



Titulación: Grado en Ingeniería de Computadores  
Asignatura: Tecnología de Computadores

## **Bloque 2:** Sistemas combinacionales

### **Tema 5:** Especificación y síntesis de circuitos combinacionales

Pablo Huerta Pellitero  
Carlos Sánchez de la Lama  
Luis Rincón Córcoles



# ÍNDICE

- Bibliografía
- Introducción
- Especificación de sistemas combinacionales
- Conceptos generales sobre puertas lógicas
- Puertas lógicas básicas
- Análisis y síntesis de circuitos combinacionales con puertas lógicas
  - Síntesis con puertas AND, OR, NOT
  - Análisis de redes de puertas AND, OR, NOT
  - Análisis de redes de puertas con lógica mixta
  - Conjuntos universales de puertas
  - Síntesis con puertas NAND, y con puertas NOR
  - Síntesis con lógica mixta



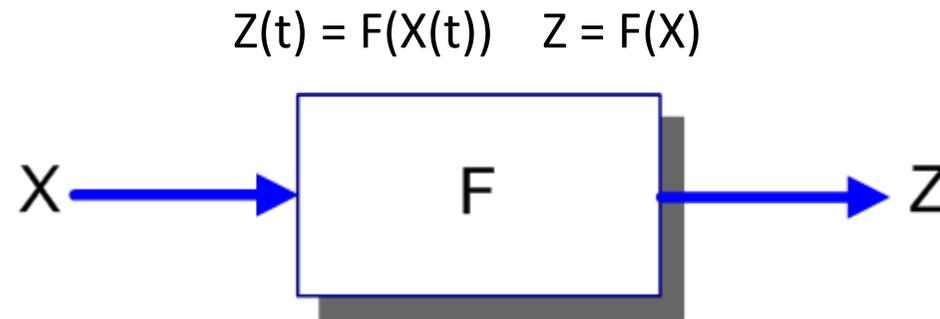
## BIBLIOGRAFÍA

- Román Hermida, Ana M<sup>o</sup> del Corral, Enric Pastor, Fermín Sánchez  
**“Fundamentos de Computadores”** , cap 2  
Editorial Síntesis
- Thomas L. Floyd  
**“Fundamentos de Sistemas Digitales”**, cap 5  
Editorial Prentice Hall
- Daniel D. Gajski  
**“Principios de Diseño Digital”**, cap 3, 4  
Editorial Prentice Hall
- M. Morris Mano  
**“Diseño Digital”**, cap 3, 4  
Editorial Prentice Hall



# INTRODUCCIÓN

- Hay dos tipos de sistemas digitales: combinacionales y secuenciales.
- En los sistemas combinacionales la salida  $Z$  del sistema en un determinado instante de tiempo  $t_i$  sólo depende del valor de la entrada  $X$  en ese mismo instante de tiempo  $t_i$ , por lo tanto se puede obviar la variable de tiempo  $t$ .



- En este tema se estudiará la forma de describir, especificar y materializar sistemas combinacionales sencillos.



# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- Bibliografía
- Introducción
- **Especificación de sistemas combinacionales**
- Conceptos generales sobre puertas lógicas
- Puertas lógicas básicas
- Análisis y síntesis de circuitos combinacionales con puertas lógicas
  - Síntesis con puertas AND, OR, NOT
  - Análisis de redes de puertas AND, OR, NOT
  - Análisis de redes de puertas con lógica mixta
  - Conjuntos universales de puertas
  - Síntesis con puertas NAND, y con puertas NOR
  - Síntesis con lógica mixta



# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- **Una especificación de alto nivel** de un sistema combinatorial se compone de un conjunto **(I,O,F)** formado por **entradas (I)**, **salidas (O)** y **la/s función/es** que realiza el sistema ( $F: I \rightarrow O$ )
- Ejemplo: especificación de alto nivel de un sistema combinatorial que calcula el módulo-5 de un dígito decimal.

$$\text{Módulo}_X(N) = \text{Resto}(N/X)$$

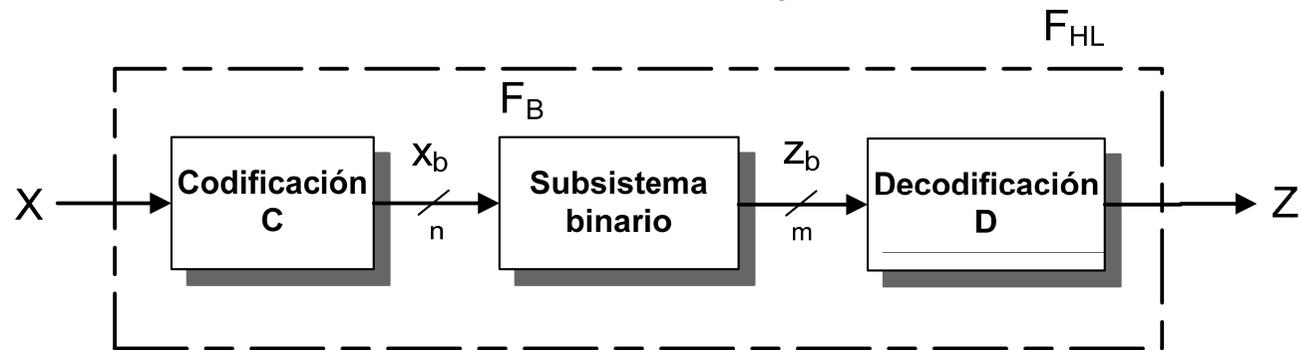
$$I = \{0, 1, 2, 3, 4, 5, 6, 7, 8, 9\}$$

$$O = \{0, 1, 2, 3, 4\}$$

<b>F: I</b>	0	1	2	3	4	5	6	7	8	9
<b>→ O</b>	0	1	2	3	4	0	1	2	3	4

# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- **Un sistema combinatorial** puede descomponerse en tres partes o subsistemas de **codificación, binario y decodificación**



- **El subsistema de codificación** traduce un valor de entrada  $X$  en un vector binario  $X_b$ , formalmente:  $C: \rightarrow \{0,1\}^n$ , donde  $\{0,1\}^n$  significa el conjunto de las combinaciones de 'n' dígitos binarios.
- Para poder representar todos los valores de  $l$  ( $k$  valores distintos), se precisan al menos  $n = \log_2 k$  bits para codificarlos.



# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- Ejemplo: una entrada X con 3 valores posibles (a, b ó c) puede codificarse con distintas combinaciones de 2 bits.

x	Codificación i	...	Codificación j
a	00		11
b	01		01
c	10		00

- Ejemplo: una entrada X con 10 valores posibles (0...9) puede codificarse con distintas combinaciones de 4 bits ( $2^4=16$  valores posibles).

Con la codificación binaria:  $\{(0,0000), (1,0001), (2,0010), \dots, (9,1001)\}$

- **El subsistema de decodificación** traduce un vector binario de entrada  $Z_b$  en un valor de salida Z, formalmente:  $D: \{0,1\}^m \rightarrow O$ . Es la operación inversa al subsistema de codificación.
- Ejemplo: una salida Z con 5 valores posibles (0...4) puede codificarse con distintas combinaciones de 3 bits ( $2^3= 8$  valores posibles).

Con codificación binaria:  $\{(0,000), (1,001), (2,010), (3,011), (4,100)\}$



# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- **El subsistema binario** traduce un vector binario  $X_b$  de  $n$  bits en un vector binario  $Z_b$  de  $m$  bits; ie,  $F_b: \{0,1\}^n \rightarrow \{0,1\}^m$ . Esta relación a través de la función binaria  $F_b$  es una **especificación binaria**
- Las funciones  $F_b$  de la especificación binaria son funciones booleanas, y se pueden representar tal como se vio en el tema anterior con: **tablas de verdad, expresiones de conmutación, mapas de Karnaugh, etc.**
- Ejemplo: el sistema combinacional que calcula el módulo-5 se puede dividir en:

<b>C</b>	0	1	2	3	4	5	6	7	8	9
	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
<b>F: I</b>	0000	0001	0010	0011	0100	0101	0110	0111	1000	1001
<b>→ O</b>	000	001	010	011	100	000	001	010	011	100
<b>D</b>	000	001	010	011	100					
	0	1	2	3	4					



## ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

- **Funciones binarias no completamente especificadas:** se puede dar el caso de que no todos los posibles valores de entrada del subsistema binario tengan sentido, por tanto las funciones de salida no importa que valor tomen para determinados casos.
- **Ejemplo:** para el sistema combinatorial que calcula el módulo-5, las combinaciones de entradas 1010, 1011, 1100, 1101, 1110 y 1111 no se utilizan. Para estos valores de entrada da igual lo que valga la salida y en la tabla de verdad esto se denota con una 'X' o un '-'.

**Estos términos se denominan  
“términos irrelevantes” o “don't care”.**

- A la hora de simplificar una función lógica que tiene términos irrelevantes, éstos se pueden utilizar como mejor convenga (como un '1' o como un '0') para hacer subcubos lo más grandes posible en el mapa de Karnaugh.



