a=1 \leftrightarrow \bar{a}=0$$

$$\text{Si } a=0 \leftrightarrow \bar{a}=1$$

$$\bar{\bar{a}}=a$$



Doble inversión

PUERTA NAND

Símbolo

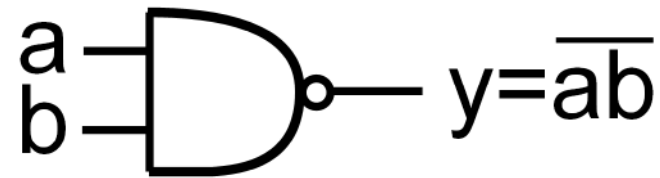
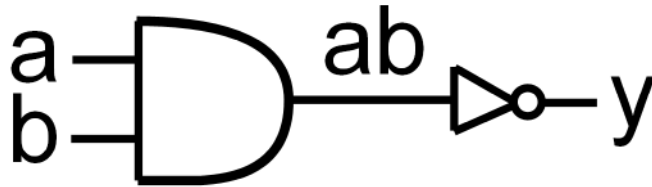


Tabla verdad

a	b	AND	NAND
0	0	0	1
0	1	0	1
1	0	0	1
1	1	1	0

PUERTA NOR

Símbolo

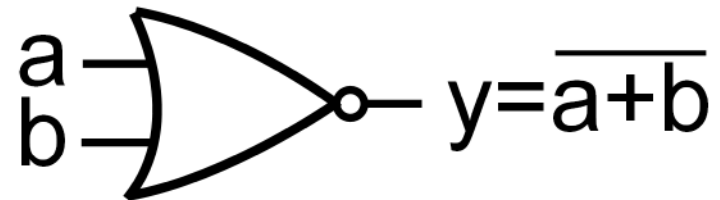
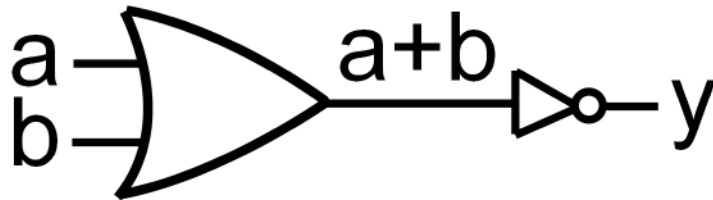
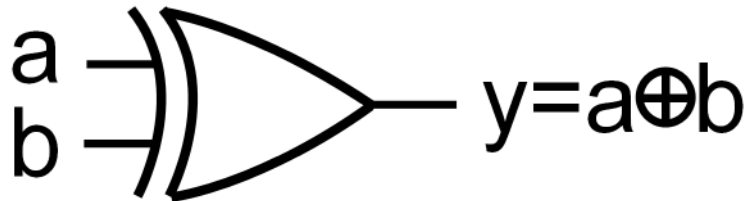


Tabla verdad

a	b	OR	NOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

PUERTA OR-EXCLUSIVA

Símbolo



Expresión booleana

$$y = \bar{a}b + a\bar{b}$$

Circuito

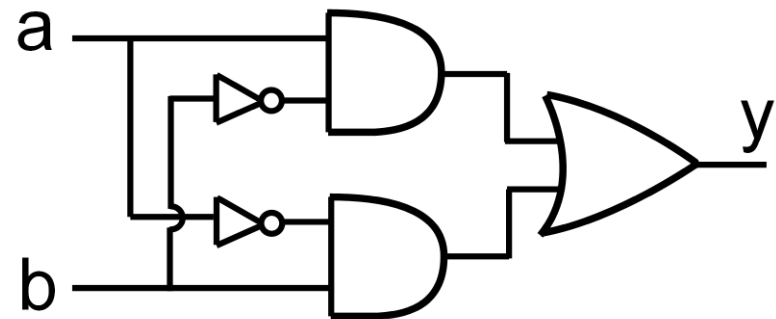


Tabla verdad

a	b	y
0	0	0
0	1	1
1	0	1
1	1	0

NOTA: Para un número de entradas mayor, la salida será uno si el número de entradas en uno es impar y cero si el número de entradas en uno es par

PUERTA NOR-EXCLUSIVA

Símbolo



Tabla verdad

a	b	XOR	XNOR
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

COMBINACIONES: INVERTIR SALIDAS

- Hemos formado la puerta NAND, NOR y XNOR basándonos en inversores a la salida.
- Se puede generalizar dando más combinaciones

Puerta lógica	+	Inversor a la salida	=	Puerta lógica
AND	+	NOT	=	NAND
OR	+	NOT	=	NOR
NAND	+	NOT	=	AND
NOR	+	NOT	=	OR

COMBINACIONES: INVERTIR ENTRADAS

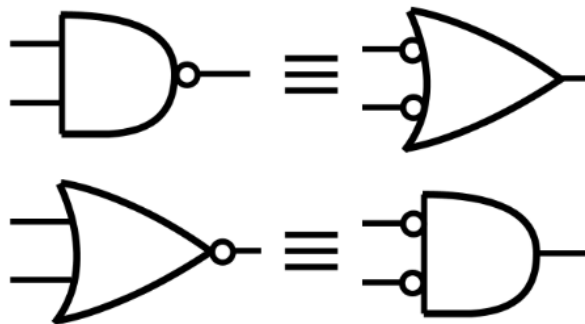
- Al incluir inversores en las entradas, las combinaciones resultantes se rigen por las leyes de Morgan
- **Cuidado: cambia la función lógica!!!**

Inversores a las entradas	+	Puerta lógica	=	Puerta lógica
NOTs	+	AND	=	NOR
NOTs	+	OR	=	NAND
NOTs	+	NAND	=	OR
NOTs	+	NOR	=	AND

Leyes de Morgan

$$\overline{a \cdot b} = \overline{a} + \overline{b}$$

$$\overline{a + b} = \overline{a} \cdot \overline{b}$$



COMBINACIONES: DOBLE INVERSIÓN

- Por tanto, al incluir inversores tanto a la entrada como a la salida de una puerta digital
- **Cuidado: cambia la función lógica!!!**

