FUNCTIONES y EXPRESIONES BOOLEANAS

Gregorio Hernández
UPM

Matemática Discreta I

ÁLGEBRAS DE BOOLE $\{0, 1\}^n$

El conjunto $B = \{0,1\}$ con las operaciones $\vee$ y $\wedge$ definidas por:

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

es un álgebra de Boole y, por tanto, también lo es $B^n = B\times \cdots \times B$

Cambiamos, a partir de ahora, la notación para las operaciones

SUMA	$a + b$	$\leftrightarrow$	$a \vee b$	DISYUNCIÓN
PRODUCTO	$a \cdot b$	$\leftrightarrow$	$a \wedge b$	CONJUNCIÓN
Complementario	a'		$0' = 1, \ 1' = 0$	

ÁLGEBRAS DE BOOLE $\{0, 1\}^n$

El conjunto B = {0,1} con las operaciones $\vee$ y $\wedge$ definidas por:

$\vee$	0	1
0	0	1
1	1	1

$\wedge$	0	1
0	0	0
1	0	1

es un álgebra de Boole y, por tanto, también lo es $B^n = B\times \cdots \times B$

Las operaciones en B^n (componente a componente)

Por ejemplo en B^2

SUMA	$(a_1, a_2) + (b_1, b_2) = (a_1 + b_1, a_2 + b_2)$
PRODUCTO	$(a_1, a_2) \cdot (b_1, b_2) = (a_1 \cdot b_1, a_2 \cdot b_2)$
Complementario	$(a_1, a_2)' = (a'_1, a'_2)$

LÓGICA PROPOSICIONAL

“Enrique va la fiesta si Carlos no va y Beatriz sí va”

Asociamos una variable booleana a cada suceso

E Enrique va a la fiesta

C Carlos va a la fiesta

B Beatriz va a la fiesta

Cada variable es
o VERDAD o FALSO

E = B **AND** (**NOT** C)

E = B **AND** C’

“Enrique va a la fiesta si Carlos va o Beatriz va”

E = B **OR** C

Significa que E es verdad si uno o ambos de B y C son verdad

ÁLGEBRAS DE BOOLE

OPERACIONES	OPERADORES LÓGICOS		
COMPLEMENTARIO	NEGACIÓN	$\neg$	**NOT**
SUMA	DISYUNCIÓN	$\vee$	**OR**
PRODUCTO	CONJUNCIÓN	$\wedge$	**AND**
0	FALSO		
1	VERDAD		

SUMA	$a + b$	a **OR** b	$a \vee b$
PRODUCTO	$a \cdot b$	a **AND** b	$a \wedge b$
Complementario	a'	**NOT** a	$\neg\, a$

$0' = 1, \quad 1' = 0$

FUNCIONES BOOLEANAS

Una función booleana es una aplicación de un álgebra de Boole en B

Una función booleana de orden n es una aplicación de B^n en B

$F: B^2 \rightarrow B$

F(1,1)
F(1,0)
F(0,1)
F(0,0)

x	y	F(x,y)
1	1	1
1	0	0
0	1	0
0	0	1

Tabla de verdad

FUNCIONES BOOLEANAS

Circuito con dos interruptores

Cuando cambia una variable, cambia el resultado

$$F: B^2 \rightarrow B$$

F(1,1)
F(1,0)
F(0,1)
F(0,0)

x	y	F(x,y)
1	1	1
1	0	0
0	1	0
0	0	1

Tabla de verdad

$$F(x,y) = xy + x'y'$$

Expresión booleana

FUNCTIONES BOOLEANAS. REPRESENTACIÓN

- Tabla de verdad

x	y	F(x,y)
1	1	0
1	0	1
0	1	0
0	0	0

- Expresiones booleanas $xyz + x'z + ((x + z)y)'$

  - Las variables x_1 , x_2 , ... , x_n son expresiones booleanas.
  - Los símbolos 0 y 1 son expresiones booleanas.
  - Si E_1 y E_2 son expresiones booleanas, entonces $E_1 + E_2$, $E_1 \cdot E_2$ y E'_1 son expresiones booleanas.
  - No hay más expresiones booleanas que las obtenidas por las reglas anteriores

FUNCIONES BOOLEANAS. REPRESENTACIÓN

1) Toda expresión booleana define una función booleana

La expresión $E(x_1, x_2, \ldots, x_n)$ en n variables define la función
$F: B^m \to B, \forall m \geq n,$ $F(b_1, b_2, \ldots, b_m) = E(b_1, \ldots, b_n)$.

x	y	z	xy	z’	F(x,y,z) = xy + z’
1	1	1	1	0	1
1	1	0	1	1	1
1	0	1	0	0	0
1	0	0	0	1	1
0	1	1	0	0	0
0	1	0	0	1	1
0	0	1	0	0	0
0	0	0	0	1	1