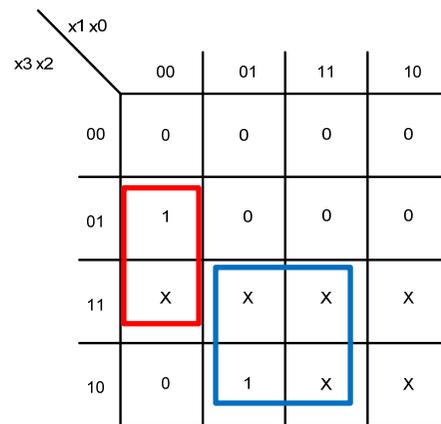
# ESPECIFICACIÓN DE SISTEMAS COMBINACIONALES

X3	X2	X1	X0	Z2	Z1	Z0
0	0	0	0	0	0	0
0	0	0	1	0	0	1
0	0	1	0	0	1	0
0	0	1	1	0	1	1
0	1	0	0	1	0	0
0	1	0	1	0	0	0
0	1	1	0	0	0	1
0	1	1	1	0	1	0
1	0	0	0	0	1	1
1	0	0	1	1	0	0
1	0	1	0	X	X	X
1	0	1	1	X	X	X
1	1	0	0	X	X	X
1	1	0	1	X	X	X
1	1	1	0	X	X	X
1	1	1	1	X	X	X

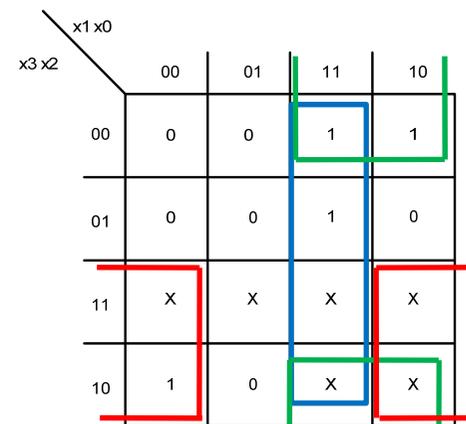
$$z_2 = \sum m(4,9) + \sum \Phi(10,11,12,13,14,15)$$

$$z_1 = \sum m(2,3,7,8) + \sum \Phi(10,11,12,13,14,15)$$

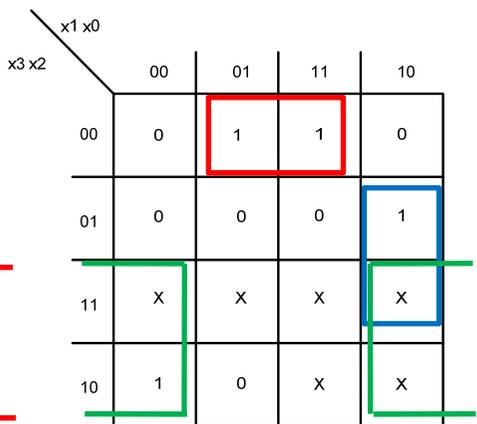
$$z_0 = \sum m(1,3,6,8) + \sum \Phi(10,11,12,13,14,15)$$



$$z_2 = x_2 \cdot \overline{x_1} \cdot \overline{x_0} + x_3 \cdot x_0$$



$$z_1 = x_3 \cdot \overline{x_0} + x_1 \cdot x_0 + \overline{x_2} \cdot x_1$$



$$z_0 = \overline{x_3} \cdot \overline{x_2} \cdot x_0 + x_2 \cdot x_1 \cdot \overline{x_0} + x_3 \cdot \overline{x_0}$$



# CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

- Bibliografía
- Introducción
- Especificación de sistemas combinacionales
- **Conceptos generales sobre puertas lógicas**
- Puertas lógicas básicas
- Análisis y síntesis de circuitos combinacionales con puertas lógicas
  - Síntesis con puertas AND, OR, NOT
  - Análisis de redes de puertas AND, OR, NOT
  - Análisis de redes de puertas con lógica mixta
  - Conjuntos universales de puertas
  - Síntesis con puertas NAND, y con puertas NOR
  - Síntesis con lógica mixta



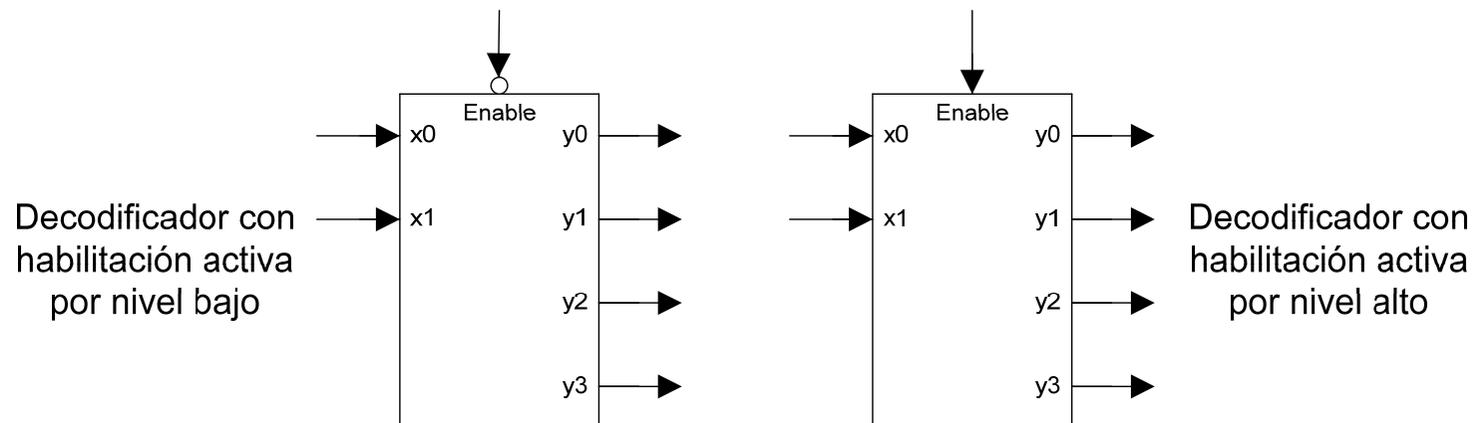
## CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

- La lógica permite la representación de la información y la realización de cálculos (mediante las operaciones lógicas).
- Valores en la lógica bivaluada: verdadero (true) y falso (false).
- En circuitos digitales se manejan voltajes con dos valores: alto (H) y bajo (L).
- La información que representan los voltajes se obtiene mediante la aplicación de un criterio de interpretación.
- Criterio más común: **lógica positiva**:
  - H – true
  - L – false
- Las señales se pueden interpretar mediante **lógica negativa**:
  - L – true
  - H – false
- A menudo ambas interpretaciones se mezclan dentro de un mismo circuito: **lógica mixta**.
  - Señales interpretadas mediante lógica positiva: **activas por nivel alto**.
  - Señales interpretadas mediante lógica negativa: **activas por nivel bajo**.



# CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

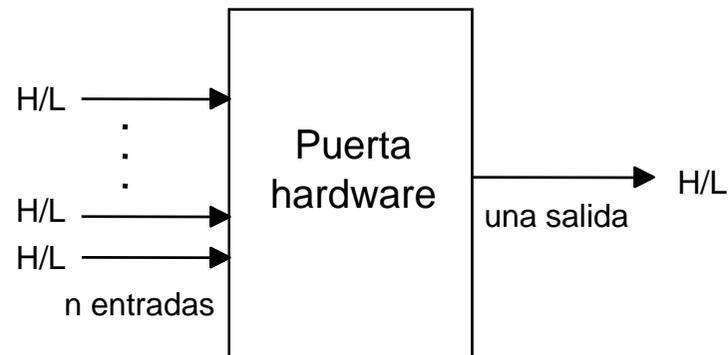
- La actividad de las señales se denota del siguiente modo:
  - Por un sufijo: .H (activa por nivel alto), .L (activa por nivel bajo).
  - Por el estado de la señal: afirmada (activa por nivel alto) o negada (activa por nivel bajo). **Esta representación es la de la lógica positiva.**
  - Ejemplos:
    - Señal activa por nivel alto:  
 $A.H, A$
    - Señal activa por nivel bajo:  
 $A.L, \bar{A}$
- En los circuitos, a menudo la actividad se muestra mediante la ausencia o presencia de una burbuja.





## CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

- Una puerta hardware es un dispositivo hardware multientrada de una salida, en el que la tensión de salida (H/L) es una función conocida para todas las combinaciones de las tensiones (H/L) de las entradas.



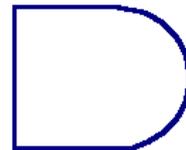
- Al aplicar una interpretación lógica (positiva, negativa o mixta) a las entradas y a las salidas, la salida de una puerta es una función booleana de las entradas  $\Rightarrow$  **puerta lógica**.



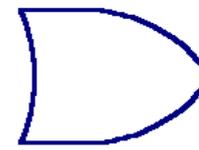
# CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

- Símbolos de las puertas lógicas en los circuitos de acuerdo con la función lógica que realizan:

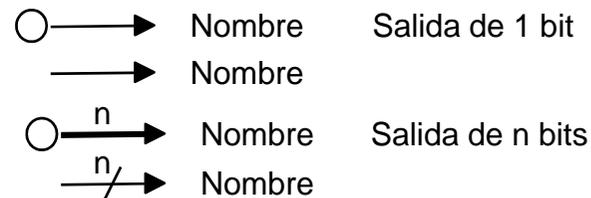
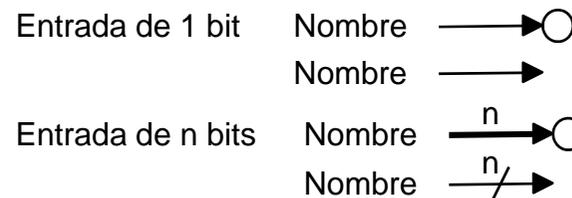
AND lógico



OR lógico



- A ambos símbolos se le añaden otros que indican el número de entradas y salidas, y los niveles de actividad.



