



Tema 3:  
**Implementación de sistemas  
combinacionales**

Fundamentos de computadores I

**José Manuel Mendías Cuadros**  
*Dpto. Arquitectura de Computadores y Automática  
Universidad Complutense de Madrid*



# Contenidos



- ✓ Puertas lógicas.
- ✓ Conjuntos universales de puertas.
- ✓ Síntesis con puertas AND-OR.
- ✓ Síntesis con puertas NAND.
- ✓ Análisis de redes de puertas AND-OR.
- ✓ Análisis de redes de puertas NAND.
  
- ✓ Apéndice tecnológico

Transparencias basadas en los libros:

- R. Hermida, F. Sánchez y E. del Corral. *Fundamentos de computadores*.
- D. Gajsky. *Principios de diseño digital*.

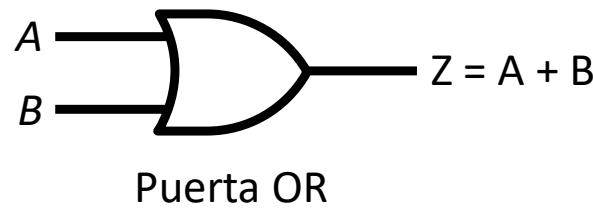


# Puertas lógicas

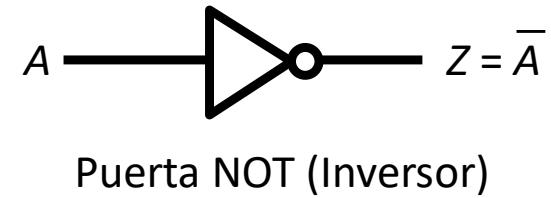
- Dispositivo que realiza **físicamente** una función de conmutación **sencilla**.



Puerta AND



Puerta OR



Puerta NOT (Inversor)

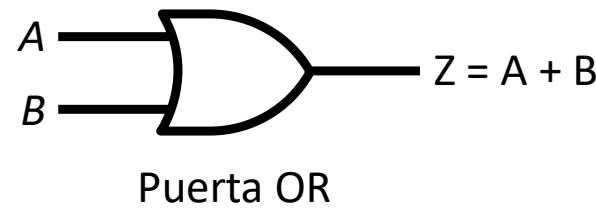
# Puertas lógicas



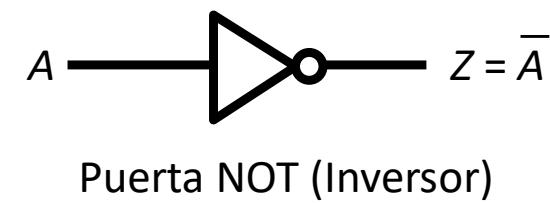
- Dispositivo que realiza **físicamente** una función de conmutación **sencilla**.



Puerta AND



Puerta OR



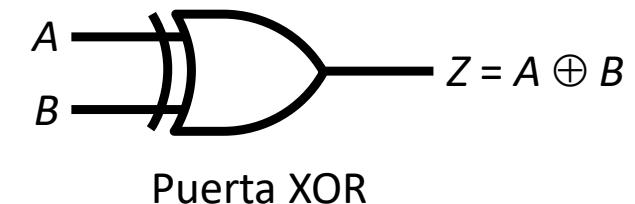
Puerta NOT (Inversor)



Puerta NAND



Puerta NOR



Puerta XOR



Puerta XNOR

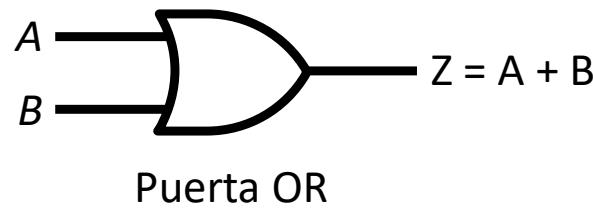
# Puertas lógicas



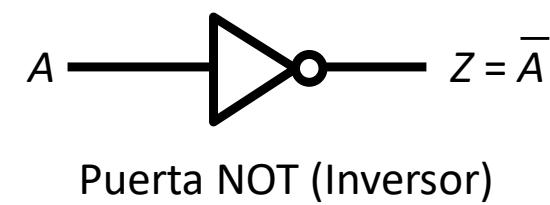
- Dispositivo que realiza **físicamente** una función de conmutación **sencilla**.



Puerta AND



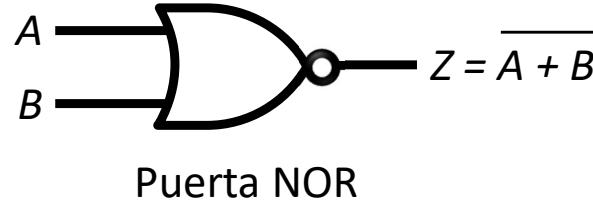
Puerta OR



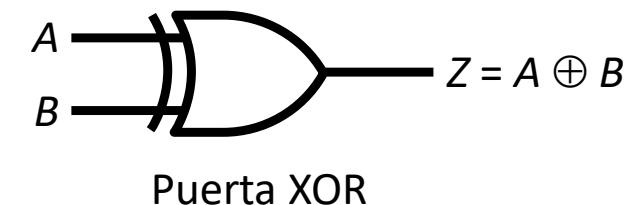
Puerta NOT (Inversor)



Puerta NAND



Puerta NOR



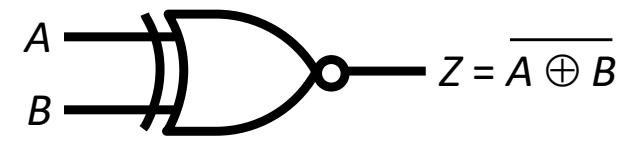
Puerta XOR

Puerta NAND  
(símbolo alternativo)

$$\overline{A \cdot B} = \overline{A} + \overline{B}$$

Puerta NOR  
(símbolo alternativo)

$$\overline{A + B} = \overline{A} \cdot \overline{B}$$

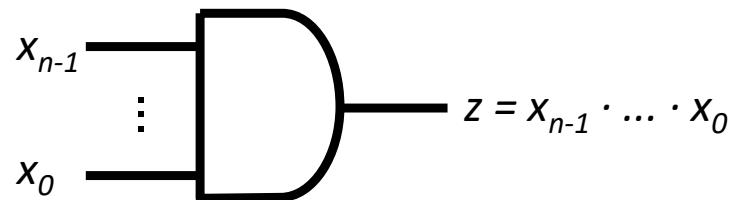


Puerta XNOR

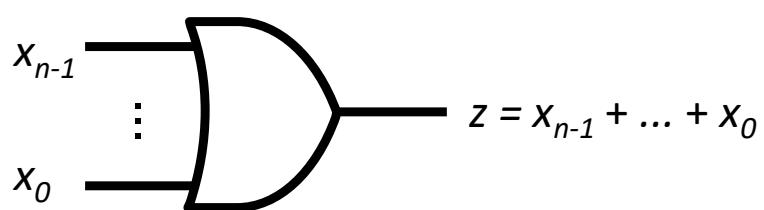
# Puertas lógicas



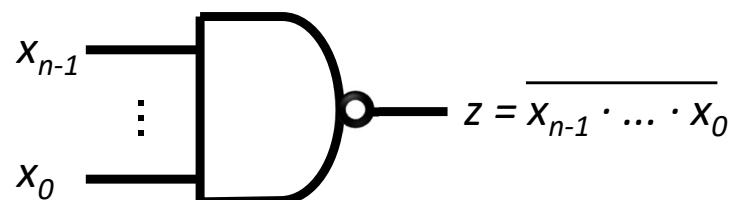
- Existen puertas con mayor número de entradas:



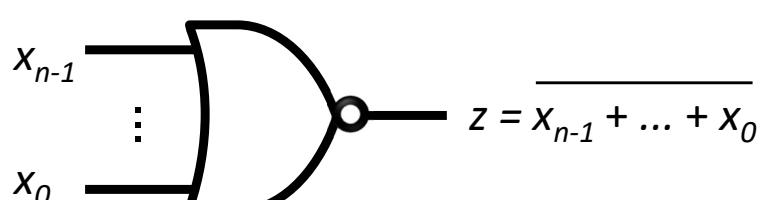
Puerta AND de  $n$  entradas



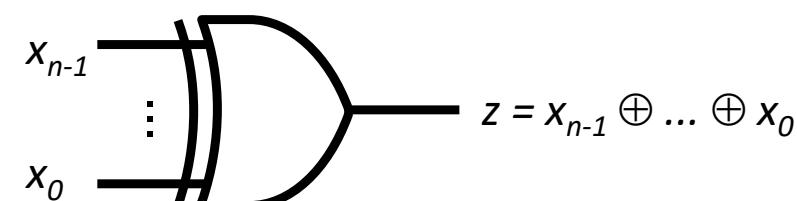
Puerta OR de  $n$  entradas



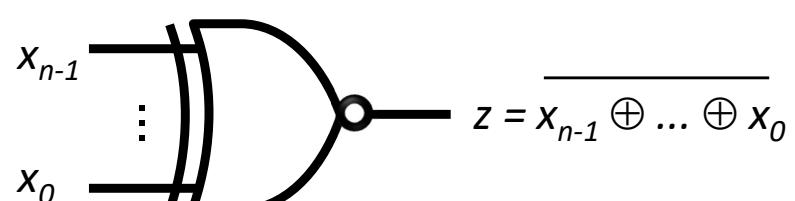
Puerta NAND de  $n$  entradas



Puerta NOR de  $n$  entradas



Puerta XOR de  $n$  entradas  
( $z=1$  si el número de  $x_i=1$  es impar)

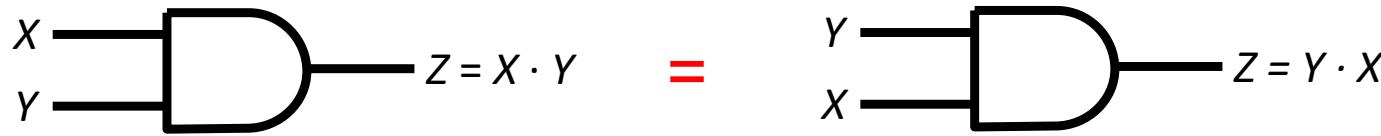


Puerta XNOR de  $n$  entradas  
( $z=1$  si el número de  $x_i=1$  es par)

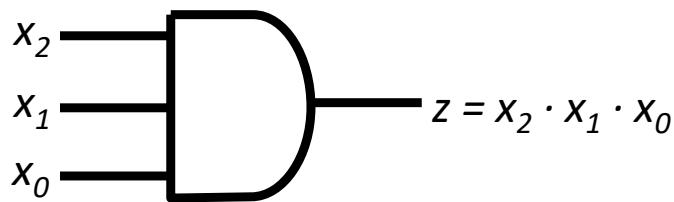
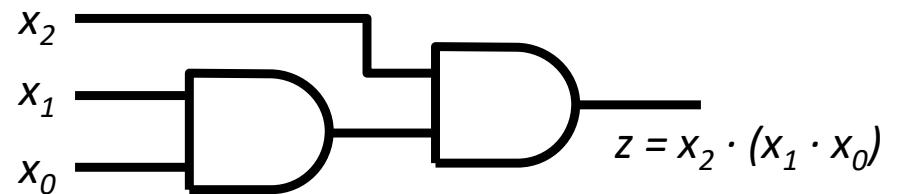
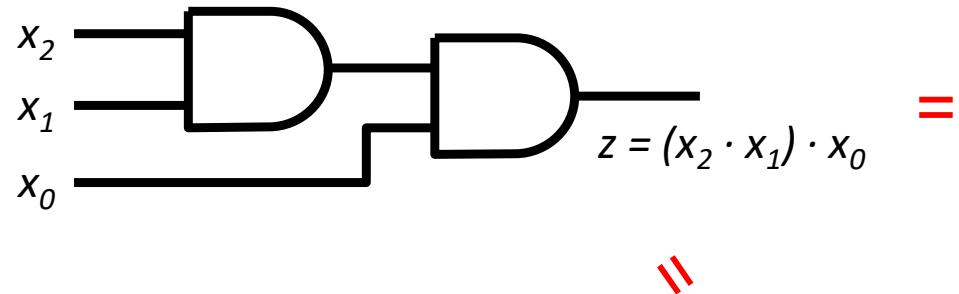


# Puertas lógicas

- Todas ellas son **comutativas**:



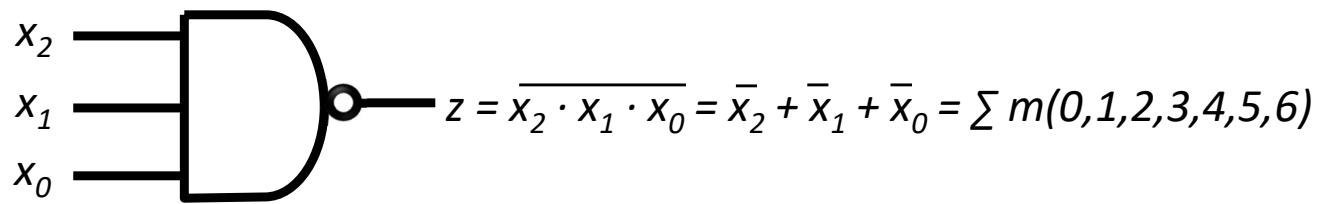
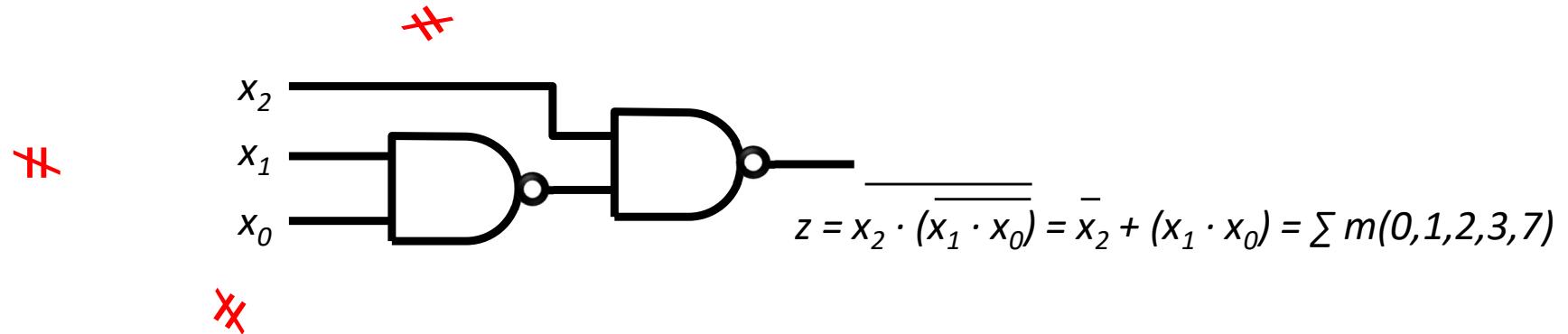
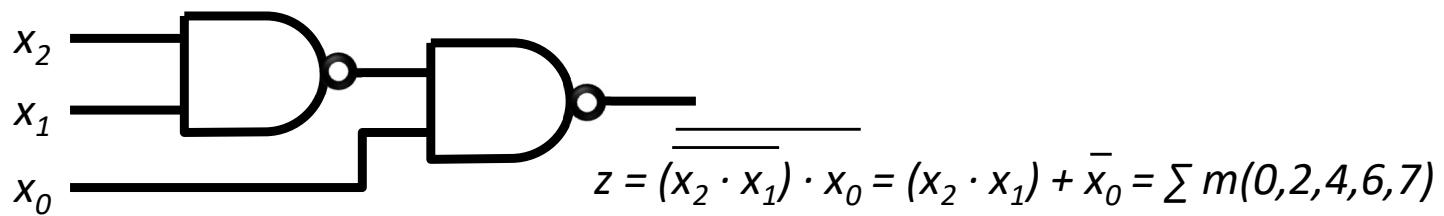
- AND, OR, XOR y XNOR son **asociativas**:





# Puertas lógicas

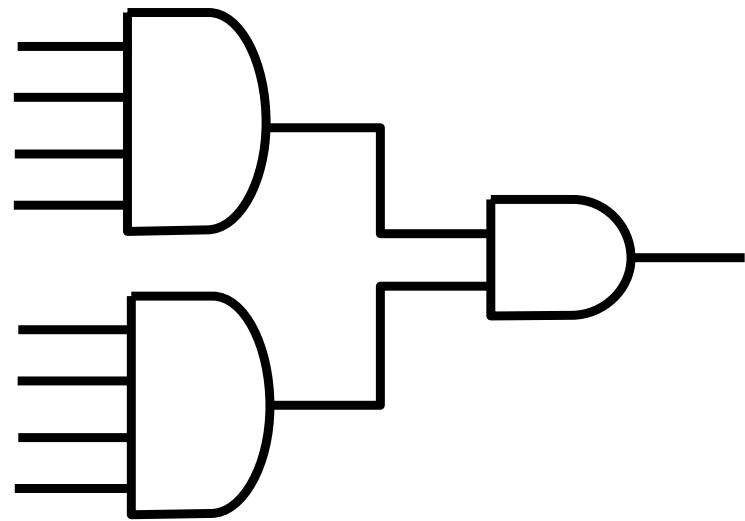
- Pero NAND y NOR no son asociativas.





# Puertas lógicas

- En la práctica no es común encontrar puertas con un número elevado de entradas.
  - Solución: implementaciones en árbol.



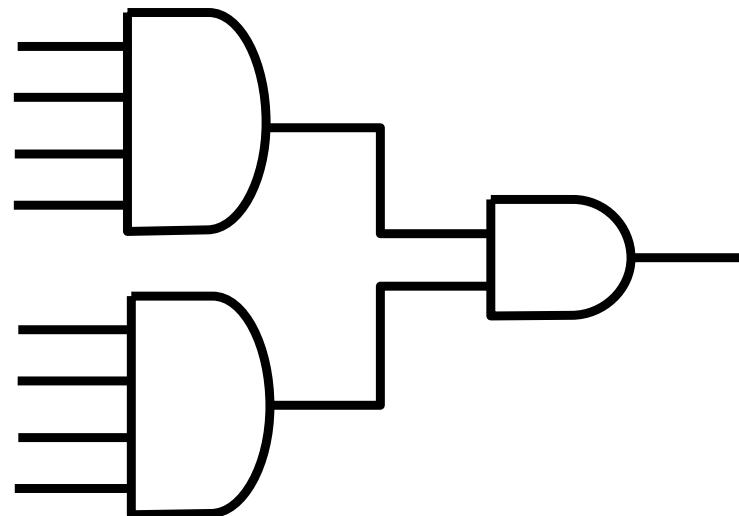
**Implementación en árbol**

Puerta AND de 8 entradas

# Puertas lógicas

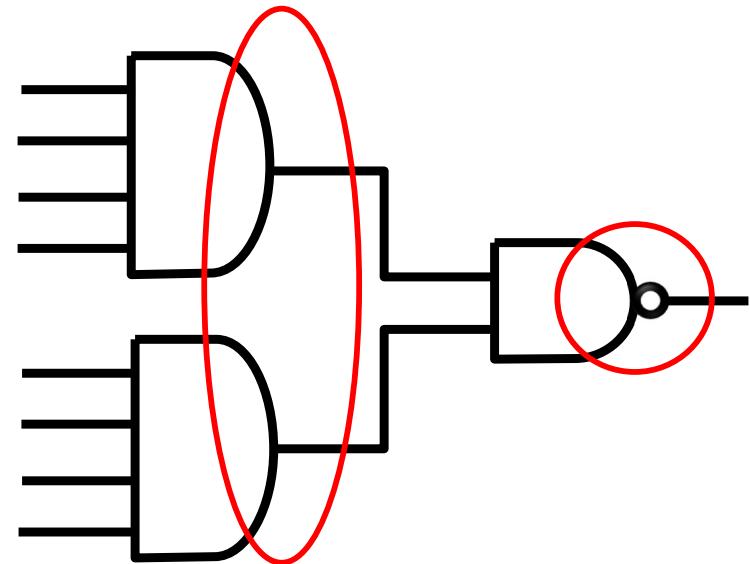


- En la práctica no es común encontrar puertas con un número elevado de entradas.
  - Solución: implementaciones en árbol.



