

Circuitos Electrónicos Digitales

E.T.S.I. Telecomunicación

Universidad Politécnica de Madrid

Álgebra de conmutación y simplificación de funciones lógicas

Álgebra Booleana. Análisis de circuitos
combinacionales. Diseño de circuitos
combinacionales. Mapas de Karnaugh

Álgebra Booleana

- También se conoce como álgebra de conmutación
 - Trabaja con valores “0” y “1”
- Existen dos convenciones
 - Lógica positiva
 - “L” o nivel bajo es equivalente a “0”
 - “H” o nivel alto es equivalente a “1”
 - Lógica negativa (poco utilizada)
 - “L” o nivel bajo es equivalente a “1”
 - “H” o nivel alto es equivalente a “0”
- Las señales se nombran con variables (por ejemplo X, Y, RST, CK, etc.)
- Las señales pueden valer solamente “0” o “1”

Operadores booleanos

- NOT: X' (negado de X. Existen varias convenciones para la negada de una variable)
- AND: $X \cdot Y$
- OR: $X + Y$

X	Y	X AND Y
0	0	0
0	1	0
1	0	0
1	1	1

X	Y	X OR Y
0	0	0
0	1	1
1	0	1
1	1	1

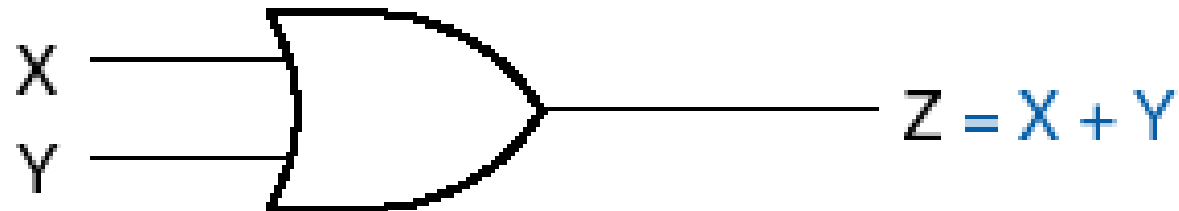
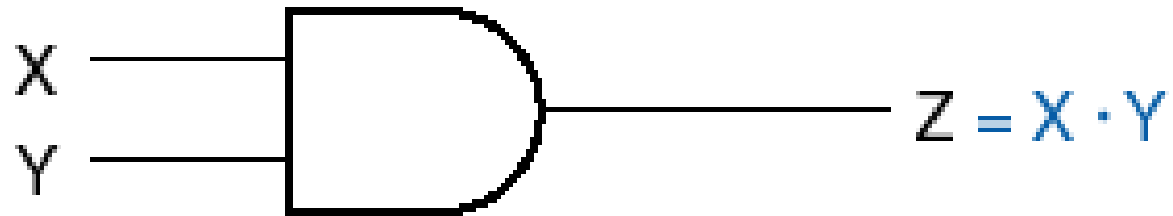
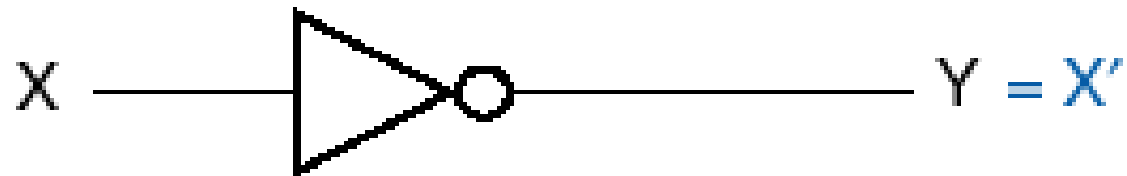
X	NOT X
0	1
1	0

- Los operadores se definen por su tabla de verdad

Más definiciones

- Literal: Una variable o su complemento (valor negado)
 - X, X', RST', CS_L
- Expresión: Literales combinados por operadores AND, OR y NOT con los paréntesis necesarios
 - $X+Y$
 - $P \cdot Q \cdot R$
 - $A + B \cdot C$
 - $((RST \cdot Z') + CS_L \cdot A \cdot B' \cdot C + Q5) \cdot RST'$
- Ecuación: Expresión del tipo “Variable=Expresión”
 - $P = ((RST \cdot Z') + CS_L \cdot A \cdot B' \cdot C + Q5) \cdot RST'$

Símbolos Lógicos



Teoremas

(T1)	$X + 0 = X$	(T1')	$X \cdot 1 = X$	(Identities)
(T2)	$X + 1 = 1$	(T2')	$X \cdot 0 = 0$	(Null elements)
(T3)	$X + X = X$	(T3')	$X \cdot X = X$	(Idempotency)
(T4)	$(X')' = X$			(Involution)
(T5)	$X + X' = 1$	(T5')	$X \cdot X' = 0$	(Complements)

(T6)	$X + Y = Y + X$	(T6')	$X \cdot Y = Y \cdot X$	(Commutativity)
(T7)	$(X + Y) + Z = X + (Y + Z)$	(T7')	$(X \cdot Y) \cdot Z = X \cdot (Y \cdot Z)$	(Associativity)
(T8)	$X \cdot Y + X \cdot Z = X \cdot (Y + Z)$	(T8')	$(X + Y) \cdot (X + Z) = X + Y \cdot Z$	(Distributivity)
(T9)	$X + X \cdot Y = X$	(T9')	$X \cdot (X + Y) = X$	(Covering)
(T10)	$X \cdot Y + X \cdot Y' = X$	(T10')	$(X + Y) \cdot (X + Y') = X$	(Combining)
(T11)	$X \cdot Y + X' \cdot Z + Y \cdot Z = X \cdot Y + X' \cdot Z$			(Consensus)
(T11')	$(X + Y) \cdot (X' + Z) \cdot (Y + Z) = (X + Y) \cdot (X' + Z)$			

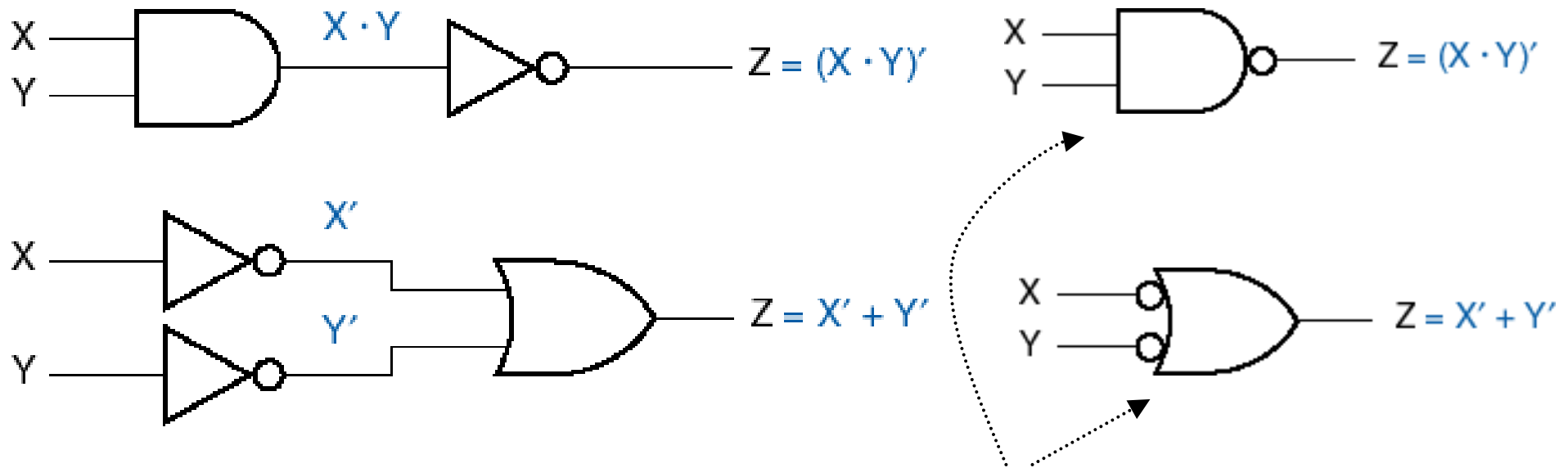
Teoremas de n variables

(T12)	$X + X + \dots + X = X$	(Generalized idempotency)
(T12')	$X \cdot X \cdot \dots \cdot X = X$	
(T13)	$(X_1 \cdot X_2 \cdot \dots \cdot X_n)' = X_1' + X_2' + \dots + X_n'$	(DeMorgan's theorems)
(T13')	$(X_1 + X_2 + \dots + X_n)' = X_1' \cdot X_2' \cdot \dots \cdot X_n'$	
(T14)	$[F(X_1, X_2, \dots, X_n, +, \cdot)]' = F(X_1', X_2', \dots, X_n', \cdot, +)$	(Generalized DeMorgan's theorem)
(T15)	$F(X_1, X_2, \dots, X_n) = X_1 \cdot F(1, X_2, \dots, X_n) + X_1' \cdot F(0, X_2, \dots, X_n)$	(Shannon's expansion theorems)
(T15')	$F(X_1, X_2, \dots, X_n) = [X_1 + F(0, X_2, \dots, X_n)] \cdot [X_1' + F(1, X_2, \dots, X_n)]$	