- **Los nombres que reciben las puertas lógicas son los que corresponden con la función lógica que realizan si interpretamos las entradas y las salidas por lógica positiva.**



# CONCEPTOS GENERALES SOBRE PUERTAS LÓGICAS

## Resumen

- Las puertas lógicas (físicas) se definen mediante **niveles de tensión** (H/L).
- Las funciones lógicas se definen mediante **niveles lógicos** (verdadero 1 / falso 0).
- La relación entre ambos conceptos viene dada por el **nivel de activación** de una señal:

$$X.H \equiv X \quad / \quad X.L \equiv \overline{X}$$

**Activa alta**

$$T \Leftrightarrow H$$

$$F \Leftrightarrow L$$

**Activa baja**

$$T \Leftrightarrow L$$

$$F \Leftrightarrow H$$

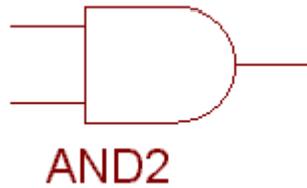


# PUERTAS LÓGICAS BÁSICAS

- Bibliografía
- Introducción
- Especificación de sistemas combinacionales
- Conceptos generales sobre puertas lógicas
- **Puertas lógicas básicas**
- Análisis y síntesis de circuitos combinacionales con puertas lógicas
  - Síntesis con puertas AND, OR, NOT
  - Análisis de redes de puertas AND, OR, NOT
  - Análisis de redes de puertas con lógica mixta
  - Conjuntos universales de puertas
  - Síntesis con puertas NAND, y con puertas NOR
  - Síntesis con lógica mixta

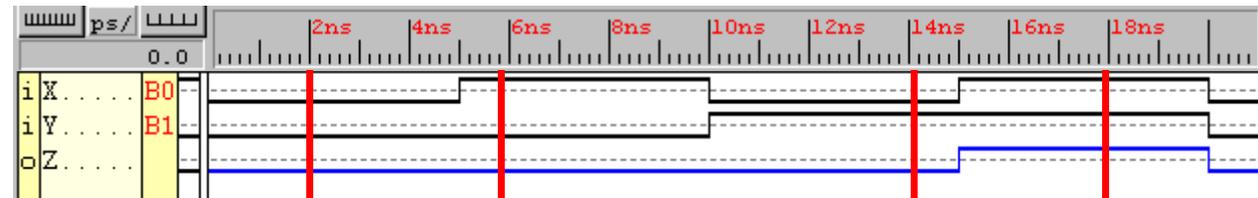
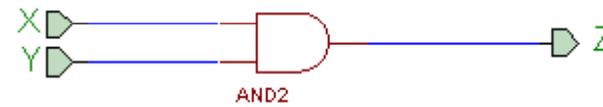


# PUERTAS LÓGICAS BÁSICAS: AND



$$z = x \cdot y$$

$$z = x \text{ and } y = \text{and}(x, y) = f_{\text{and}}(x, y)$$



x	y	z
L	L	L
L	H	L
H	L	L
H	H	H



## PUERTAS LÓGICAS BÁSICAS: AND

- Interpretando entradas y salidas por lógica positiva: función AND

X	Y	Z
L	L	L
L	H	L
H	L	L
H	H	H



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



$$Z = X \cdot Y$$

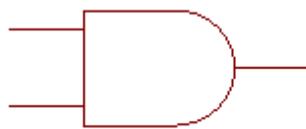


# PUERTAS LÓGICAS BÁSICAS: AND

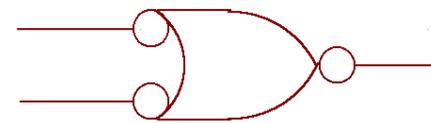
- Negando la función obtenemos que:

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

- Siguiendo el criterio de que una señal negada es en realidad una señal activa por nivel bajo (interpretada por lógica negativa) vemos que la puerta AND también puede realizar la función OR, sin más que interpretar sus entradas y su salida por nivel bajo.



AND2



X	Y	Z
L	L	L
L	H	L
H	L	L
H	H	H

X.L	Y.L	Z.L
1	1	1
1	0	1
0	1	1
0	0	0

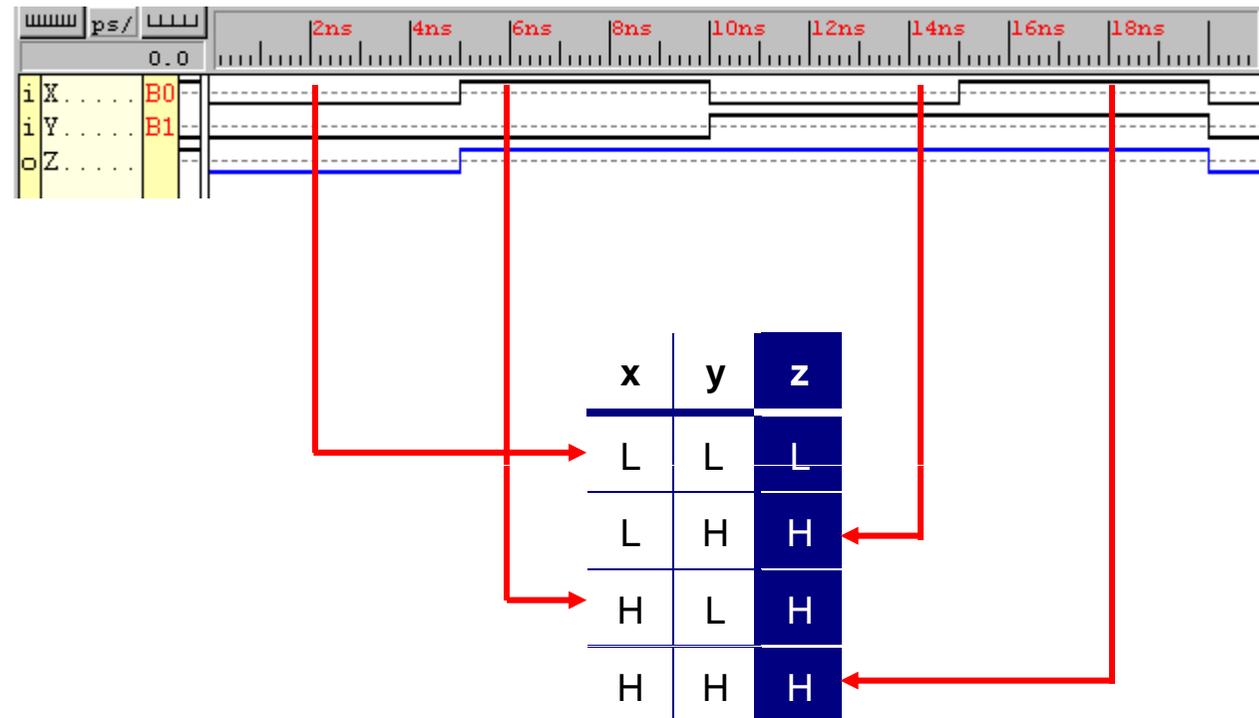


# PUERTAS LÓGICAS BÁSICAS: OR



$$z = x + y$$

$$z = x \text{ or } y = \text{or}(x, y) = f_{\text{or}}(x, y)$$





# PUERTAS LÓGICAS BÁSICAS: OR

- Interpretando entradas y salidas por lógica positiva: función OR

X	Y	Z
L	L	L
L	H	H
H	L	H
H	H	H



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



$$Z = X + Y$$

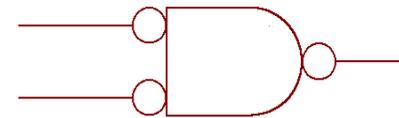


# PUERTAS LÓGICAS BÁSICAS: OR

- Negando la función obtenemos que:

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

- Siguiendo el criterio de que una señal negada es en realidad una señal activa por nivel bajo (interpretada por lógica negativa) vemos que la puerta OR también puede realizar la función AND, sin más que interpretar sus entradas y su salida por nivel bajo.



X	Y	Z
L	L	L
L	H	H
H	L	H
H	H	H

X.L	Y.L	Z.L
1	1	1
1	0	0
0	1	0
0	0	0

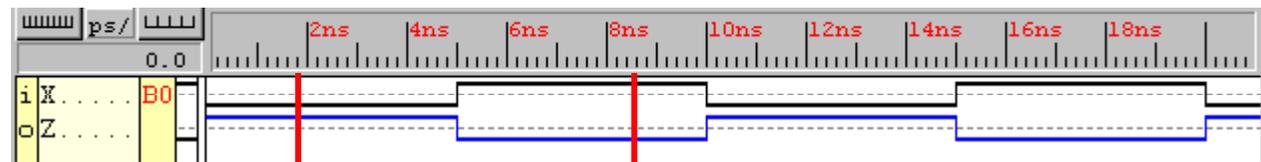


# PUERTAS LÓGICAS BÁSICAS: NOT



$$z = \bar{x}$$

$$z = \text{not}(x) = f_{\text{not}}(x)$$



x	z
L	H
H	L



## PUERTAS LÓGICAS BÁSICAS: NOT



- El símbolo de la puerta NOT se explica del siguiente modo:
  - El triángulo representa un **driver** (regenerador del nivel físico de la señal), que no realiza ninguna función lógica.
  - La burbuja indica que la salida es activa por nivel bajo, o lo que es lo mismo, tiene el nivel físico contrario al esperado.
- La presencia de la burbuja sugiere que debemos interpretar la salida como una señal activa por nivel bajo, mientras que la entrada se debería interpretar como una señal activa por nivel alto.



## PUERTAS LÓGICAS BÁSICAS: NOT

- Interpretando entradas y salidas por lógica positiva (ambas activas por nivel alto): función NOT

X	Z
L	H
H	L

X	Z
0	1
1	0



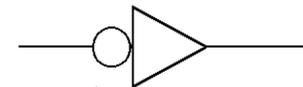
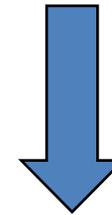
$$Z = \overline{X}$$

Leyendo el dibujo



$$\overline{Z} = X$$

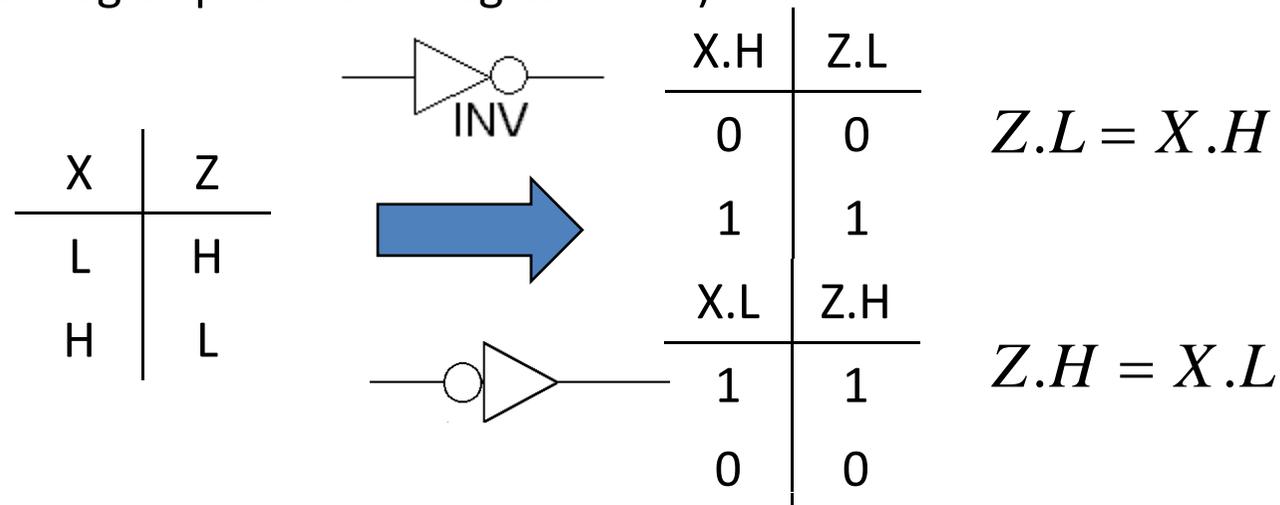
Leyendo la ecuación





## PUERTAS LÓGICAS BÁSICAS: NOT

- La presencia de la burbuja sugiere que debemos interpretar la señal del lado en que se encuentra la misma por lógica negativa (manteniendo la otra señal en lógica positiva  $\Rightarrow$  lógica mixta):