La expresión E representa la función F

FUNCIONES BOOLEANAS. REPRESENTACIÓN

2) Dada una función booleana siempre existe una expresión booleana que la representa

x	y	z	F(x,y,z)
1	1	1	1
1	1	0	1
1	0	1	0
1	0	0	1
0	1	1	0
0	1	0	1
0	0	1	0
0	0	0	1

$$S(F) = \{b \in B^n \,/\, F(b)=1\}$$

$$S(F) = \{111, 110, 100, 010, 000\}$$

$$\Downarrow$$

xyz, xyz', $xy'z'$, $x'yz'$, $x'y'z'$

$$E_F(x,y,z) = xyz + xyz' + xy'z' + x'yz' + x'y'z'$$

Suma de productos elementales

FUNCIONES BOOLEANAS. REPRESENTACIÓN

$E = xy + z'$ $\longrightarrow$ F $\longrightarrow$

$$E_F(x,y,z) = xyz + xyz' + xy'z' + x'yz' + x'y'z'$$

E y E_F representan la misma función booleana F

Expresiones booleanas son equivalentes si definen la misma función

Dada una expresión booleana E, siempre existe otra E' equivalente a E y en forma de suma de productos elementales

PROBLEMA de SIMPLIFICACIÓN
Elegir, de todas las expresiones que representan una función booleana, la más sencilla de todas

EXPRESIONES BOOLEANAS. SIMPLIFICACIÓN

Elegir, de todas las expresiones que representan una función booleana, la más sencilla de todas

Aplicar las propiedades del álgebra de Boole

- Mapas de Karnaugh
- Método de Quine-McCluskey

Ambos procedimientos parten de una expresión en forma de suma de productos elementales y minimizan con estos objetivos:

menor número de sumandos

menor número de factores en cada sumando

EXPRESIONES BOOLEANAS. SIMPLIFICACIÓN

Teorema.

Si E es una expresión booleana en n variables entonces las expresiones E y $zE + z'E$ son equivalentes.

$$xyz + x'yz = (x + x')yz = 1yz = yz$$

$E = xy + z'$ $\Rightarrow$ F $\Rightarrow$

$$E_F(x,y,z) = xyz + xyz' + xy'z' + x'yz' + x'y'z'$$

$$= xyz + xz'(y + y') + x'z'(y + y') = xyz + xz' + x'z' =$$

$$= xyz + (x + x')z' = xyz + z'$$

$$E_F(x,y,z) = xyz + xyz' + xyz' + xy'z' + x'yz' + x'y'z'$$

$$= xy(z + z') + z' = xy + z'$$

MAPA DE KARNAUGH

Válido para simplificar expresiones en 2, 3 ó 4 variables

xy	xy’
x’y	x’y’

	y		y’	
x	xyz’	xyz	xy’z	xy’z’
x’	x’yz’	x’yz	x’y’z	x’y’z’
		z		

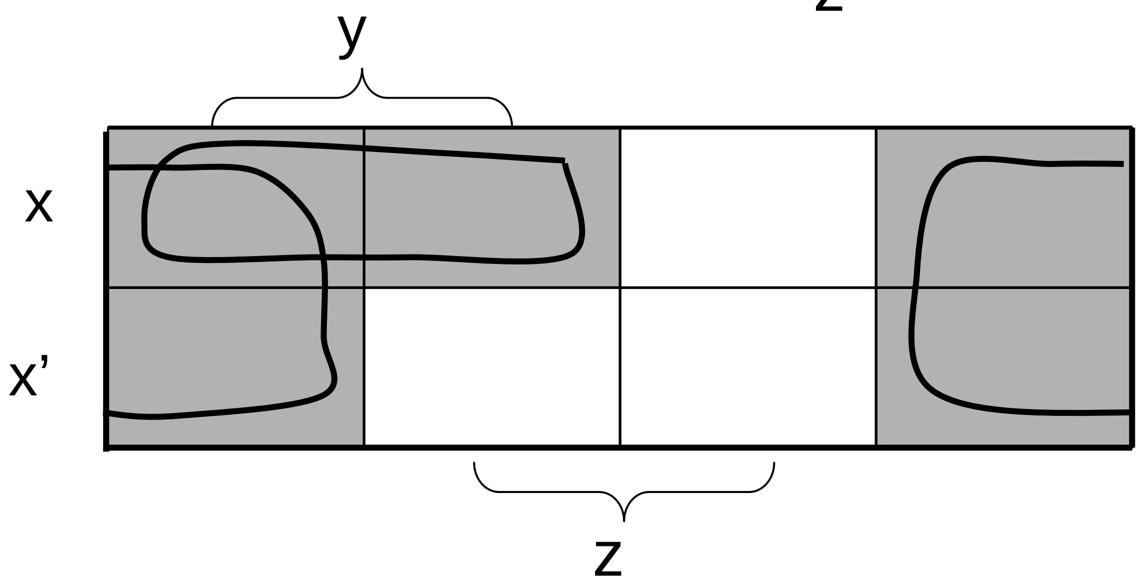
¿ xyz + x’yz = yz ?