- Prueba utilizando inducción finita
- Los más importantes son los de teoremas de DeMorgan

Equivalencia de símbolos resultante del teorema de DeMorgan T13

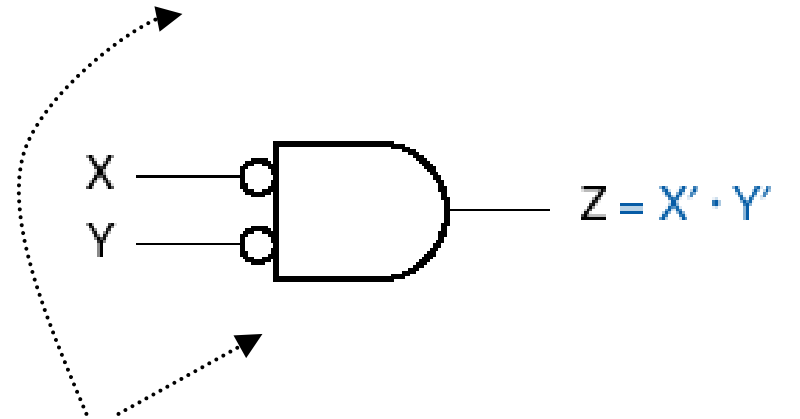
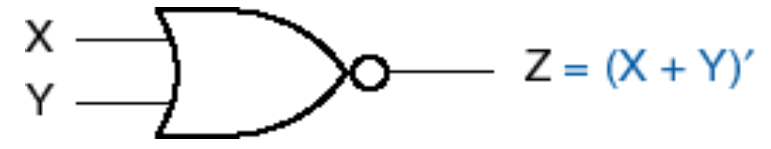
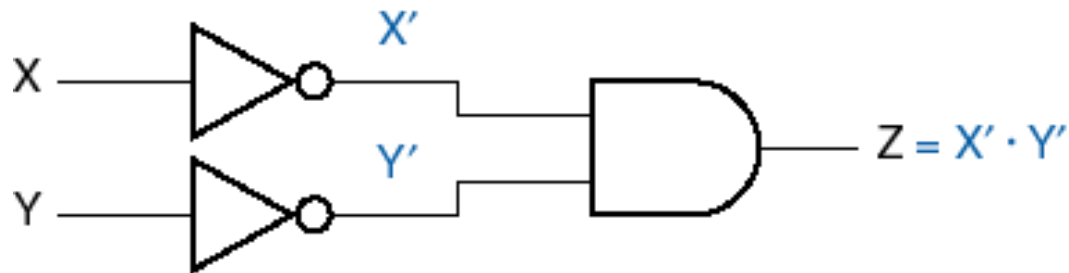
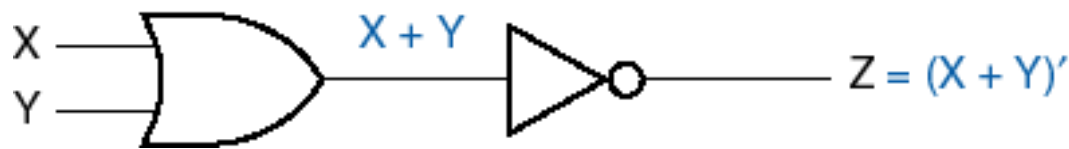
- Puertas NAND



Puertas NAND

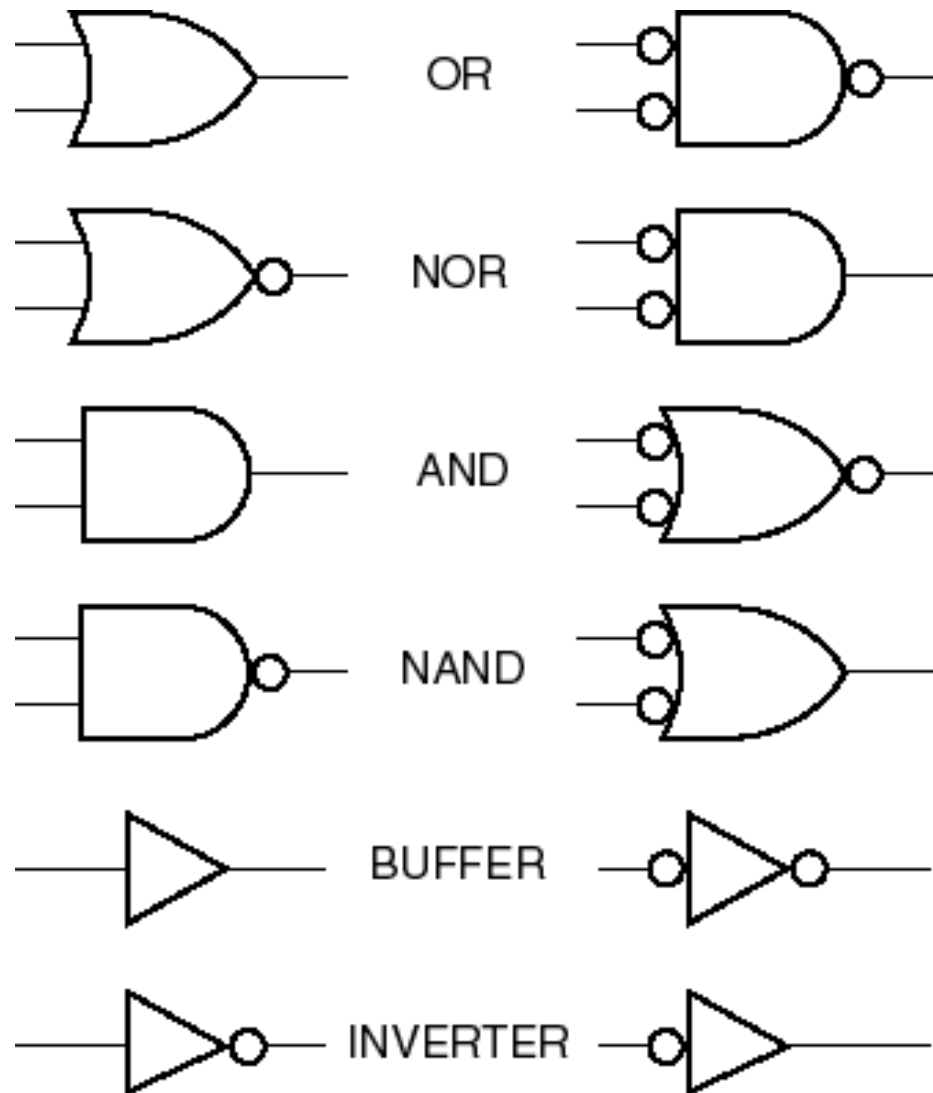
Equivalencia de símbolos resultante del teorema de DeMorgan T13'

- Puertas NOR



Puertas NOR

Símbolos utilizando DeMorgan



Más definiciones

- Término producto:
 - Un literal o un producto de dos o más literales
- Expresión de suma de productos:
 - Suma de términos producto
- Término suma:
 - Un literal o una suma de dos o más literales
- Expresión producto de sumas:
 - Producto de términos suma
- Término normal:
 - Término suma o producto en el que ninguna variable aparece más de una vez.
- Minterm de n variables:
 - Término producto normal con n literales (existen 2^n)
- Maxterm de n variables:
 - Término suma normal con n literales (existen 2^n)