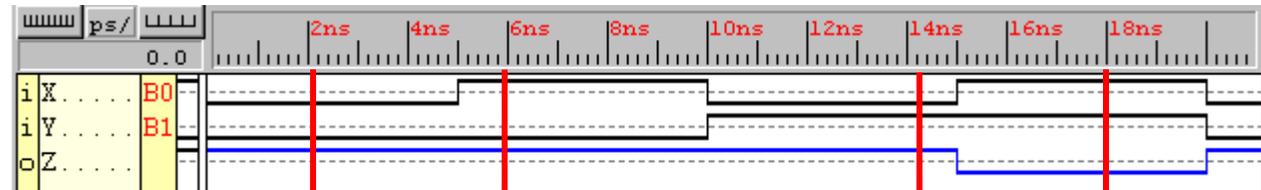
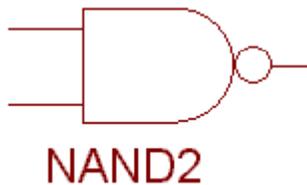
- ¡En lógica mixta, la puerta NOT no realiza ninguna función lógica!
  - La operación que realiza una puerta NOT interpretada por lógica mixta es invertir el nivel físico de la señal.
- Si usamos la notación de la lógica positiva, tenemos que la puerta NOT realiza la función NOT (como ya habíamos visto):

$$Z.L = X.H \Rightarrow \bar{Z} = X \quad Z.H = X.L \Rightarrow Z = \bar{X}$$



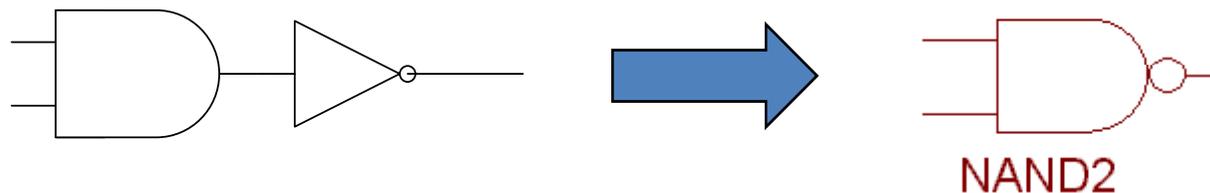
# PUERTAS LÓGICAS BÁSICAS: NAND

$$z = \overline{x \cdot y} = \bar{x} + \bar{y}$$



x	y	z
L	L	H
L	H	H
H	L	H
H	H	L

- Una puerta NAND equivale física y lógicamente a una puerta AND seguida de una NOT:





# PUERTAS LÓGICAS BÁSICAS: NAND

- Interpretando entradas y salidas por lógica positiva: función NAND

X	Y	Z
L	L	H
L	H	H
H	L	H
H	H	L



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



NAND2

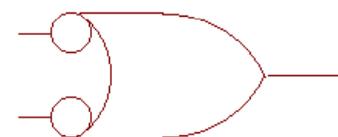
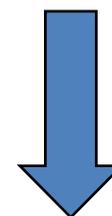
Leyendo el dibujo



$$\bar{Z} = X \cdot Y$$

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

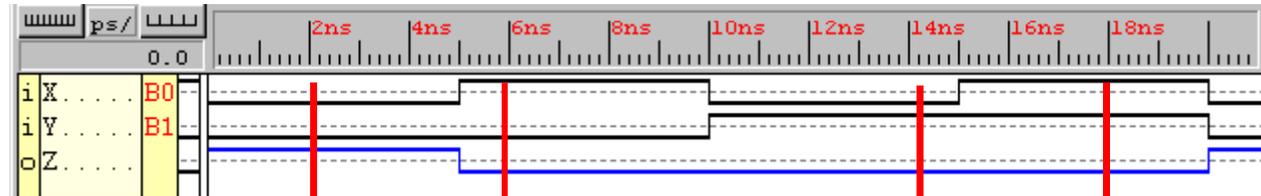
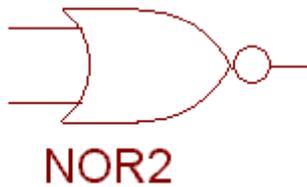
Leyendo la ecuación





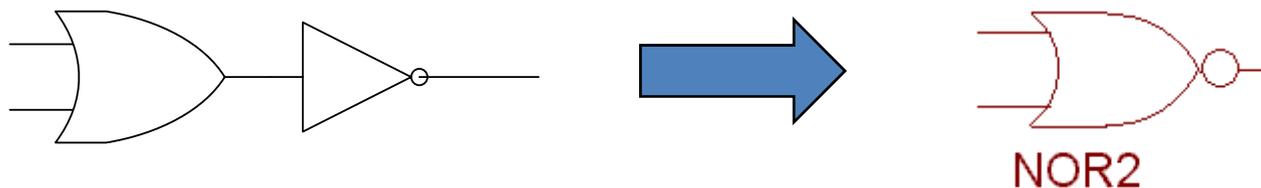
# PUERTAS LÓGICAS BÁSICAS: NOR

$$z = \overline{x + y} = \bar{x} \cdot \bar{y}$$



x	y	z
L	L	H
L	H	L
H	L	L
H	H	L

- Una puerta NOR equivale física y lógicamente a una puerta OR seguida de una NOT:





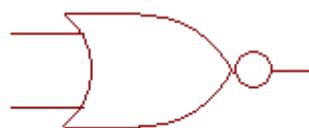
# PUERTAS LÓGICAS BÁSICAS: NOR

- Interpretando entradas y salidas por lógica positiva: función NOR

X	Y	Z
L	L	H
L	H	L
H	L	L
H	H	L



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



NOR2

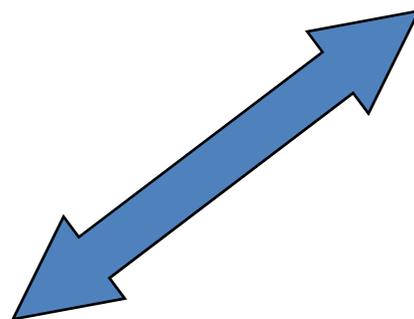
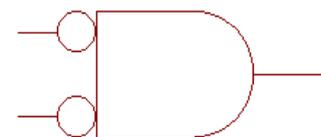
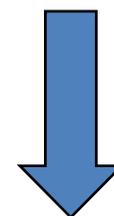
$$Z = \overline{X + Y} = \overline{X} \cdot \overline{Y}$$

Leyendo el dibujo



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

Leyendo la ecuación



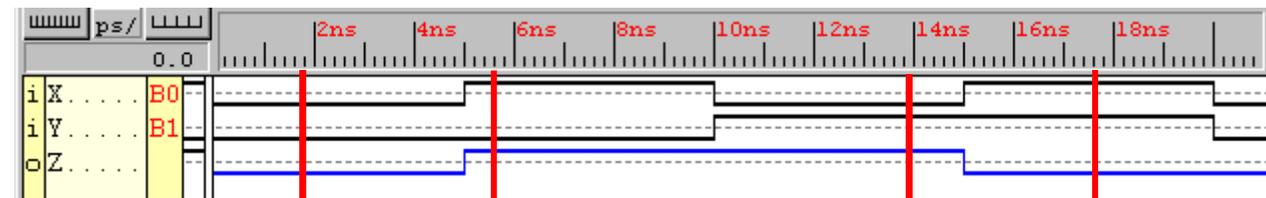


# PUERTAS LÓGICAS BÁSICAS: XOR

$$z = x \oplus y$$



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



x	y	z
L	L	L
L	H	H
H	L	H
H	H	L

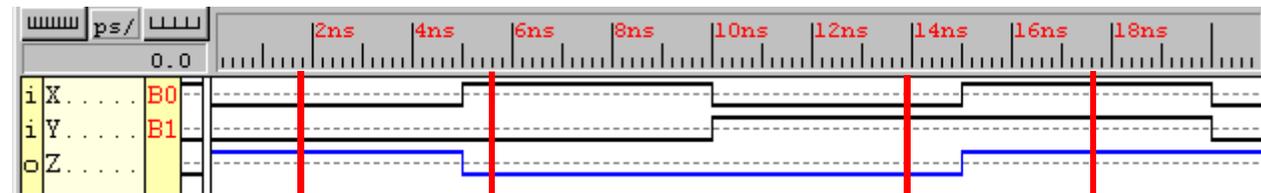


# PUERTAS LÓGICAS BÁSICAS: XNOR

$$z = \overline{x \oplus y}$$



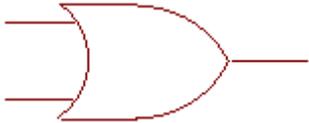
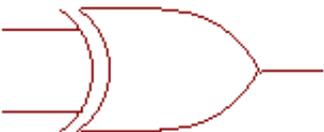
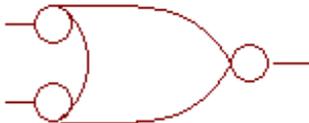
X	Y	Z
0	0	1
0	1	0
1	0	0
1	1	1



x	y	z
L	L	H
L	H	L
H	L	L
H	H	H



# PUERTAS LÓGICAS BÁSICAS

	Suma lógica	Producto lógico	Suma lógica exclusiva
Entradas y salida por lógica positiva	 OR	 AND	 XOR
Entradas por lógica positiva y salida por lógica negativa	 NOR	 NAND	 XNOR
Entradas por lógica negativa y salida por lógica positiva	 NAND	 NOR	 XOR
Entradas y salida por lógica negativa	 AND	 OR	 XNOR