xyz’	xyz	xy’z	xy’z’
x’yz’	x’yz	x’y’z	x’y’z’

MAPA DE KARNAUGH

$E_F(x,y,z) = xyz + xyz' + xy'z' + x'yz' + x'y'z'$

	y	y	y'	y'
x	xyz'	xyz	xy'z	xy'z'
x'	x'yz'	x'yz	x'y'z	x'y'z'
		z	z	



xy + z'

MAPA DE KARNAUGH

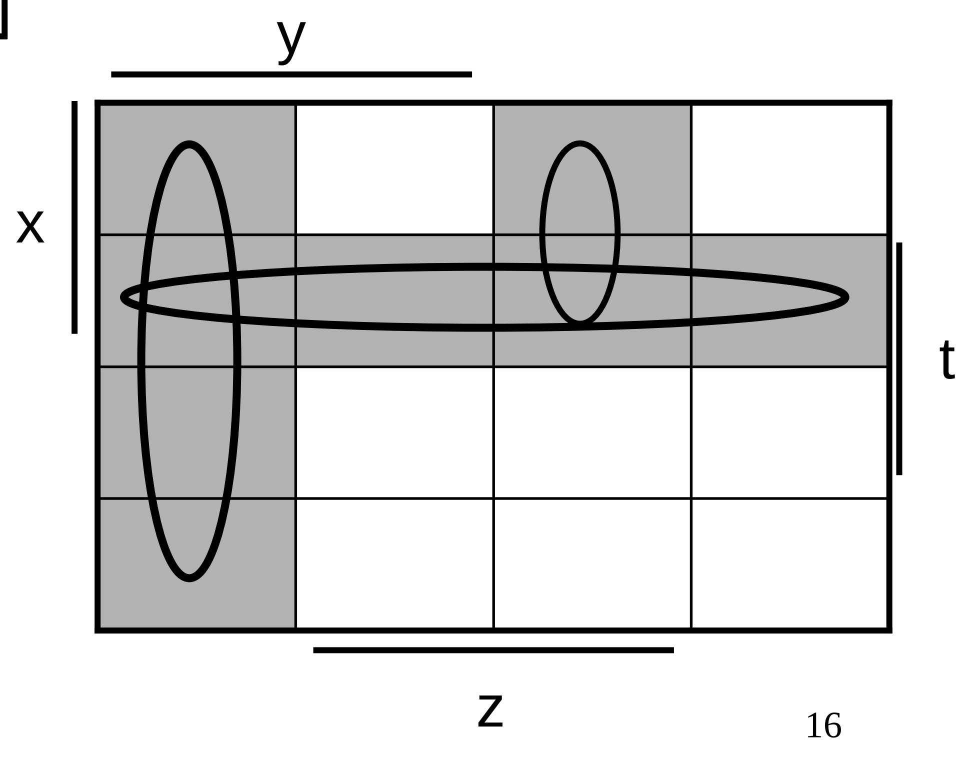
y

x

xyz't'	xyzt'	xy'zt'	xyz't'
xyz't	xyzt	xy'zt	xy'z't'
x'yz't	x'yzt	x'y'zt	x'y'z't
x'yz't'	x'yzt'	x'y'zt'	x'y'z't'

t

z



yz' + xt + xy'z

MÉTODO DE QUINE – McCLUSKEY

Válido para cualquier número de variables

E = xyz’t + x’yzt + xyzt’ + x’yz’t + xy’zt’ + x’yzt’ +x’y’zt’

1101 0111 1110 0101 1010 0110 0010

Se hallan los unos de la función booleana asociada f_E
Se ordenan, en orden decreciente de unos

1101
0111
1110
0101
1010
0110
0010

Se compara cada fila con las del bloque siguiente

Si sólo varían en un dígito, se marcan y se pone en otra columna, colocando un guión en el lugar de los dígitos distintos

* 1101
* 0101

→ –101

MÉTODO DE QUINE – McCLUSKEY

*1101	–101	– – 10
*0111	01–1	
*1110	011–	
*0101	* 1–10	
*1010	* –110	
*0110	* –010	
*0010	* 0–10	

Se continúa el proceso hasta que no se pueda construir más bloques

Se consideran todos los elementos no marcados

–101 01–1 011– – – 10

Para cada elemento de S(f), los unos de la función f, se elige uno de los no marcados, que lo incluya, que tenga el mayor número de guiones y repitiendo elección siempre que se pueda.

