

# Implementación de funciones booleanas

Tema 4



**Universidad**  
de La Laguna

©Sistemas Electrónicos Digitales  
Grado en Ingeniería Informática

# Contenido

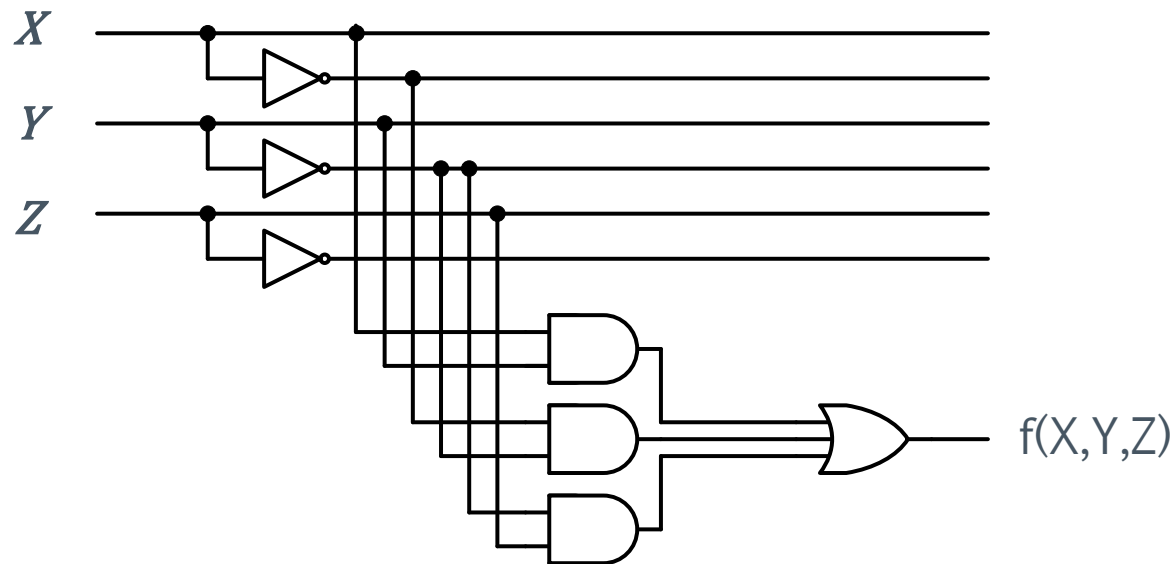
- › Implementación de funciones POS y SOP
- › Síntesis de funciones con puertas NAND y NOR
- › Otras implementaciones
- › Funciones lógicas según el número de variables

# Implementación de funciones POS y SOP

- › POS: Product of sums
- › SOP: Sum of products
- › Características:
  - Popular: uso de puertas básicas NOT, AND y OR
  - Implementación en 2 etapas:
    - › Red AND-OR para funciones SOP
    - › Red OR-AND para funciones POS