# ANÁLISIS Y SÍNTESIS DE CIRCUITOS COMBINACIONALES CON PUERTAS LÓGICAS

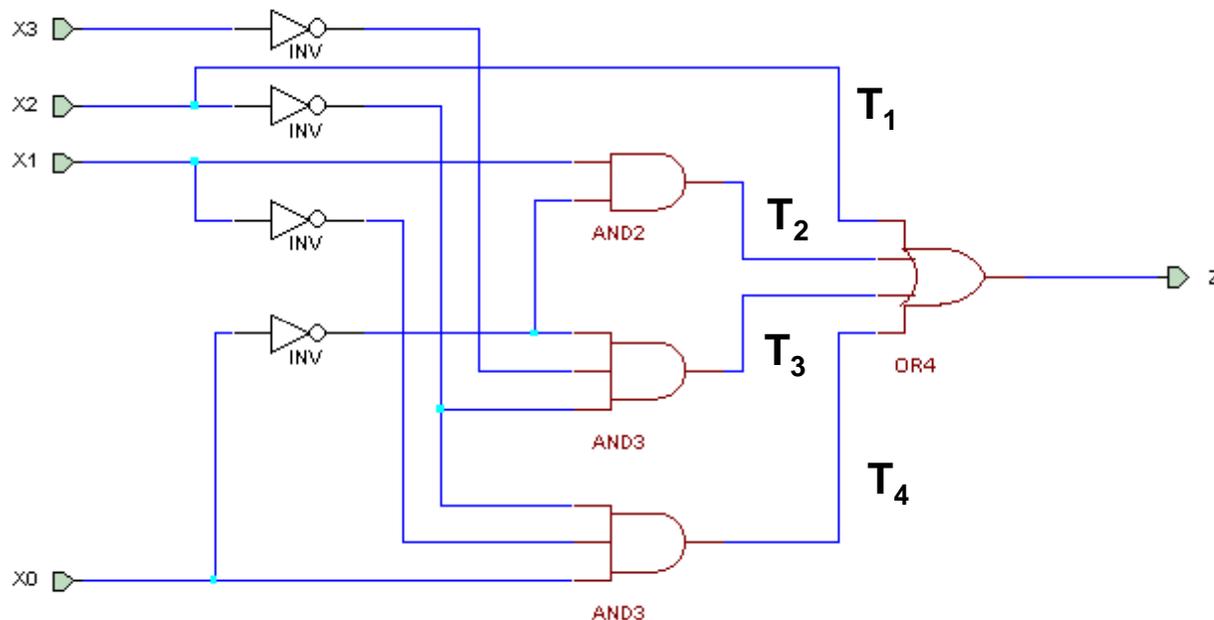
- Bibliografía
- Introducción
- Especificación de sistemas combinacionales
- Conceptos generales sobre puertas lógicas
- Puertas lógicas básicas
- **Análisis y síntesis de circuitos combinacionales con puertas lógicas**
  - Síntesis con puertas AND, OR, NOT
  - Análisis de redes de puertas AND, OR, NOT
  - Análisis de redes de puertas con lógica mixta
  - Conjuntos universales de puertas
  - Síntesis con puertas NAND, y con puertas NOR
  - Síntesis con lógica mixta



# SÍNTESIS CON REDES DE PUERTAS AND-OR-NOT

- La síntesis de un sistema combinacional en forma de SdeP con puertas AND-OR-NOT es directa materializando los operadores que aparecen en la EC con sus equivalentes hardware.
- Ejemplo: sintetizar Z mediante una red de puertas AND-OR-NOT de dos niveles.

$$Z = x_2 + x_1 \cdot \bar{x}_0 + \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_2 \cdot \bar{x}_1 \cdot x_0$$

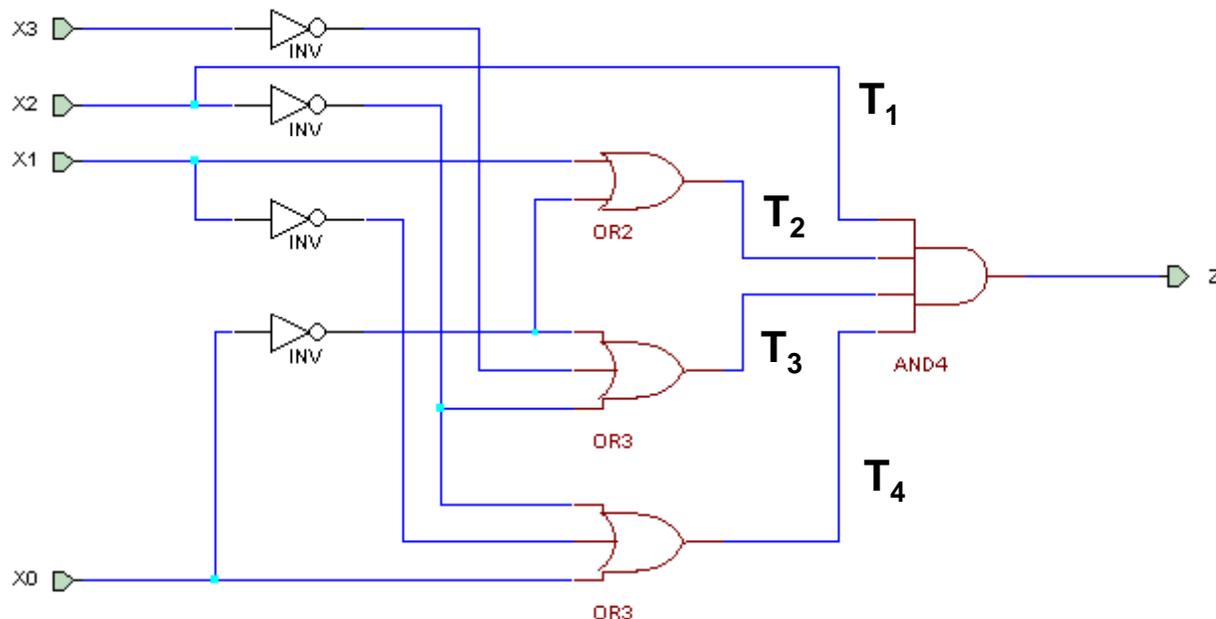




# SÍNTESIS CON REDES DE PUERTAS AND-OR-NOT

- La síntesis de un sistema combinacional en forma de PdeS con puertas AND-OR-NOT es directa materializando los operadores que aparecen en la EC con sus equivalentes hardware.
- Ejemplo: sintetizar Z mediante una red de puertas AND-OR-NOT de dos niveles.

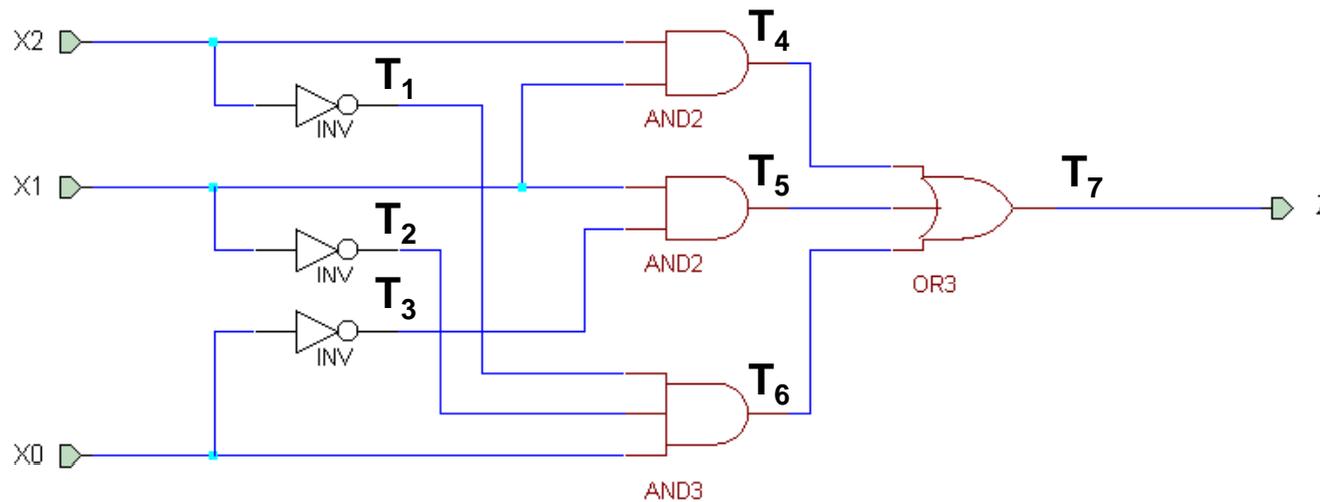
$$Z = x_2 \cdot (x_1 + \bar{x}_0) \cdot (\bar{x}_3 + \bar{x}_2 \cdot \bar{x}_0) \cdot (\bar{x}_2 + \bar{x}_1 + x_0)$$