MÉTODO DE QUINE – McCLUSKEY

	1101	0111	1110	0101	1010	0110	0010
– –10			✓		✓	✓	✓
–101	✓			✓			
01–1		✓		✓			
011–		✓				✓	

Con los elementos – – 10, –101 y 01–1 se cubren todos los elementos de S(f)

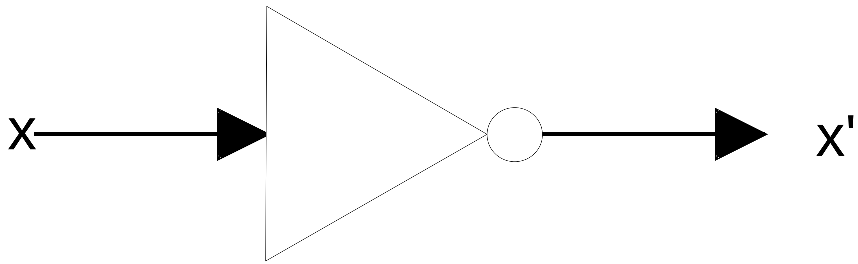
La expresión simplificada es zt’ + yz’t + x’yt

PUERTAS LÓGICAS

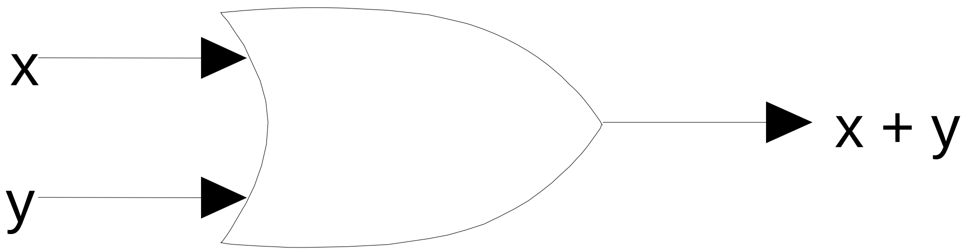
Álgebras de Boole --- Circuitos de dispositivos electrónicos