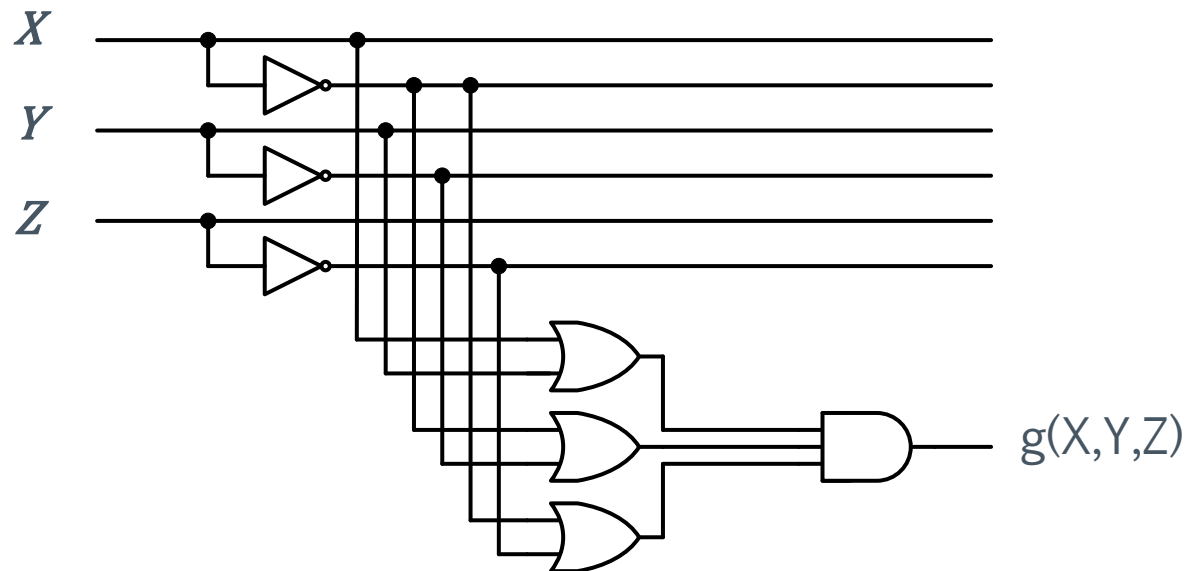
# Ejemplo función Suma de Productos

$$f(X,Y,Z) = X \cdot Y + \bar{X} \cdot \bar{Y} + \bar{Y} \cdot Z$$



# Ejemplo función Producto de Sumas

$$g(X,Y,Z) = (X + Y) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{X} + \bar{Z})$$



# Síntesis de funciones con puertas NAND y NOR

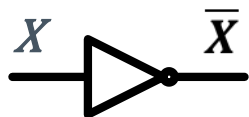
- › Cualquier función se puede implementar usando exclusivamente puertas NAND ó NOR
- › La alternativa anterior requiere de 3 tipos de puertas (AND, OR, NOT)
- › Condición necesaria y suficiente:
  - Basta con que las 3 operaciones básicas NOT, AND y NOR sean implementables sólo con NAND o sólo con NOR
  - Para esto se usa el teorema de Morgan