Implementación en árbol

Puerta AND de 8 entradas



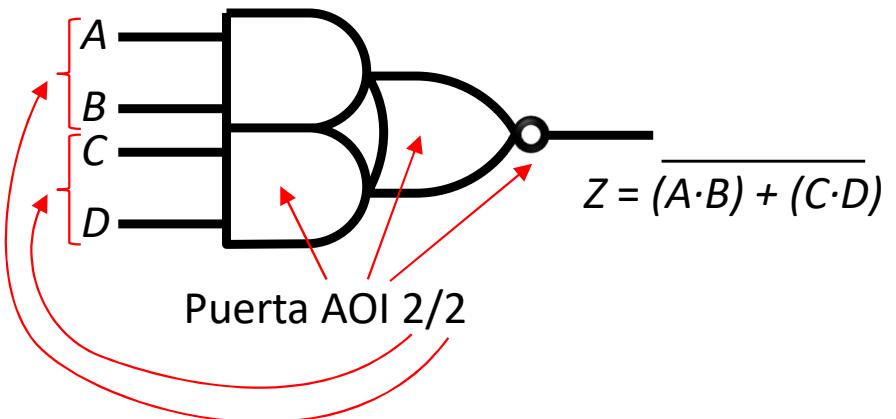
Implementación en árbol

Puerta NAND de 8 entradas



# Puertas lógicas

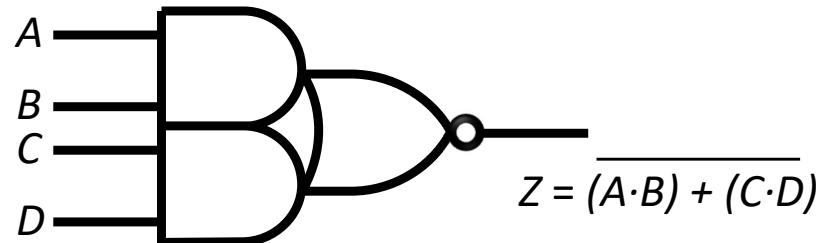
- Existen puertas compuestas:



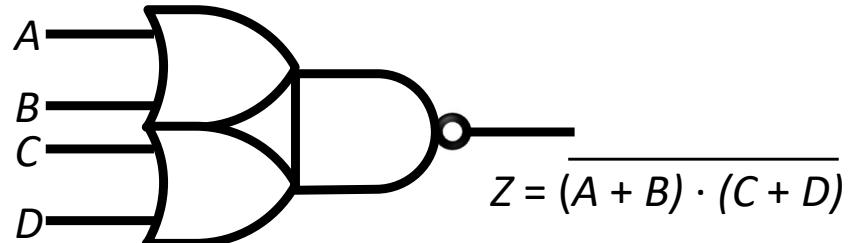
# Puertas lógicas



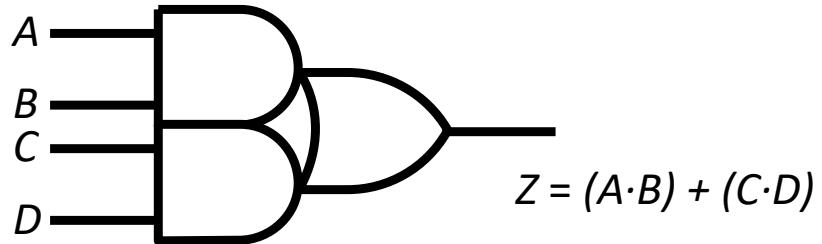
- Existen puertas compuestas:



Puerta AOI 2/2



Puerta OAI 2/2

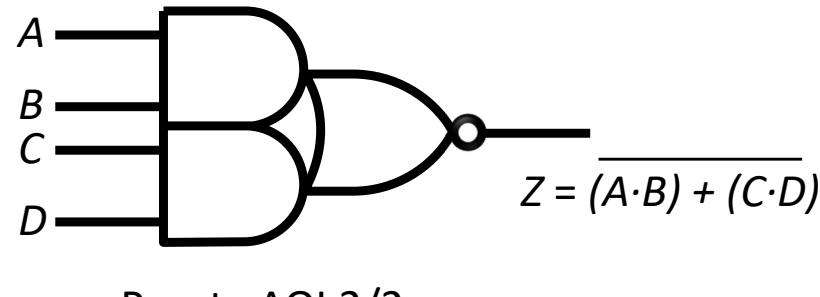


Puerta AO 2/2

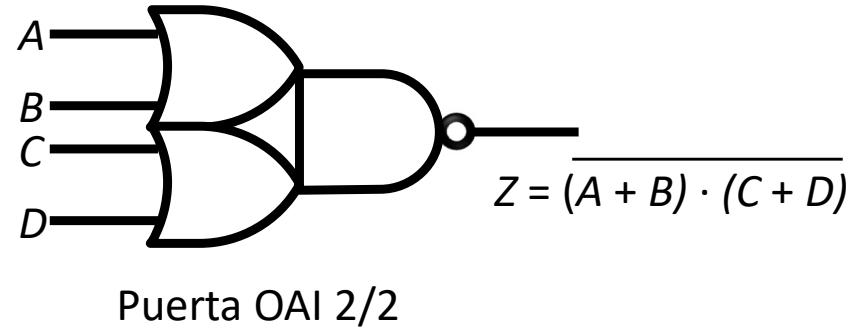
# Puertas lógicas



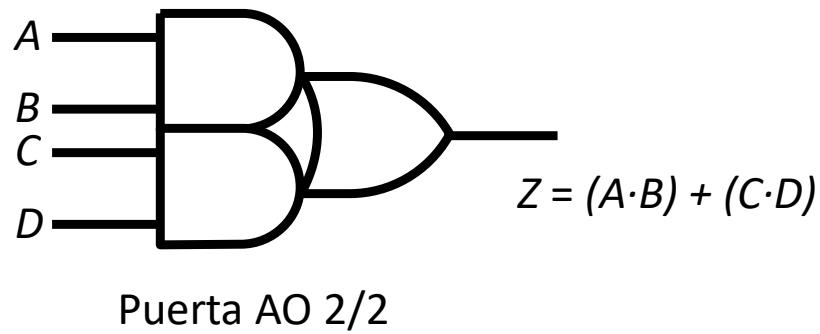
- Existen puertas compuestas:



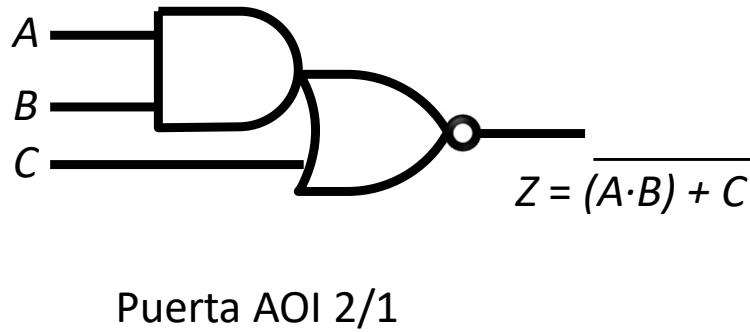
Puerta AOI 2/2



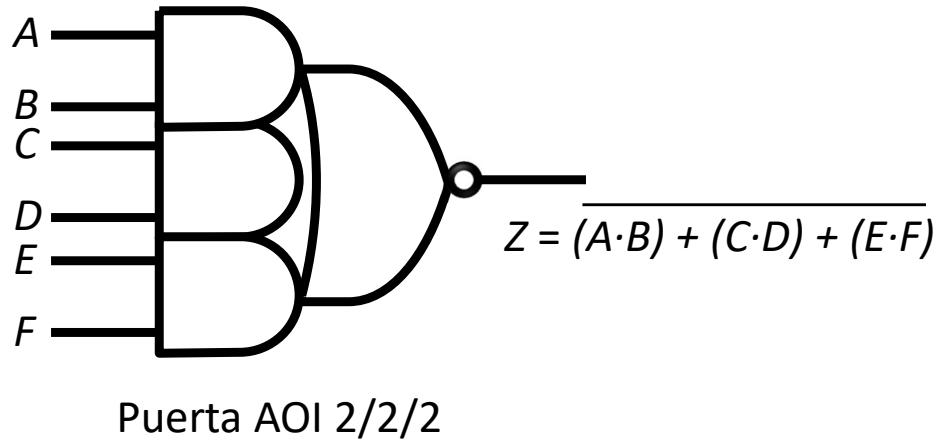
Puerta OAI 2/2



Puerta AO 2/2



Puerta AOI 2/1



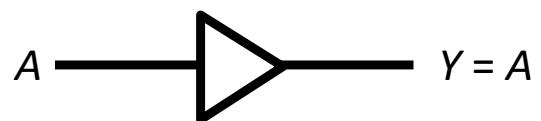
Puerta AOI 2/2/2

y algunas más...

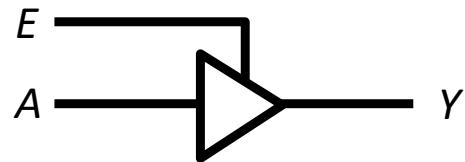
# Buffers



- Existen otros dispositivos sin funcionalidad lógica:
  - Buffer no inversor: permite compensar la atenuación eléctrica de una señal.
  - Buffer triestado: permite desconectar selectivamente una señal.



Buffer no inversor



Buffer triestado

A	Y
0	0
1	1

E	A	Y
0	0	Z
0	1	Z
1	0	0
1	1	1

Alta impedancia  
(desconecta  $Y$  de  $A$ )

# Algunas definiciones



- **Módulo:** dispositivo que realiza físicamente una función conocida de cualquier complejidad.
  - Los hay combinacionales y secuenciales
- **Puerto:** cada una de las líneas de entrada/salida que comunica un módulo con el exterior.
- **Interconexión:** unión de 2 o más puertos entre sí.
- **Red:** colección de módulos interconectados de manera que **toda entrada solo está conectada a una salida** (una salida sí puede estar conectada a varias entradas).
  - Las interconexiones 1:1 y 1:n están permitidas.
  - Las interconexiones n:1 están prohibidas (a menos que se utilicen buffers triestado).

# Algunas definiciones

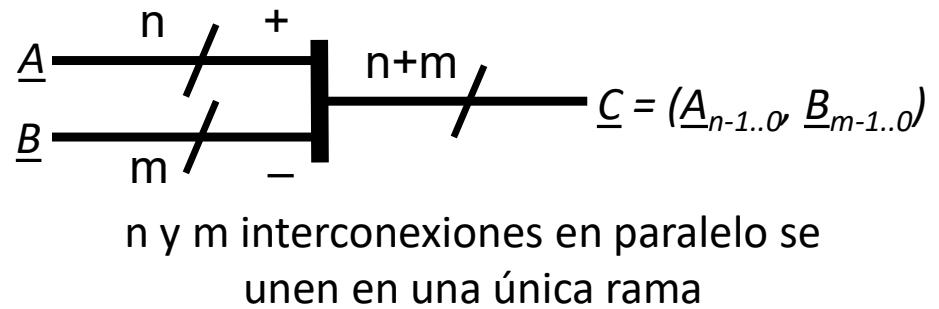
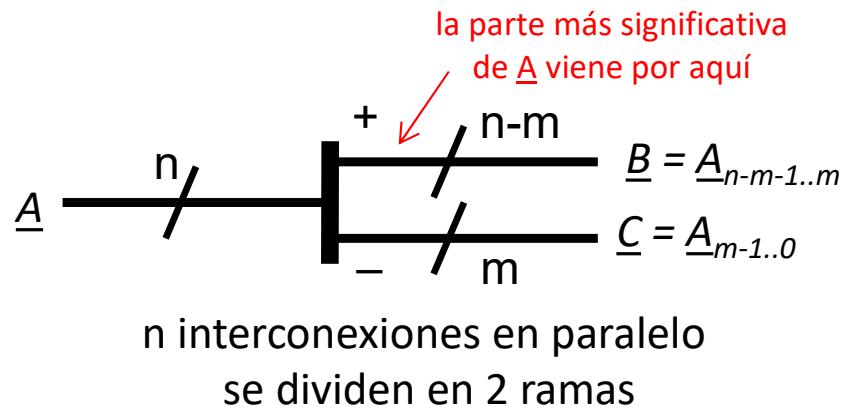
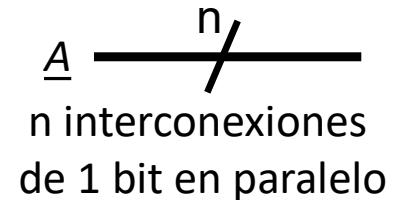
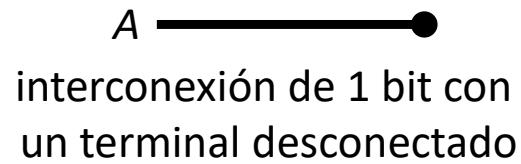


- **Red combinacional:** red de módulos combinacionales en las que **no existen realimentaciones**.
  - no hay ningún camino dentro de la red que pase 2 veces por el mismo punto.
  - toda red combinacional es un módulo combinacional.
- **Nivel de una red:** número máximo de módulos que atraviesa cualquier camino que conecte una entrada con una salida
  - cuando la red es de puertas no se suelen contar los inversores.

# Interconexiones



- Al dibujar el **esquema** de un circuito usaremos alguna notación adicional para las interconexiones:





# Conjunto universal

- Se dice que un **conjunto de módulos combinacionales es universal** si permite implementar cualquier FC
  - Un conjunto lo es, si con sus módulos pueden implementarse todos los operadores del álgebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de puertas :
  - { AND, NOT }

$$a + b = \overline{(\bar{a} + \bar{b})} = \bar{a} \cdot \bar{b}$$



# Conjunto universal

- Se dice que un **conjunto de módulos combinacionales es universal** si permite implementar cualquier FC
  - Un conjunto lo es, si con sus módulos pueden implementarse todos los operadores del álgebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de puertas :

- { AND, NOT }

$$a + b = \overline{(a + b)} = \overline{\bar{a} \cdot \bar{b}}$$

- { NAND }

$$\bar{a} = \overline{(a \cdot a)} = a \uparrow a$$

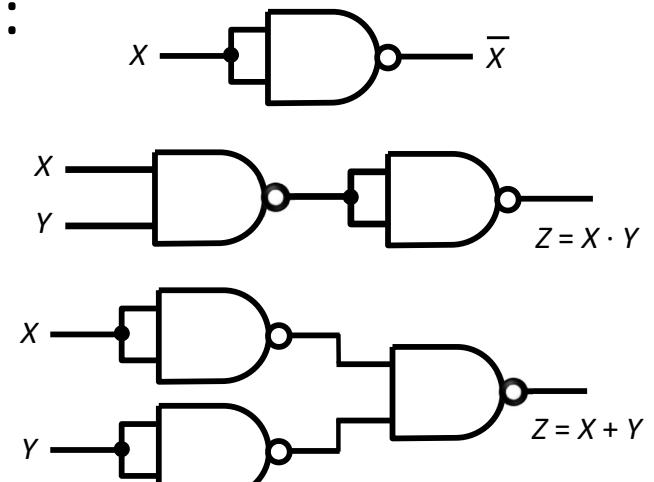
$$a \cdot b = \overline{\overline{(a \cdot b)}} = \overline{a \uparrow b} = (a \uparrow b) \uparrow (a \uparrow b)$$

$$a + b = \overline{\overline{(a + b)}} = \overline{(\bar{a} \cdot \bar{b})} = \bar{a} \uparrow \bar{b} = (a \uparrow a) \uparrow (b \uparrow b)$$



# Conjunto universal

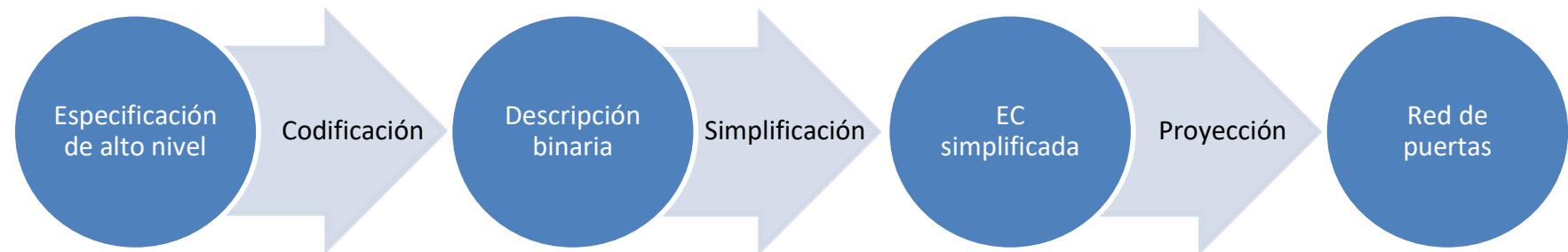
- Se dice que un **conjunto** de módulos combinacionales es **universal** si permite implementar cualquier FC
  - Un conjunto lo es, si con sus módulos pueden implementarse todos los operadores del álgebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de puertas :
  - { AND, NOT }  
$$a + b = \overline{(\overline{a} \cdot \overline{b})} = \overline{\overline{a} \cdot \overline{b}}$$
  - { NAND }  
$$\overline{a} = \overline{(a \cdot a)} = a \uparrow a$$
$$a \cdot b = \overline{(\overline{a} \cdot \overline{b})} = \overline{\overline{a} \uparrow \overline{b}} = (a \uparrow b) \uparrow (a \uparrow b)$$
$$a + b = \overline{(\overline{a} + \overline{b})} = \overline{(\overline{a} \cdot \overline{b})} = \overline{\overline{a} \uparrow \overline{b}} = (a \uparrow a) \uparrow (b \uparrow b)$$
  - { OR, NOT }, { NOR }, { XOR, AND } ...



# Síntesis de redes de puertas



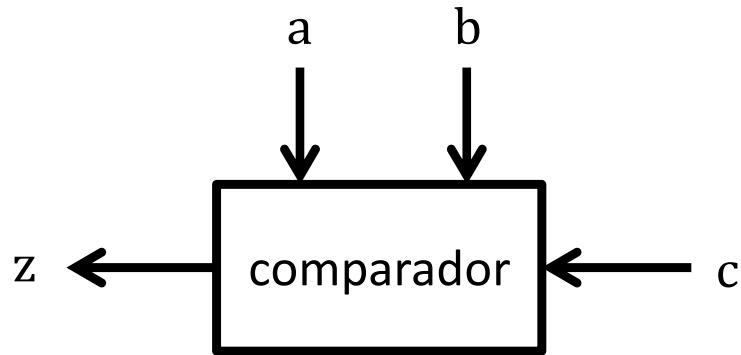
- Dada una especificación de una conducta combinacional implementarla usando puertas.



- **Implementaciones a 2 niveles**
  - **Implementación canónica:** implementa la SPC con 2 niveles AND-OR.
  - **Implementación mínima:** implementa una  $EC_{min}$  con 2 niveles AND-OR.
    - La red resultante tiene un número mínimo de puertas y éstas tienen un número mínimo de entradas.
- **Implementaciones multnivel**
  - Tienen un número arbitrario de niveles y se reutilizan cálculos intermedios.
  - Para obtenerlas, se parte de un conjunto de SP y se **factorizan** heurísticamente .