Operaciones --- Puertas lógicas

Inversor



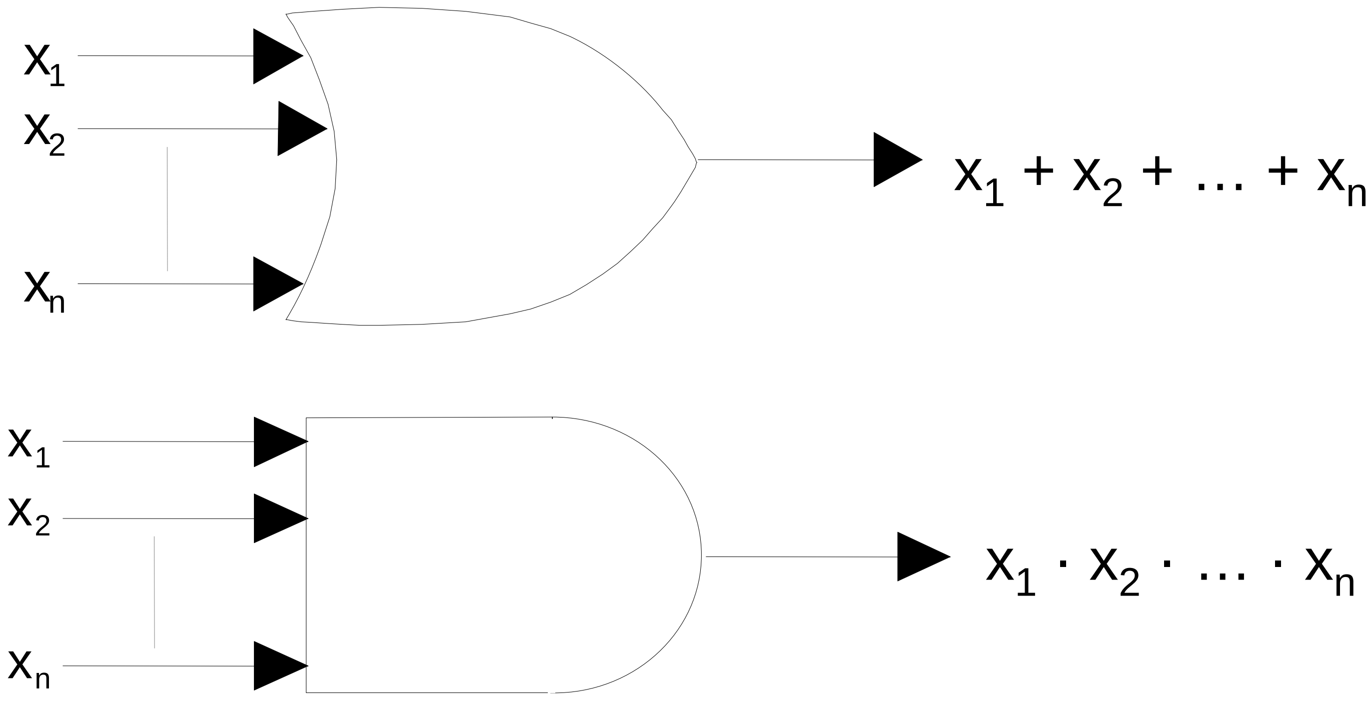
OR

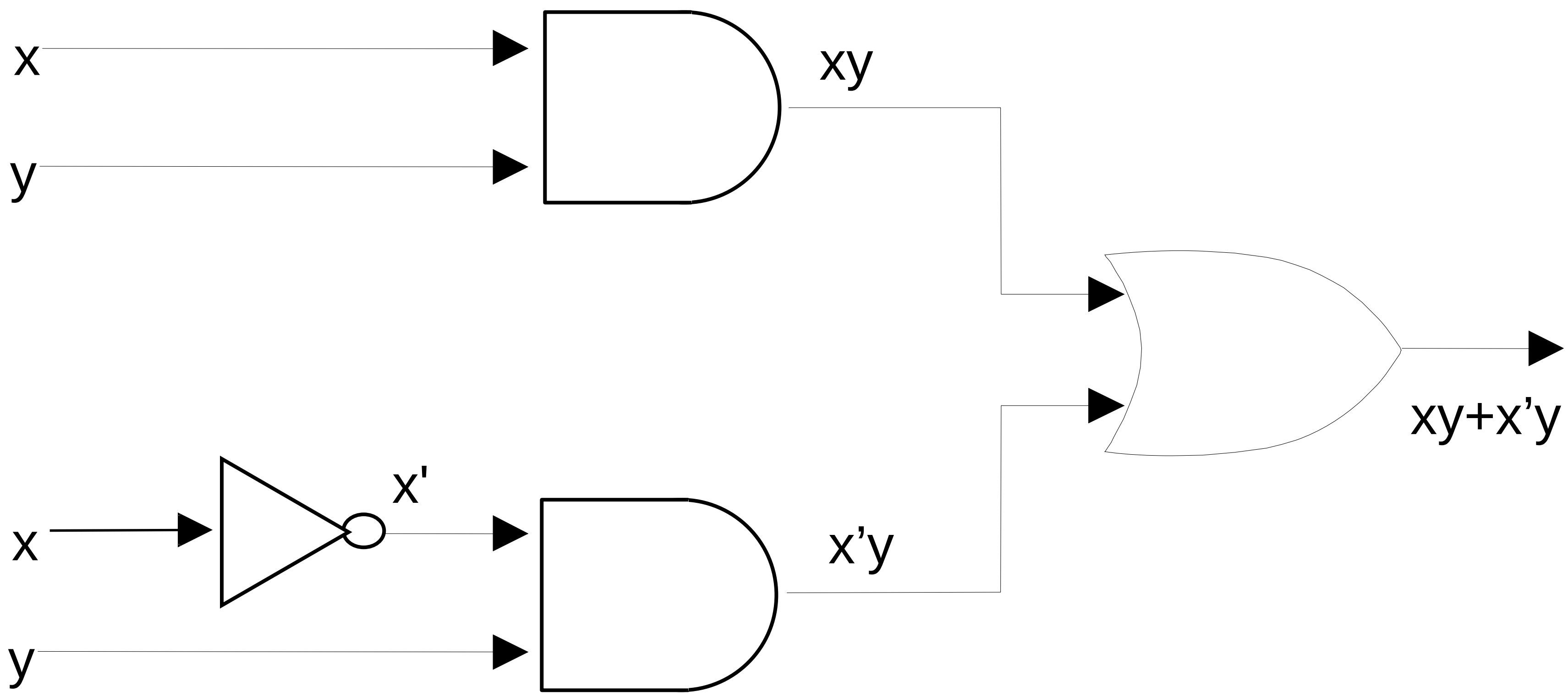