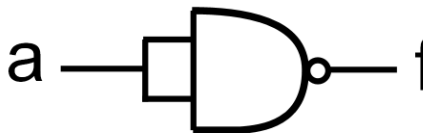
Inversores a las entradas	+	Puerta lógica	+	Inversor a la salida	=	Puerta lógica
NOTs	+	AND	+	NOT	=	OR
NOTs	+	OR	+	NOT	=	AND
NOTs	+	NAND	+	NOT	=	NOR
NOTs	+	NOR	+	NOT	=	NAND

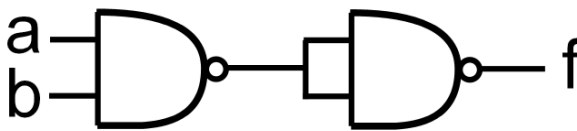
PUERTA UNIVERSAL

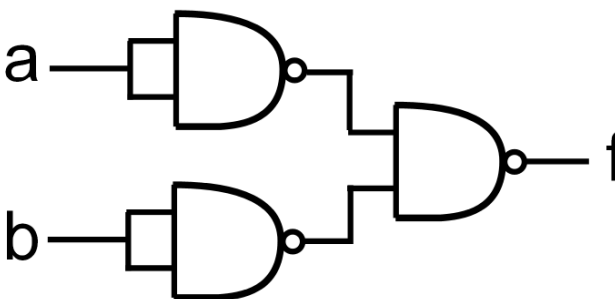
- La puerta NAND y la puerta NOR son universales, es decir, cualquier función lógica se puede representar solo con una de ellas.
- A menudo es más sencillo y económico a la hora de realizar un circuito emplear sólo un tipo de puerta lógica. En varias familias lógicas las puertas NAND son las más simples, por lo que resulta útil poder construir circuitos usando sólo éstas.
- Utilizando las leyes de Morgan podemos intercambiar OR por AND o viceversa.
- La negación se implementa con una puerta (NAND o NOR) con las dos entradas unidas

NAND COMO PUERTA UNIVERSAL

➤ Basándonos en puerta NAND podemos implementar:

– NOT:  $f = \overline{a \cdot a} = \bar{a}$

– AND:  $f = \overline{\overline{a \cdot b}} = a \cdot b$

– OR:  $f = \overline{\bar{a} \cdot \bar{b}} = a + b$

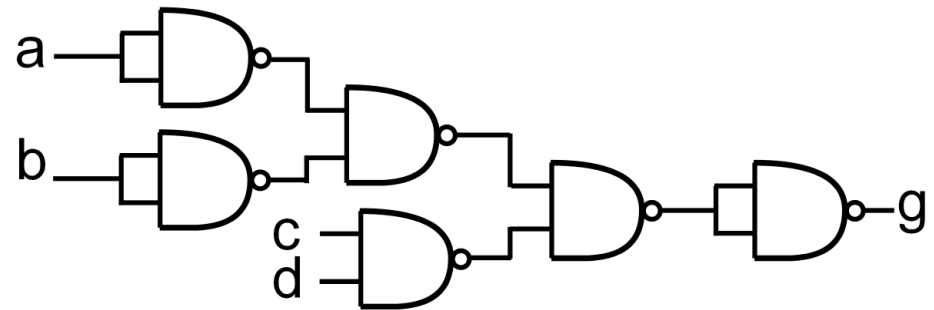
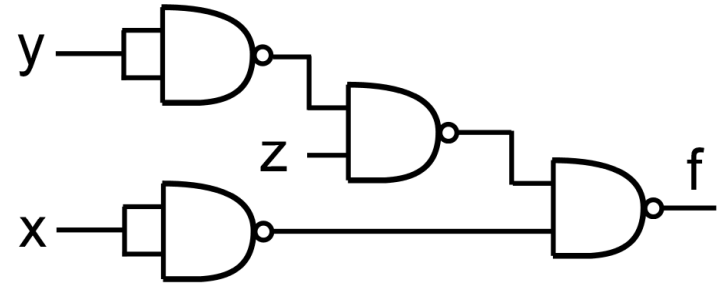
➤ Y, por lo tanto, cualquier función booleana.

NAND COMO PUERTA UNIVERSAL

- Mediante doble inversión y las leyes de Morgan conseguimos una expresión equivalente sin sumas booleanas:

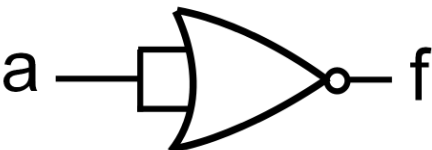
$$\begin{aligned} f(x, y, z) &= x + \bar{y}z = \\ &= \overline{\overline{x + \bar{y}z}} = \overline{\bar{x} \cdot \bar{\bar{y}z}} \end{aligned}$$

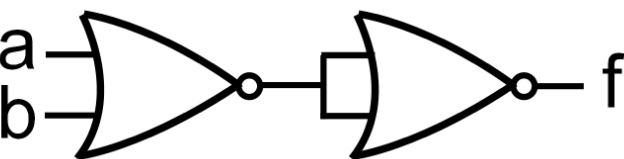
$$\begin{aligned} g(a, b, c, d) &= (a + b)(\bar{c} + \bar{d}) = \\ &= \overline{\overline{a + b}} \cdot \overline{\overline{\bar{c} + \bar{d}}} = \overline{\bar{a}\bar{b}} \cdot \overline{cd} \end{aligned}$$

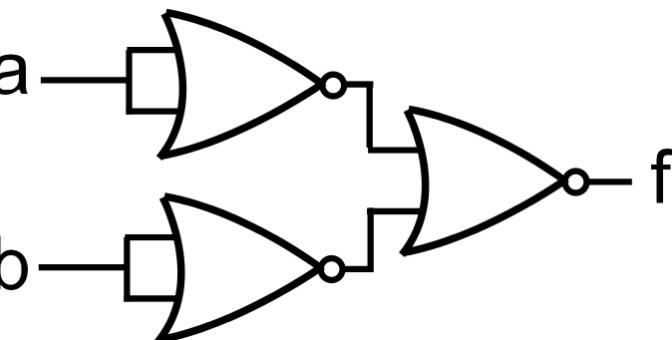


NOR COMO PUERTA UNIVERSAL

➤ Análogamente con puertas NOR:

– NOT:  $f = \overline{a+a} = \bar{a}$

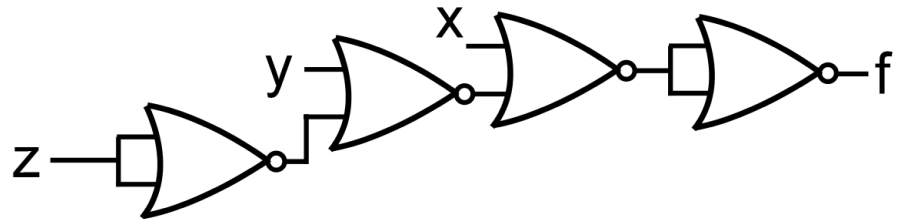
– OR:  $f = \overline{\overline{a+b}} = a+b$

– AND:  $f = \overline{\overline{a} + \overline{b}} = a \cdot b$

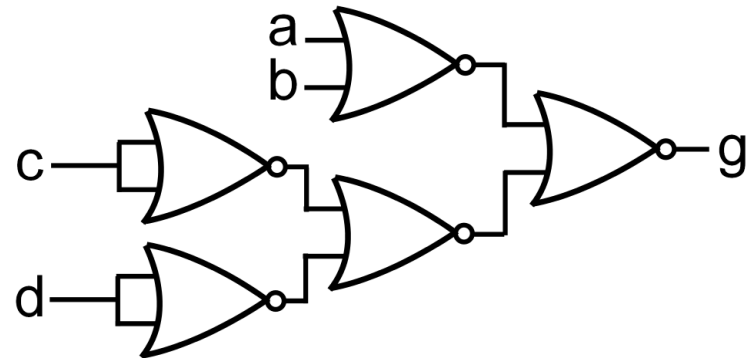
NOR COMO PUERTA UNIVERSAL

- Mediante doble inversión y las leyes de Morgan conseguimos una expresión equivalente sin productos booleanos:

$$\begin{aligned} f(x, y, z) &= x + \bar{y}z = \\ &= x + \overline{\bar{y}z} = x + \overline{\bar{y} + \bar{z}} \end{aligned}$$



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

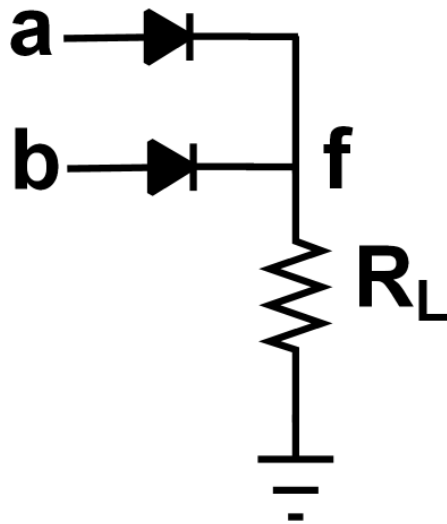


IMPLEMENTACIONES

- Las puertas lógicas se pueden implementar con diversas combinaciones de elementos electrónicos:
 - **DL: Lógica diodo**
 - **RTL: Lógica transistor-resistencia**
 - DTL: Lógica diodo-transistor
 - HTL: Lógica de alto umbral (incorpora zeners)
 - TTL: Lógica transistor-transistor
 - ECL: Lógica de emisor acoplado
 - **NMOS: Solo transistores NMOS**
 - **CMOS: transistores NMOS y PMOS**
 - BiCMOS: transistores BJT, NMOS y PMOS

LÓGICA DIODO

- Se pueden formar la puerta AND y OR
- No se puede invertir (puerta NOT)
- Ejemplo:



Análisis

a	b	f
0V	0V	0V
0V	5V	4,3V
5V	0V	4,3V
5V	5V	4,3V

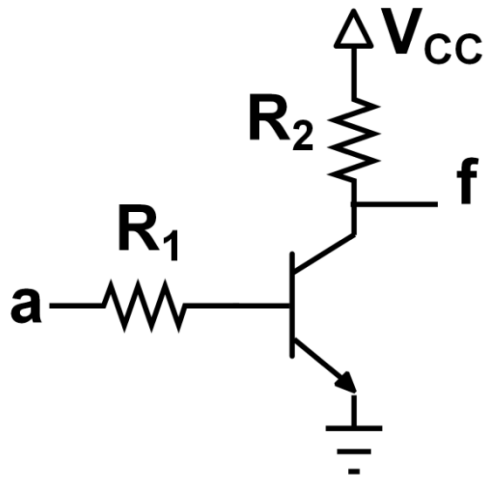
Tabla verdad

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

Puerta OR

LÓGICA TRANSISTOR

- Se puede invertir



Análisis

a	BJT	f
0V	Corte	V_{CC} (5V)
5V	Saturación	0,2V

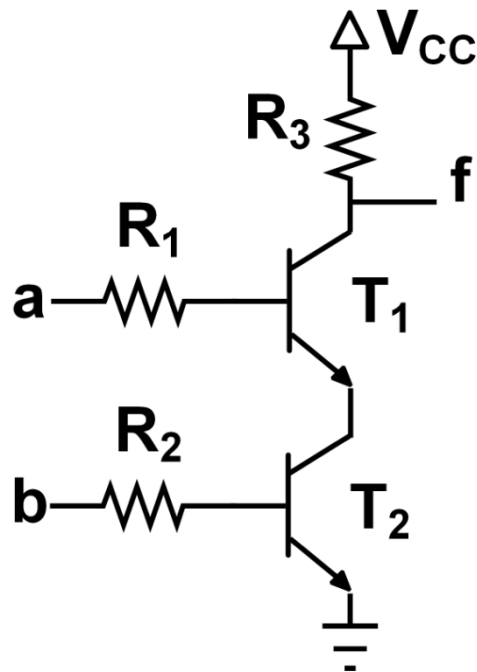
Tabla verdad

a	f
0	1
1	0

Puerta NOT

LÓGICA TRANSISTOR

➤ La puerta NAND es muy simple



Puerta NAND

Análisis

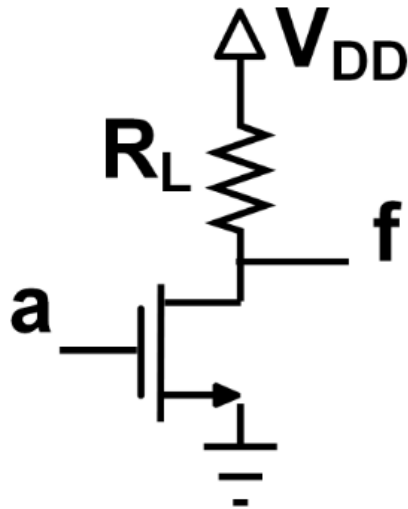
a	b	BJTs	f
0V	0V	Corte	V_{CC} (5V)
0V	5V	T_1 Corte	V_{CC} (5V)
5V	0V	T_2 Corte	V_{CC} (5V)
5V	5V	Saturación	0,4 V

Tabla verdad

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

TECNOLOGÍA NMOS

- Consiste en implementar circuitos digitales solo con transistores NMOS
- Nos basamos en el siguiente circuito, formado por un transistor NMOS y una resistencia R_L



Puerta NOT

Análisis

a	NMOS	f
0V	Corte	V_{DD} (5V)
5V	Triodo	V_{OL} ($\approx 1V$)

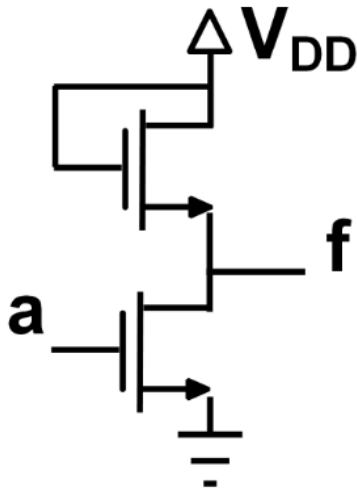
Tabla verdad

a	f
0	1
1	0

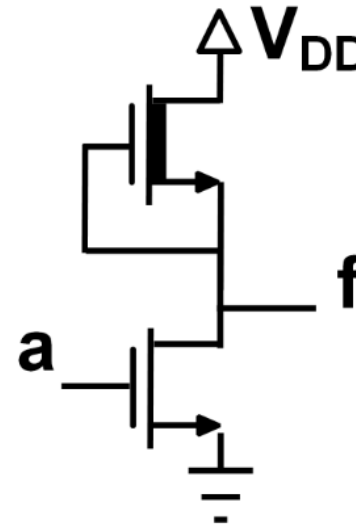
TECNOLOGÍA NMOS

- La resistencia R_L se puede emular con un NMOS por medio de dos configuraciones:

Carga saturada

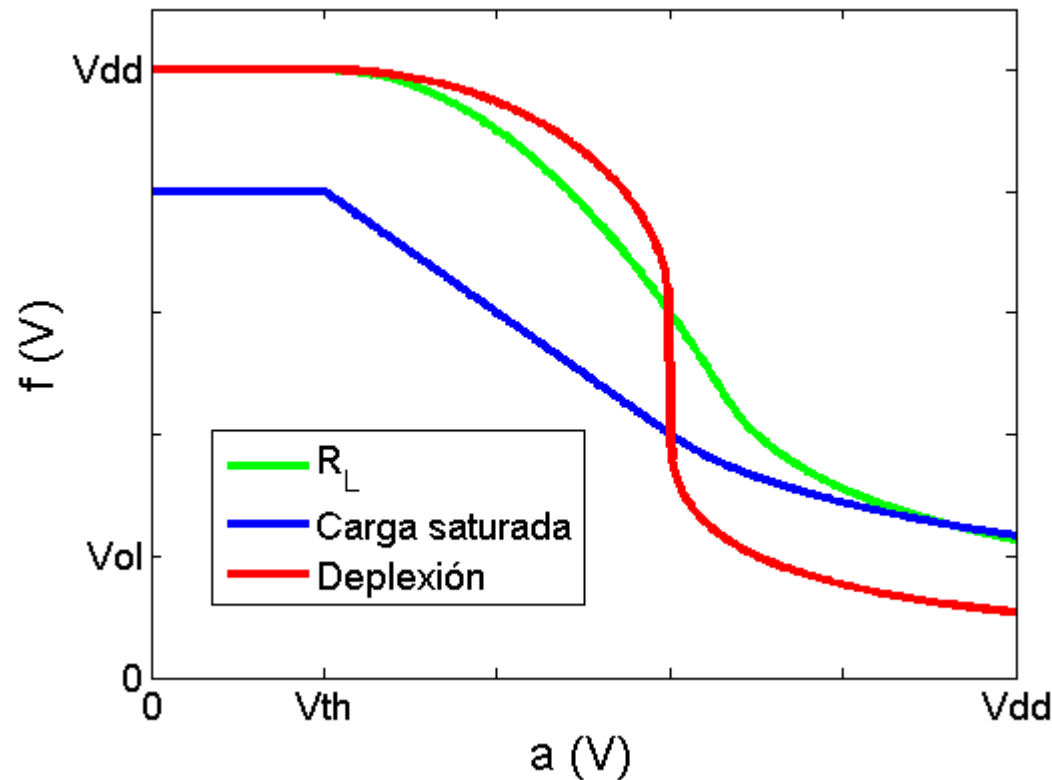


NMOS de deplexión



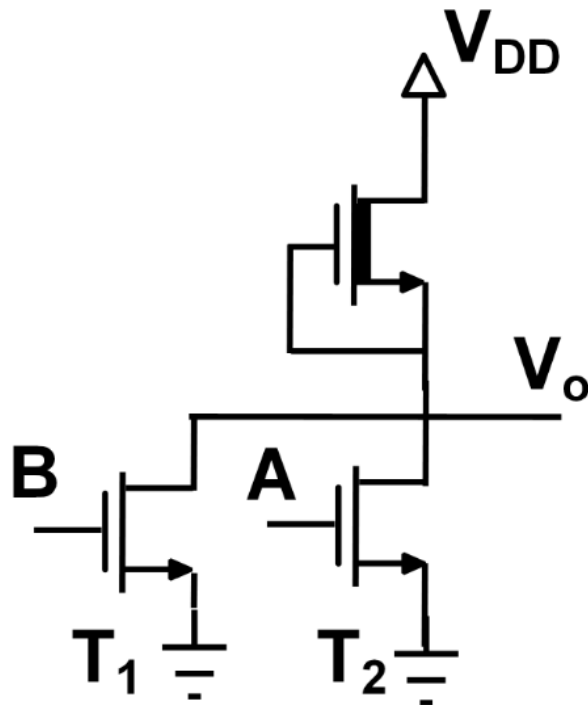
TECNOLOGÍA NMOS

- Comparando ambas propuestas, el inversor con carga saturada tiene el inconveniente de que para tensiones de entrada (V_i) pequeñas, la salida toma un valor inferior a V_{DD}



TECNOLOGÍA NMOS

- Interconectando varios transistores NMOS podemos implementar funciones lógicas.
- Ejemplo:



A	B	NMOS	V_o
0V	0V	Corte	V_{DD} (5V)
0V	5V	T_1 Triodo T_2 Corte	V_{OL}
5V	0V	T_1 Corte T_2 Triodo	V_{OL}
5V	5V	Triodo	V_{OL}

Tabla verdad

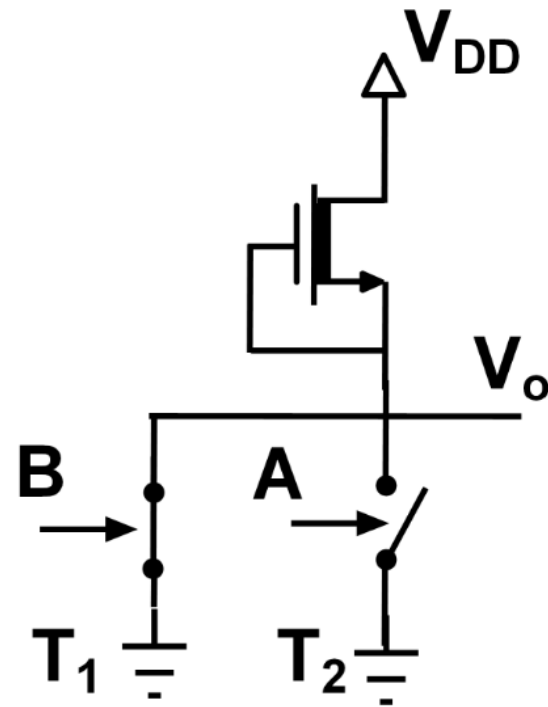
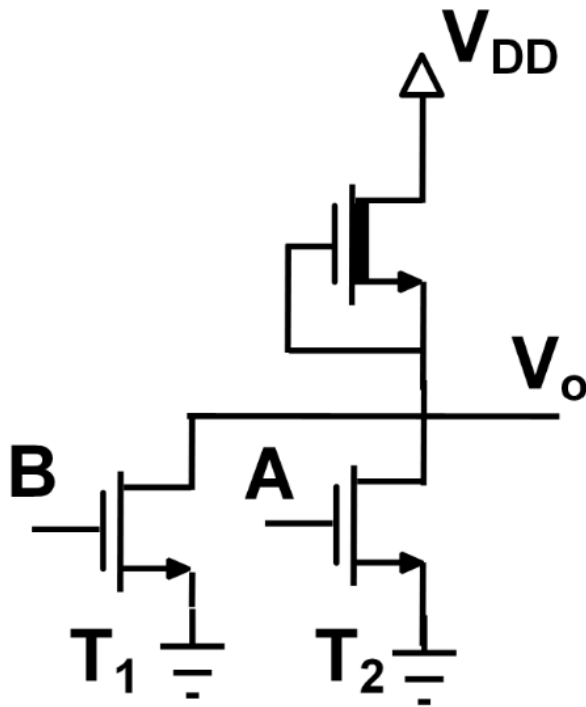
Puerta NOR

A	B	V_o
0	0	1
0	1	0
1	0	0
1	1	0

27

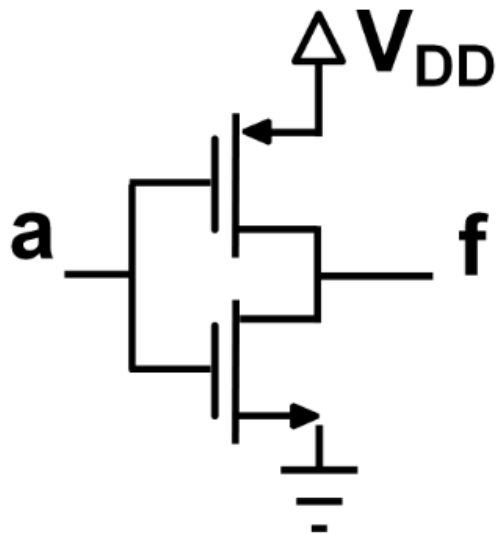
TECNOLOGÍA NMOS

- Los transistores NMOS controlados por las señales digitales se comportan como conmutadores
 - Si existe camino desde la salida hasta tierra, la salida toma valor 0
 - Sino, la salida toma valor 1



TECNOLOGÍA CMOS

- Consiste en implementar circuitos digitales con transistores NMOS y PMOS, que sustituyen al NMOS de vaciamiento o carga saturada
- Una misma señal digital controla un par NMOS-PMOS, los cuales tendrán diferentes estados (corte-triodo o triodo-corte)