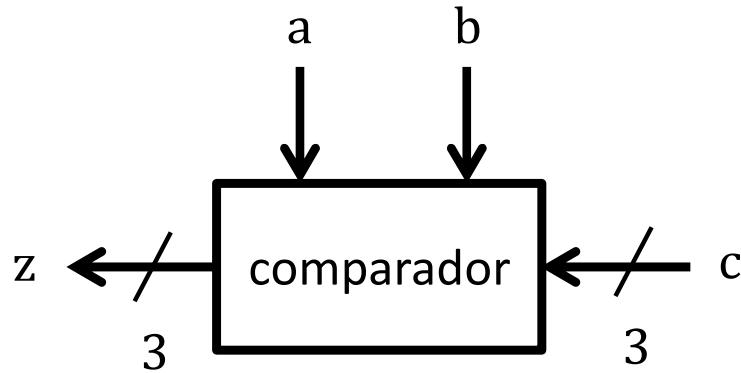
# Síntesis de redes AND-OR



$$a, b \in \{ 0, 1 \} \quad c, z \in \{ aM, IG, bM \}$$

$$z = \begin{cases} aM & \text{si } (a>b) \text{ o } (a=b \text{ y } c=aM) \\ IG & \text{si } (a=b) \text{ y } (c=IG) \\ bM & \text{si } (a<b) \text{ o } (a=b \text{ y } c=bM) \end{cases}$$

Codificación:  $aM = (100)$ ,  $IG = (010)$ ,  $bM = (001)$



$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR

Implementación  
a 2 niveles

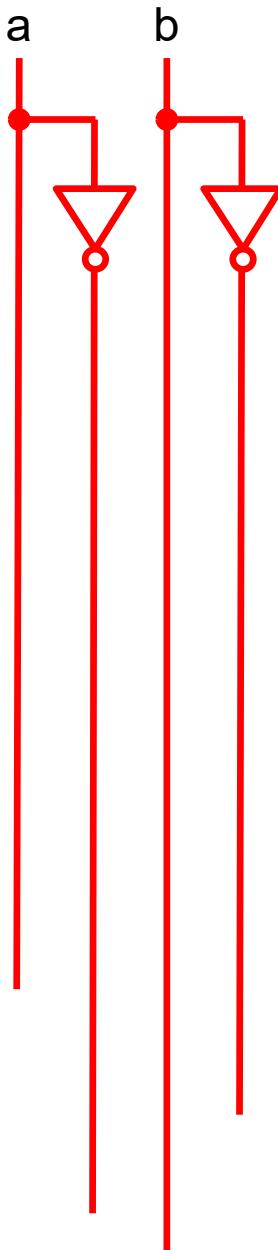
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

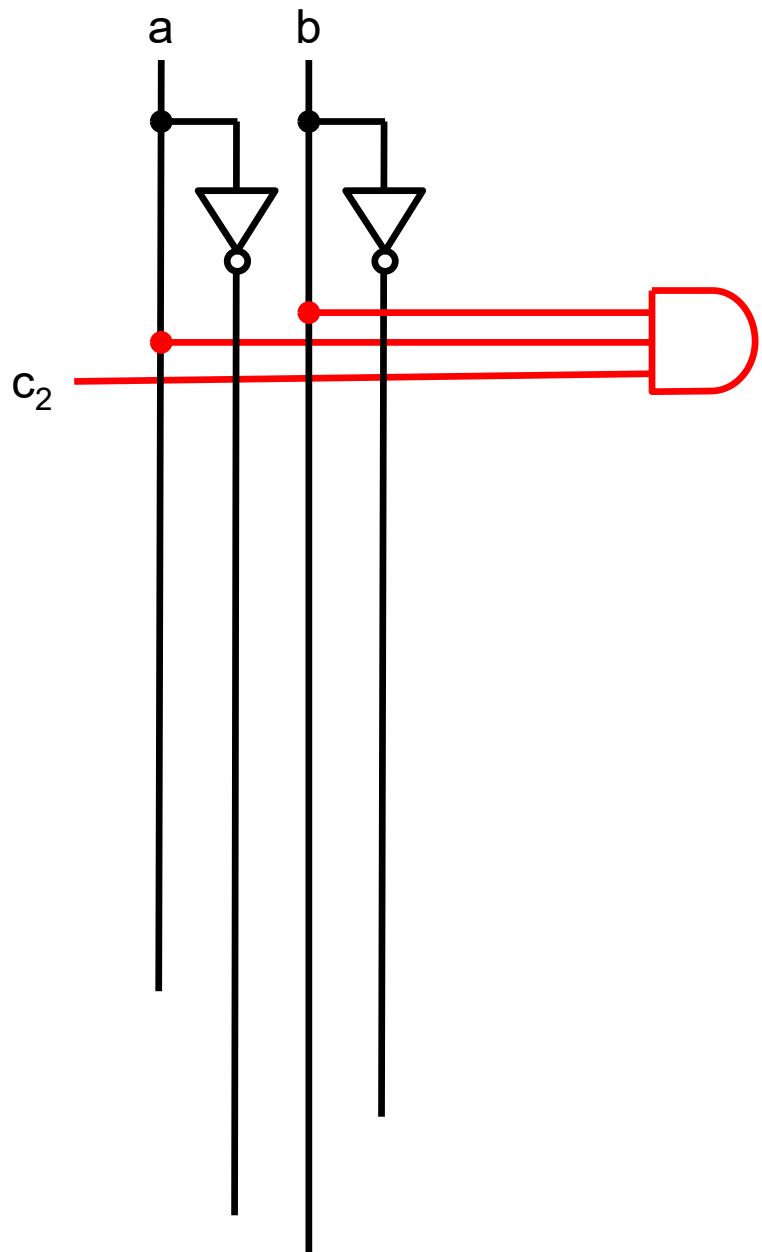
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

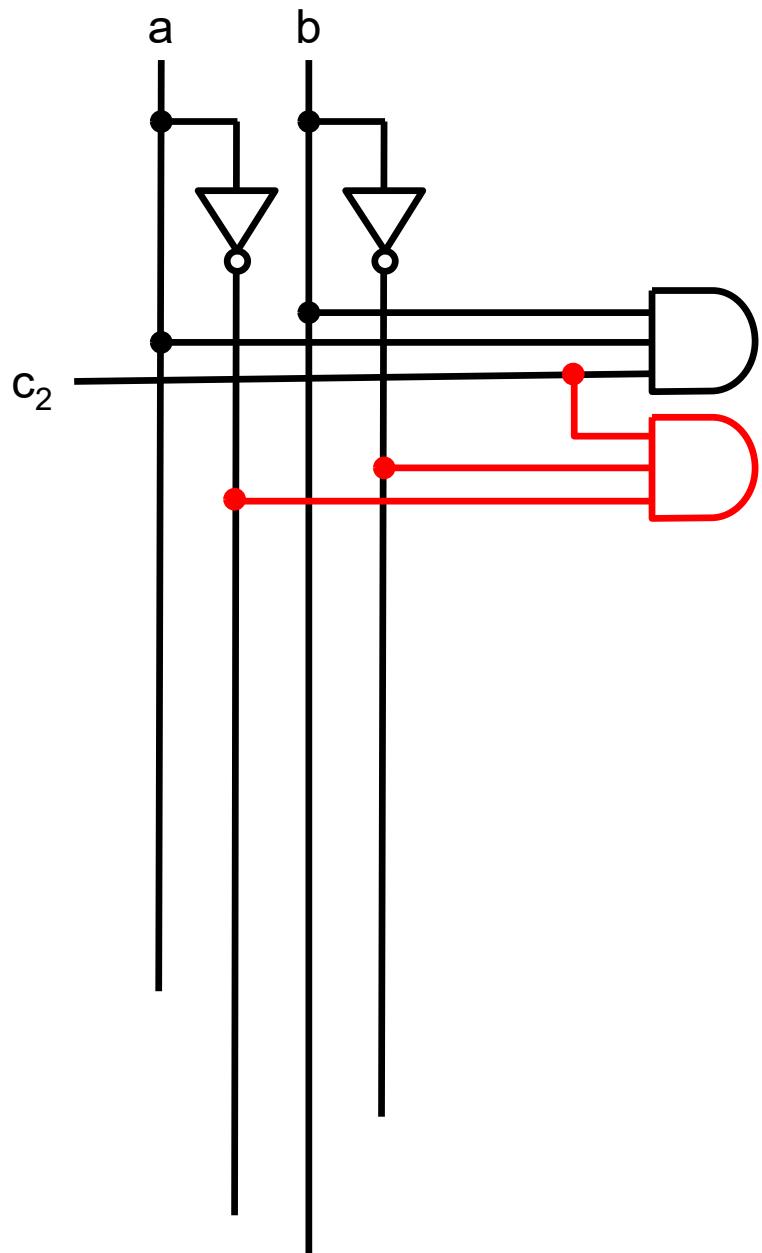
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

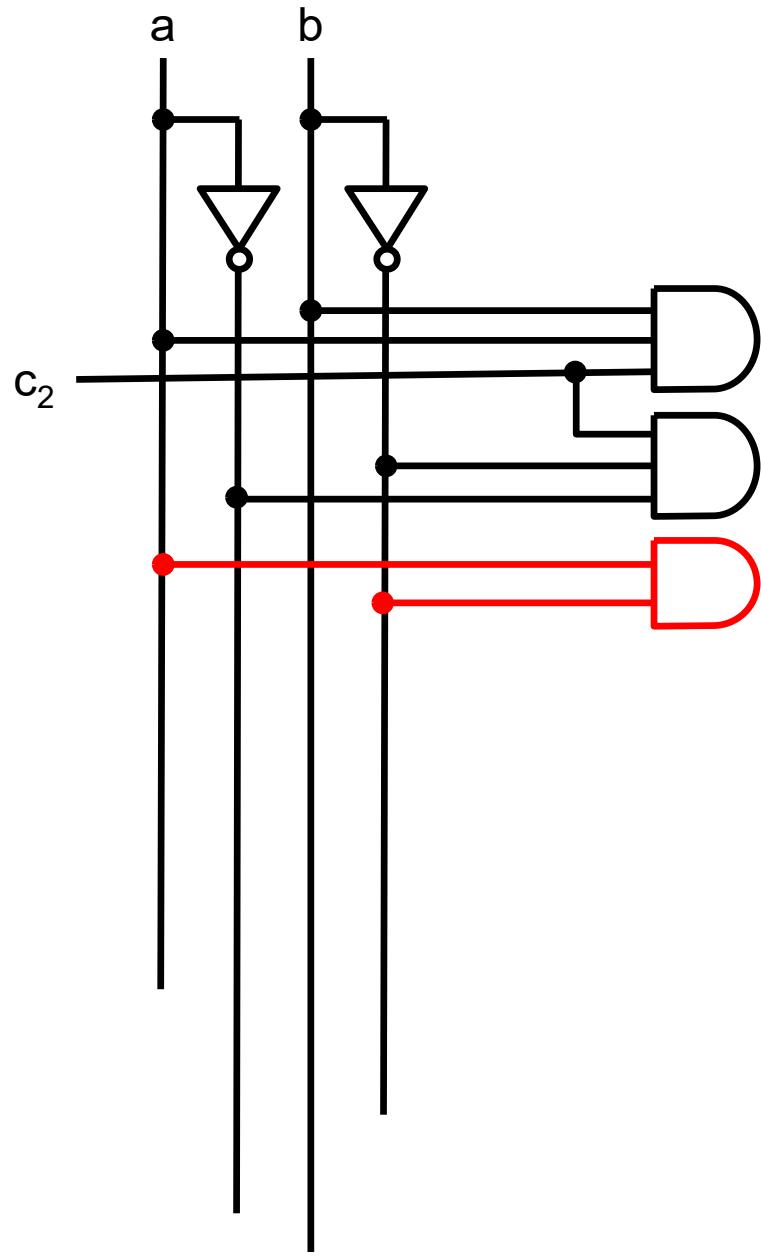
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

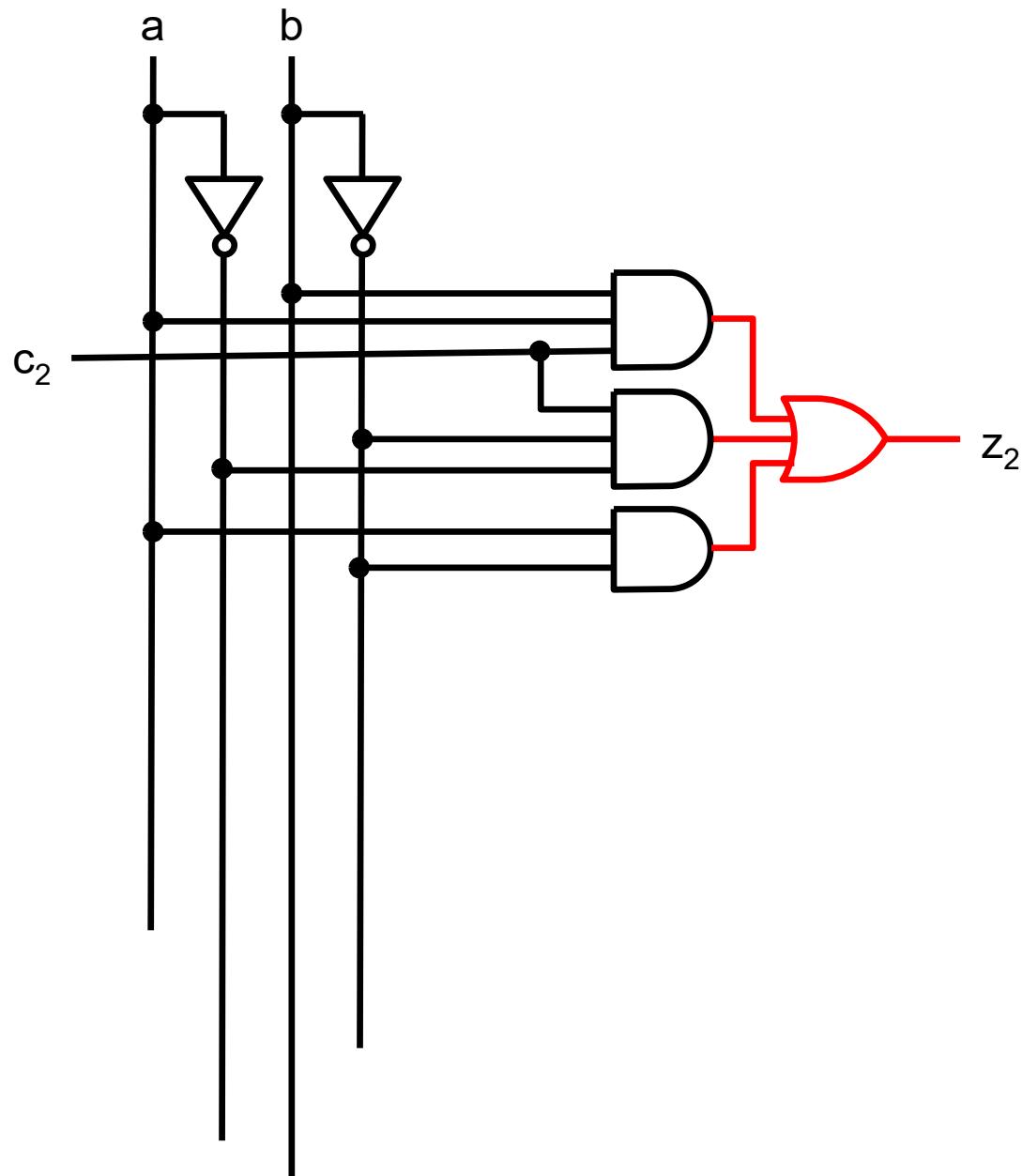
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

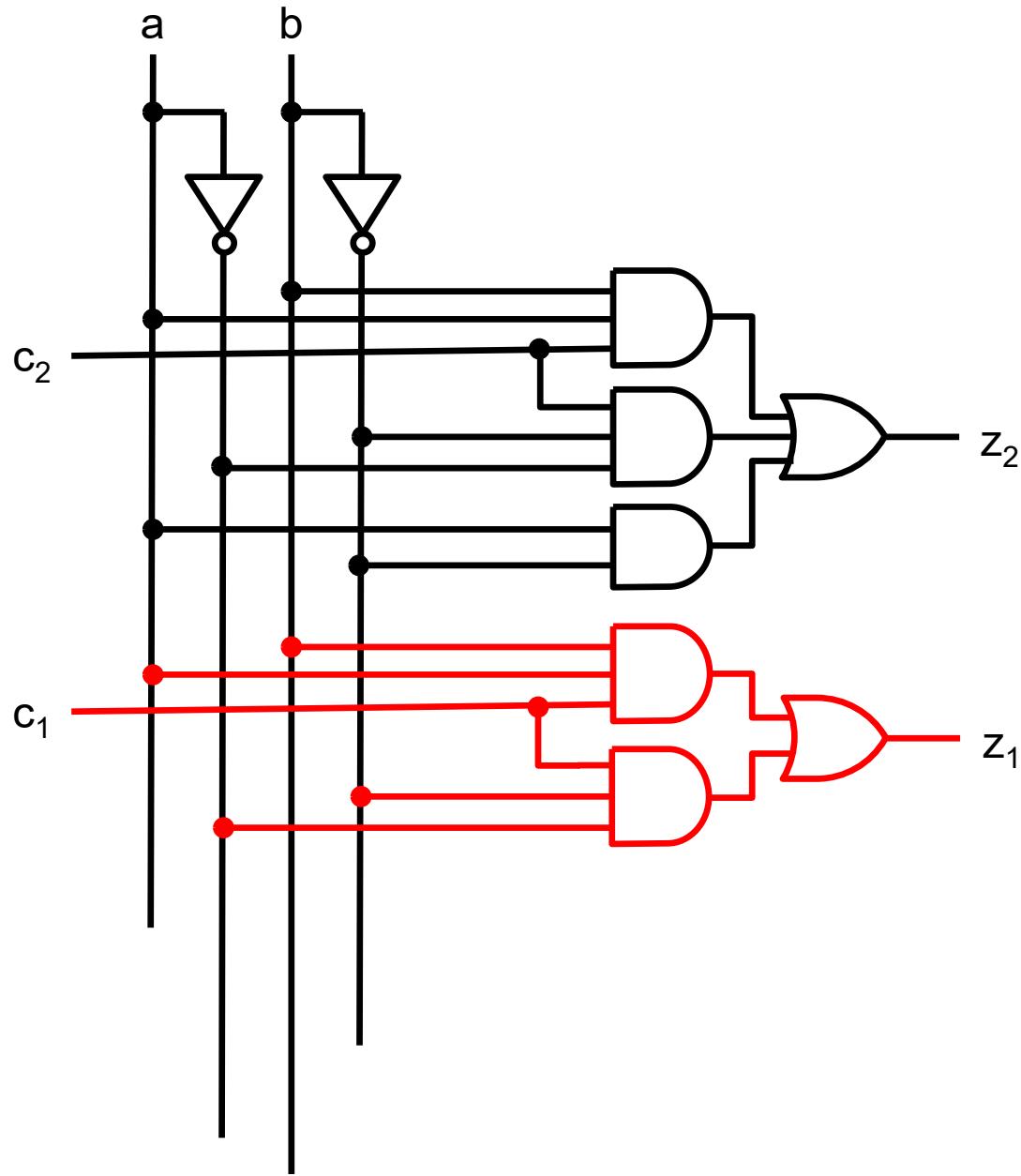
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

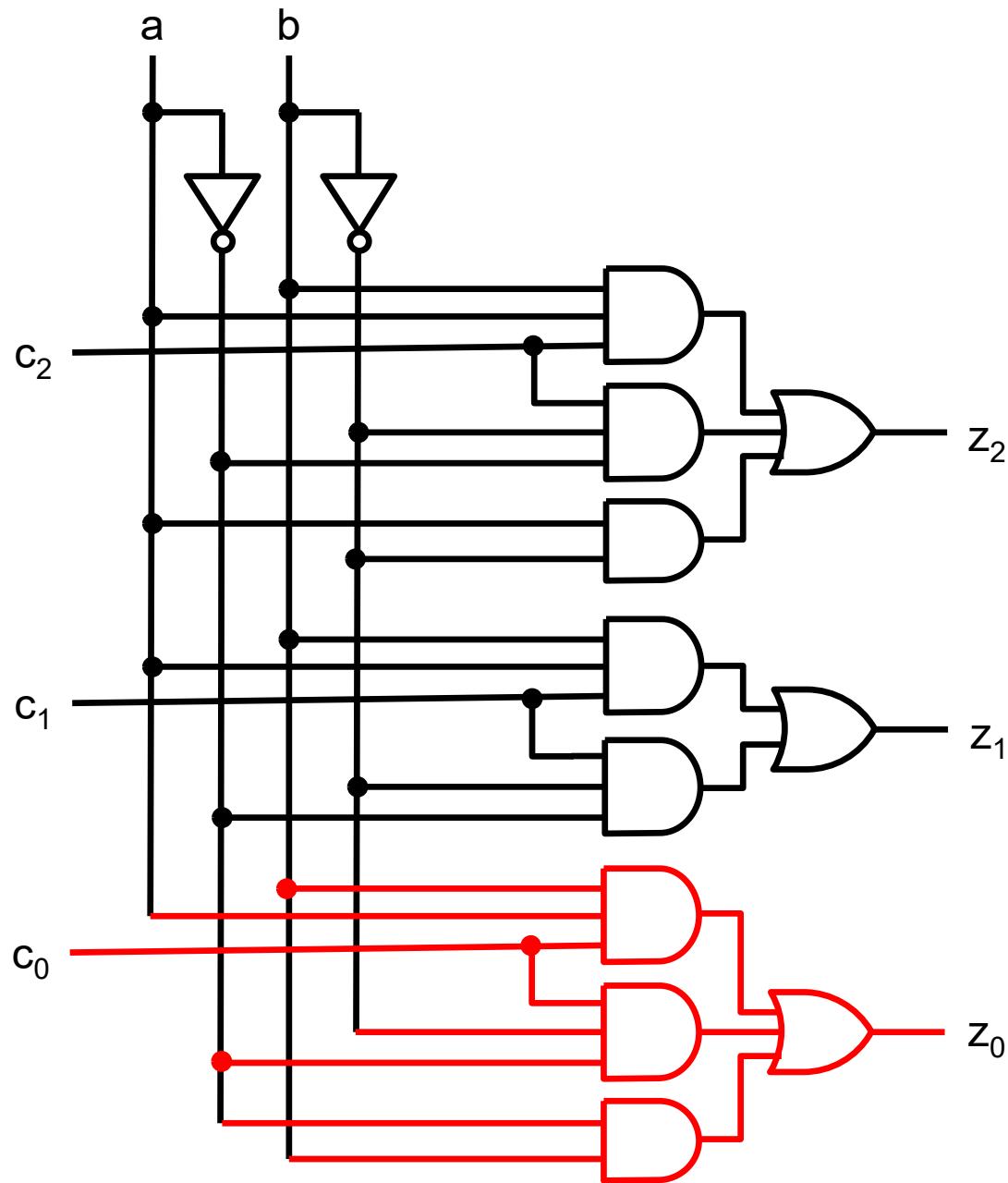
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