Tabla de verdad con minterms y maxterms

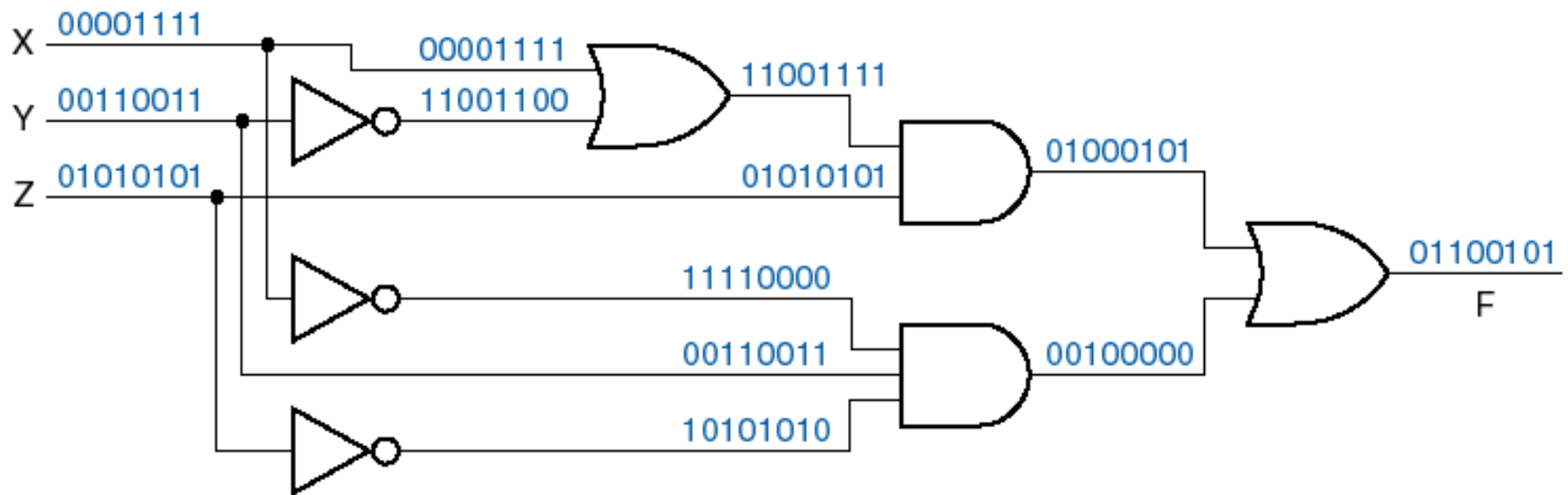
<i>Row</i>	<i>X</i>	<i>Y</i>	<i>Z</i>	<i>F</i>	<i>Minterm</i>	<i>Maxterm</i>
0	0	0	0	$F(0,0,0)$	$X' \cdot Y' \cdot Z'$	$X + Y + Z$
1	0	0	1	$F(0,0,1)$	$X' \cdot Y' \cdot Z$	$X + Y + Z'$
2	0	1	0	$F(0,1,0)$	$X' \cdot Y \cdot Z'$	$X + Y' + Z$
3	0	1	1	$F(0,1,1)$	$X' \cdot Y \cdot Z$	$X + Y' + Z'$
4	1	0	0	$F(1,0,0)$	$X \cdot Y' \cdot Z'$	$X' + Y + Z$
5	1	0	1	$F(1,0,1)$	$X \cdot Y' \cdot Z$	$X' + Y + Z'$
6	1	1	0	$F(1,1,0)$	$X \cdot Y \cdot Z'$	$X' + Y' + Z$
7	1	1	1	$F(1,1,1)$	$X \cdot Y \cdot Z$	$X' + Y' + Z'$

Análisis de circuitos combinacionales

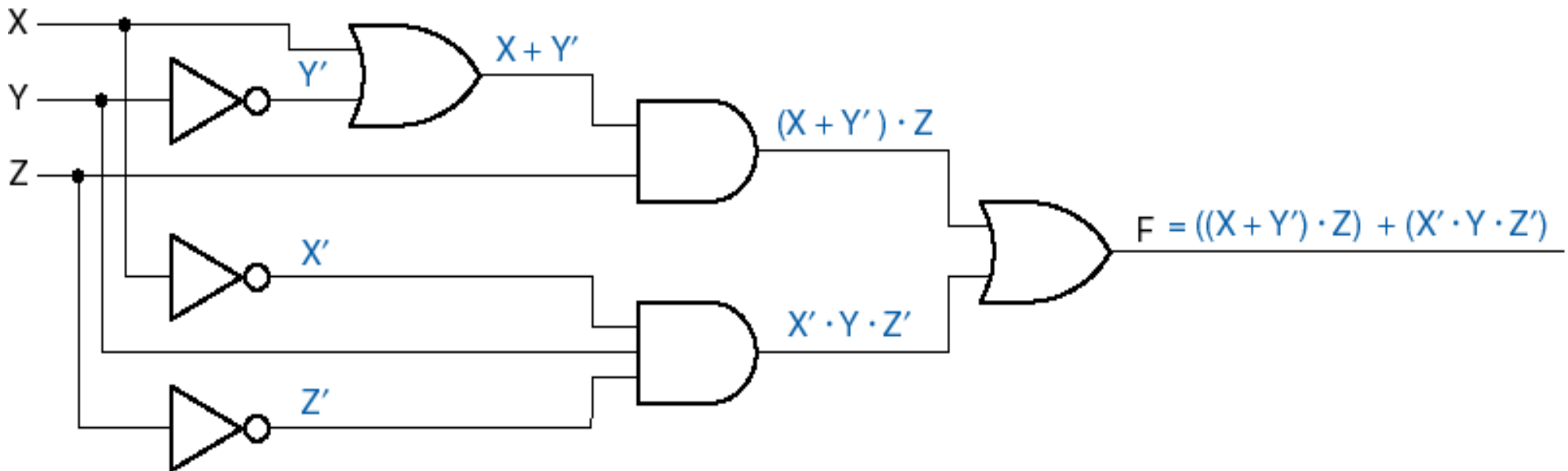
- Los valores de las señales de salida dependen exclusivamente de los valores de las de entrada, no de su historia
- Formas de análisis de circuitos combinacionales
 - Exhaustivo (tabla de verdad)
 - Algebraico mediante expresiones (Álgebra de Boole)
 - Simulación o emulación utilizando herramientas CAD
 - Escribir una descripción funcional en un lenguaje de descripción hardware (HDL)
 - Definir la condiciones en las que realizar la prueba
 - Comparar los valores obtenidos a la salida con los esperados del diseño
 - Repetir con valores hasta que se dé por válido el circuito diseñado (punto muy crítico)

Análisis exhaustivo de un circuito combinacional

- Para cada una de las posibles combinaciones de valores a las entradas se obtienen los valores de las variables intermedias y las salidas.
 - Si tenemos “n” entradas habrá 2^n combinaciones



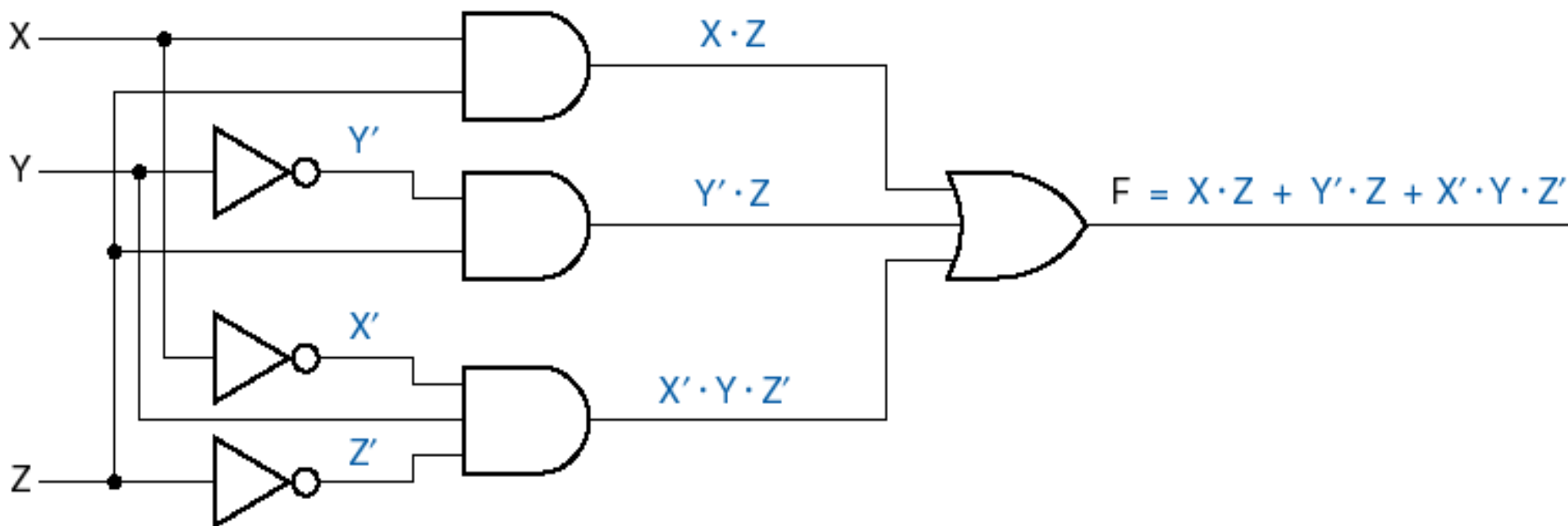
Análisis obteniendo las expresiones de las señales



- La función de salida F es:

$$\begin{aligned} F &= ((X + Y') \cdot Z) + (X' \cdot Y \cdot Z') \\ &= (X \cdot Z) + (Y' \cdot Z) + (X' \cdot Y \cdot Z') \end{aligned}$$