# ANÁLISIS DE REDES DE PUERTAS AND-OR-NOT

- El análisis persigue obtener las expresiones de conmutación que describen el comportamiento del circuito expresando la salida en función de las entradas.
- Ejemplo:



$$T_1 = \bar{x}_2$$

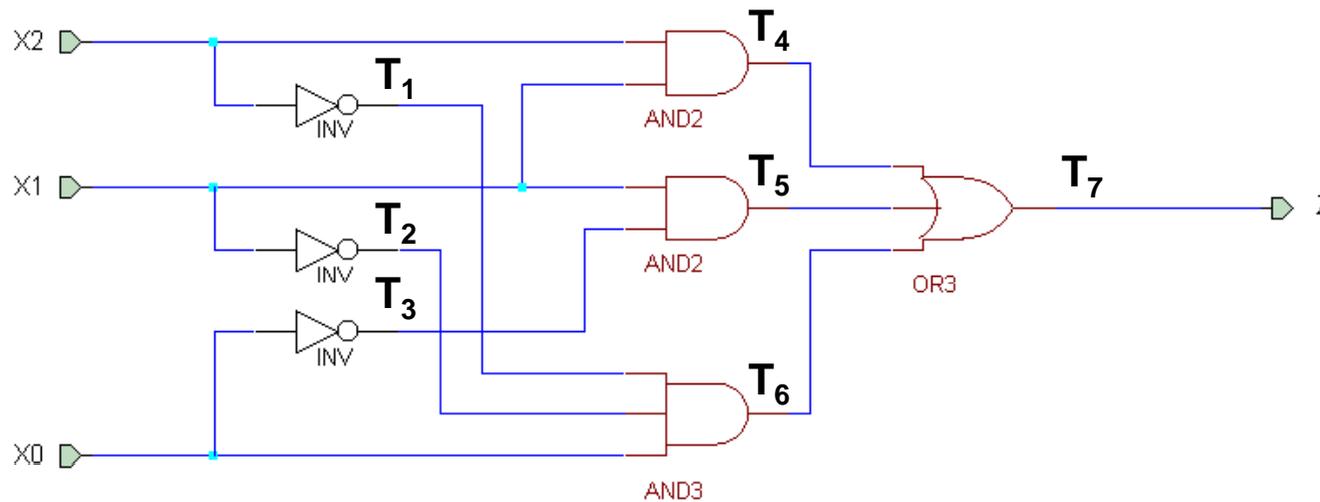
$$T_2 = \bar{x}_1$$

$$T_3 = \bar{x}_0$$



# ANÁLISIS DE REDES DE PUERTAS AND-OR-NOT

- El análisis persigue obtener las expresiones de conmutación que describen el comportamiento del circuito expresando la salida en función de las entradas.
- Ejemplo:

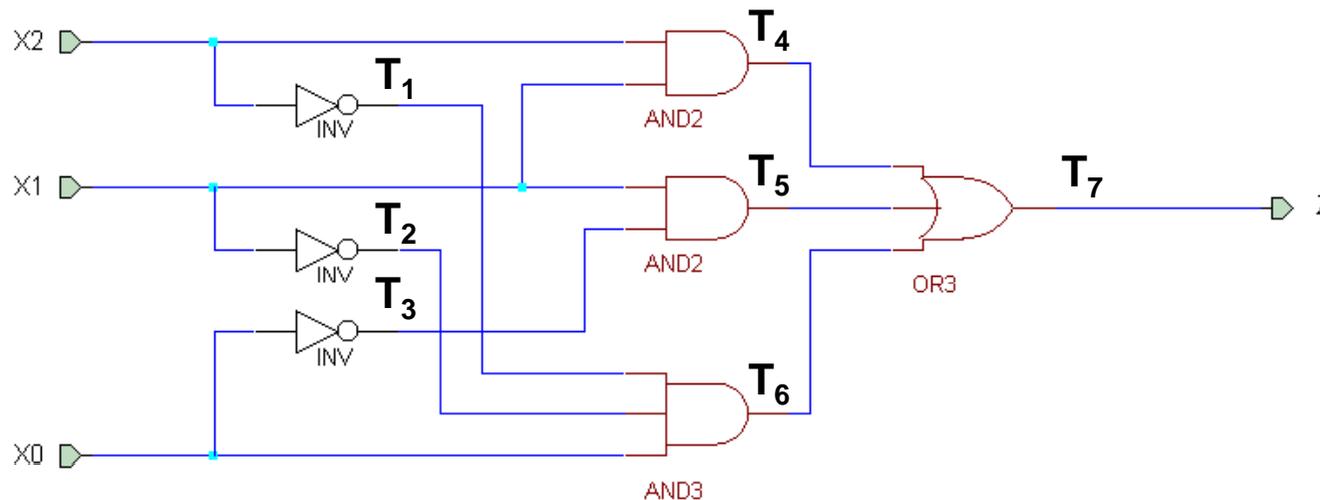


$$\begin{aligned} T_1 &= \bar{x}_2 & T_4 &= x_2 \cdot x_1 \\ T_2 &= \bar{x}_1 & T_5 &= x_1 \cdot T_3 = x_1 \cdot \bar{x}_0 \\ T_3 &= \bar{x}_0 & T_6 &= T_1 \cdot T_2 \cdot x_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0 \end{aligned}$$



# ANÁLISIS DE REDES DE PUERTAS AND-OR-NOT

- El análisis persigue obtener las expresiones de conmutación que describen el comportamiento del circuito expresando la salida en función de las entradas.
- Ejemplo:



$$T_1 = \bar{x}_2$$

$$T_4 = x_2 \cdot x_1$$

$$T_2 = \bar{x}_1$$

$$T_5 = x_1 \cdot T_3 = x_1 \cdot \bar{x}_0$$

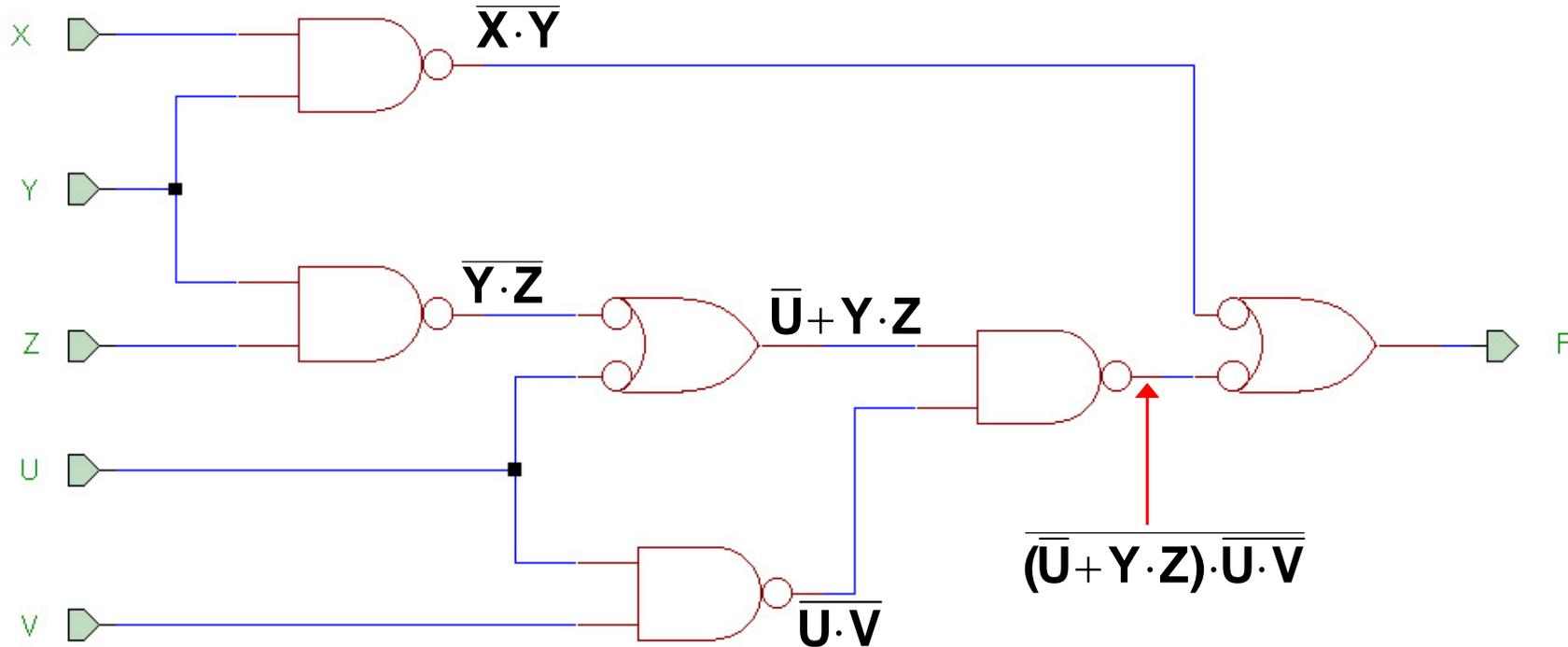
$$T_3 = \bar{x}_0$$

$$T_6 = T_1 \cdot T_2 \cdot x_0 = \bar{x}_2 \cdot \bar{x}_1 \cdot x_0$$

$$T_7 = T_4 + T_5 + T_6 = x_2 \cdot x_1 + x_1 \cdot \bar{x}_0 + \bar{x}_2 \cdot \bar{x}_1 \cdot x_0$$

# ANÁLISIS DE REDES DE PUERTAS CON LÓGICA MIXTA

- Cada vez que aparece una burbuja, negamos la señal.
  - Si en los dos extremos de un hilo aparece burbuja, tendríamos que aplicar doble negación, con lo cual nos ahorramos las dos.
- Ejemplo:



$$F = X \cdot Y + (\bar{U} + Y \cdot Z) \cdot \overline{U \cdot V}$$



## CONJUNTOS UNIVERSALES DE PUERTAS

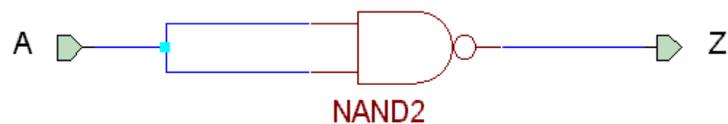
- Un sistema combinatorial puede representarse mediante expresiones de conmutación y éstas están formadas por variables y operadores **and**, **or** y **not**.
- Por tanto **cualquier sistema combinatorial puede materializarse mediante las puertas hardware que realizan esas funciones lógicas**. Por eso, al conjunto de puertas formado por la AND, OR y NOT se le conoce como conjunto universal.
- **En determinadas circunstancias**, por ejemplo diseñando con circuitos SSI, diseñando *Full Custom ASICs*, o debido a restricciones del problema, **se deben utilizar otros conjuntos de puertas más reducidos que también son universales**: {And, Not}, {Or, Not}, {Nand}, {Nor} o {Xor, and}.
- Para demostrar que cualquiera de estos conjuntos es universal basta comprobar que con las puertas integrantes del conjunto se pueden conseguir las tres funciones lógicas básicas And, Or y Not.