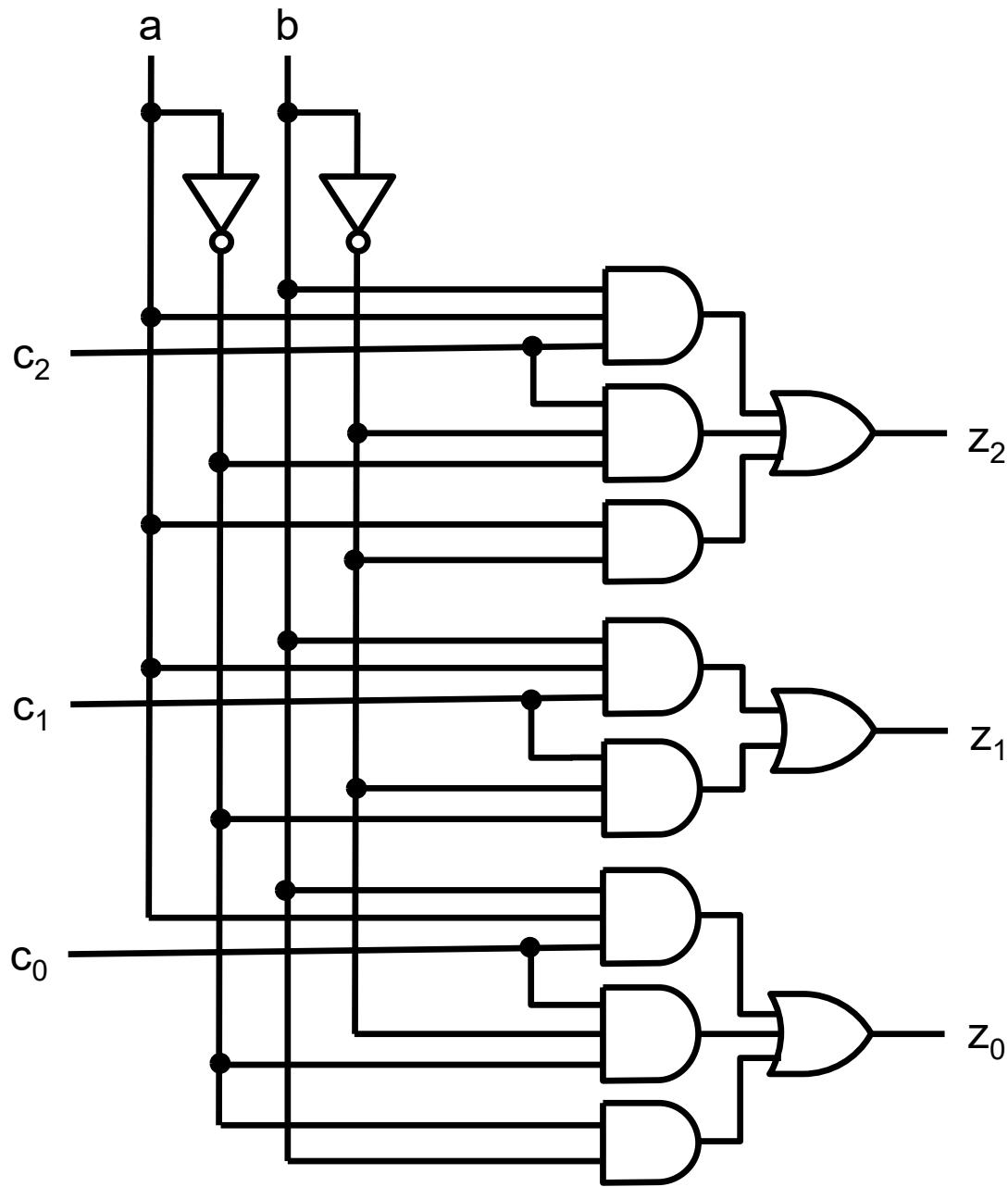
$$z_2 = ab + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$



# Síntesis de redes AND-OR



Implementación  
a 2 niveles

$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

$$z_1 = \bar{a}\bar{b}c_1 + abc_1$$

$$z_0 = \bar{a}b + \bar{a}\bar{b}c_0 + abc_0$$

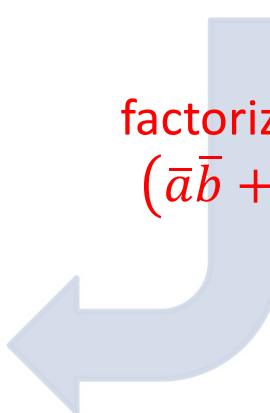
## Implementación multinivel

# Síntesis de redes AND-OR



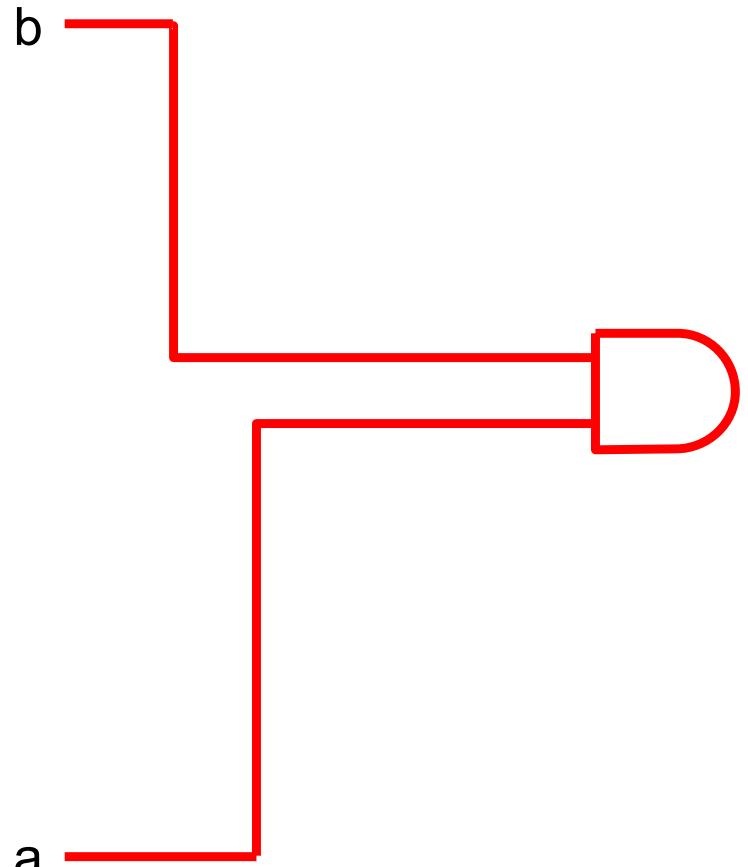
$$\begin{aligned}z_2 &= a\bar{b} + \bar{a}\bar{b}c_2 + abc_2 \\z_1 &= \bar{a}\bar{b}c_1 + abc_1 \\z_0 &= \bar{a}b + \bar{a}\bar{b}c_0 + abc_0\end{aligned}$$

factorizando  
 $(\bar{a}\bar{b} + ab)$


$$\begin{aligned}z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

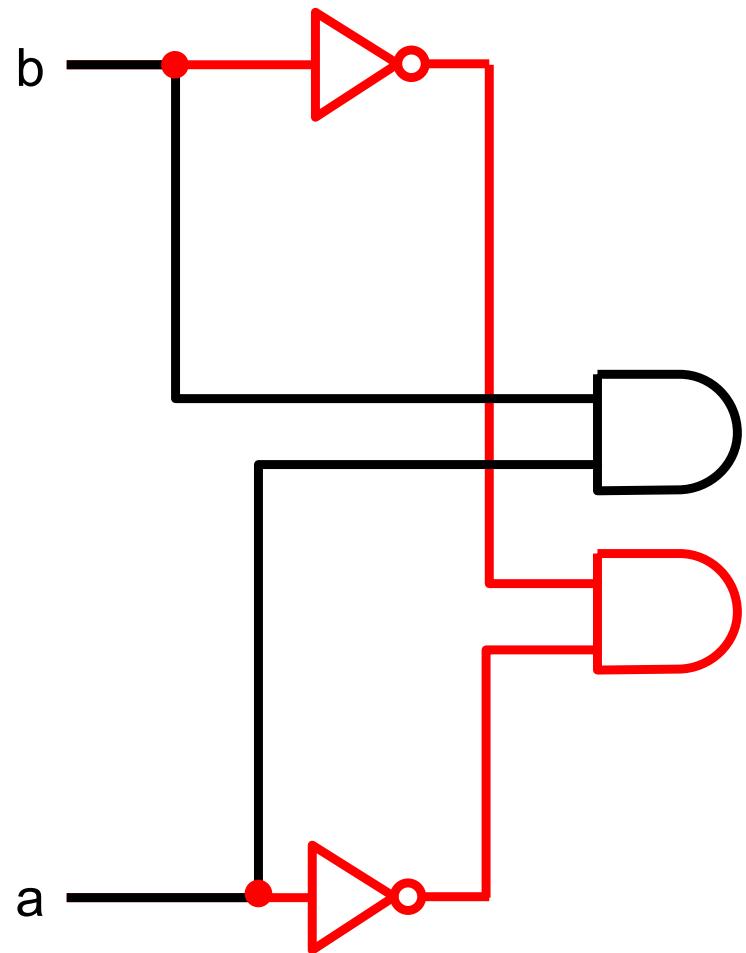


Implementación  
multinivel

$$\begin{aligned}z_2 &= ab + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR



Implementación  
multinivel

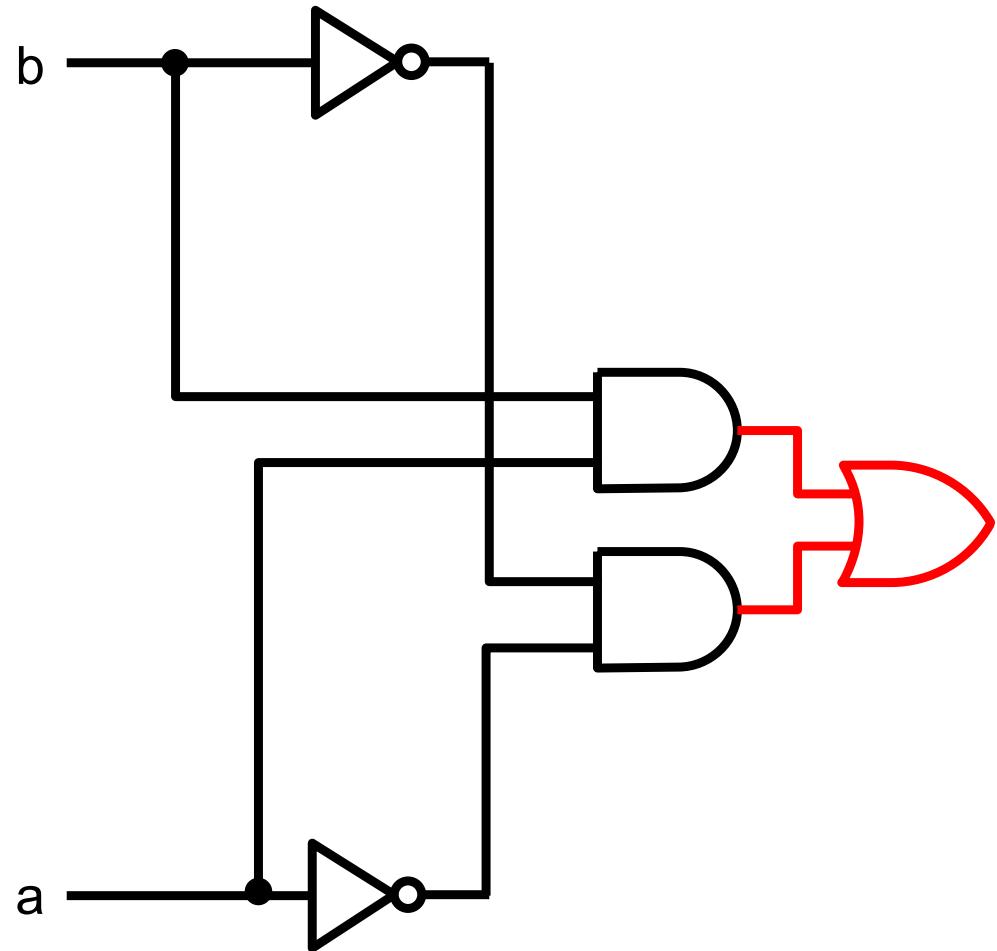
$$z_2 = a\bar{b} + (\bar{a}\bar{b} + ab)c_2$$

$$z_1 = (\bar{a}\bar{b} + ab)c_1$$

$$z_0 = \bar{a}b + (\bar{a}\bar{b} + ab)c_0$$



# Síntesis de redes AND-OR

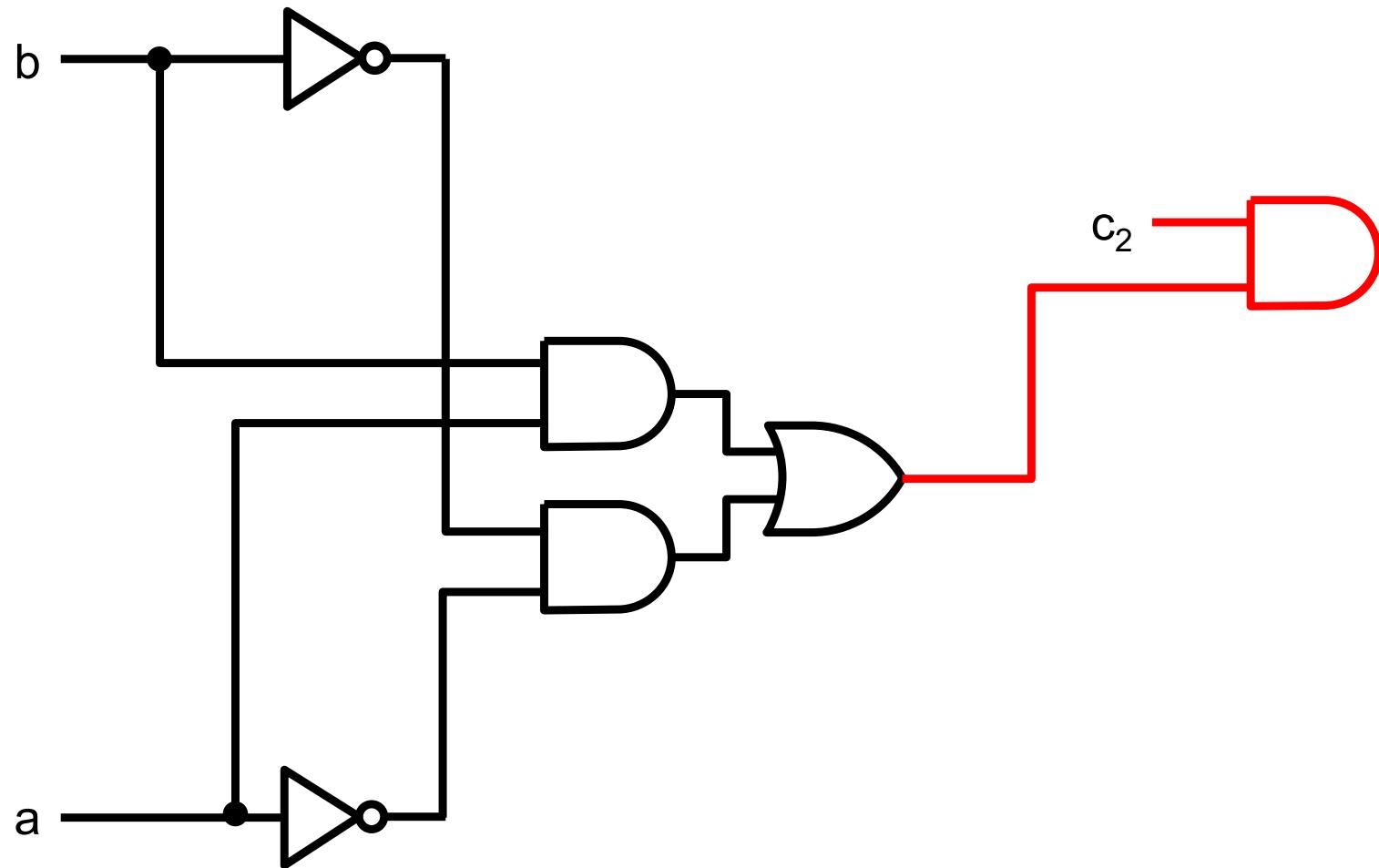


Implementación  
multinivel

$$\begin{aligned}z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

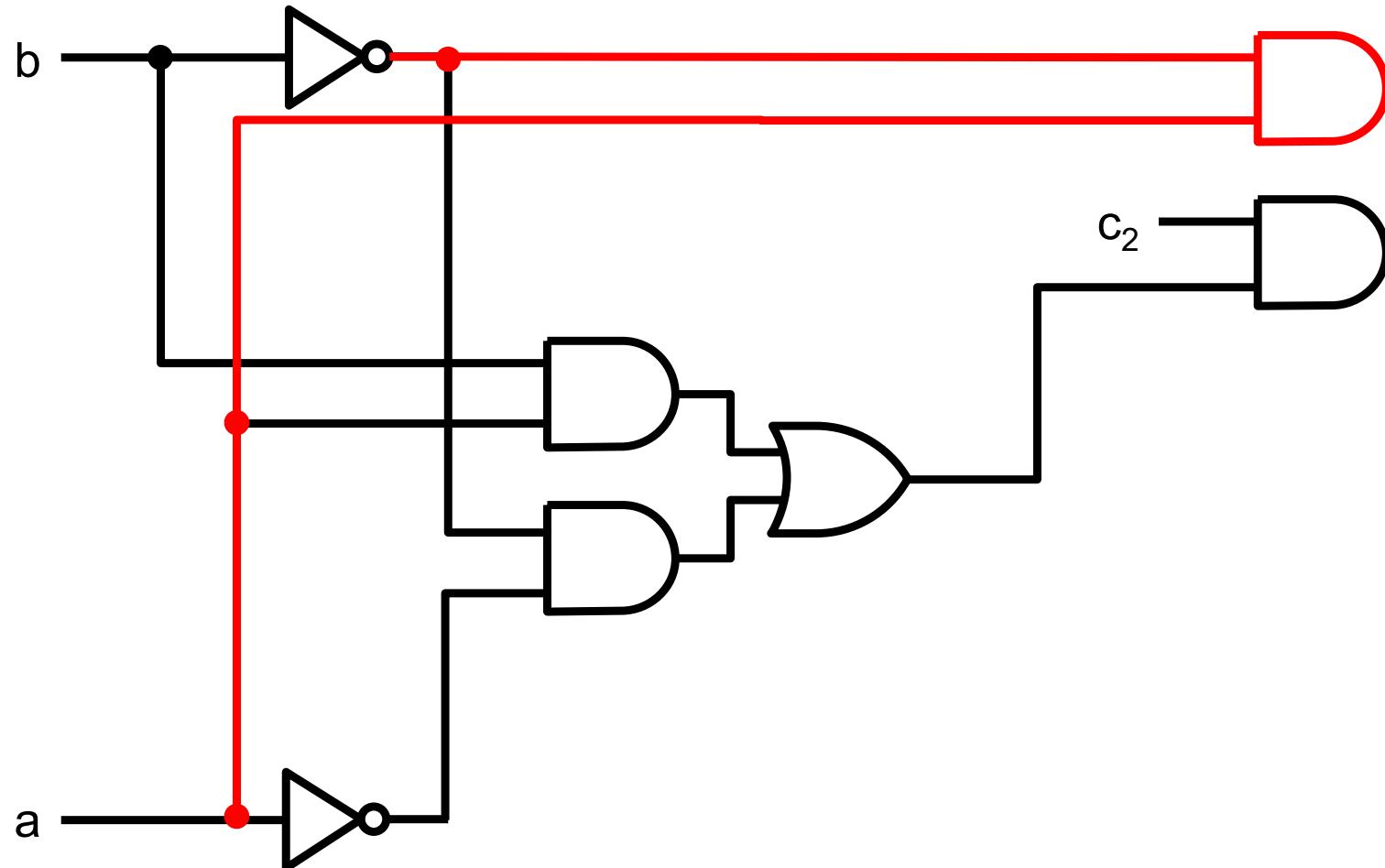


Implementación  
multinivel

$$\begin{aligned}z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

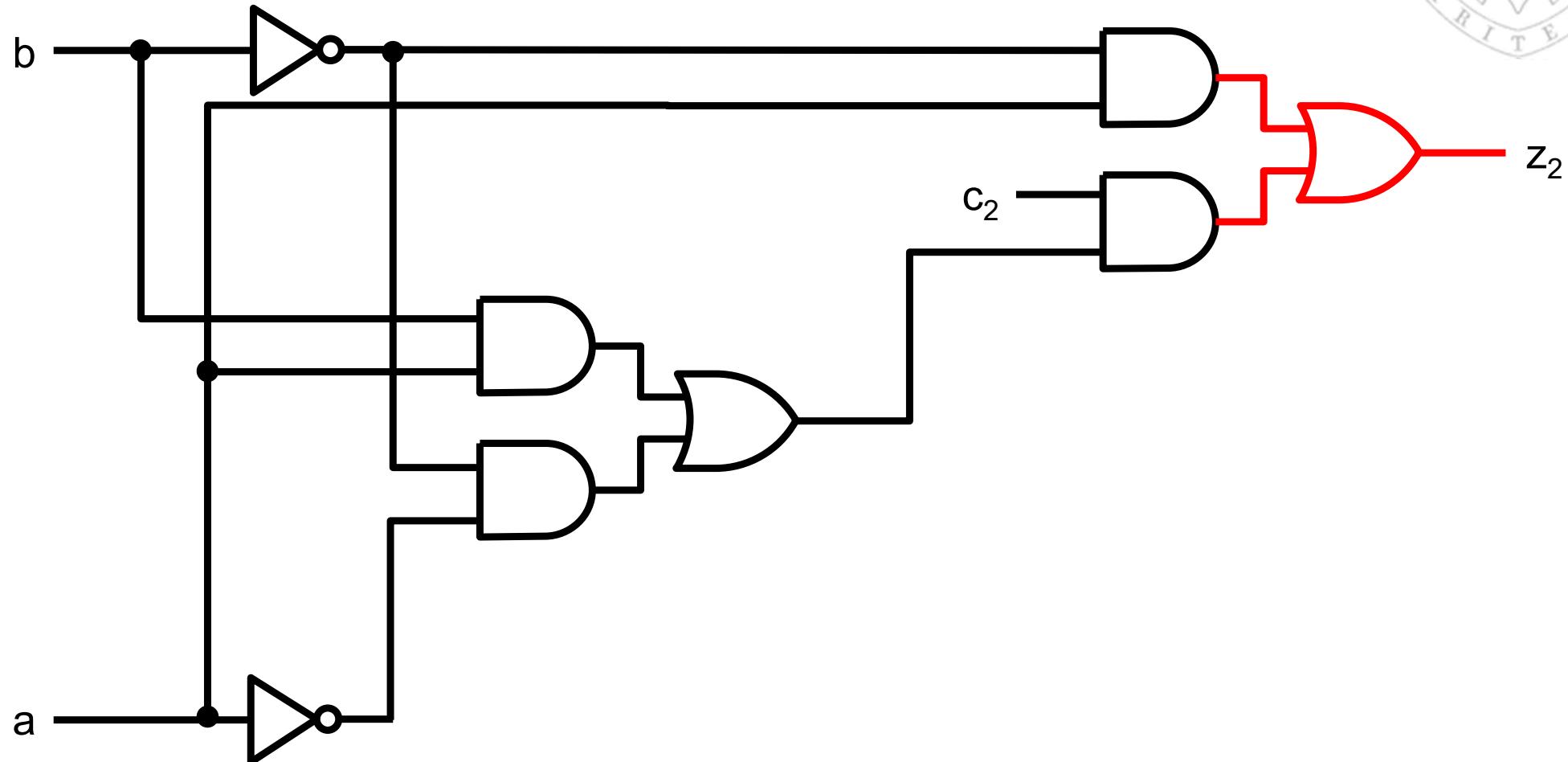


Implementación  
multinivel

$$\begin{aligned}z_2 &= \bar{a}\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