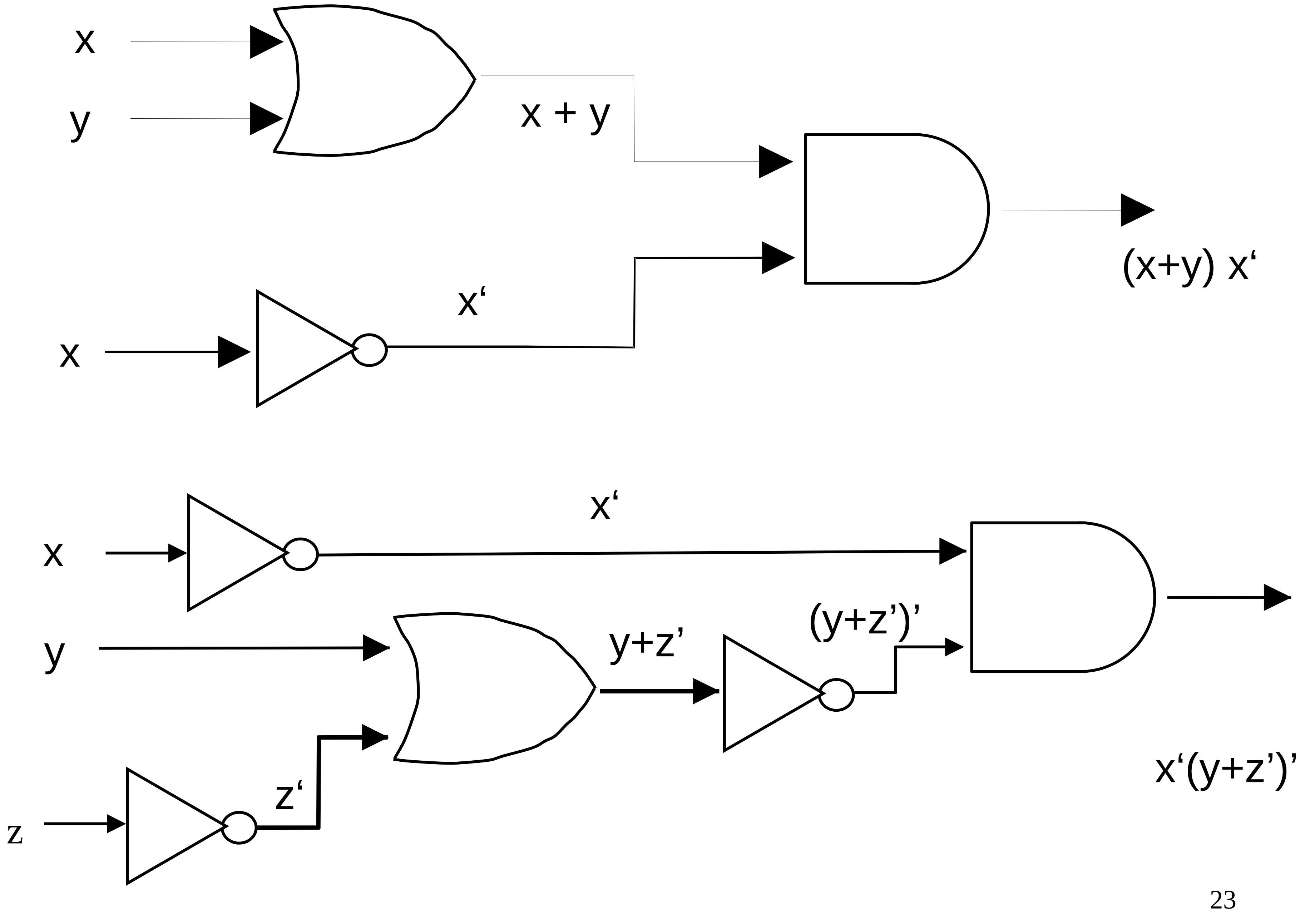
AND

PUERTAS LÓGICAS

Se permiten múltiples entradas en las puertas AND y OR.