Puerta NOT

Análisis

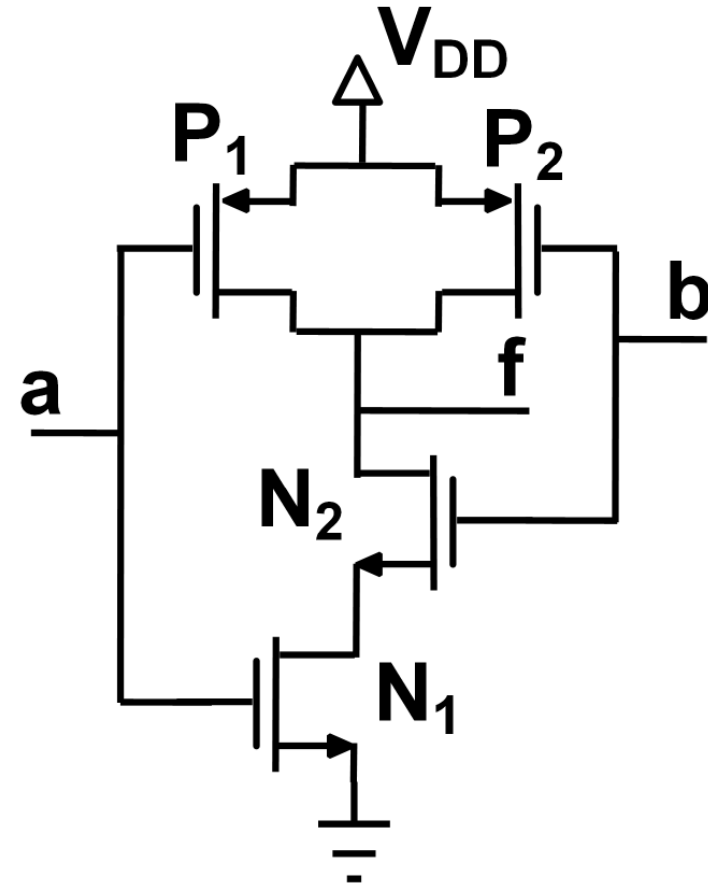
a	NMOS	PMOS	f
0V	Corte	Triodo	V_{DD} (5V)
5V	Triodo	Corte	0V

Tabla verdad

a	f
0	1
1	0

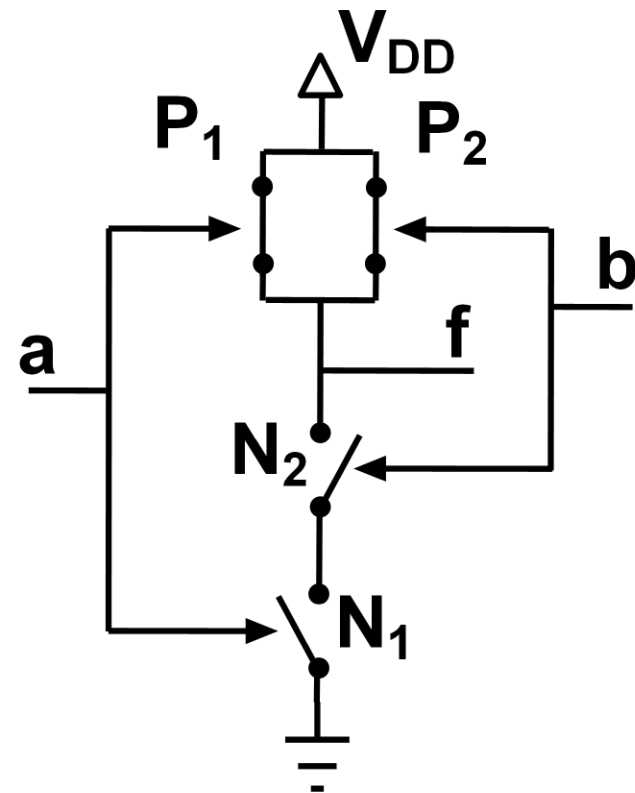
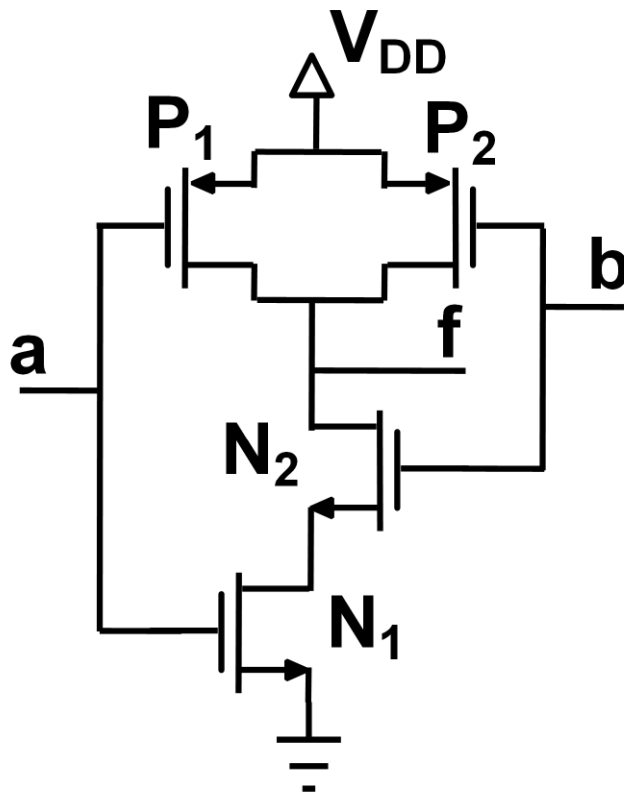
TECNOLOGÍA CMOS

- Al igual que en la tecnología NMOS, las funciones lógicas se implementan con varios transistores NMOS, todos en el plano inferior
- Por cada transistor NMOS, habrá un transistor PMOS en el plano superior
- Los transistores PMOS se interconectan de manera complementaria (serie-paralelo) a los NMOS



TECNOLOGÍA CMOS

- Los transistores NMOS y PMOS se comportan como conmutadores
- Está garantizado que existe camino desde la salida a alimentación o a tierra, pero nunca a ninguno o a ambos