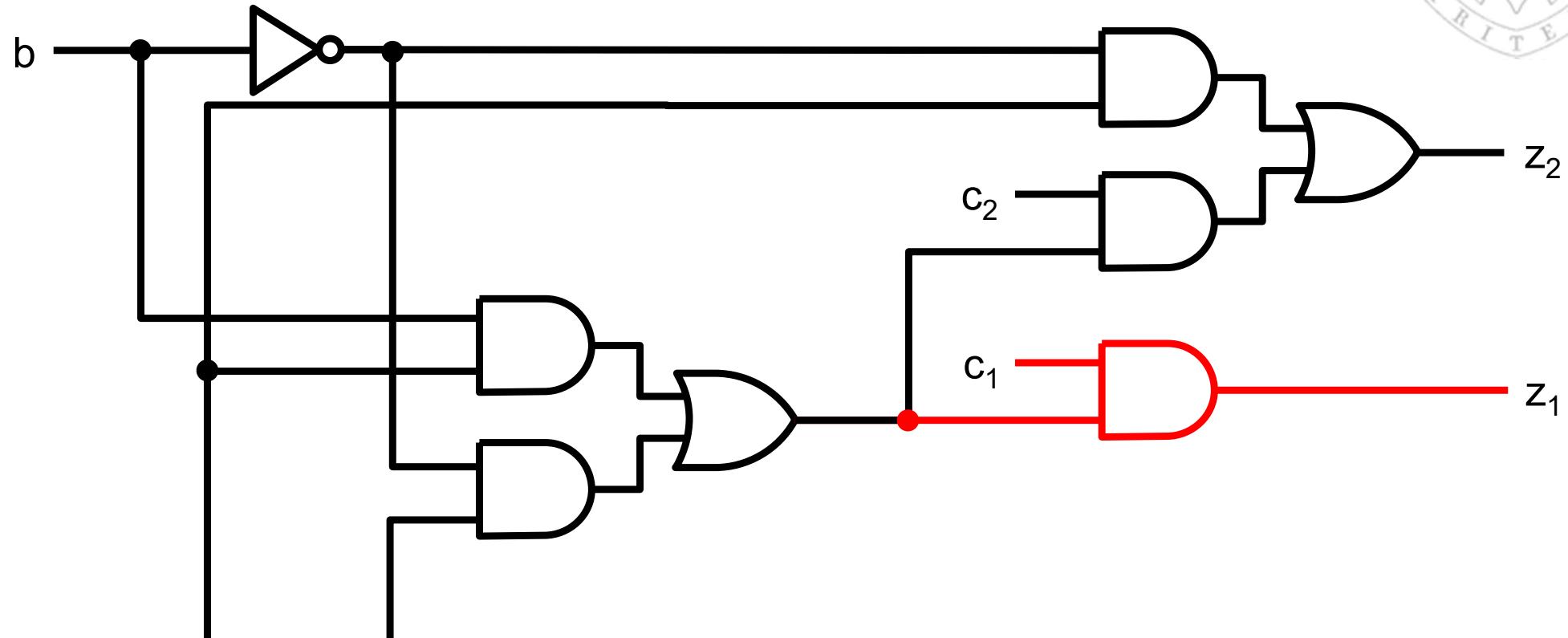


Implementación  
multinivel

$$\begin{aligned} z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\ z_1 &= (\bar{a}\bar{b} + ab)c_1 \\ z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0 \end{aligned}$$



# Síntesis de redes AND-OR

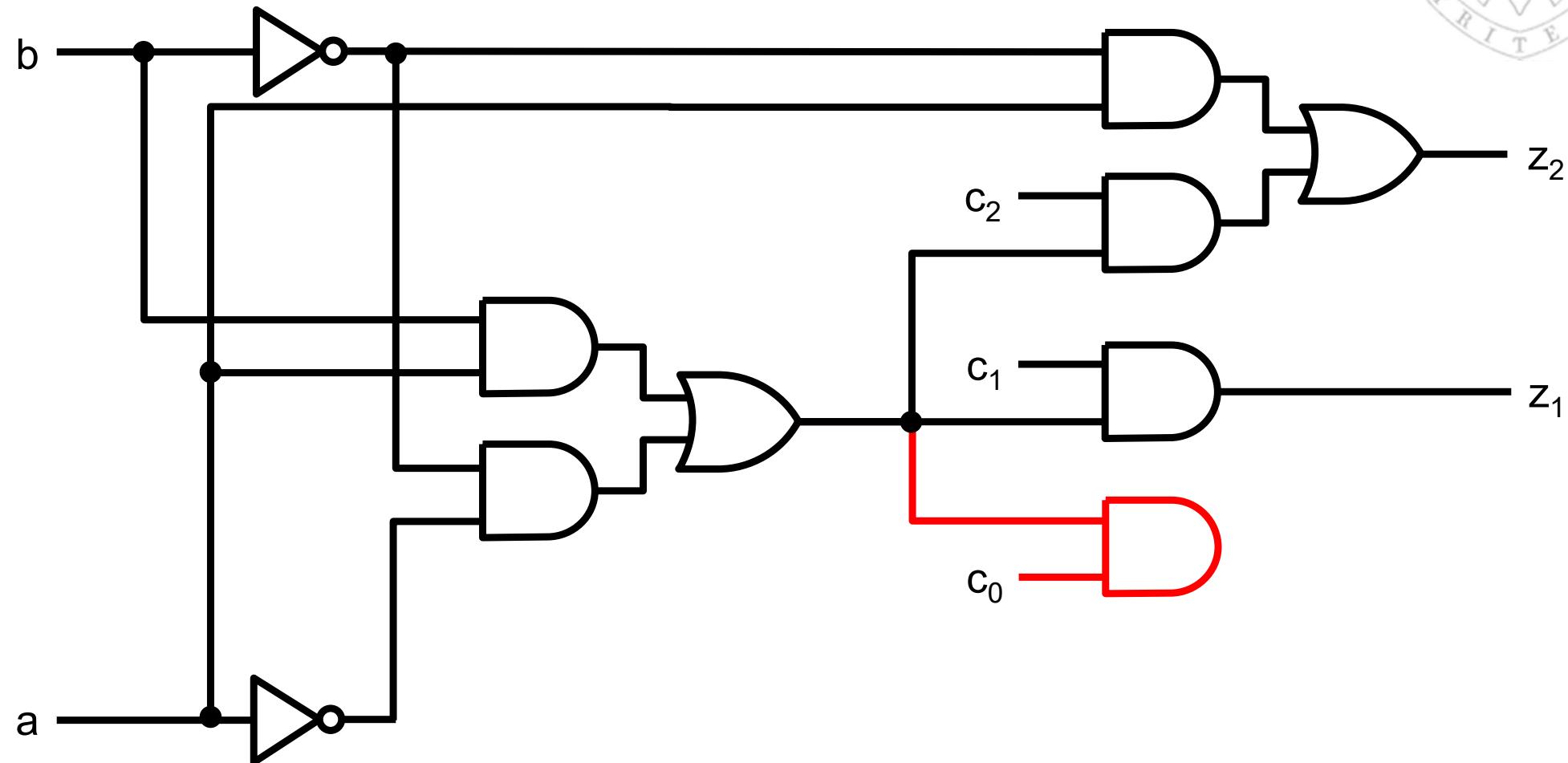


Implementación  
multinivel

$$\begin{aligned}z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

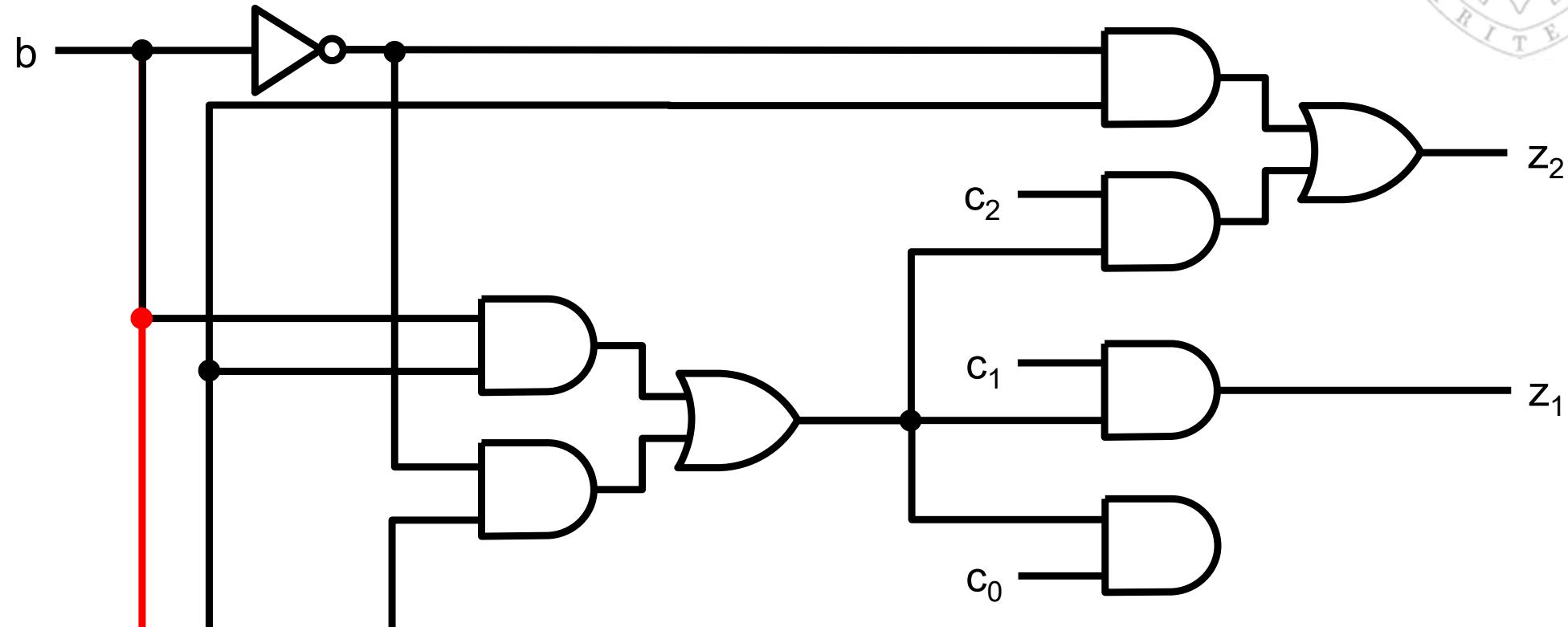


Implementación  
multinivel

$$\begin{aligned} z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\ z_1 &= (\bar{a}\bar{b} + ab)c_1 \\ z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0 \end{aligned}$$



# Síntesis de redes AND-OR

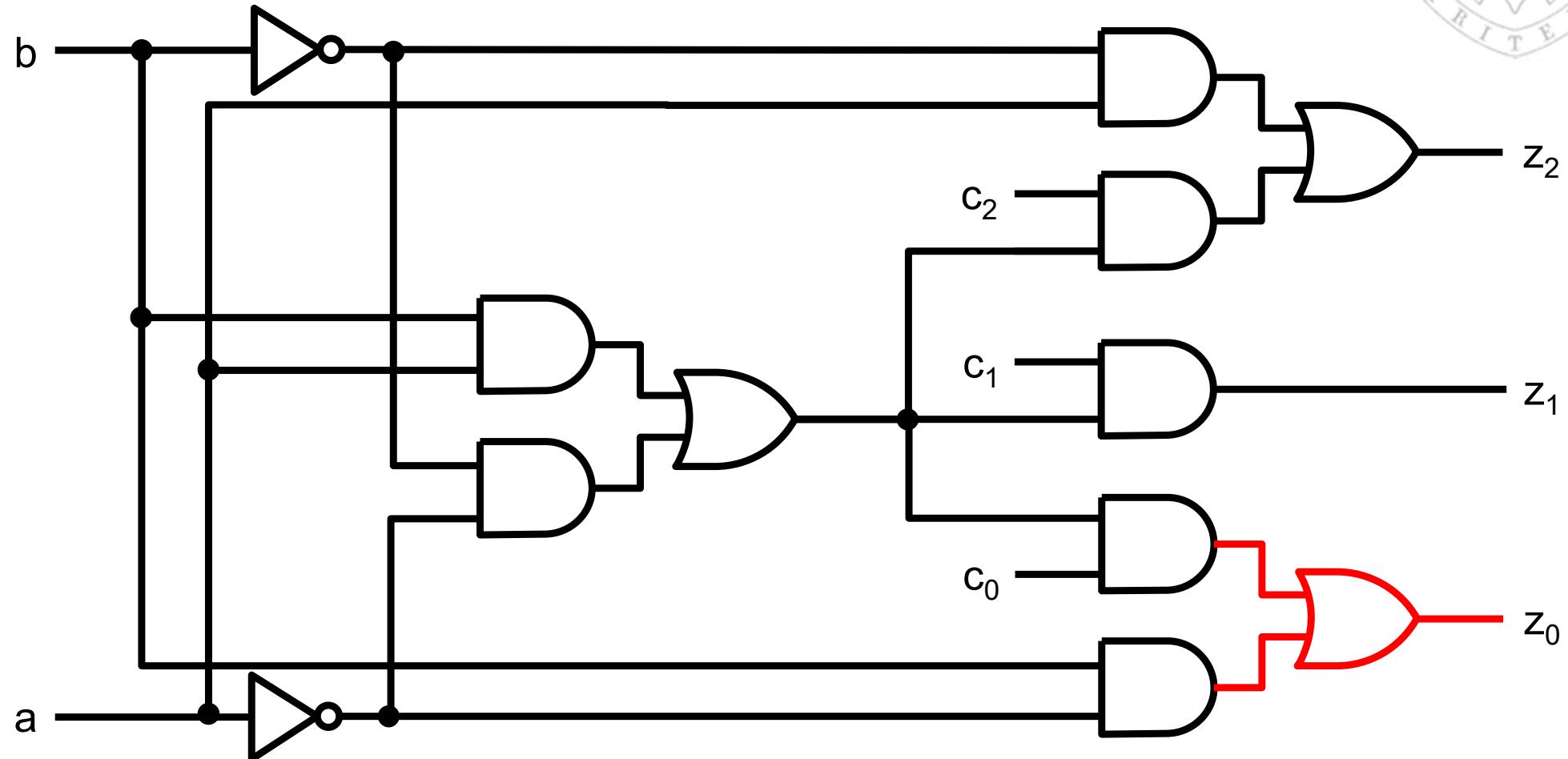


Implementación  
multinivel

$$\begin{aligned}z_2 &= ab + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR

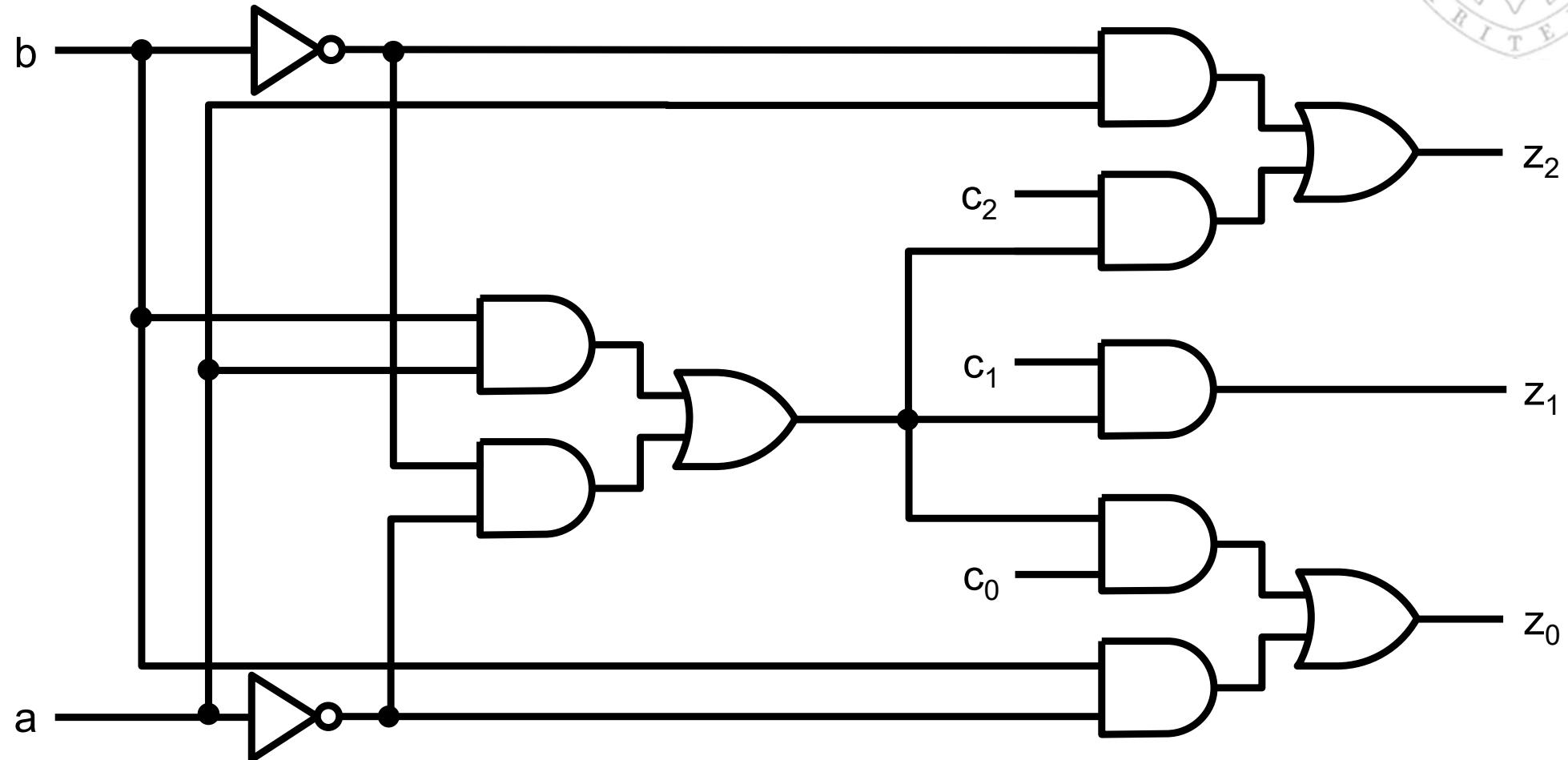


Implementación  
multinivel

$$\begin{aligned}z_2 &= a\bar{b} + (\bar{a}\bar{b} + ab)c_2 \\z_1 &= (\bar{a}\bar{b} + ab)c_1 \\z_0 &= \bar{a}b + (\bar{a}\bar{b} + ab)c_0\end{aligned}$$