x
y
xy
x
x'
y
x’y
xy+x’y

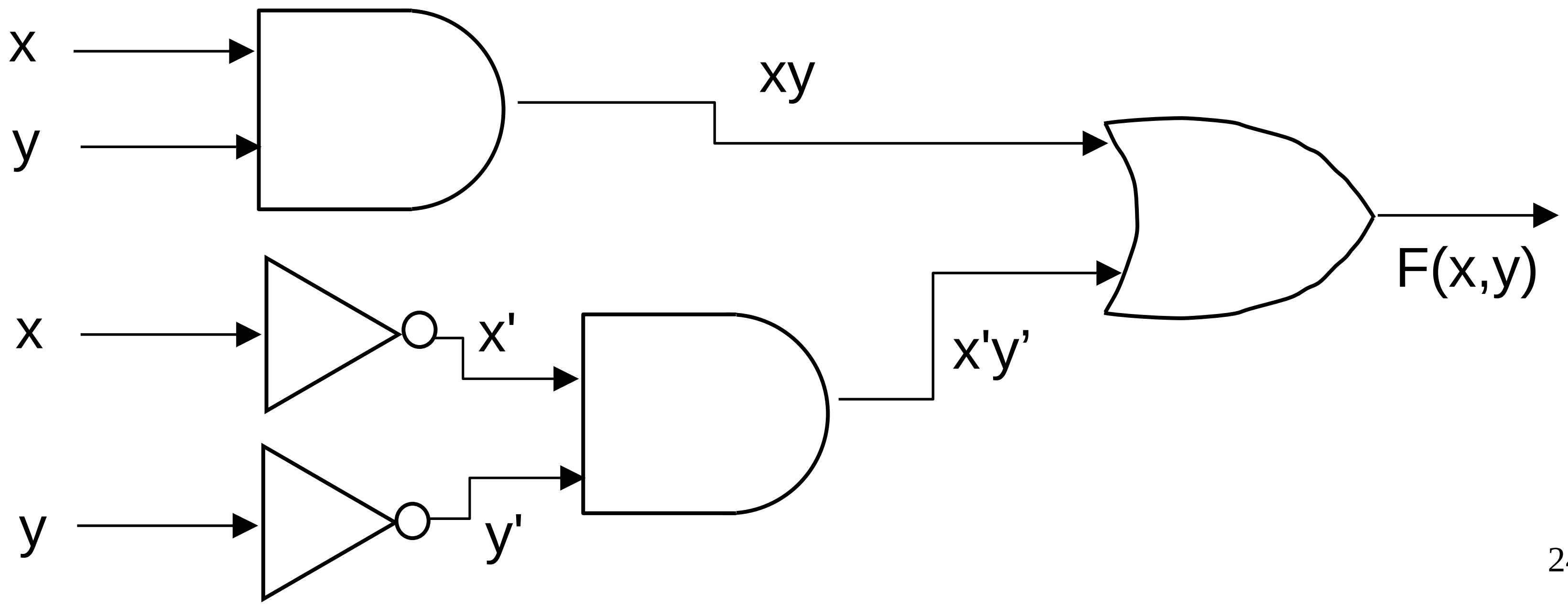

x
y
x + y
(x+y) x‘
x‘
x
x‘
x
y
(y+z’)’
y+z’
x‘(y+z’)’
z‘
z

Circuito con dos interruptores

Cuando cambia una variable,
cambia el resultado

$$F(x,y) = xy + x'y'$$

x	y	F(x,y)
1	1	1
1	0	0
0	1	0
0	0	1



Suma en base 2

Circuito lógico para sumar dos dígitos binarios

Entrada dos bits, X, Y valores 0, 1

Salida dos bits S, C suma y acarreo

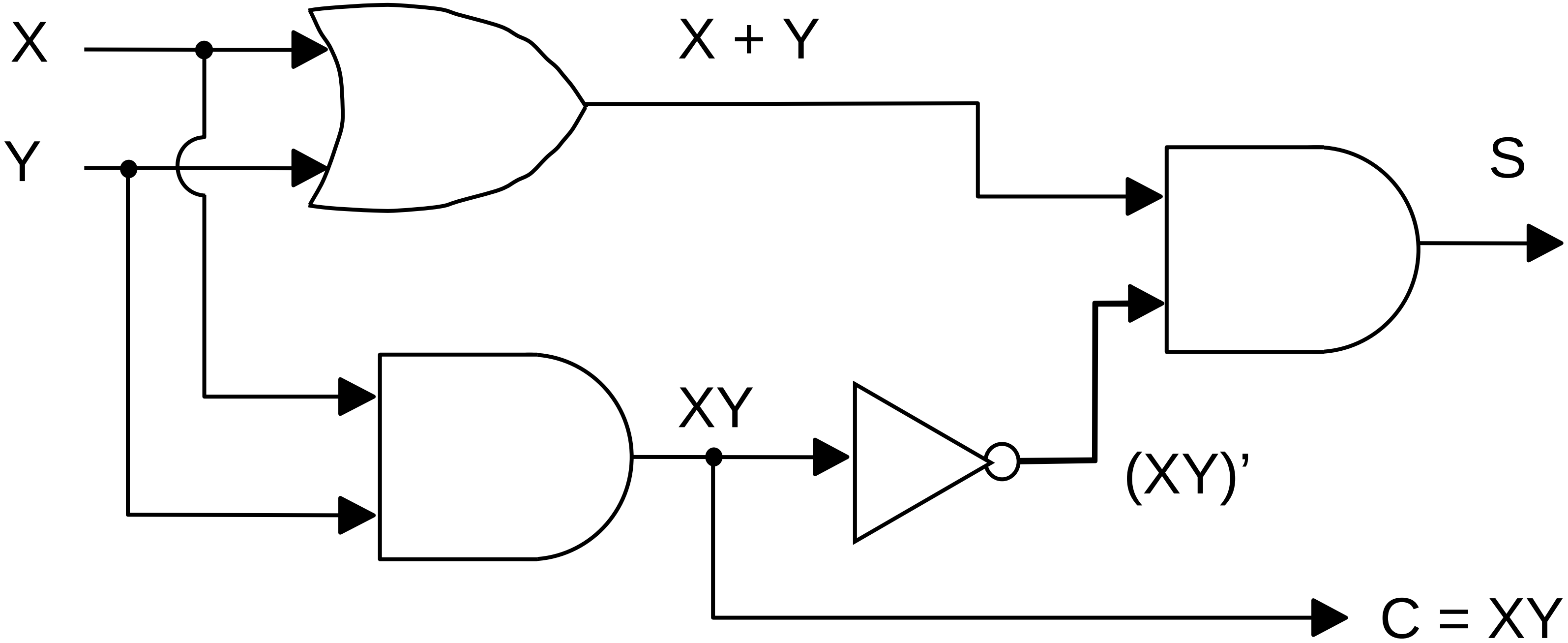
SEMISUMADOR (sin acarreo previo)