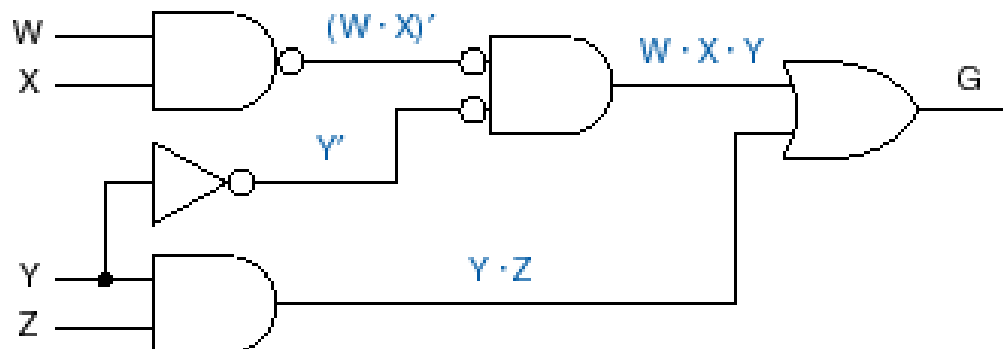
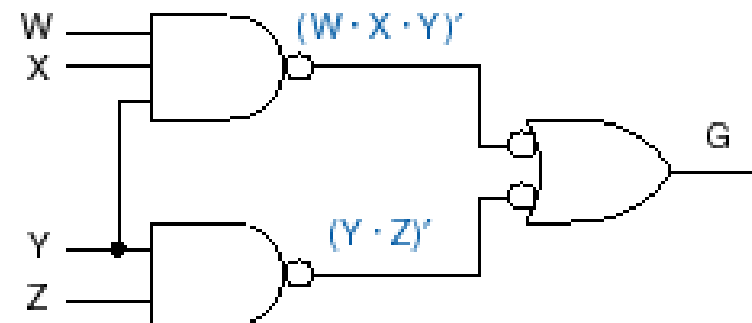
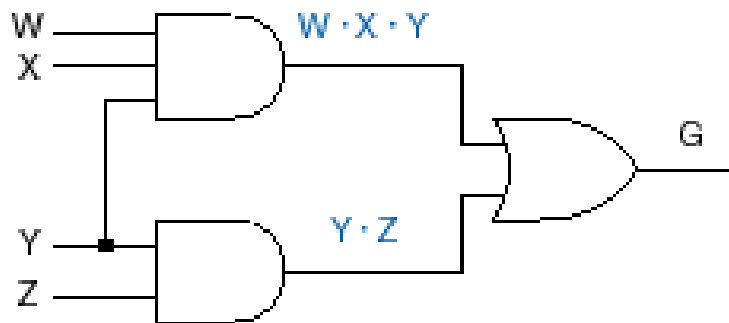
Otro ejemplo de análisis de un circuito que tiene la misma función que el anterior



Ejemplo de sustitución de símbolos

- Se emplea para minimizar el número de puertas o el número de entradas de las mismas (mayor velocidad)

$$G(W, X, Y, Z) = W \cdot X \cdot Y + Y \cdot Z$$



Diseño de circuitos combinacionales

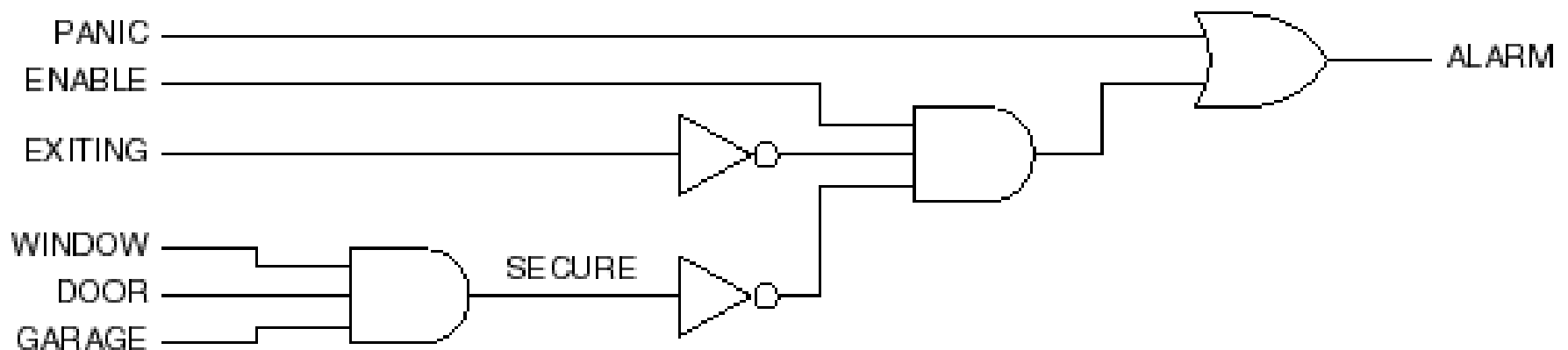
- En algunas ocasiones puede escribirse en ecuaciones el funcionamiento del circuito
- Ejemplo, el circuito de una alarma

$$\text{ALARM} = \text{PANIC} + \text{ENABLE} \cdot \text{EXITING}' \cdot \text{SECURE}'$$

$$\text{SECURE} = \text{WINDOW} \cdot \text{DOOR} \cdot \text{GARAGE}$$

$$\text{ALARM} = \text{PANIC} + \text{ENABLE} \cdot \text{EXITING}' \cdot (\text{WINDOW} \cdot \text{DOOR} \cdot \text{GARAGE})'$$

- Implementación directa de la ecuación del circuito

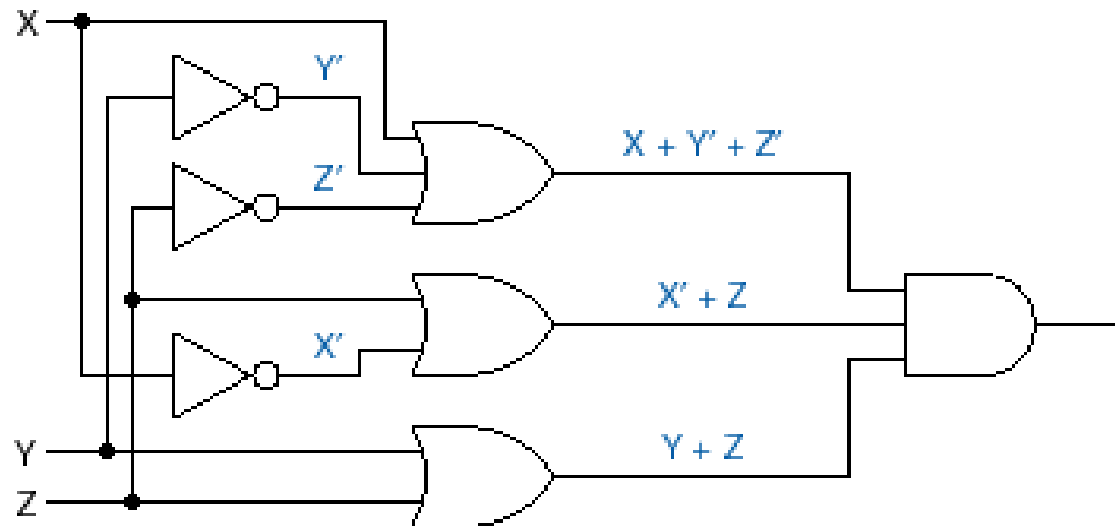


Ejemplo de implementación

- Es conveniente manipular la función lógica a F para minimizar el número de puertas

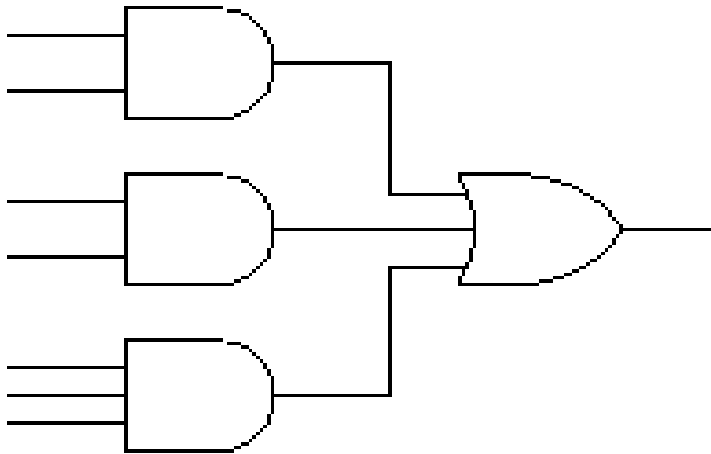
$$\begin{aligned} F &= ((X + Y') \cdot Z) + (X' \cdot Y \cdot Z') \\ &= (X + Y' + X') \cdot (X + Y' + Y) \cdot (X + Y' + Z') \cdot (Z + X') \cdot (Z + Y) \cdot (Z + Z') \\ &= 1 \cdot 1 \cdot (X + Y' + Z') \cdot (X' + Z) \cdot (Y + Z) \cdot 1 \\ &= (X + Y' + Z') \cdot (X' + Z) \cdot (Y + Z) \end{aligned}$$

- Circuito:

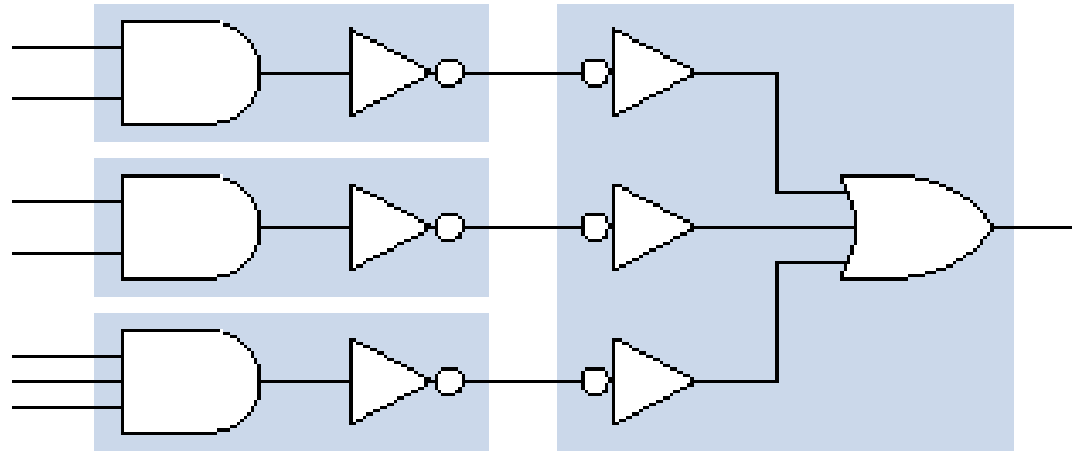


Implementación formal de funciones

- Implementación como suma de productos



AND-OR

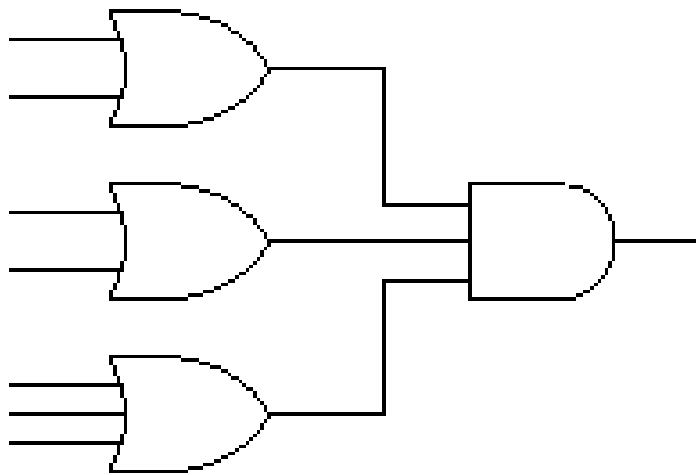


NAND-NAND



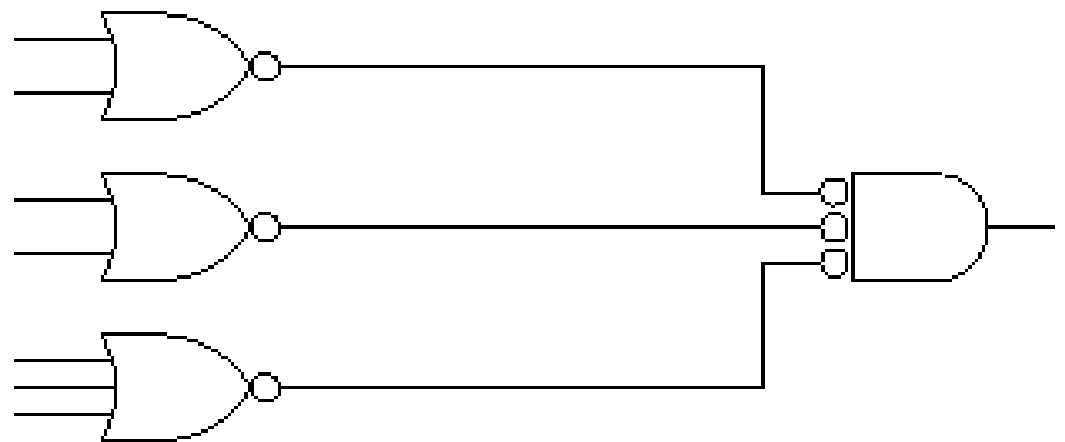
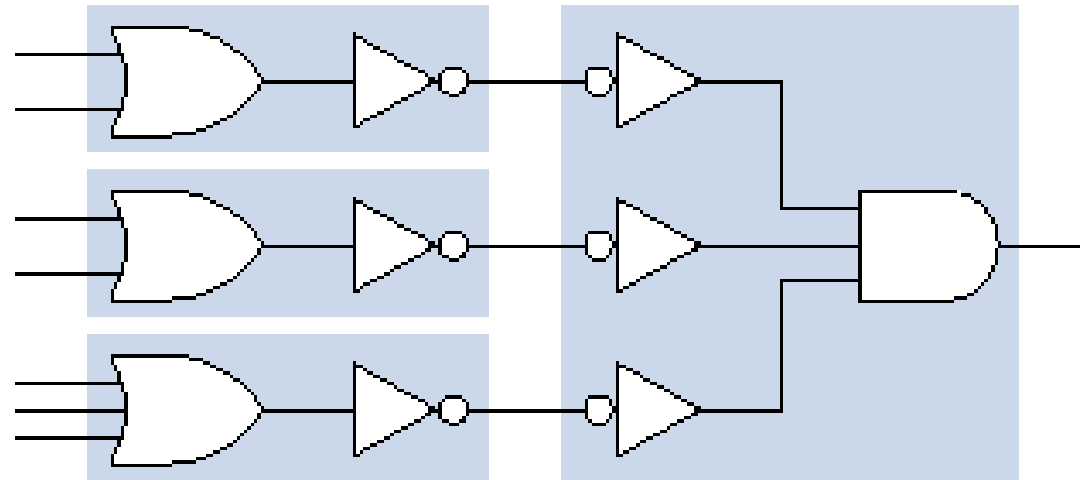
Implementación formal de funciones

- Implementación como producto de sumas



OR-AND

NOR-NOR



Diseño “por las bravas”

- Tabla de verdad →
suma canónica
(suma de minterms)
- Ejemplo:
detector de números primos
– 4-bits de entrada, $N_3N_2N_1N_0$

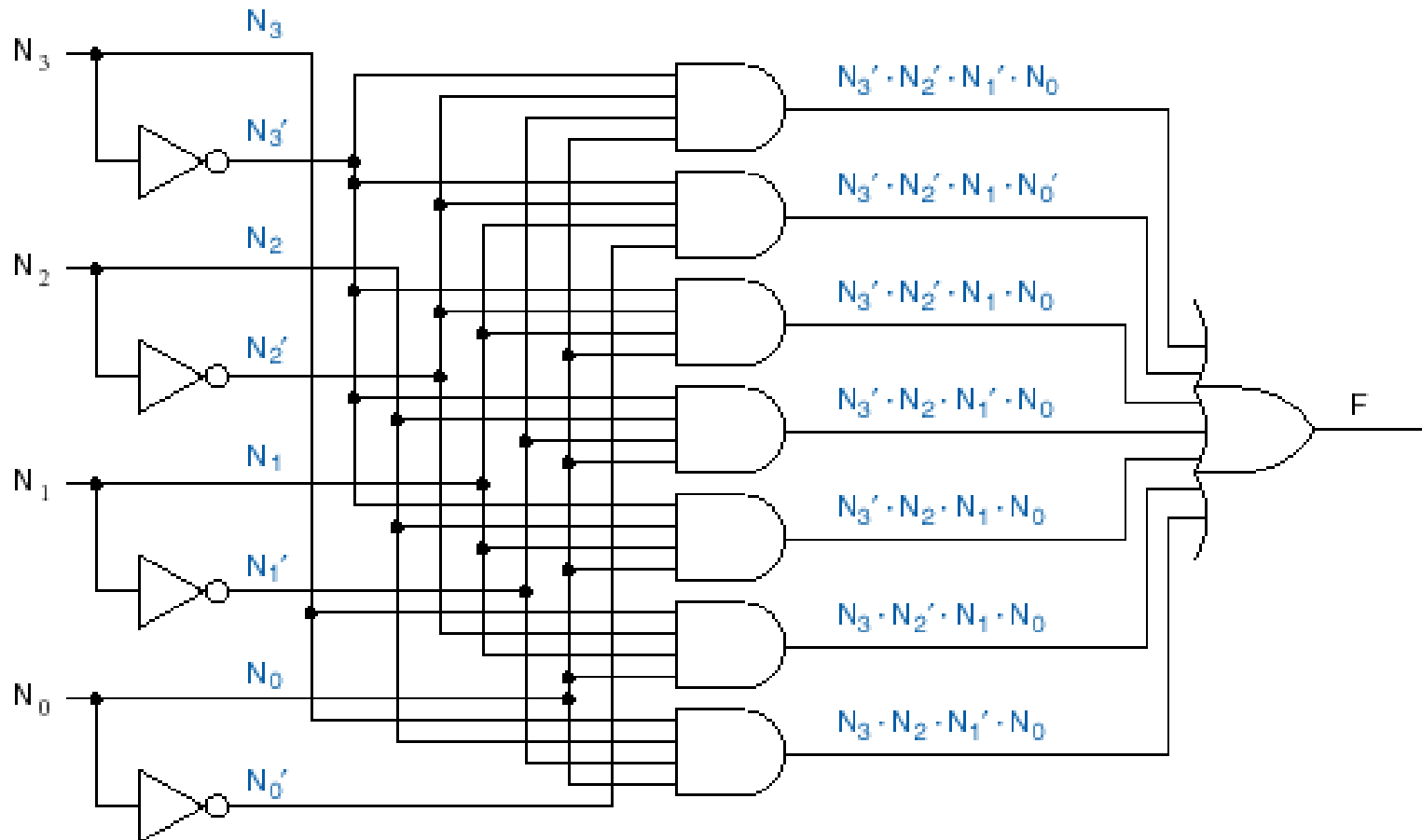
$$F = S_{N_3N_2N_1N_0}(1,2,3,5,7,11,13)$$