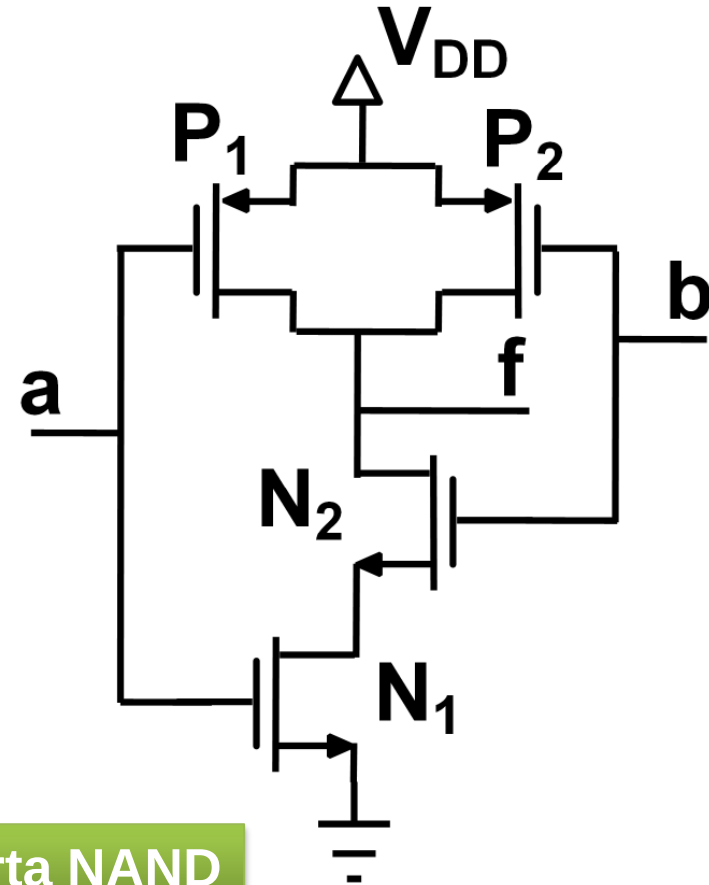


TECNOLOGÍA CMOS

a	b	N1	P1	N2	P2	f
0V	0V	OFF	ON	OFF	ON	V_{DD} (5V)
0V	5V	OFF	ON	ON	OFF	V_{DD} (5V)
5V	0V	ON	OFF	OFF	ON	V_{DD} (5V)
5V	5V	ON	OFF	ON	OFF	0V

Tabla verdad

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



Puerta NAND

VENTAJAS TECNOLOGÍA CMOS

- Los transistores MOS se imponen en el diseño de electrónica digital gracias a: bajo consumo de potencia, alta densidad de integración y bajo coste
- La tecnología CMOS presenta dos ventajas:
 - La salida toma los valores más diferenciados posibles (0 – 5V), por lo que no presenta problemas derivados de encadenar varios circuitos en cascada.
 - Nunca existe conexión entre la alimentación (V_{DD}) y tierra (o el PMOS o el NMOS están en corte) por lo que se reduce el consumo de potencia.
 - El circuito no consume potencia mientras permanezca en un estado (con salida en 1 o en 0)
 - El circuito solo consumirá potencia cuando conmute entre los dos valores de la salida

TEMA 5.2 – FUNCIONES LÓGICAS

- Puertas lógicas
- Simplificación de funciones lógicas

SIMPLIFICACIÓN DE FUNCIONES LÓGICAS

- Una función lógica tiene múltiples expresiones equivalentes. La forma más sencilla dará lugar a una implementación mejor.
- Criterios de optimización:
 - Tamaño:
 - Menor número de puertas lógicas
 - Puertas lógicas con el menor número de entradas
 - Retardo:
 - Menor número de puertas lógicas desde una entrada hasta la salida

MINITÉRMINO

- Expresión: un producto en el que aparecen todas las variables, negadas o no
- Una función que se define con un único minitérmino tiene una tabla de verdad con un 1 en una posición y 0 en todas las demás
- Regla para obtener la expresión:
 - 0 → variable negada
 - 1 → variable sin negar

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



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

MAXITÉRMINO

- Expresión: una suma en la que aparecen todas las variables, negadas o no
- Una función que se define con un único maxitérmino tiene una tabla de verdad con un 0 en una posición y 1 en todas las demás
- Regla para obtener la expresión:
 - 1 → variable negada
 - 0 → variable sin negar

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



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

Cuidado: Al contrario que los minitérminos!!!!

FORMA CANÓNICA

- Una función se puede expresar como la suma de los minitérminos para los que la función vale 1
- o como el producto de los maxitérminos para los que la función vale 0

Primera forma canónica

$$f(a,b,c) = \sum m(0,3,5) = (\bar{a}\bar{b}\bar{c}) + (\bar{a}bc) + (a\bar{b}c)$$

Segunda forma canónica

$$f(a,b,c) = \prod M(1,2,4,6,7) = (a+b+\bar{c})(a+\bar{b}+c)(\bar{a}+b+c)(\bar{a}+\bar{b}+c)(\bar{a}+\bar{b}+\bar{c})$$

	a b c	f
0	0 0 0	1
1	0 0 1	0
2	0 1 0	0
3	0 1 1	1
4	1 0 0	0
5	1 0 1	1
6	1 1 0	0
7	1 1 1	0

ADYACENCIAS

- Dos términos son adyacentes si son idénticos excepto por un literal, que aparece negado en un término y no negado en el otro.
- Los dos términos se simplifican en uno sólo con eliminación del literal que los diferencia.

$$G = x\bar{y}\bar{z}u + x\bar{y}zu = x\bar{y}u(\bar{z} + z) = x\bar{y}u1 = x\bar{y}u$$

- Por dualidad

$$\begin{aligned} H &= (x + \bar{y} + \bar{z} + u)(x + \bar{y} + z + u) = (x + \bar{y} + u) + (\bar{z}z) \\ &= x + \bar{y} + u + 0 = x + \bar{y} + u \end{aligned}$$