# Síntesis de funciones con puertas NAND y NOR

## › Funciones básicas con NAND/NOR

**Operación**

$$f = \bar{X}$$



$$f = X + Y$$

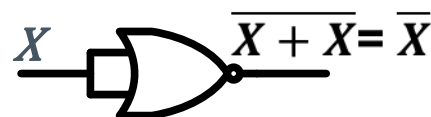


$$f = X \cdot Y$$

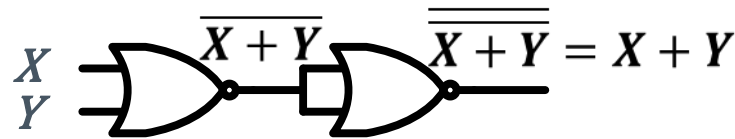


**Con NOR**

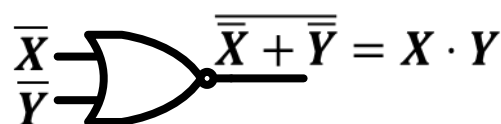
$$f = \overline{X + X}$$



$$f = \overline{\overline{X + Y}}$$

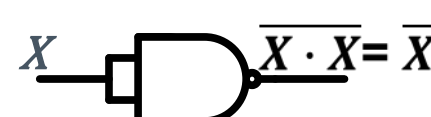


$$f = \overline{\overline{X} + \overline{Y}}$$

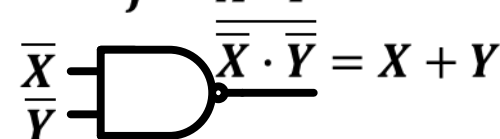


**Con NAND**

$$f = \overline{X \cdot X}$$



$$f = \overline{\overline{X} \cdot \overline{Y}}$$



$$f = \overline{\overline{X} \cdot \overline{Y}}$$

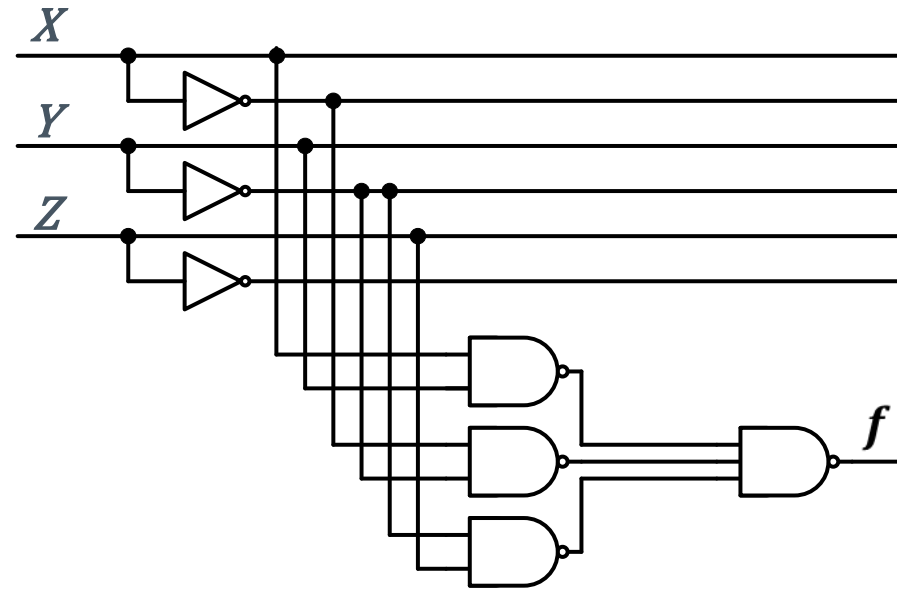
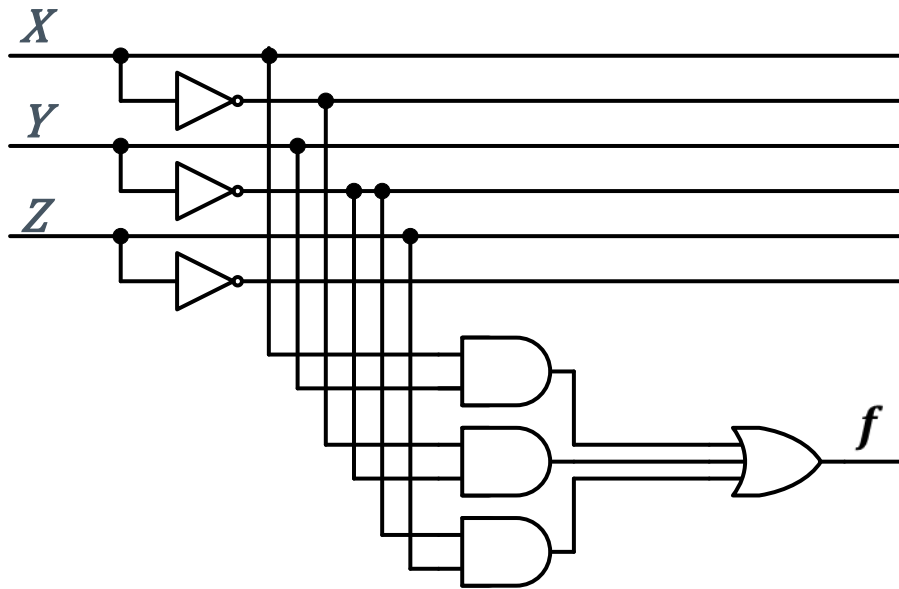