row	N_3	N_2	N_1	N_0	F
0	0	0	0	0	0
1	0	0	0	1	1
2	0	0	1	0	1
3	0	0	1	1	1
4	0	1	0	0	0
5	0	1	0	1	1
6	0	1	1	0	0
7	0	1	1	1	1
8	1	0	0	0	0
9	1	0	0	1	0
10	1	0	1	0	0
11	0	0	1	1	1
12	1	1	0	0	0
13	1	1	0	1	1
14	1	1	1	0	0
15	1	1	1	1	0

Lista de minterms \rightarrow suma canónica

$$F = \Sigma_{N_3, N_2, N_1, N_0} (1, 2, 3, 5, 7, 11, 13)$$

$$\begin{aligned} &= N_3' \cdot N_2' \cdot N_1' \cdot N_0 + N_3' \cdot N_2' \cdot N_1 \cdot N_0' + N_3' \cdot N_2' \cdot N_1 \cdot N_0 + N_3' \cdot N_2 \cdot N_1' \cdot N_0 \\ &\quad + N_3' \cdot N_2 \cdot N_1 \cdot N_0 + N_3 \cdot N_2' \cdot N_1 \cdot N_0 + N_3 \cdot N_2 \cdot N_1' \cdot N_0 \end{aligned}$$



Simplificación algebraica

- Teorema T8, $X \cdot Y + X \cdot Y' = X$

$$F = \Sigma_{N_3, N_2, N_1, N_0}(1, 3, 5, 7, 2, 11, 13)$$

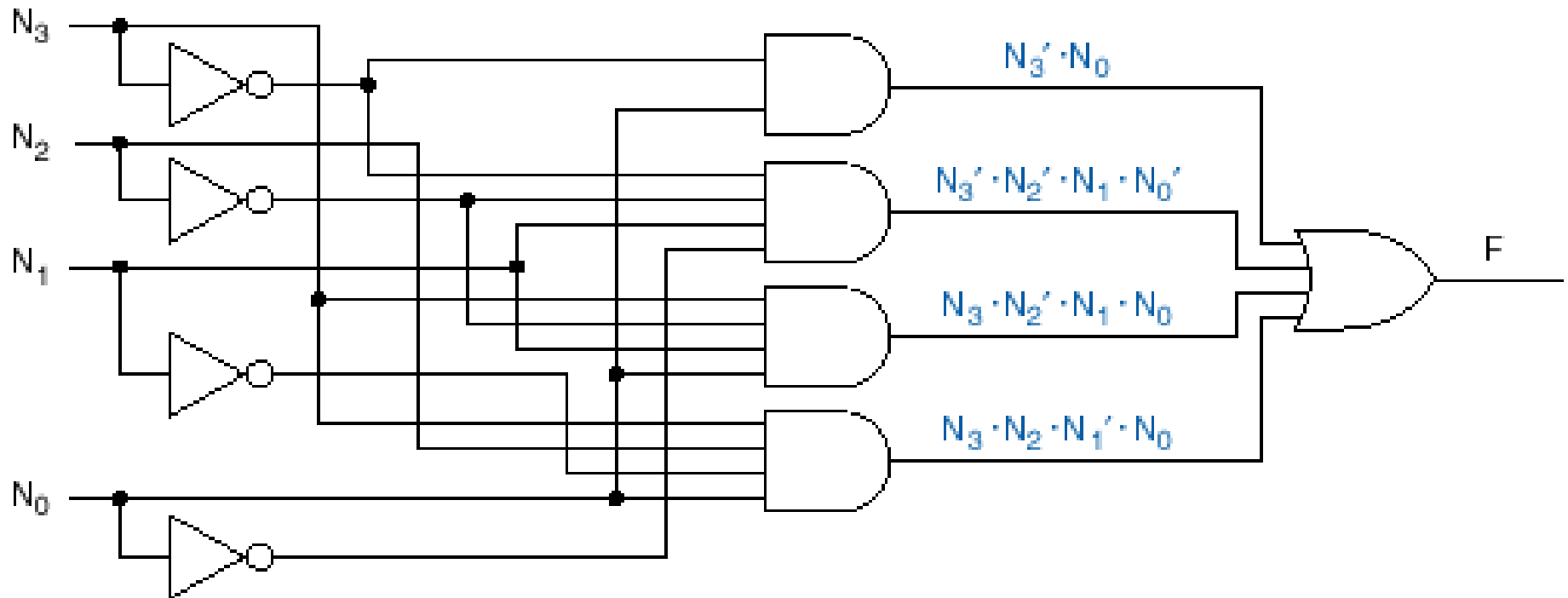
$$= N_3' \cdot N_2' \cdot N_1' \cdot N_0 + N_3' \cdot N_2' \cdot N_1 \cdot N_0 + N_3' \cdot N_2 \cdot N_1' \cdot N_0 + N_3' \cdot N_2 \cdot N_1 \cdot N_0 + \dots$$

$$= (N_3' \cdot N_2' \cdot N_1' \cdot N_0 + N_3' \cdot N_2' \cdot N_1 \cdot N_0) + (N_3' \cdot N_2 \cdot N_1' \cdot N_0 + N_3' \cdot N_2 \cdot N_1 \cdot N_0) + \dots$$

$$= N_3' \cdot N_2' \cdot N_0 + N_3' \cdot N_2 \cdot N_0 + \dots$$

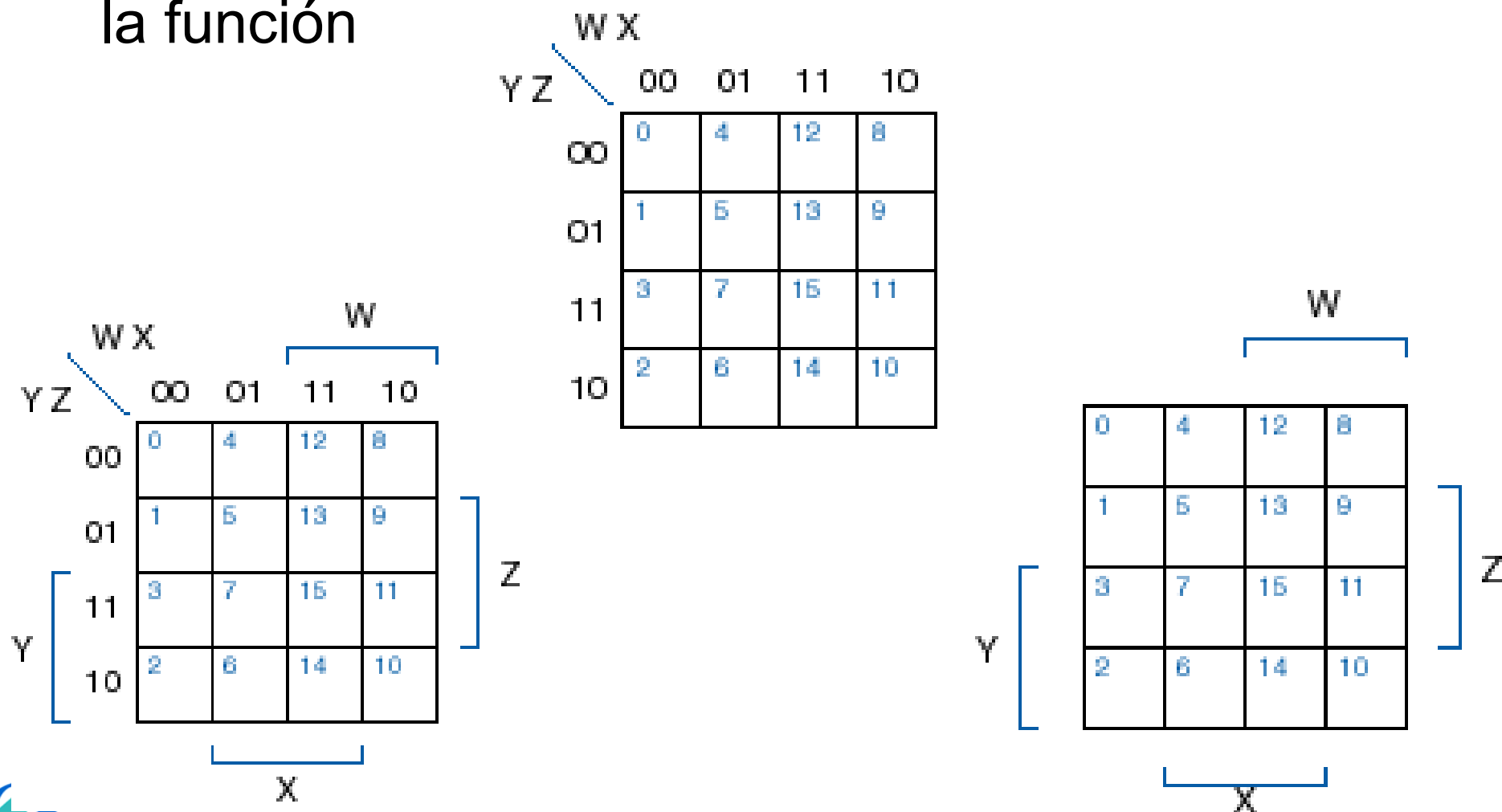
- Reduce el número de puertas y el de entradas por puerta
- Las simplificaciones no suelen ser sistemáticas, por lo que no puede garantizarse un resultado óptimo