# Síntesis de redes AND-OR



Implementación  
multinivel

$$z_2 = ab + (\bar{a}\bar{b} + ab)c_2$$

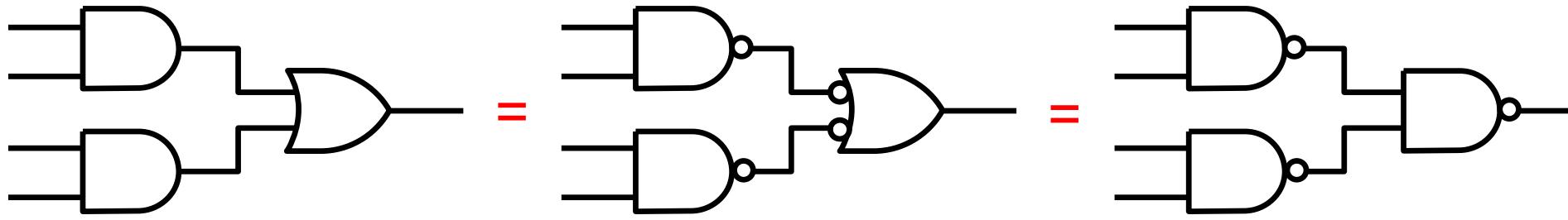
$$z_1 = (\bar{a}\bar{b} + ab)c_1$$

$$z_0 = \bar{a}b + (\bar{a}\bar{b} + ab)c_0$$

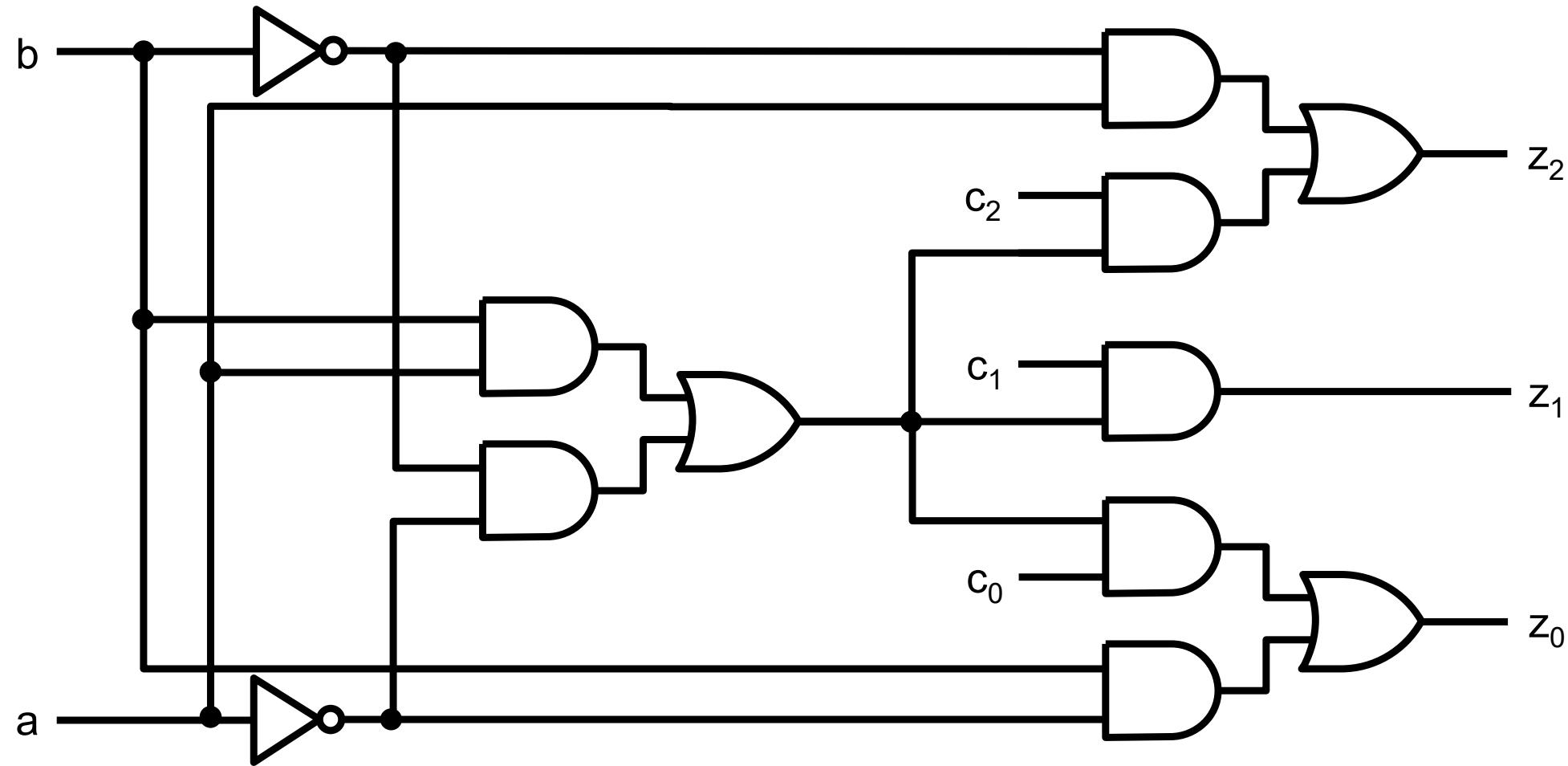
# Síntesis de redes NAND



- 2 niveles AND-OR equivalen a 2 niveles NAND-NAND



- Método:
  - Obtener una red AND-OR.
  - Añadir pares de inversores a las salidas de las puertas AND (o a las entradas de las puertas OR).
  - Uniformizar la notación de las puertas NAND.
  - Eliminar dobles inversores donde sea posible.
  - Reemplazar inversores por su implementación con NAND.

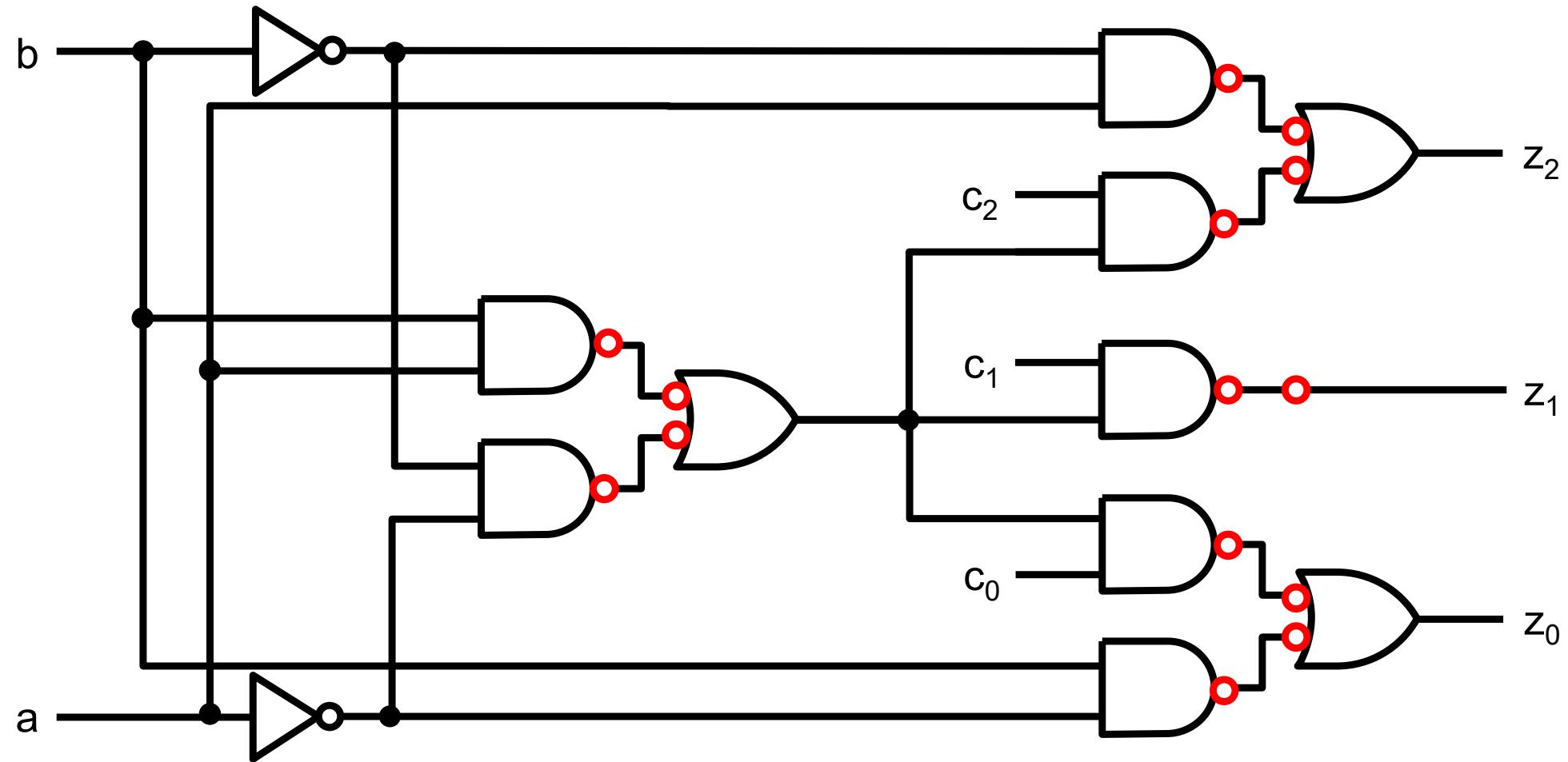




# Síntesis de redes NAND

tema 3:  
Implementación de sistemas combinacionales

vérion 14/07/23

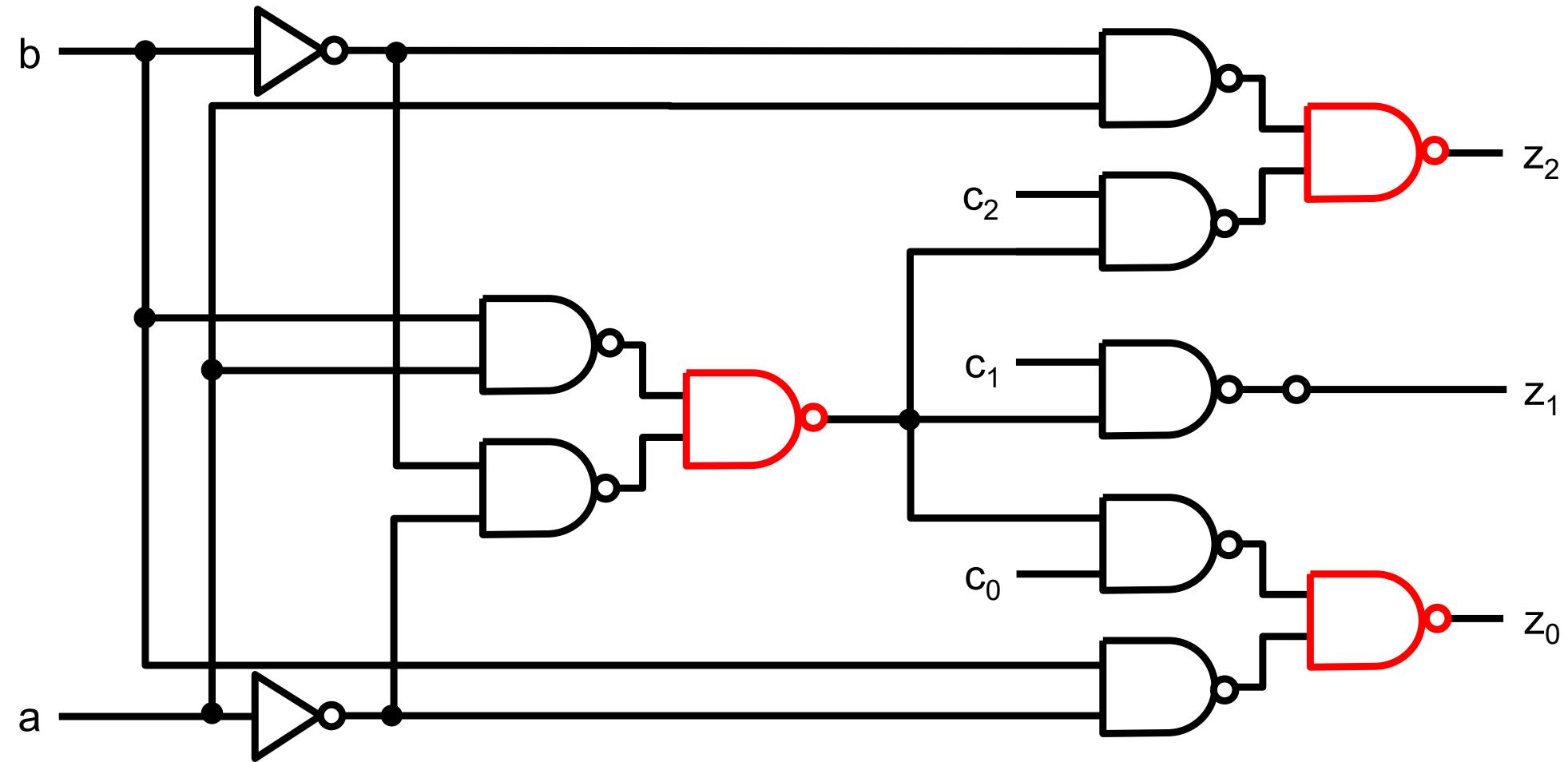




# Síntesis de redes NAND

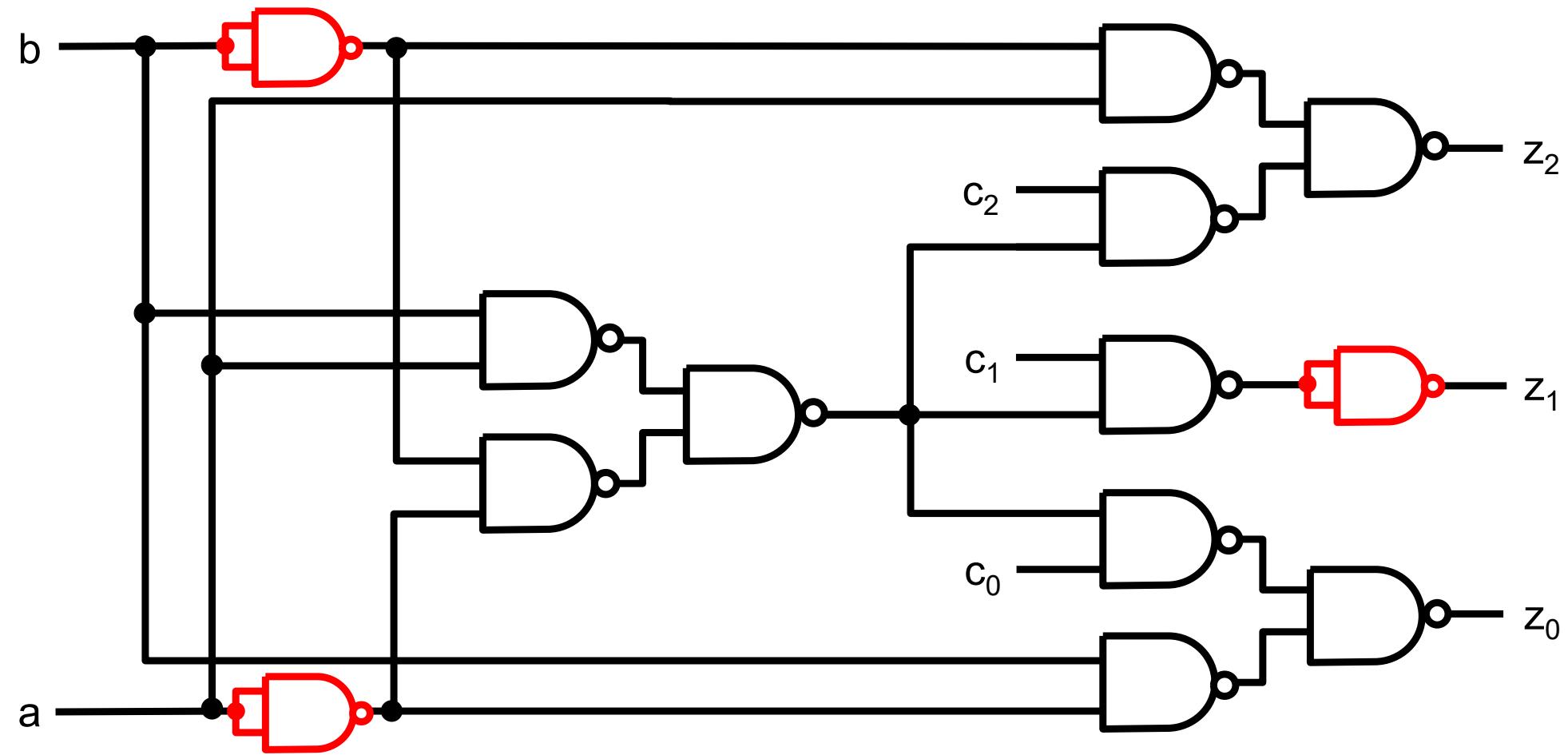
tema 3:  
Implementación de sistemas combinacionales

vérion 14/07/23



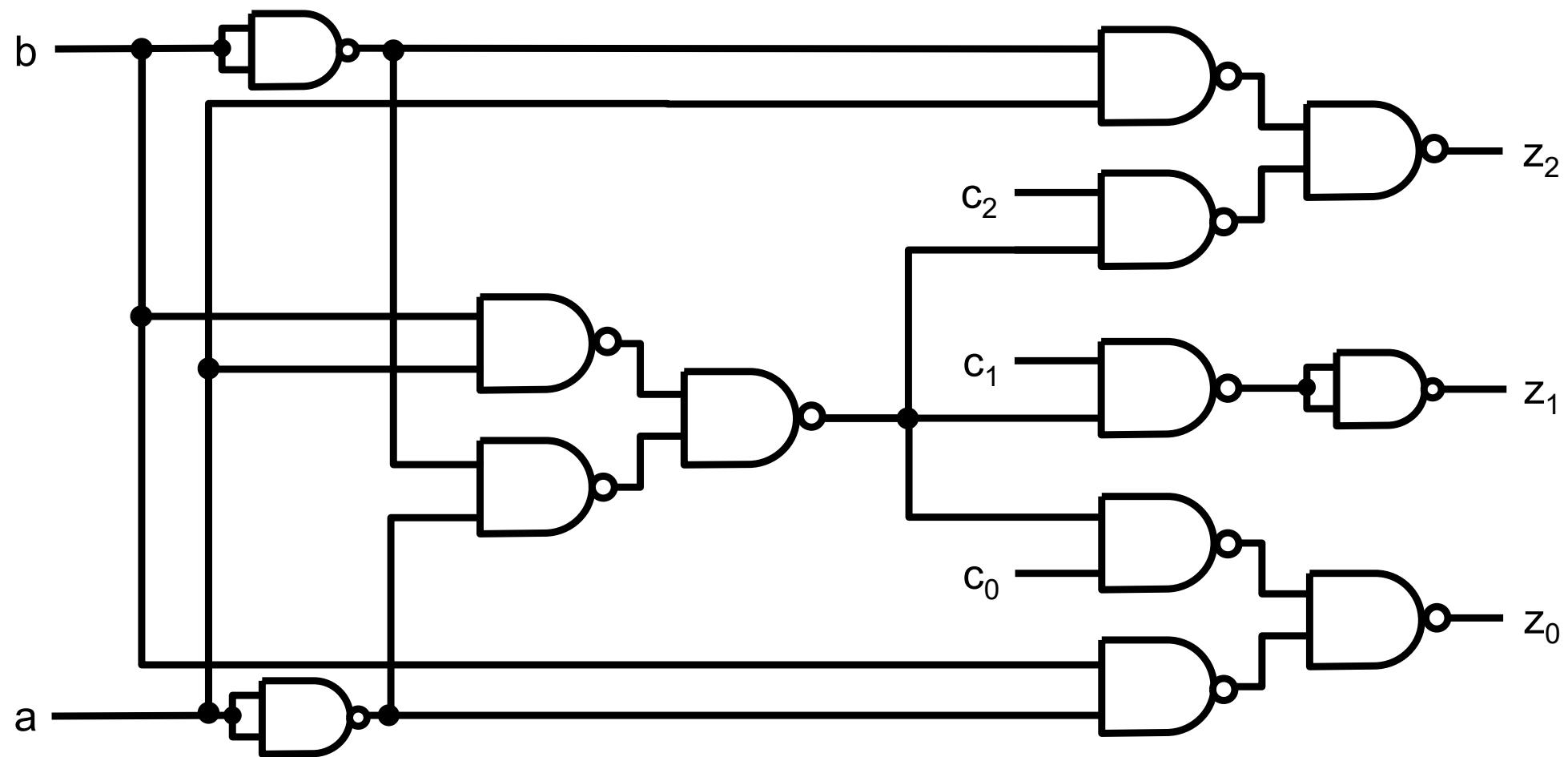


# Síntesis de redes NAND





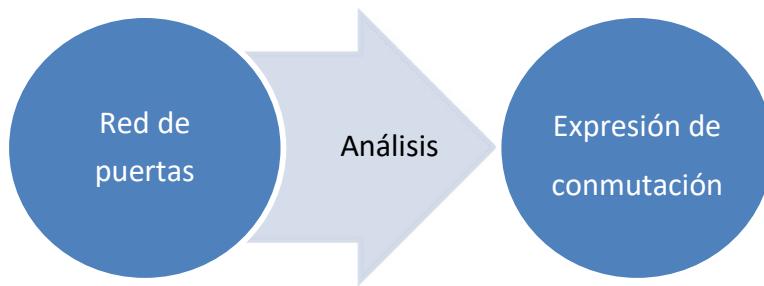
# Síntesis de redes NAND



# Análisis de redes de puertas



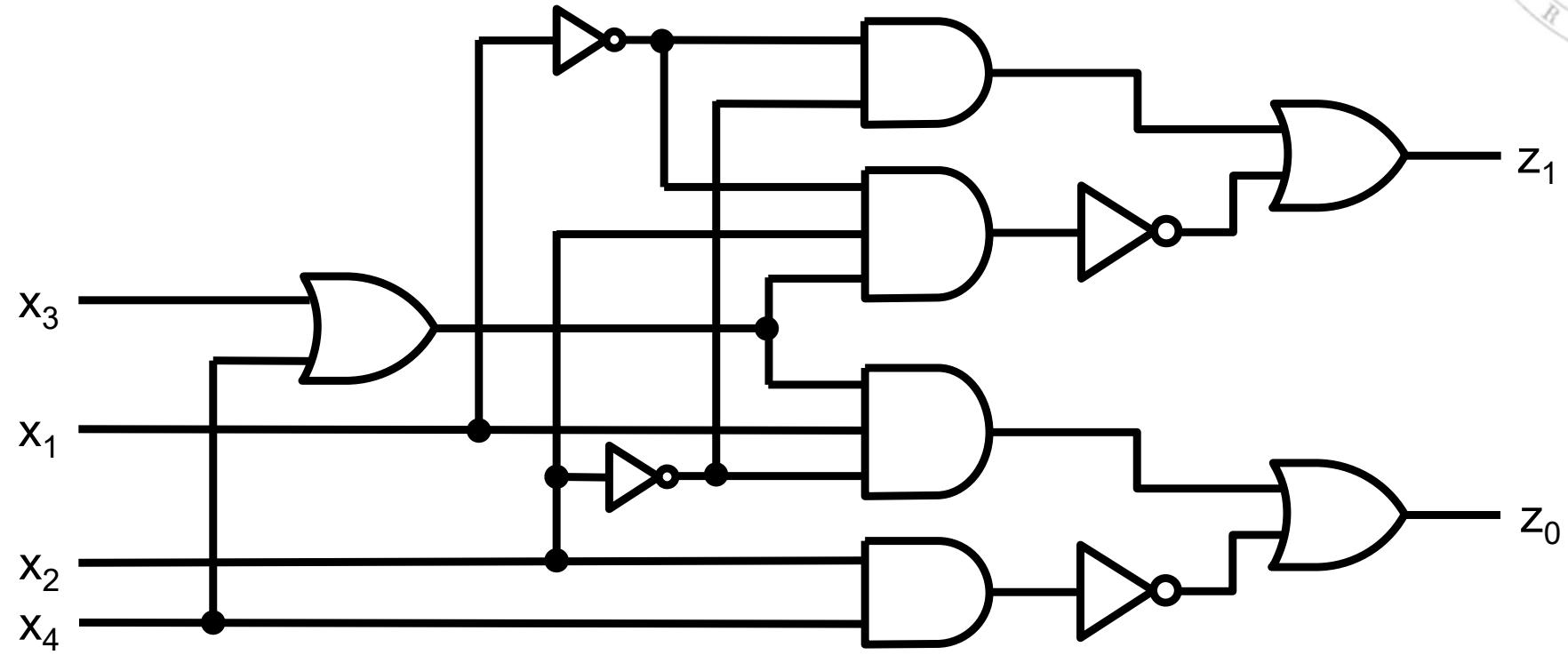
- Dada una red de puertas obtener una descripción de su conducta