# SOP con puertas NAND

Implementar con una red NAND la función  $f = X \cdot Y + \bar{X} \cdot \bar{Y} + \bar{Y} \cdot Z$

$$f = X \cdot Y + \bar{X} \cdot \bar{Y} + \bar{Y} \cdot Z = f_1 + f_2 + f_3$$

$$f = \bar{\bar{f}} = \overline{\overline{f_1 + f_2 + f_3}} = \overline{\overline{f_1} \cdot \overline{f_2} \cdot \overline{f_3}} = \overline{\bar{X} \cdot \bar{Y} \cdot \bar{\bar{X}} \cdot \bar{\bar{Y}} \cdot \bar{\bar{Y}} \cdot Z}$$





# SOP con puertas NOR

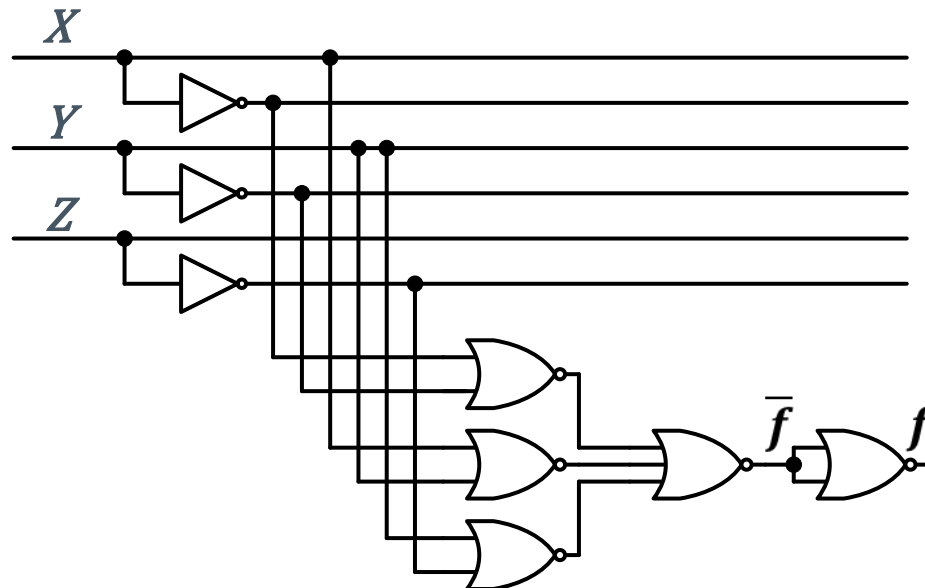
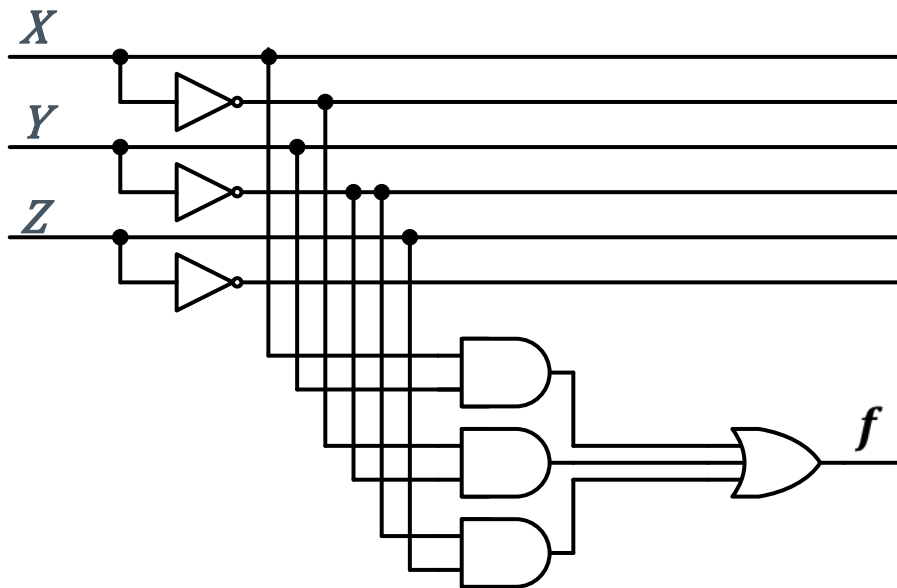
Implementar con una red NOR la función  $f = X \cdot Y + \bar{X} \cdot \bar{Y} + \bar{Y} \cdot Z$