Circuito resultante



Mapas de Karnaugh (teorema T10)

- Permiten realizar una minimización sistemática de la función



Mapa de Karnaugh de tres variables

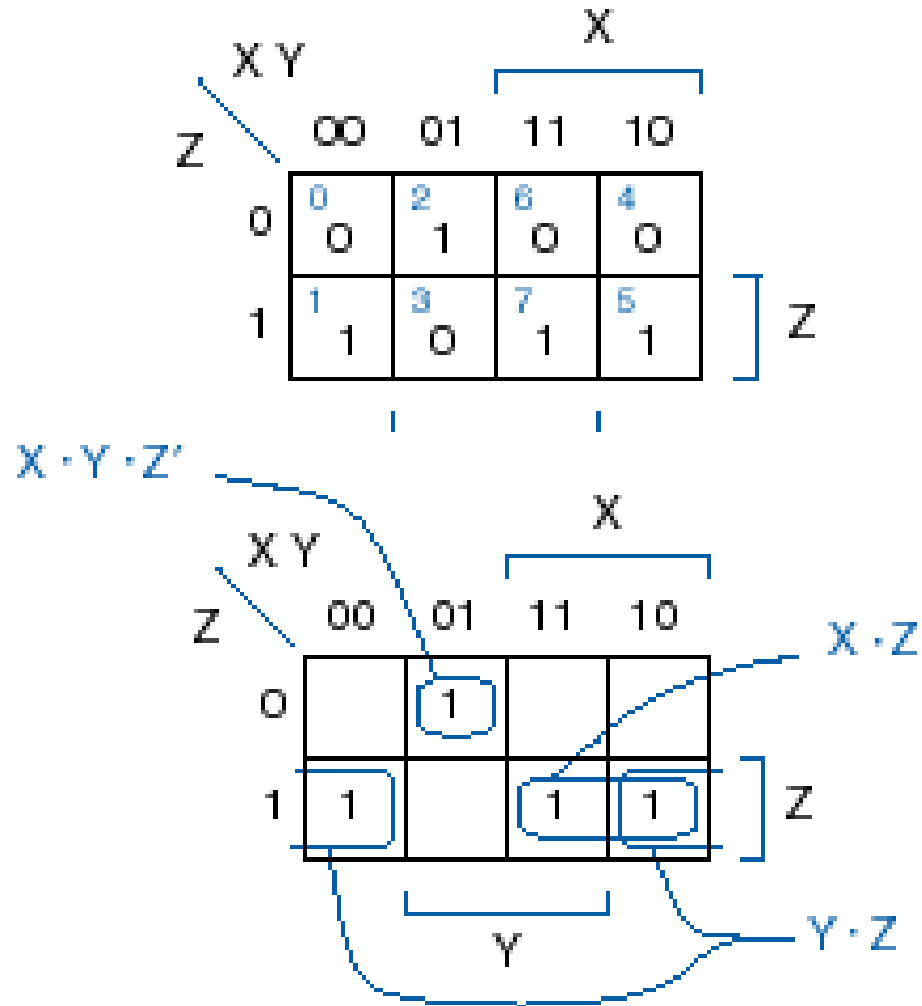
- En cualquier mapa de Karnaugh, los valores solamente pueden ser “1” y “0”.
- En caso de que existan “X”, habrá que asignarles un valor “1” ó “0” según el criterio de máxima minimización (rectángulos más grandes)

A 3-variable Karnaugh map diagram. The vertical axis is labeled 'Z' with values 0 and 1. The horizontal axis is labeled 'X Y' with values 00, 01, 11, and 10. The map is a 2x4 grid of cells. The cells are numbered 0 through 7 in blue. A blue bracket above the grid is labeled 'X', spanning the 11 and 10 columns. A blue bracket below the grid is labeled 'Y', spanning the 01 and 11 columns. A blue bracket to the right of the grid is labeled 'Z', spanning the 0 and 1 rows.

Z \ X Y	00	01	11	10
0	0	2	6	4
1	1	3	7	5

Ejemplo: $\Sigma(1,2,5,7)$

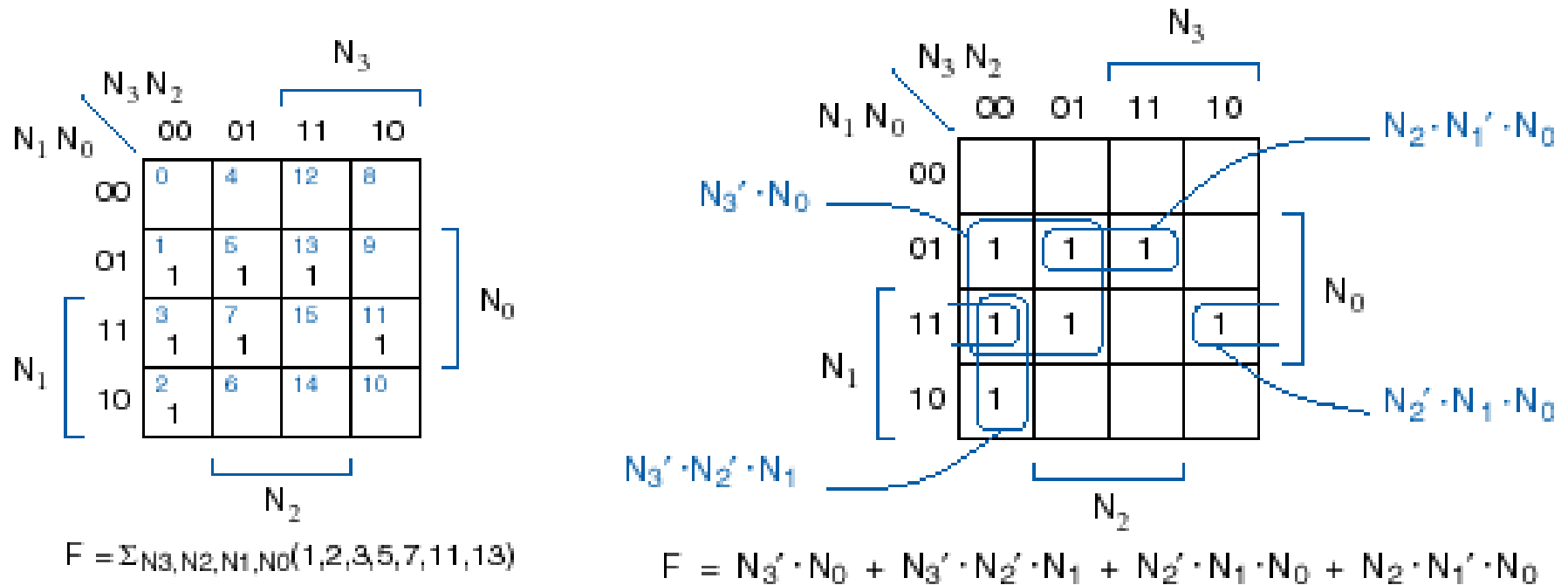
X	Y	Z	F
0	0	0	0
0	0	1	1
0	1	0	1
0	1	1	0
1	0	0	0
1	0	1	1
1	1	0	0
1	1	1	1