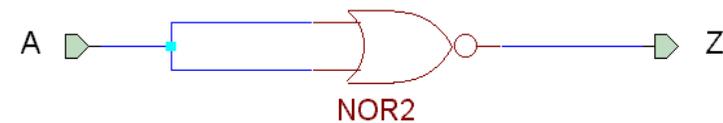
# CONJUNTOS UNIVERSALES DE PUERTAS

- Existen dos conjuntos universales útiles en diseño SSI por utilizar una única puerta lógica: la **Nand** o la **Nor**.

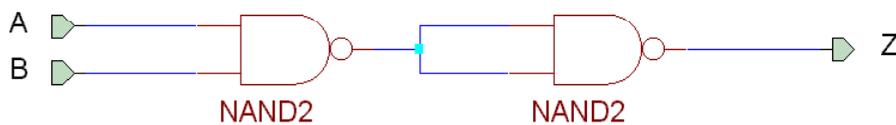
Not:  $\bar{a} = \bar{a} + \bar{a} = \overline{a \cdot a}$



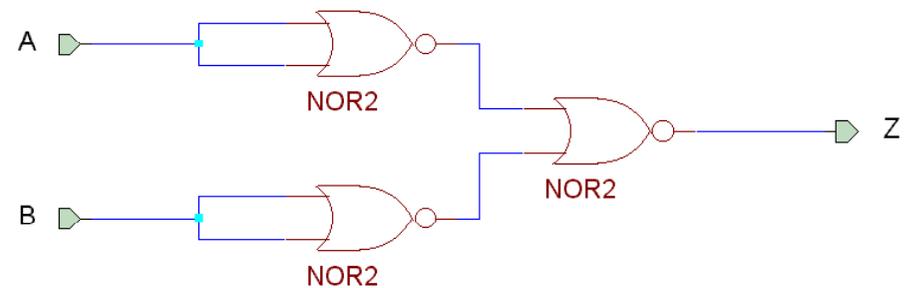
Not:  $\bar{a} = \bar{a} \cdot \bar{a} = \overline{a + a}$



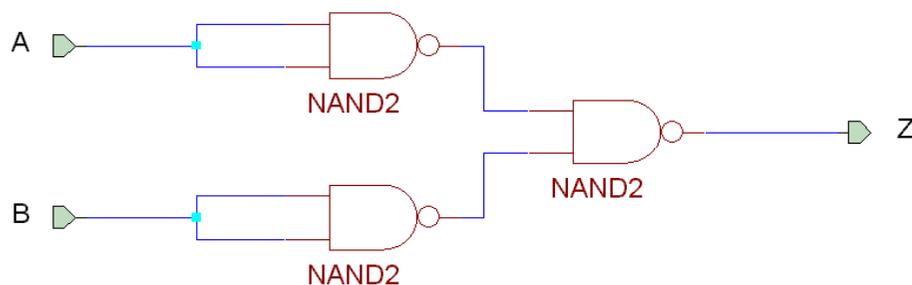
And:  $a \cdot b = \overline{\overline{a \cdot b}}$



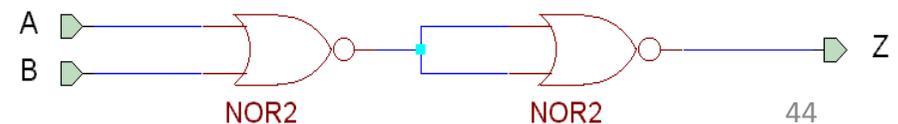
And:  $a \cdot b = \overline{\overline{a \cdot b}} = \overline{a + b}$



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



Or:  $a + b = \overline{\overline{a + b}}$

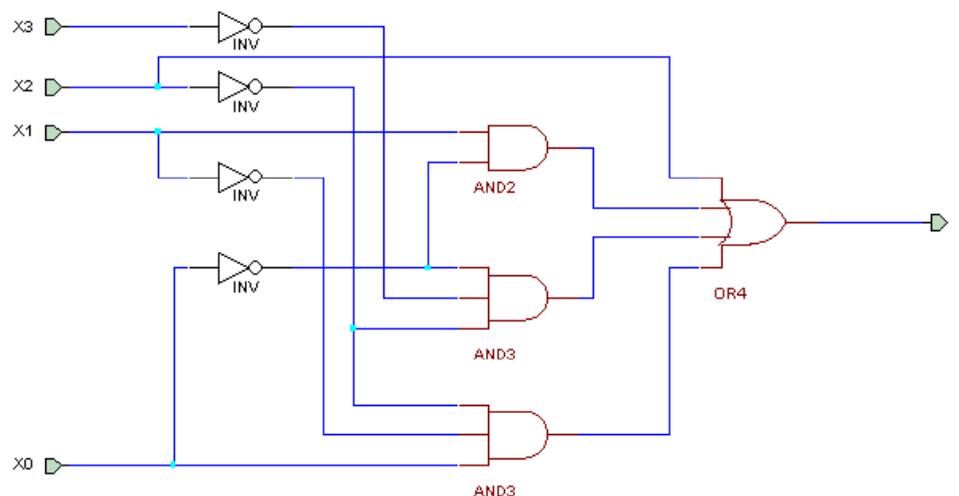




# SÍNTESIS CON REDES DE PUERTAS NAND

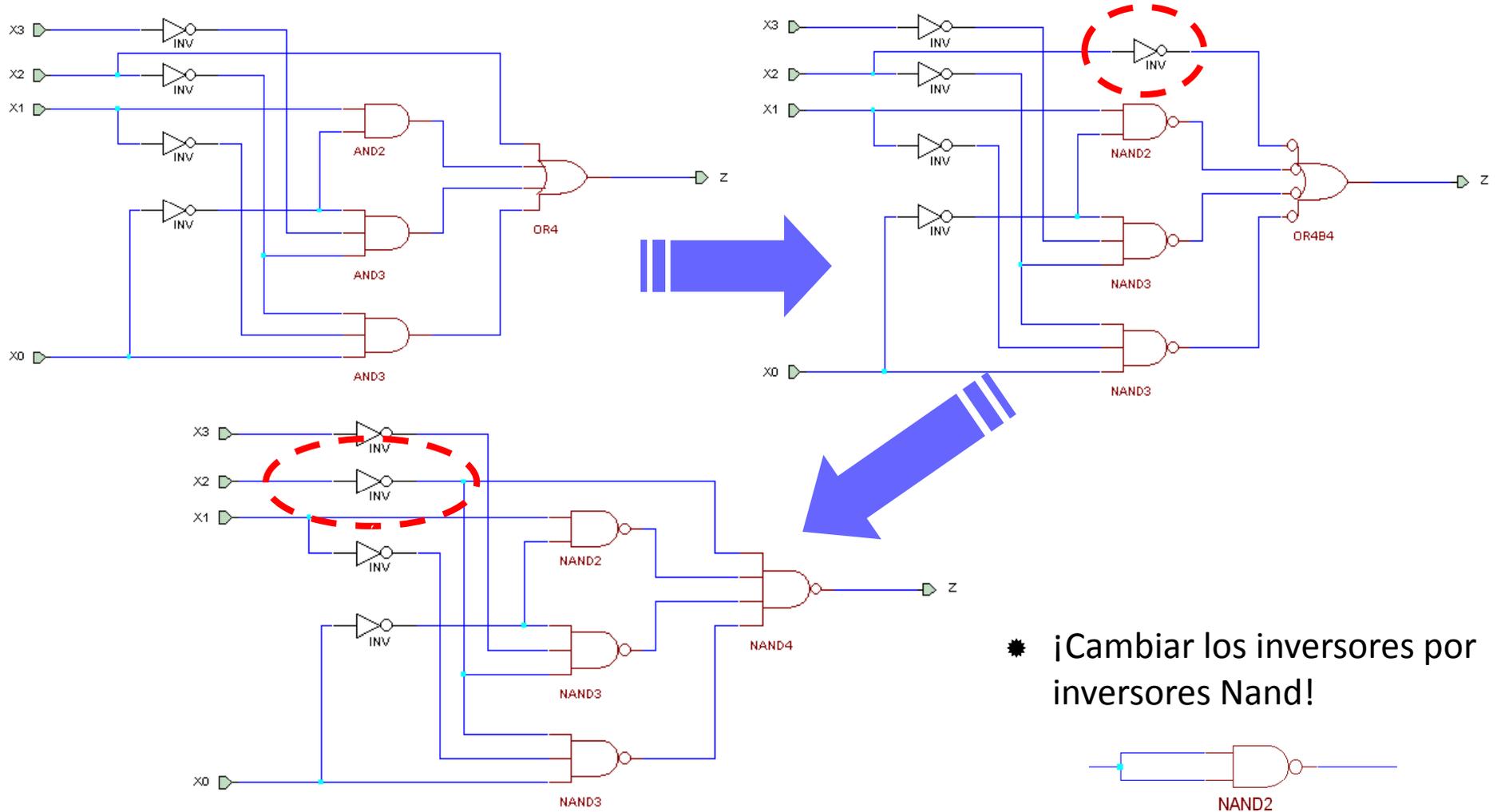
- La síntesis de un sistema combinacional en forma de suma de productos mediante sólo puertas Nand se puede realizar:
  - partiendo de una red And-or-not de dos niveles
  - sustituyendo todas las puertas por Nand
  - añadiendo inversores en los términos formados por un único literal
- Ejemplo: sintetizar Z mediante una red de puertas Nand:

$$Z = x_2 + x_1 \cdot \bar{x}_0 + \bar{x}_3 \cdot \bar{x}_2 \cdot \bar{x}_0 + \bar{x}_2 \cdot \bar{x}_1 \cdot x_0$$