$$f = X \cdot Y + \bar{X} \cdot \bar{Y} + \bar{Y} \cdot Z = f_1 + f_2 + f_3 = \overline{\overline{f_1 + f_2 + f_3}} = \overline{\overline{X + Y} + \overline{X + Y} + \overline{X + Z}}$$

$$f_1 = X \cdot Y = \overline{\overline{X} + \overline{Y}}$$

$$f_2 = \bar{X} \cdot \bar{Y} = \overline{X + Y}$$

$$f_3 = \bar{Y} \cdot Z = \overline{X + \bar{Z}}$$



# POS con puertas NAND

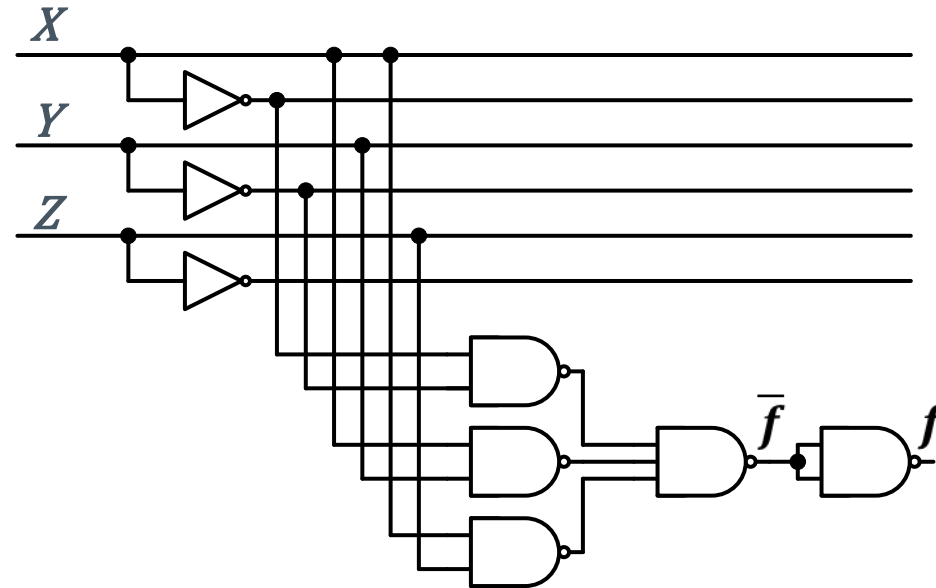
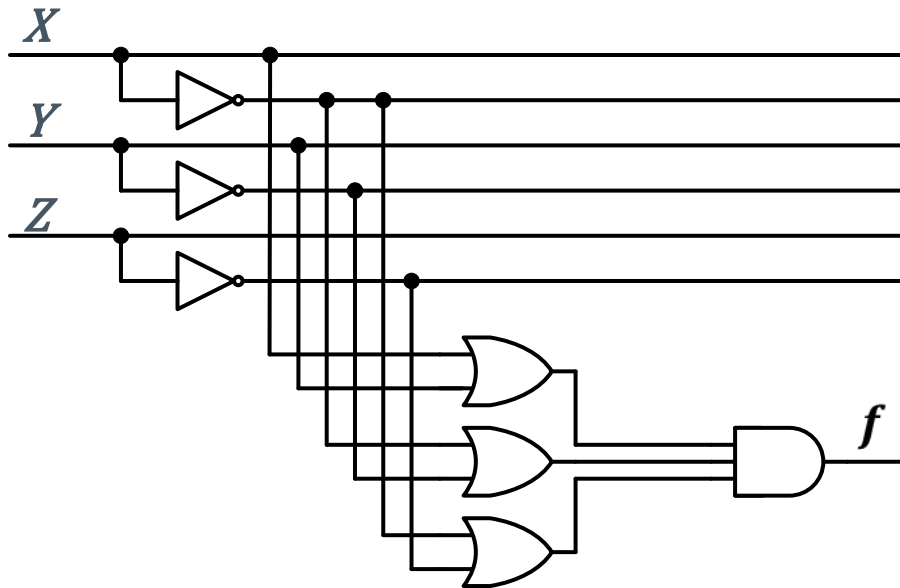
Implementar con una red NAND la función  $f = (X + Y) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{X} + \bar{Z})$