Uso de los mapas de Karnaugh

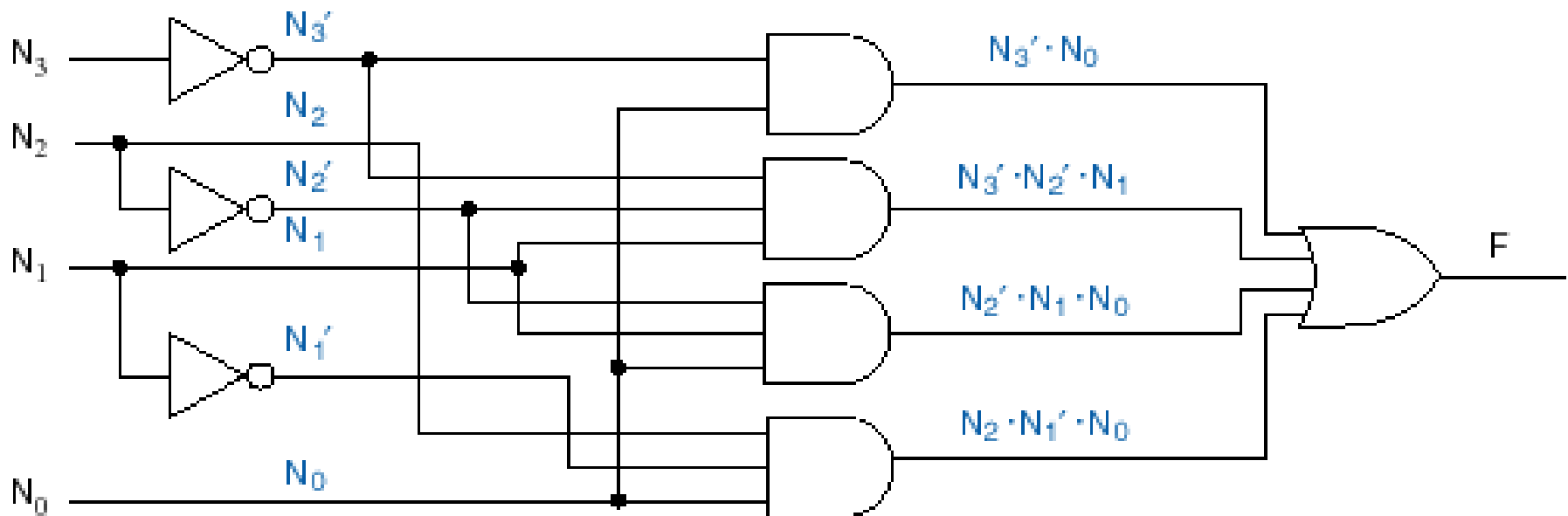
- Dibujar los “1” correspondientes al los minterms de la función
- Encuadrar el mayor número posible de grupos rectangulares de “1” hasta que no quede ninguno libre
 - El número de “1” en cada rectángulo tiene que ser múltiplo de 2
 - Está permitido pasar los bordes del mapa (en vertical y horizontal)
- Para cada uno de los rectángulos, obtener el término producto
 - Si la variable es “1” -> incluir la variable
 - Si la variable es “0” -> incluir la variable complementada
 - Si la variable vale a la vez “0” y “1” -> no incluir la variable
- Los rectángulos marcados y sus correspondientes términos producto se llaman “principales implicados”
- Esta minimización permite mínimo número de puertas y de entradas por puerta

Ejemplo: Detector de números primos I



Ejemplo: Detector de números primos II

- Este circuito tiene tres entradas de puerta menos que en la solución algebraica anterior



Otro ejemplo

W X		W			
		00	01	11	10
Y Z	00	0 1	4 1	12 1	8
	01	1 1	5 1	13 1	9 1
	11	3 1	7	15 1	11 1
	10	2	6	14 1	10

Brackets in the original image indicate groupings: X for columns 01 and 11, Z for rows 01 and 11, and W for columns 11 and 10.

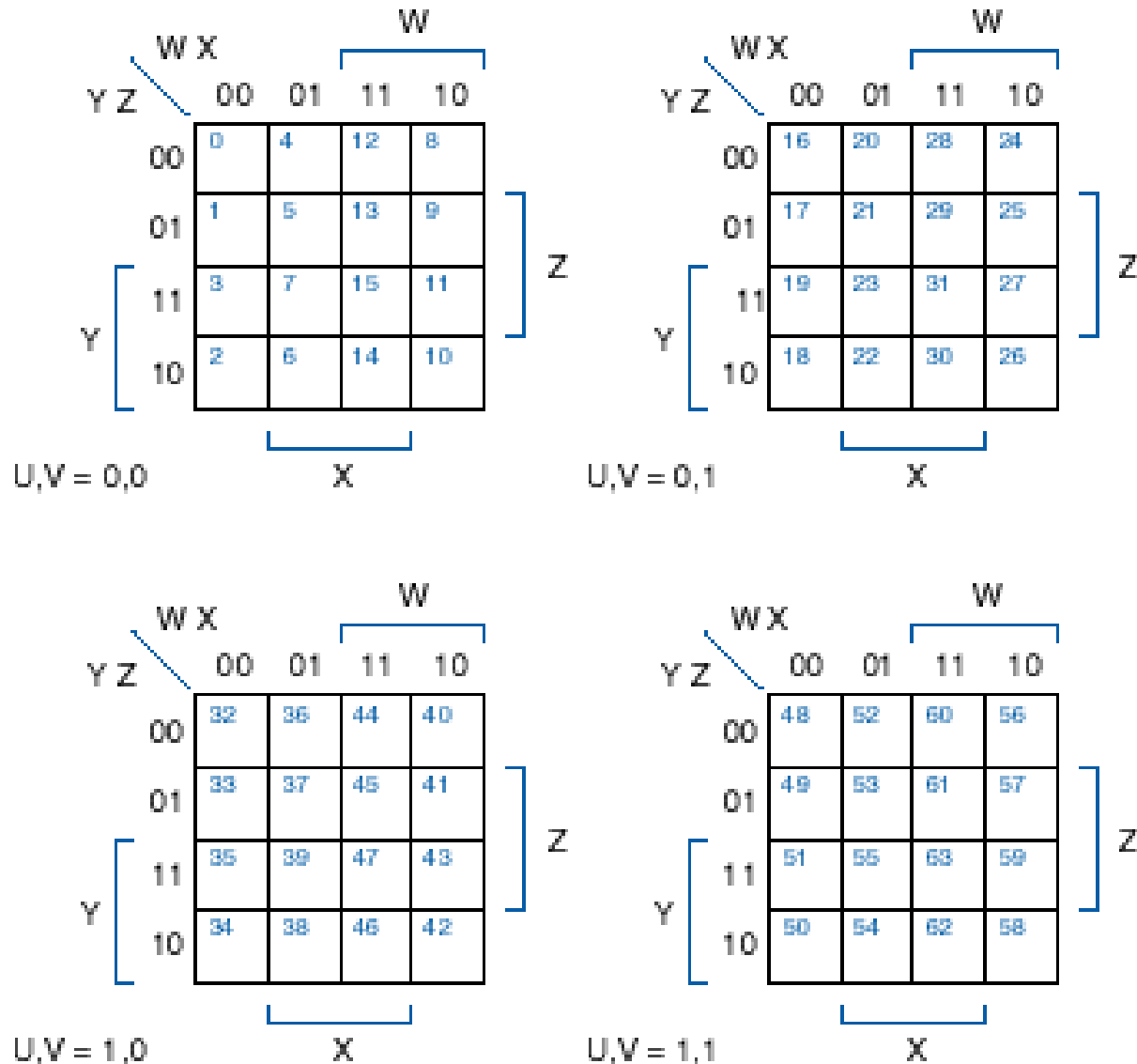
$$F = \Sigma_{W,X,Y,Z}(1,3,4,5,9,11,12,13,14,15)$$

W X		W			
		00	01	11	10
Y Z	00		1	1	
	01	1	1	1	1
	11	1		1	1
	10			1	

Brackets and labels in the original image indicate groupings for the simplified expression: $X \cdot Y'$ (columns 01, 11), $W \cdot Z$ (rows 00, 01), $X' \cdot Z$ (columns 00, 01), and $W \cdot X$ (columns 11, 10).

$$F = X \cdot Y' + X' \cdot Z + W \cdot X$$

Mapa de Karnaugh de seis variables



Diseño lógico en el mundo real

- Los circuitos tienen muchas entradas, por lo que los mapas de Karnaugh no pueden utilizarse, ya que es difícil manejar manualmente más de 6 entradas
- La corrección del diseño es mucho más importante que la minimización de puertas
 - Utilización de lenguajes de descripción de alto nivel para especificar operaciones (HDL, C++, etc.)
- Utilización de programas para manipulación de expresiones lógicas y minimización de la lógica
 - PALASM, ABEL, CUPL – desarrollado para PLDs
 - VHDL, Verilog – desarrollado para ASICs