# SÍNTESIS CON REDES DE PUERTAS NAND

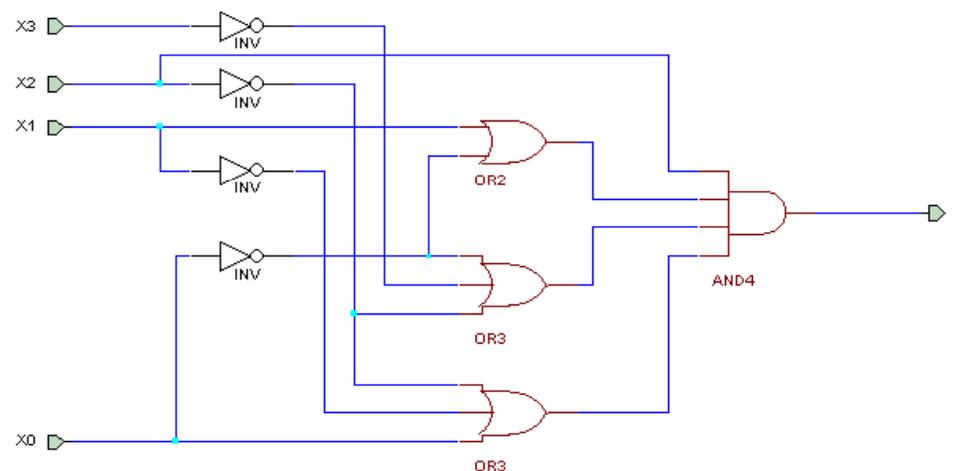




# SÍNTESIS CON REDES DE PUERTAS NOR

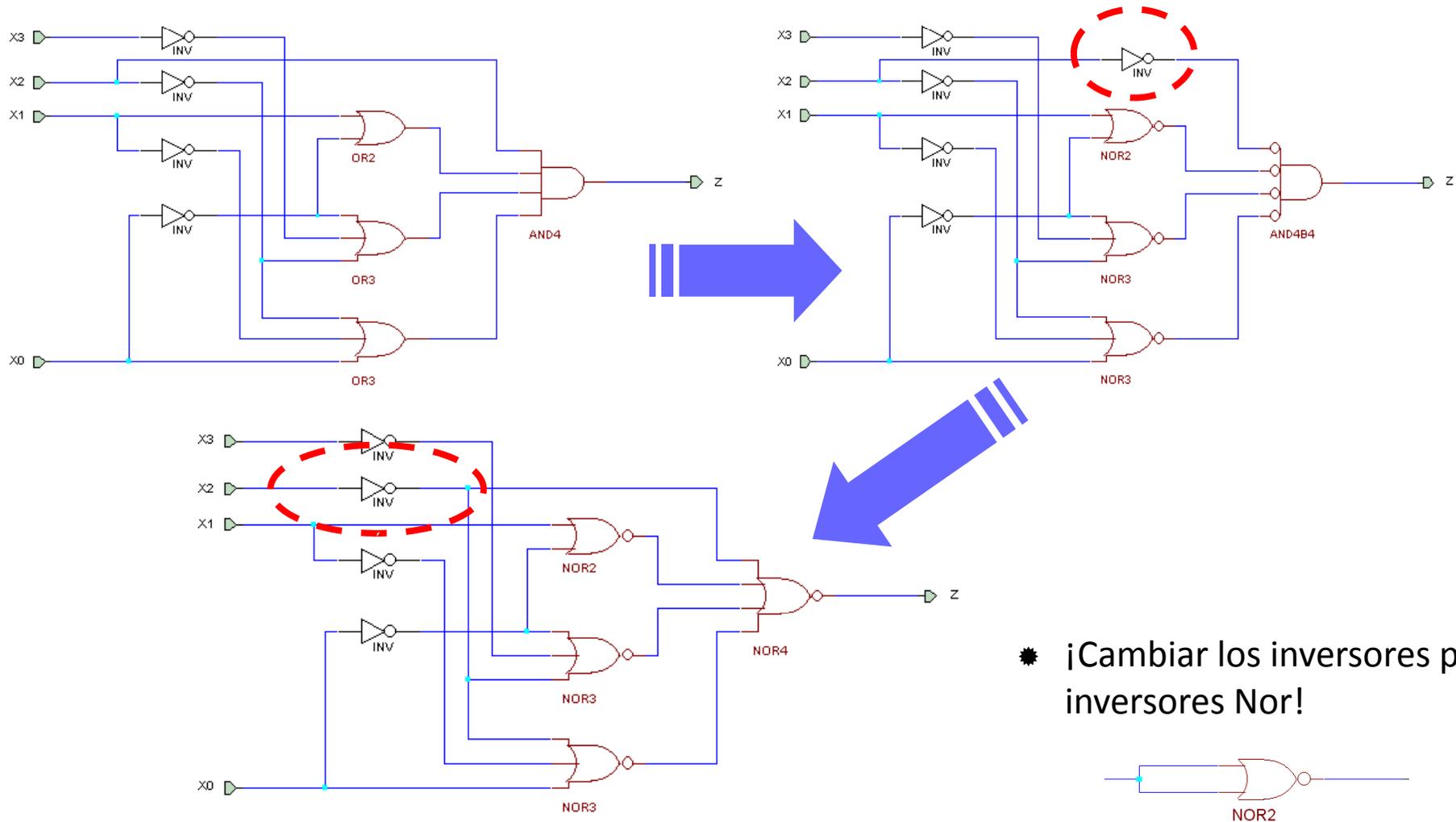
- La síntesis de un sistema combinacional en forma de producto de sumas mediante sólo puertas Nor se puede realizar:
  - partiendo de una red Or-And-Not de dos niveles.
  - sustituyendo todas las puertas por Nor.
  - añadiendo inversores en los términos formados por un único literal.
- Ejemplo: sintetizar Z mediante una red de puertas Nor.

$$Z = x_2 \cdot (x_1 + \bar{x}_0) \cdot (\bar{x}_3 + \bar{x}_2 \cdot \bar{x}_0) \cdot (\bar{x}_2 + \bar{x}_1 + x_0)$$





# SÍNTESIS CON REDES DE PUERTAS NOR



✳ ¡Cambiar los inversores por inversores Nor!



## SÍNTESIS CON REDES DE PUERTAS CON LÓGICA MIXTA

1. Dibujo del circuito básico (puertas de suma, producto o XOR sin indicar la lógica de las entradas ni de las salidas). Donde haya que realizar una negación lógica se dibujará una línea que cruce el hilo.
2. Selección del tipo de puertas según el conjunto universal empleado.
3. Asignación de la lógica de las entradas según las restricciones.
4. Verificación de la función implementada y adición de inversores en caso necesario.
5. Verificación de todas las restricciones impuestas. Si es preciso, rediseñar el circuito volviendo atrás a cualquiera de los pasos previos con objeto de obtener otra solución mejor (más sencilla, más rápida, más barata o, en definitiva, más de acuerdo con las restricciones).

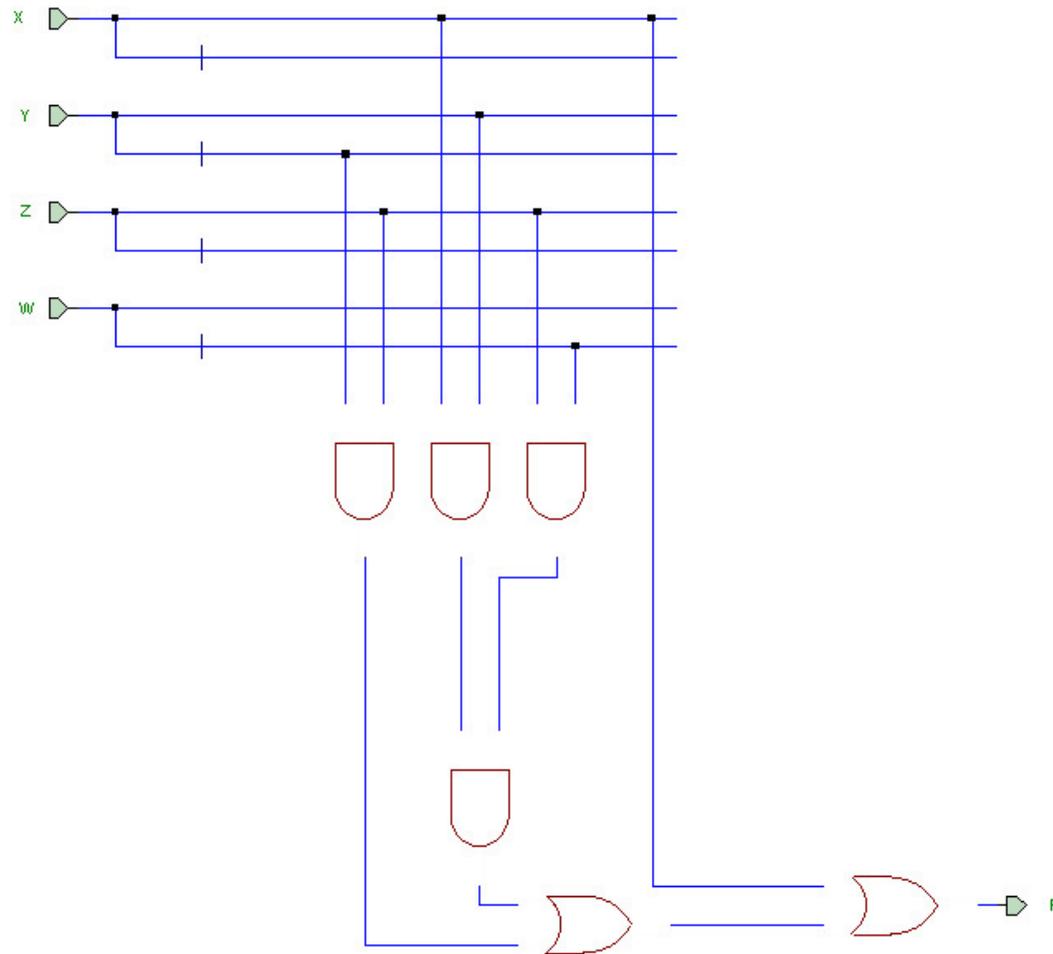
**Los pasos 2 y 3 son intercambiables según las restricciones**



# SÍNTESIS CON REDES DE PUERTAS CON LÓGICA MIXTA

- Sintetizar usando puertas NAND de 2 entradas: circuito básico

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





# SÍNTESIS CON REDES DE PUERTAS CON LÓGICA MIXTA

- Sintetizar usando puertas NAND de 2 entradas:

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