$$f = (X + Y) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{X} + \bar{Z}) = f_1 + f_2 + f_3 = \overline{\overline{f_1 \cdot f_2 \cdot f_3}} = \overline{\overline{\bar{X} \cdot \bar{Y} \cdot \bar{X} \cdot \bar{Y} \cdot \bar{X} \cdot \bar{Y}}}$$

$$f_1 = X + Y = \overline{\bar{X} \cdot \bar{Y}}$$

$$f_2 = \bar{X} + \bar{Y} = \overline{X \cdot Y}$$

$$f_3 = X + Z = \overline{\bar{X} \cdot \bar{Z}}$$

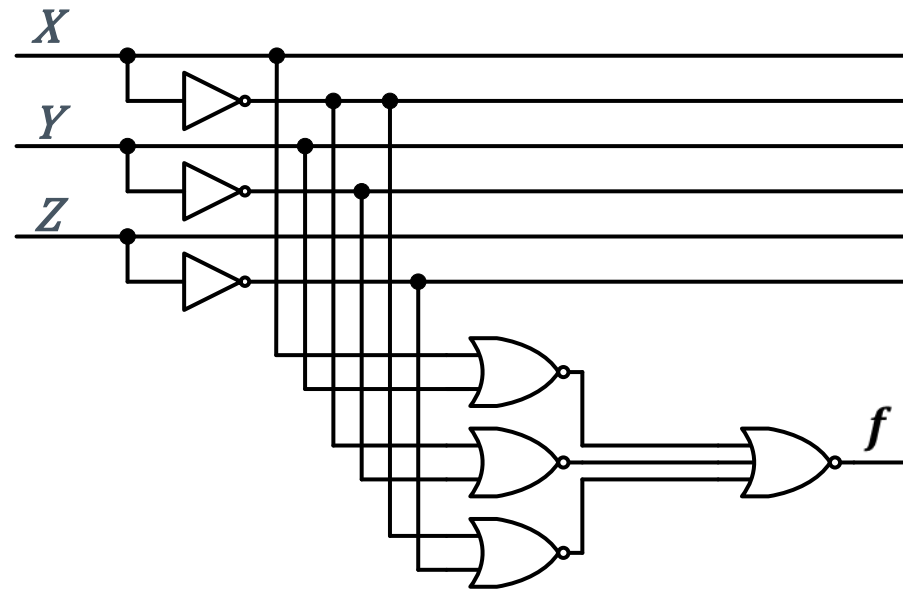
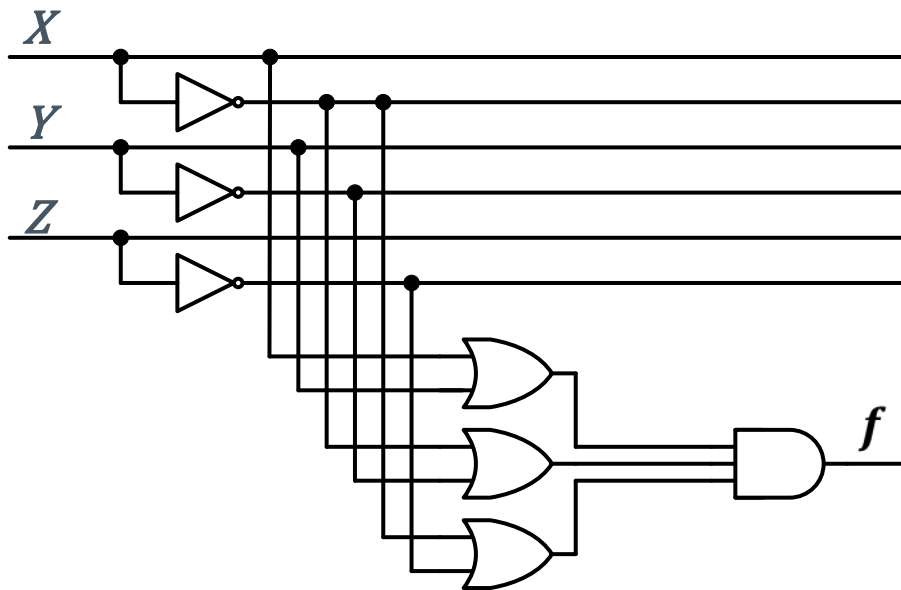