- **Método:**
  - Dar nombre a cada una de las interconexiones intermedias.
  - En dirección de entradas a salidas, obtener una EC de cada una de dichas interconexiones como función de las entradas.
  - Simplificar las expresiones obtenidas.

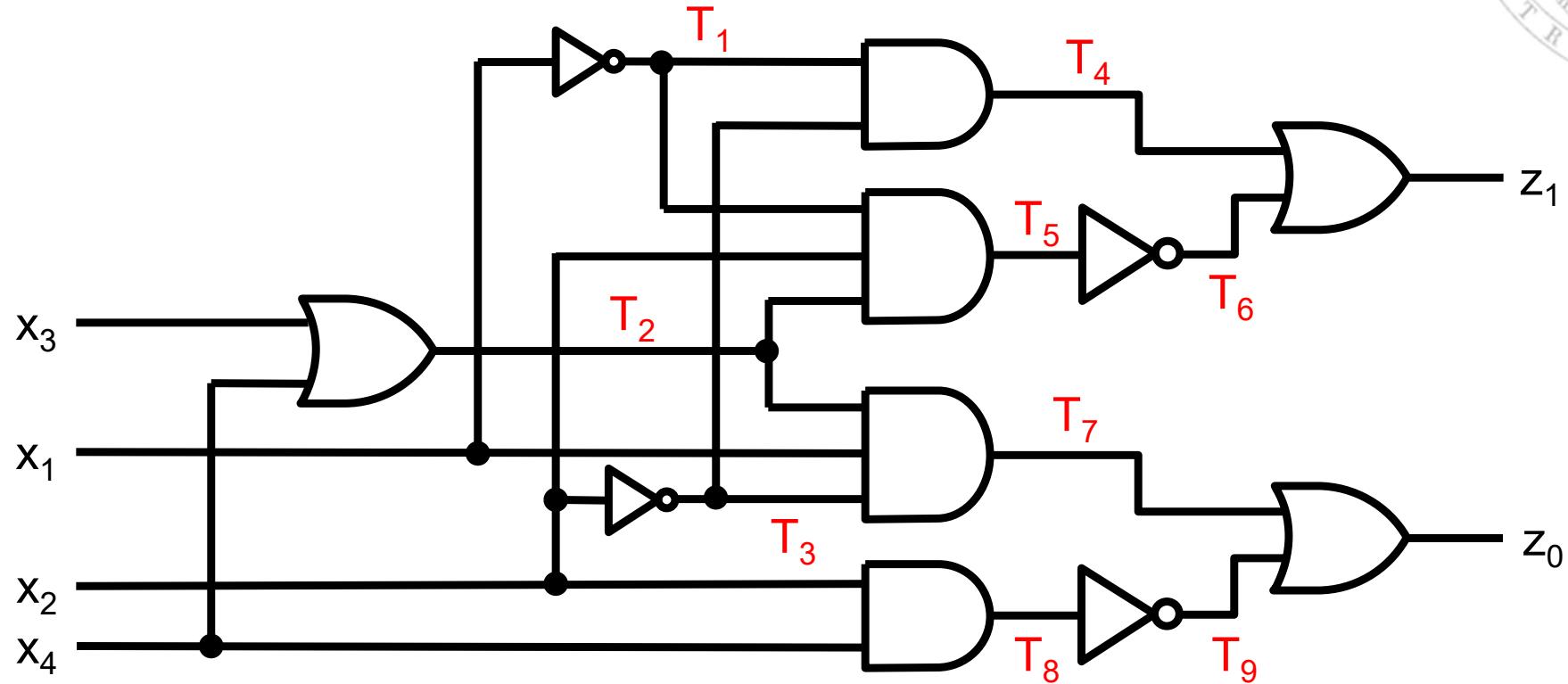


# Análisis de redes AND-OR



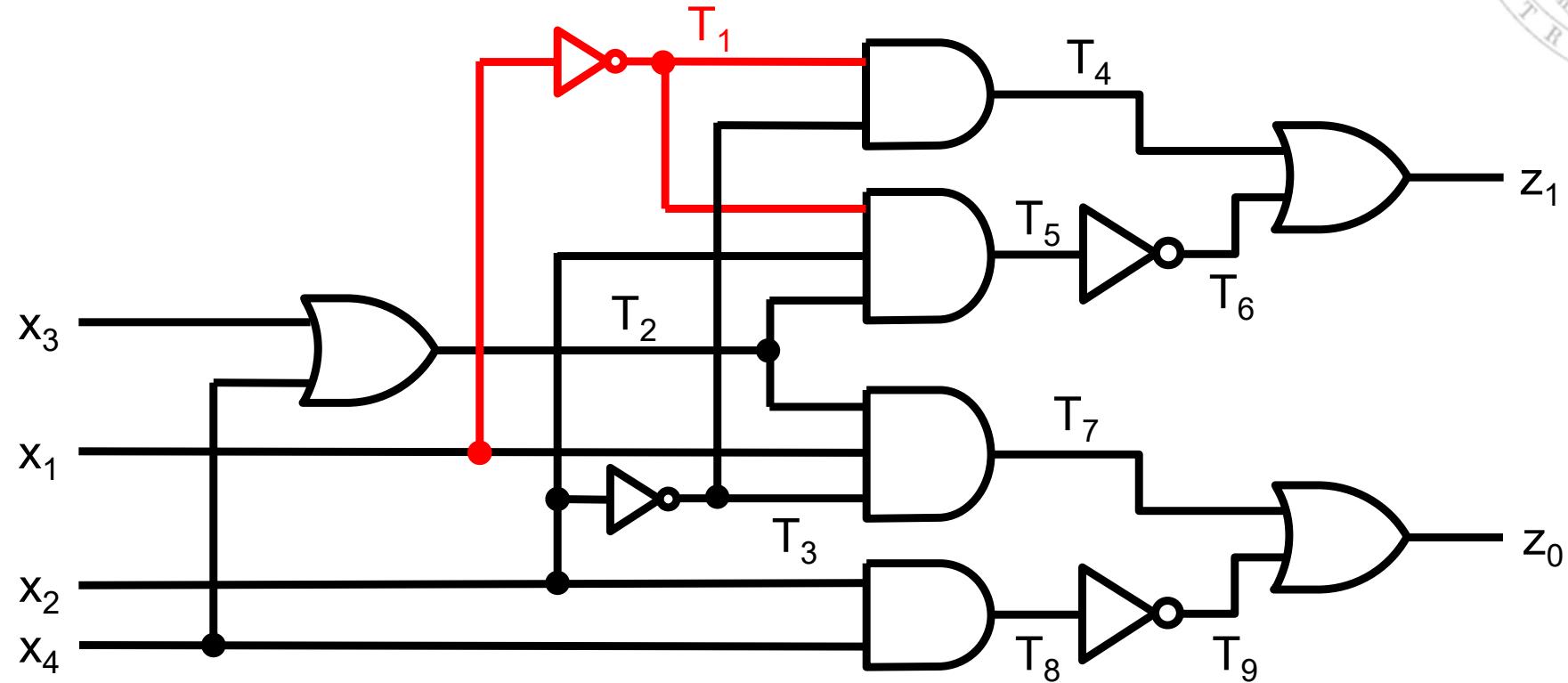


# Análisis de redes AND-OR





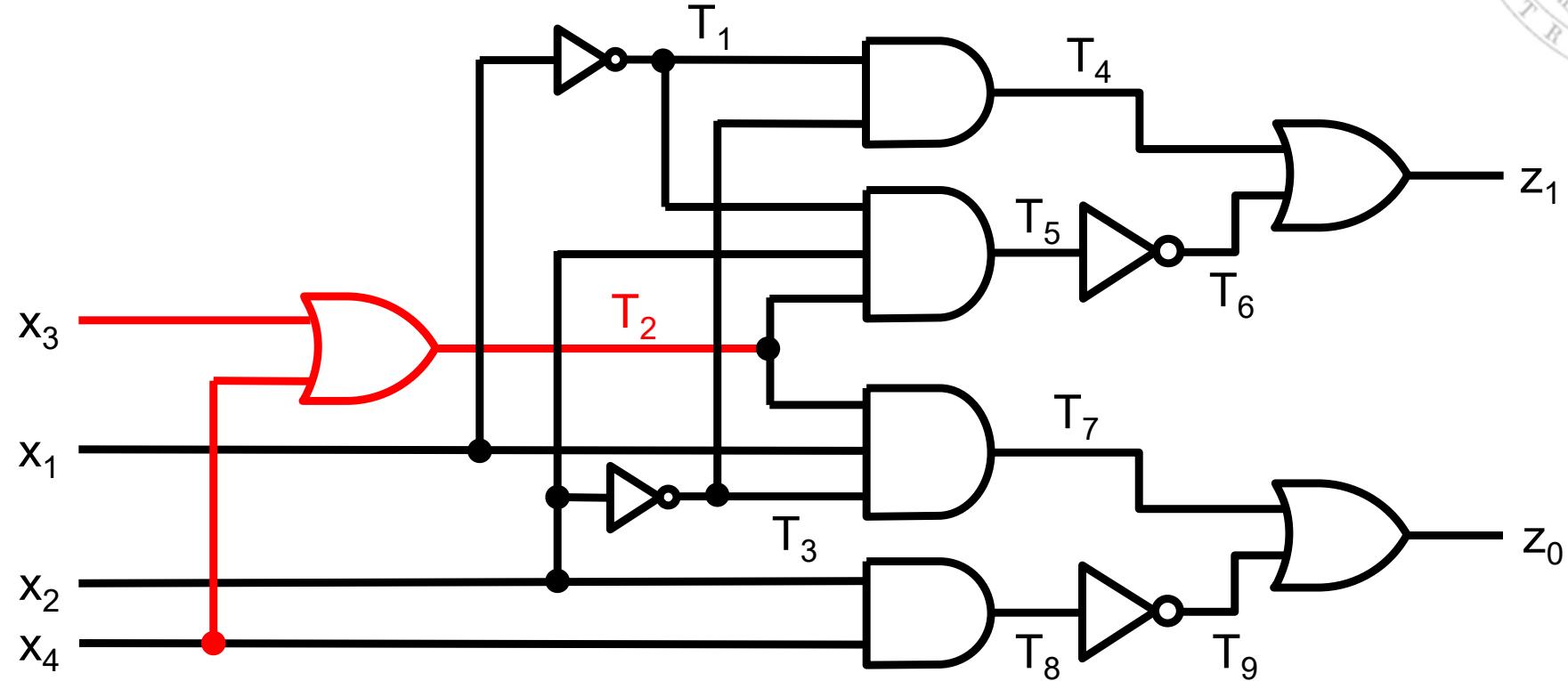
# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$



# Análisis de redes AND-OR

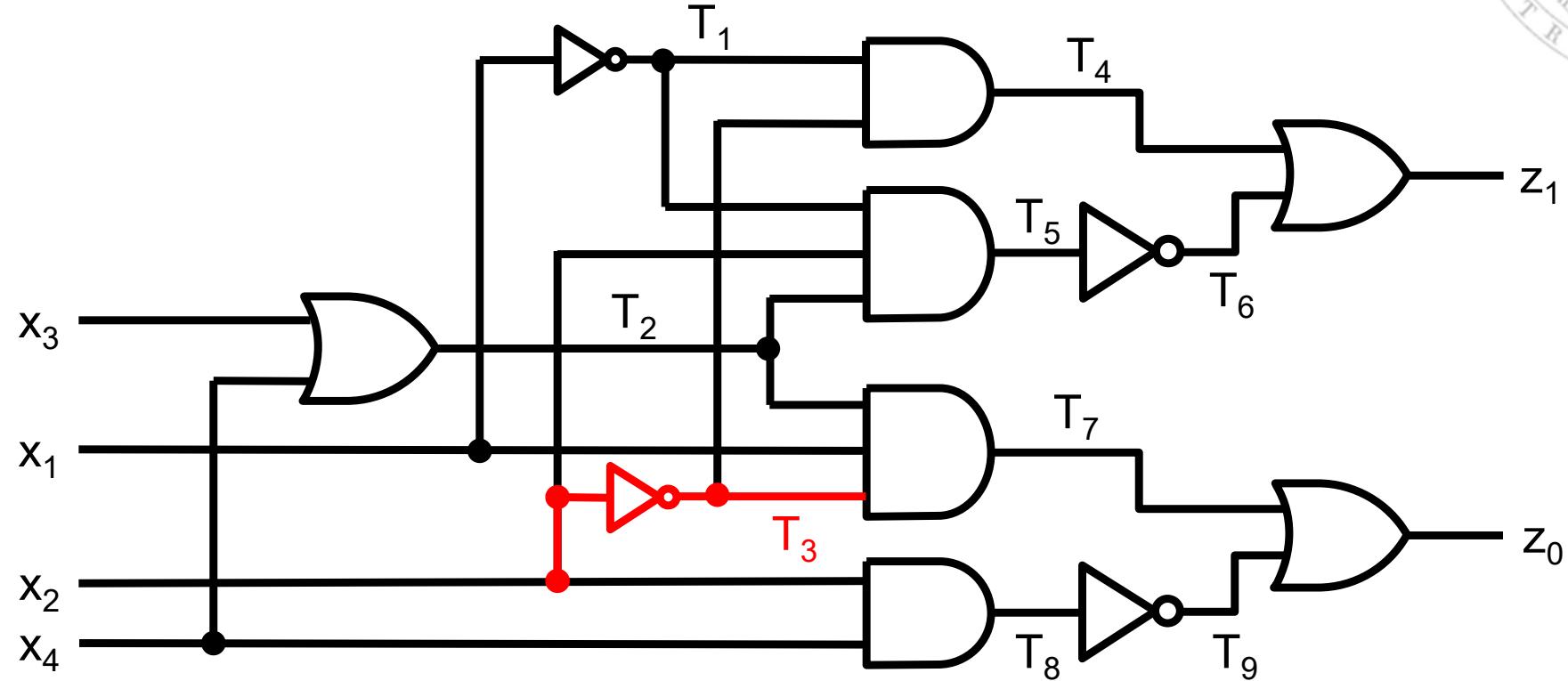


$$T_1 = \overline{x_1}$$

$$T_2 = x_3 + x_4$$



# Análisis de redes AND-OR



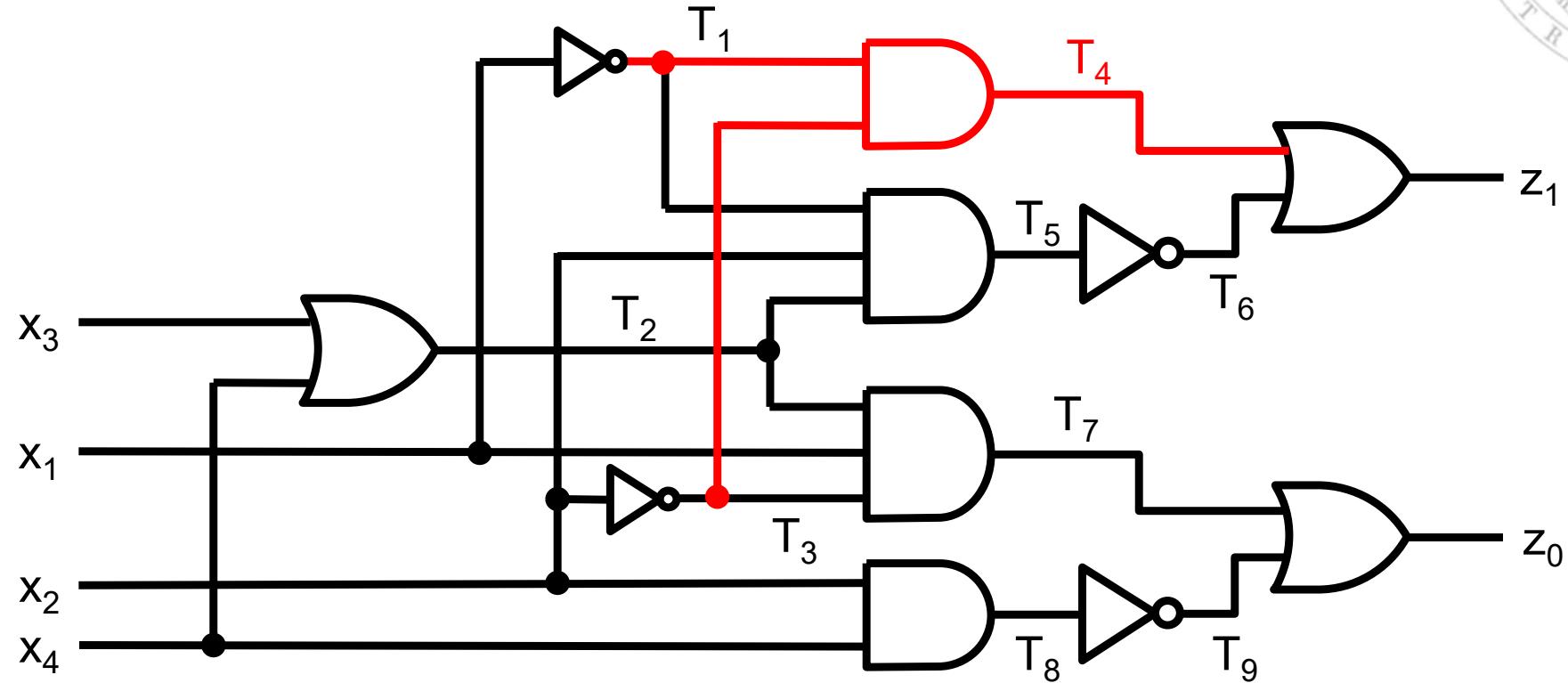
$$T_1 = \overline{x_1}$$

$$T_2 = x_3 + x_4$$

$$T_3 = \overline{x_2}$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

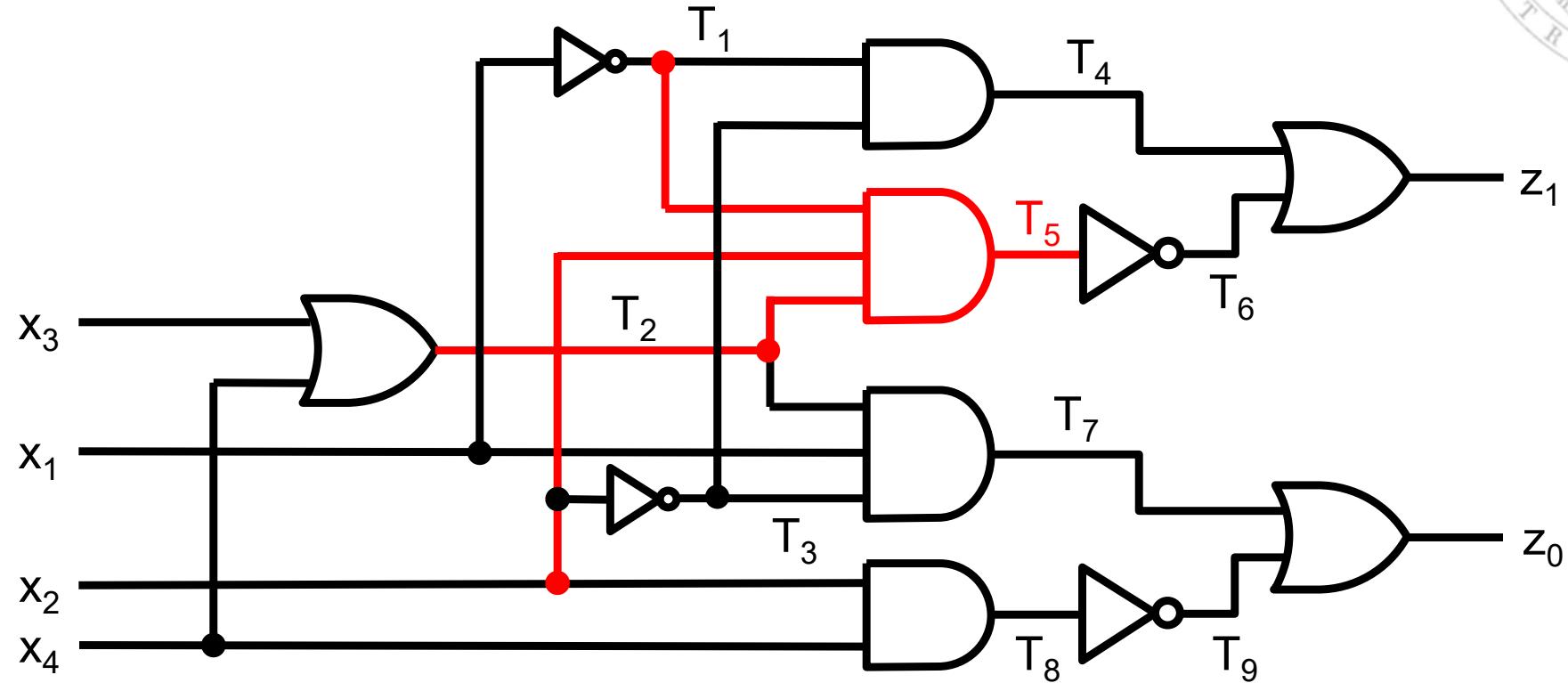
$$T_4 = T_1 \cdot T_3 = \overline{x_1} \cdot \overline{x_2}$$