Entrada		Salida	
x	y	s	c
1	1	0	1
1	0	1	0
0	1	1	0
0	0	0	0

$S = XY' + X'Y \quad = (X + Y)(XY)'$

$C = XY$

SEMISUMADOR



$S = XY' + X'Y \quad = (X + Y)(XY)'$

$C = XY$

SUMADOR COMPLETO

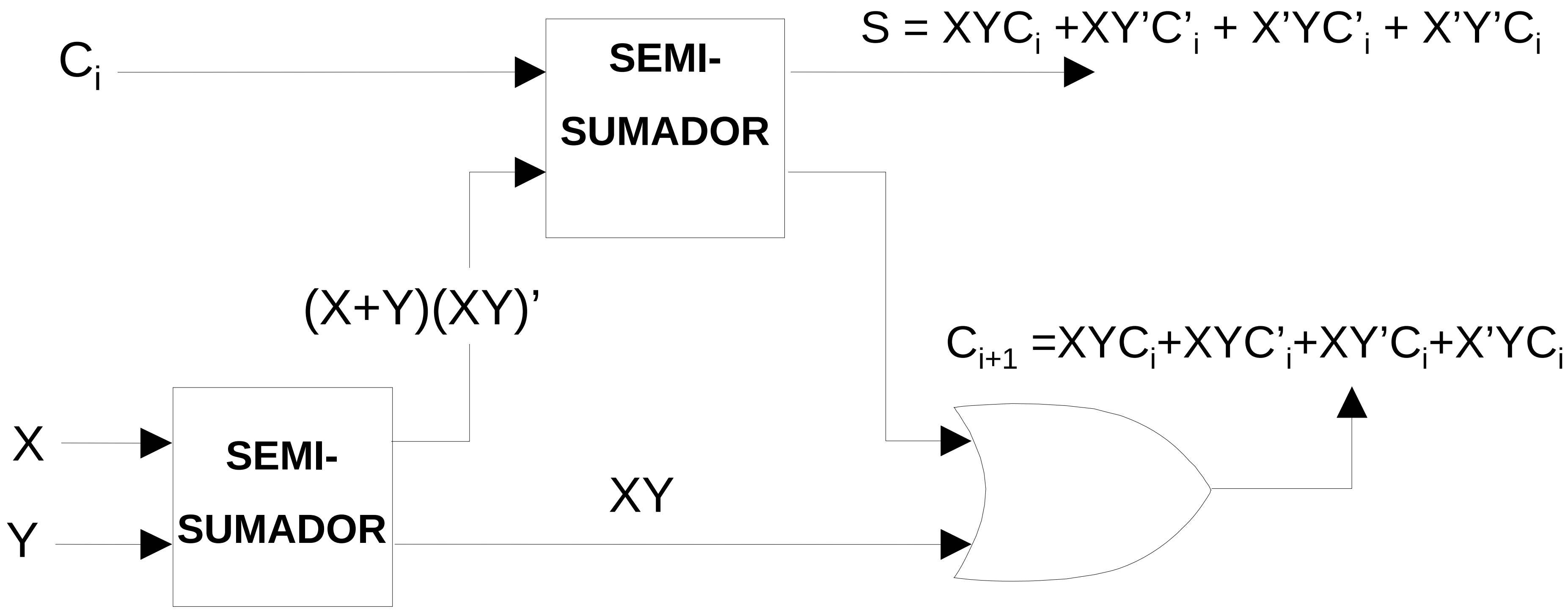
Suma con acarreo de la suma anterior
Entrada: Sumandos X, Y más el acarreo anterior C_i
Salida: La suma S y el nuevo acarreo C_{i+1}

Entrada			Salida	
x	y	c_i	s	c_{i+1}
1	1	1	1	1
1	1	0	0	1
1	0	1	0	1
1	0	0	1	0
0	1	1	0	1
0	1	0	1	0
0	0	1	1	0
0	0	0	0	0

$$S = XYC_i + XY'C'_i + X'YC'_i + X'Y'C_i$$

$$C_{i+1} = XYC_i + XYC'_i + XY'C_i + X'YC_i$$

SUMADOR COMPLETO



SUMADOR VARIAS CIFRAS

$$(X_2X_1X_0)_2 + (Y_2Y_1Y_0)_2 = (S_3S_2S_1S_0)_2$$