MAPA DE KARNAUGH

- Mapa que presenta la tabla de verdad de una función de manera que los términos adyacentes son contiguos:
 - Una casilla para cada combinación o término
 - Las casillas se numeran en código Gray
 - **Para dos variables: 00 01 11 10**
 - En un mapa de n variables, cada casilla tiene n casillas adyacentes que se corresponden con las combinaciones que resultan de invertir el valor de cada una de las n variables

NUMERACIÓN DE LAS CASILLAS

➤ De dos variables

a \ b	0	1
0	0	1
1	2	3

➤ De tres variables

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

➤ De cuatro variables

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

NUMERACIÓN DE LAS CASILLAS

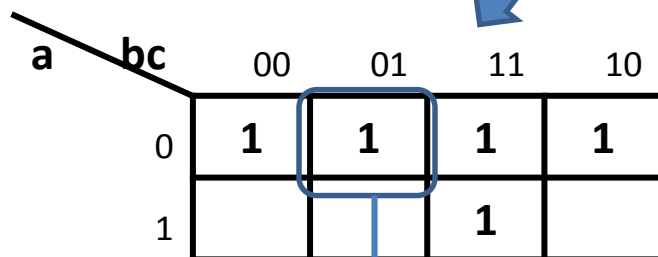
➤ De 6 variables (para examen, máx. 4 variables)

		d=0				d=1			
abc	def	000	001	011	010	110	111	101	100
a=0	000	0	1	3	2	6	7	5	4
	001	8	9	11	10	14	15	13	12
	011	24	25	27	26	30	31	29	28
	010	16	17	19	18	22	23	21	20
a=1	110	48	49	51	50	54	55	53	52
	111	56	57	59	58	62	63	61	60
	101	40	41	43	42	46	47	45	44
	100	32	33	35	34	38	39	37	36

REPRESENTACIÓN DE UNA FUNCIÓN

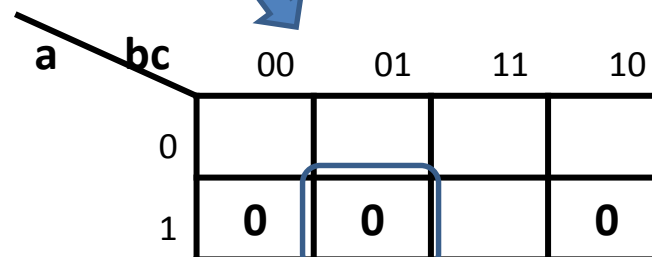
- Se marcan las casillas que corresponden a los minitérminos de la función con unos y el resto con ceros
- O se marcan los maxitérminos de la función con ceros y el resto con unos

$$f = \sum (0,1,2,3,7) = \prod (4,5,6)$$



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

$$m_1 = \bar{a}\bar{b}c$$



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

$$M_5 = \bar{a} + b + \bar{c}$$

SIMPLIFICACIÓN MEDIANTE KARNAUGH

- Dos opciones
 - Por minitérminos (unos): se obtiene una suma de productos
 - Por maxitérminos (ceros): se obtiene un producto de sumas
- Buscar grupos de casillas adyacentes
 - Un grupo de 2 casillas adyacentes elimina 1 variable
 - Un grupo de 4 casillas adyacentes elimina 2 variables
 - Un grupo de 8 casillas adyacentes elimina 3 variables
 - Un grupo de 16 casillas adyacentes elimina 4 variables
 -
- Objetivo: cubrir todos los minitérminos (maxitérminos) con los grupos más grandes posibles y con el menor número de grupos

SIMPLIFICACIÓN MEDIANTE KARNAUGH

➤ Algoritmo sistemático

1. Cubrir las casillas que no pueden formar grupos de 2
2. Cubrir las casillas que pueden formar grupos de 2, pero no de 4
3. Cubrir las casillas que pueden formar grupos de 4, pero no de 8
4. Cubrir las casillas que pueden formar grupos de 8, pero no de 16
5. ...

➤ Si en algún paso hay más de una opción:

- Comenzar siempre cubriendo las casillas que tienen menos opciones

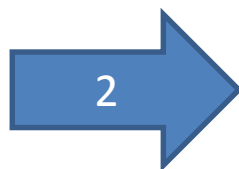
EJEMPLO

$$f = \sum (1,6,7,11,14,15) = \bar{a}\bar{b}\bar{c}d + \bar{a}bcd + \bar{a}bcd + abcd + abcd + \bar{a}bcd$$

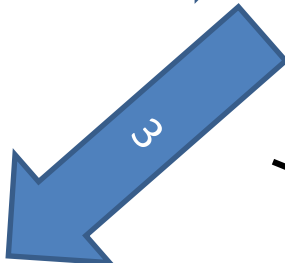
ab \ cd	00	01	11	10
00		1		
01			1	1
11			1	1
10			1	



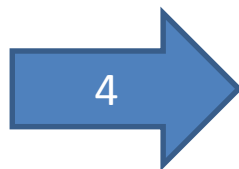
ab \ cd	00	01	11	10
00		1		
01			1	1
11			1	1
10			1	



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



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



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

EJEMPLO

➤ $f = \sum(3,5,6,7,9,13) = \bar{a}\bar{b}cd + \bar{a}b\bar{c}d + \bar{a}bcd\bar{c} + \bar{a}bcd + ab\bar{c}d + a\bar{b}\bar{c}d$

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



- $g_1: 3-7: \bar{a}cd$
- $g_2: 5-7: \bar{a}bd$
- $g_3: 6-7: \bar{a}bc$
- $g_4: 5-13: b\bar{c}d$
- $g_5: 9-13: a\bar{c}d$

- Al agrupar aparecen 5 grupos de dos, de los cuales solo son necesarios cuatro para coger todos los unos, y por tanto hay dos soluciones equivalentes

$$f = g_1 + g_3 + g_5 + g_2 = \bar{a}cd + \bar{a}bc + a\bar{c}d + \bar{a}bd$$

$$f = g_1 + g_3 + g_5 + g_4 = \bar{a}cd + \bar{a}bc + a\bar{c}d + b\bar{c}d$$