# POS con puertas NOR

Implementar con una red NOR la función  $f = (X + Y) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{X} + \bar{Z})$

$$f = (X + Y) \cdot (\bar{X} + \bar{Y}) \cdot (\bar{X} + \bar{Z}) = f_1 \cdot f_2 \cdot f_3$$

$$f = \overline{\overline{f}} = \overline{\overline{f_1 \cdot f_2 \cdot f_3}} = \overline{\overline{f_1} + \overline{f_2} + \overline{f_3}} = \overline{\overline{X + Y} + \overline{\bar{X} + \bar{Y}} + \overline{\bar{X} + \bar{Z}}}$$



# Otras implementaciones

- › En lugar de puertas lógicas básicas, se puede usar:
  - Multiplexores y decodificadores (tema 10)
  - Memorias: almacenan la Tabla de verdad de la función
  - Dispositivos lógicos configurables, como los dispositivos FPGA (Field Programmable Gate Array) que se usan en las prácticas de la asignatura

# Funciones lógicas según el número de variables

- › Para 2 variables, hemos visto las operaciones lógicas: NOT, AND, NAND, OR, NOR, XOR y XNOR
- › ¿Cuántas hay?
  - El número de operaciones distintas que pueden plantearse entre un número “n” de variables es  $2^{2^n}$

n	$2^n$	$2^{2^n}$	Operaciones no redundantes
1	2	4	3
2	4	16	6
3	8	256	22
4	16	65.536	402
5	32	4.294.967.296	1.228.158

# Funciones lógicas según el número de variables

- › Para 2 variables, tenemos:
  - 2 constantes: 0 y 1
  - 4 operaciones de una variable, transferencia y complemento
  - 10 operaciones con 2 variables, 6 de ellas son AND, OR, NAND, NOR, XOR y XNOR

xy	0	$\bar{x} \cdot \bar{y}$	$\bar{x}y$	$\bar{x}$	$x\bar{y}$	$\bar{y}$	$x \oplus y$	$\bar{x} + \bar{y}$	xy	$\overline{x \oplus y}$	y	$\bar{x} + y$	x	$x + \bar{y}$	x+y	1
00	0	1	0	1	0	1	0	1	0	1	0	1	0	1	0	1
01	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1
10	0	0	0	0	1	1	1	1	0	0	0	0	1	1	1	1
11	0	0	0	0	0	0	0	0	1	1	1	1	1	1	1	1

# Referencias

Angulo Usategui, García Zubía, “Sistemas Digitales y Tecnología de Computadores”, Thomson, 2003, ISBN 84-9732-042-5