$$T_2 = x_3 + x_4$$

$$T_3 = \overline{x_2}$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

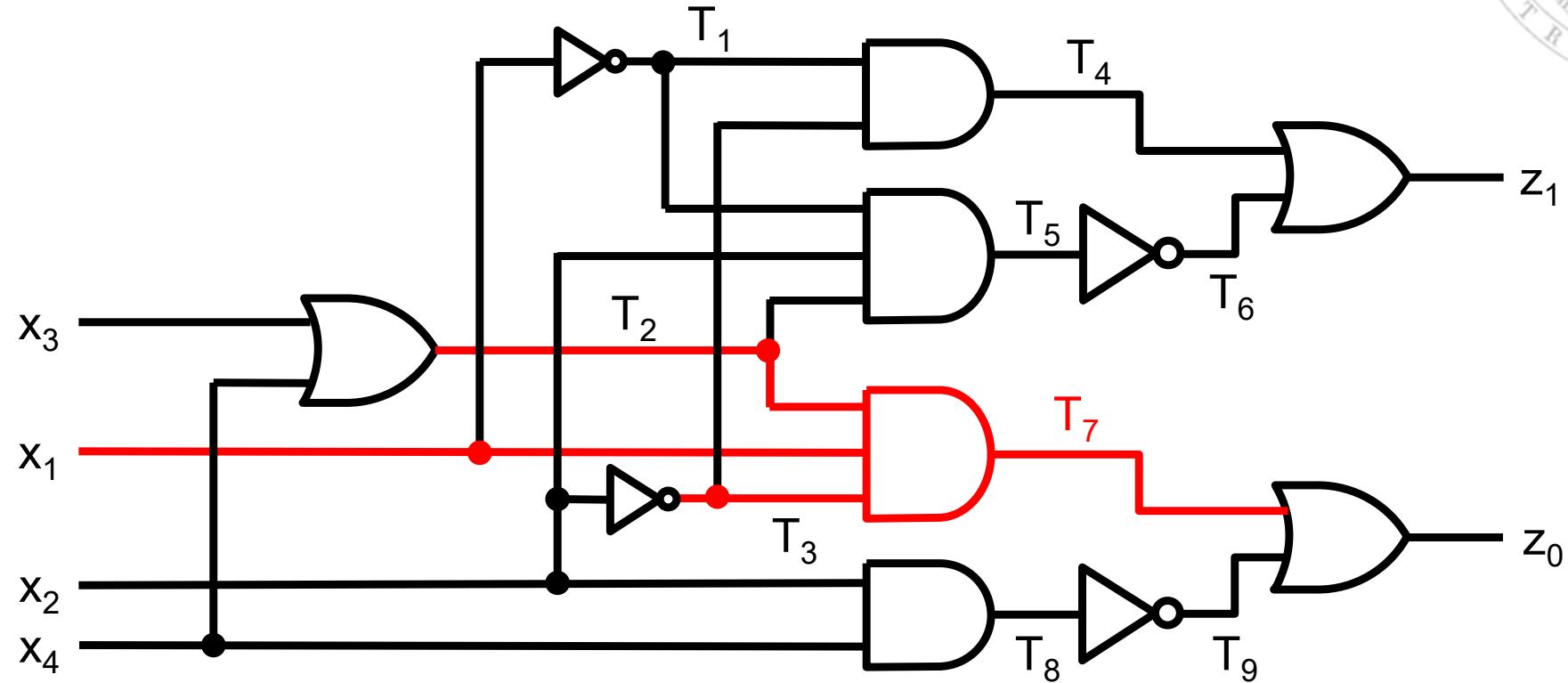
$$T_2 = x_3 + x_4$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_3 = \overline{x_2}$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_2 = x_3 + x_4$$

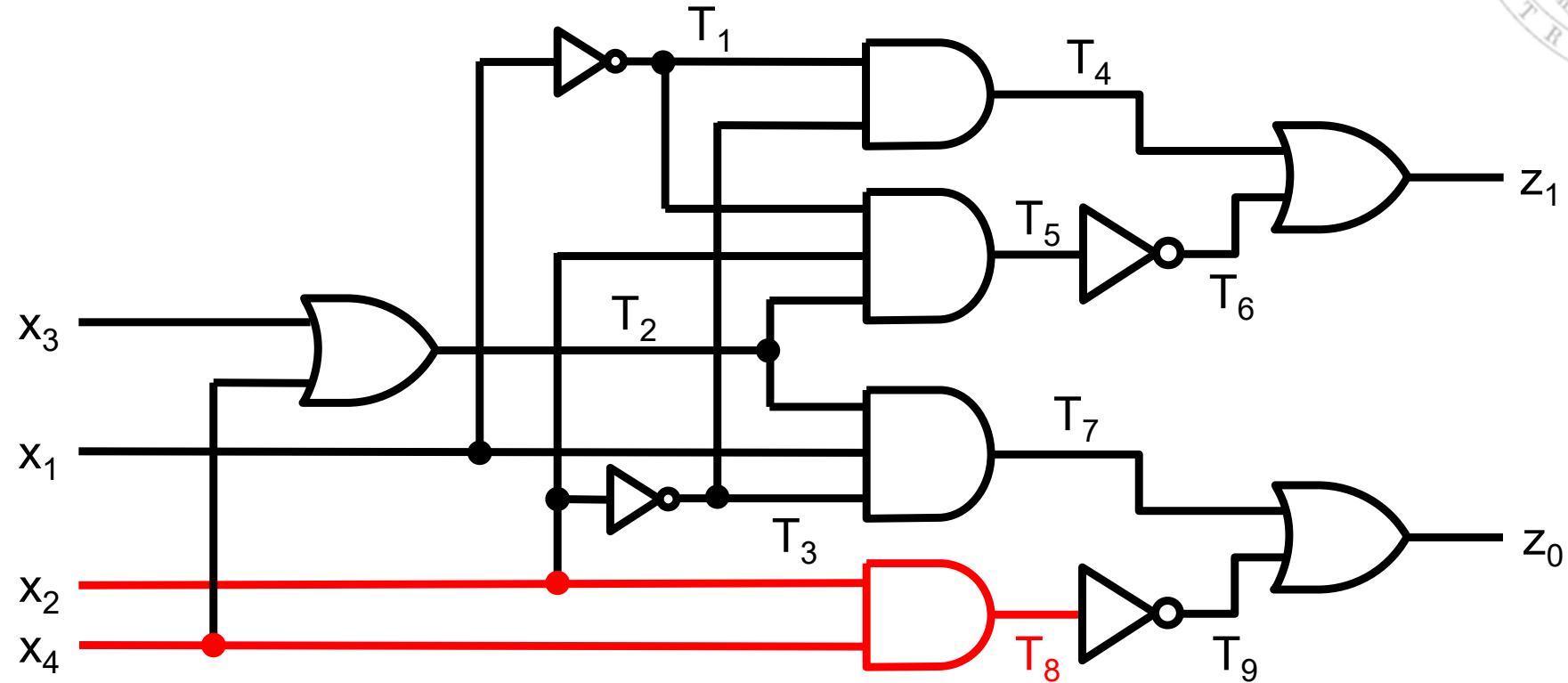
$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_3 = \overline{x_2}$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_2 = x_3 + x_4$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

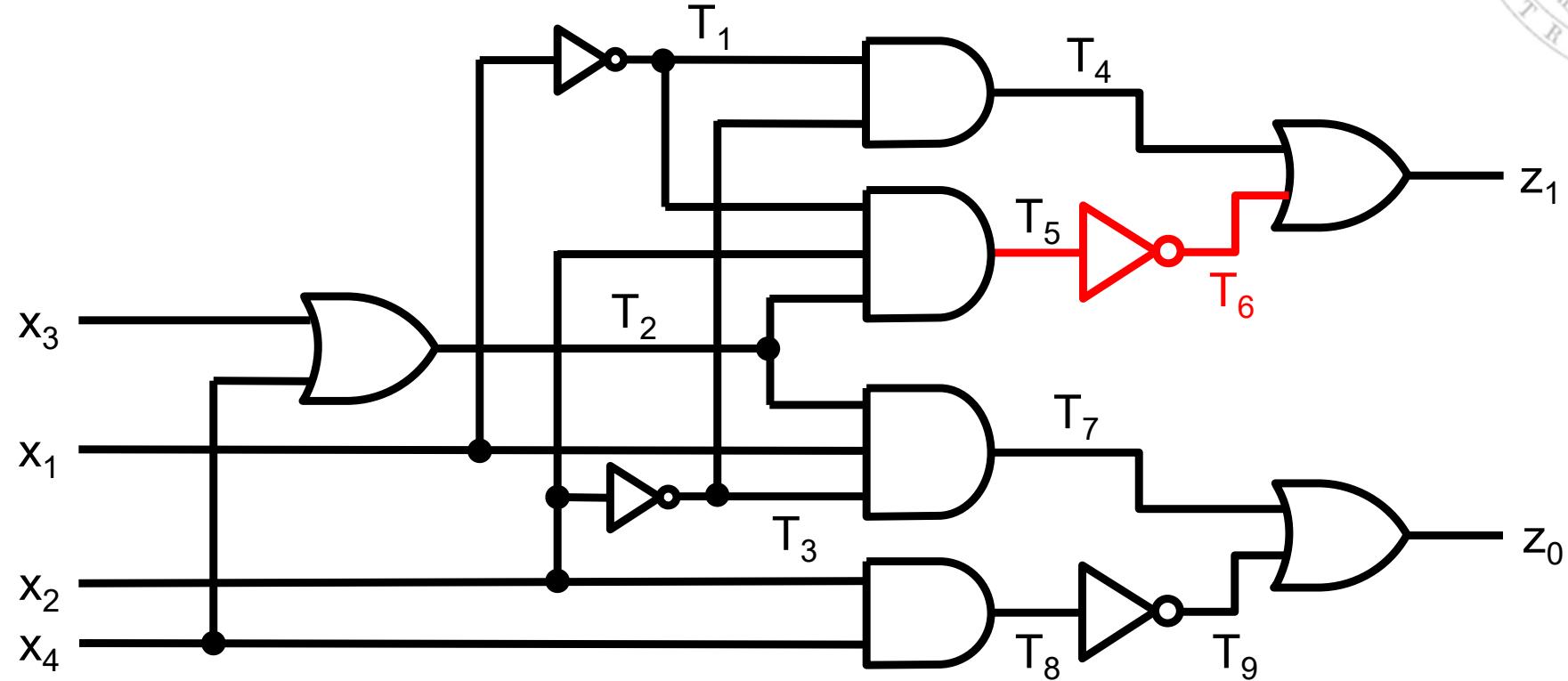
$$T_3 = \overline{x_2}$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$

$$T_8 = x_2 x_4$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_6 = \overline{T_5} = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_2 = x_3 + x_4$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

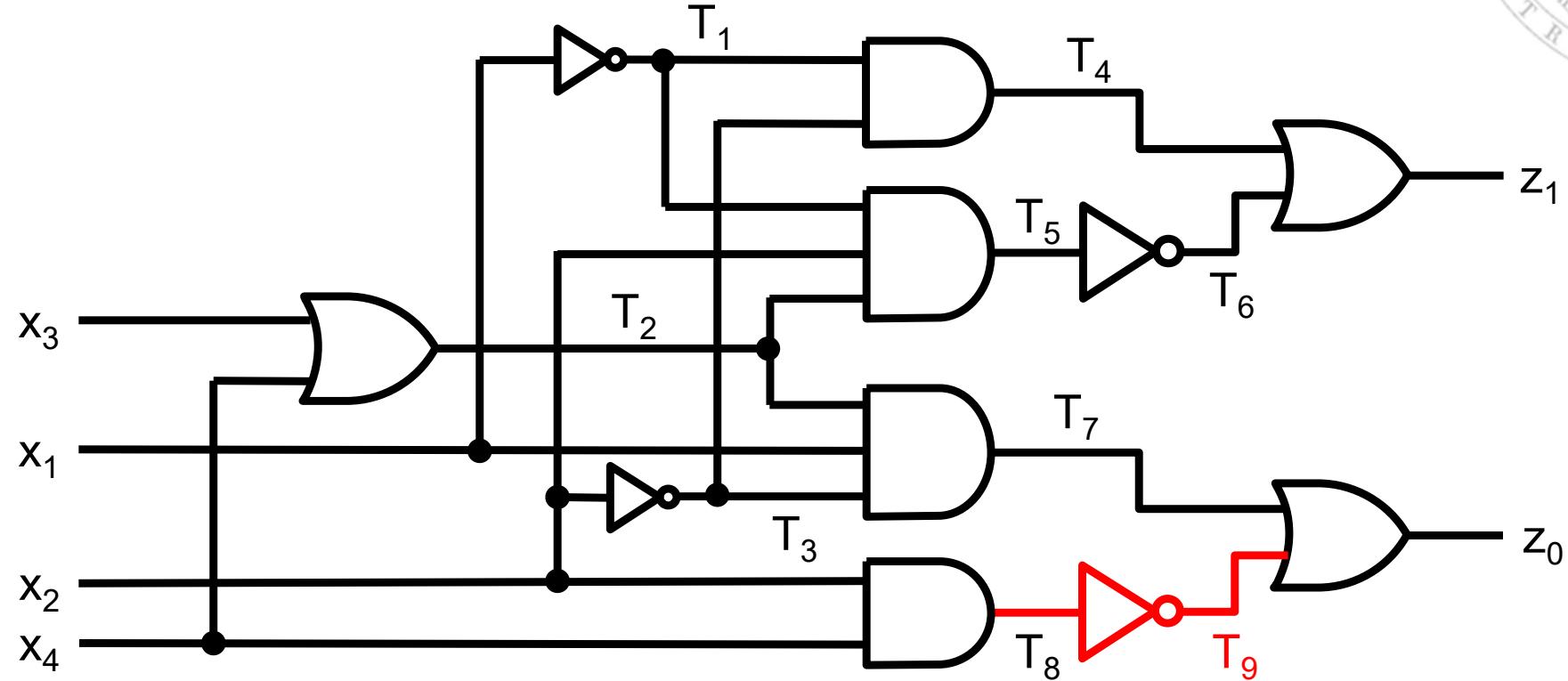
$$T_3 = \overline{x_2}$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$

$$T_8 = x_2 x_4$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_2 = x_3 + x_4$$

$$T_3 = \overline{x_2}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$

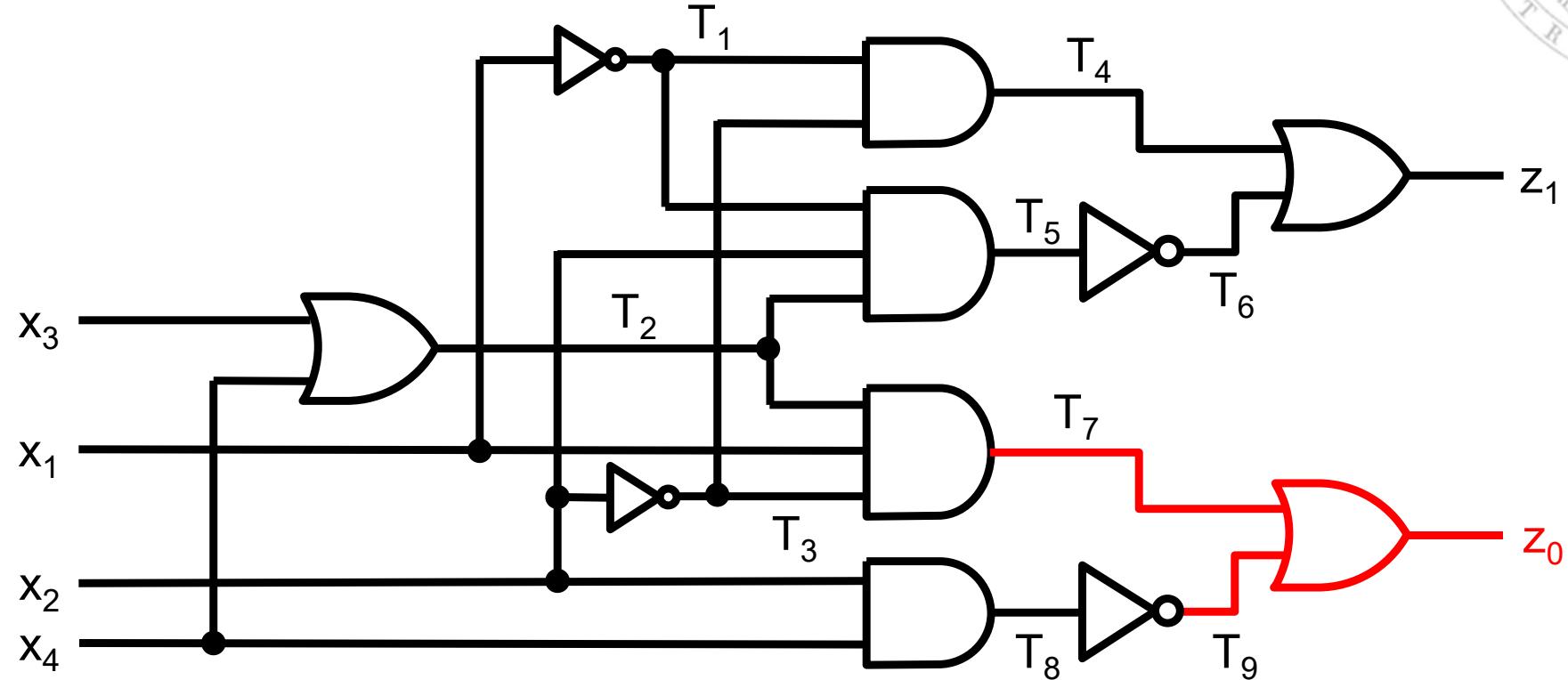
$$T_8 = x_2 x_4$$

$$T_6 = \overline{T_5} = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_9 = \overline{T_8} = \overline{x_2 x_4}$$



# Análisis de redes AND-OR



$$T_1 = \bar{x}_1$$

$$T_2 = x_3 + x_4$$

$$T_3 = \bar{x}_2$$

$$T_4 = T_1 T_3 = \bar{x}_1 \bar{x}_2$$

$$T_5 = T_1 x_2 T_2 = \bar{x}_1 x_2 (x_3 + x_4)$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \bar{x}_2$$

$$T_8 = x_2 x_4$$

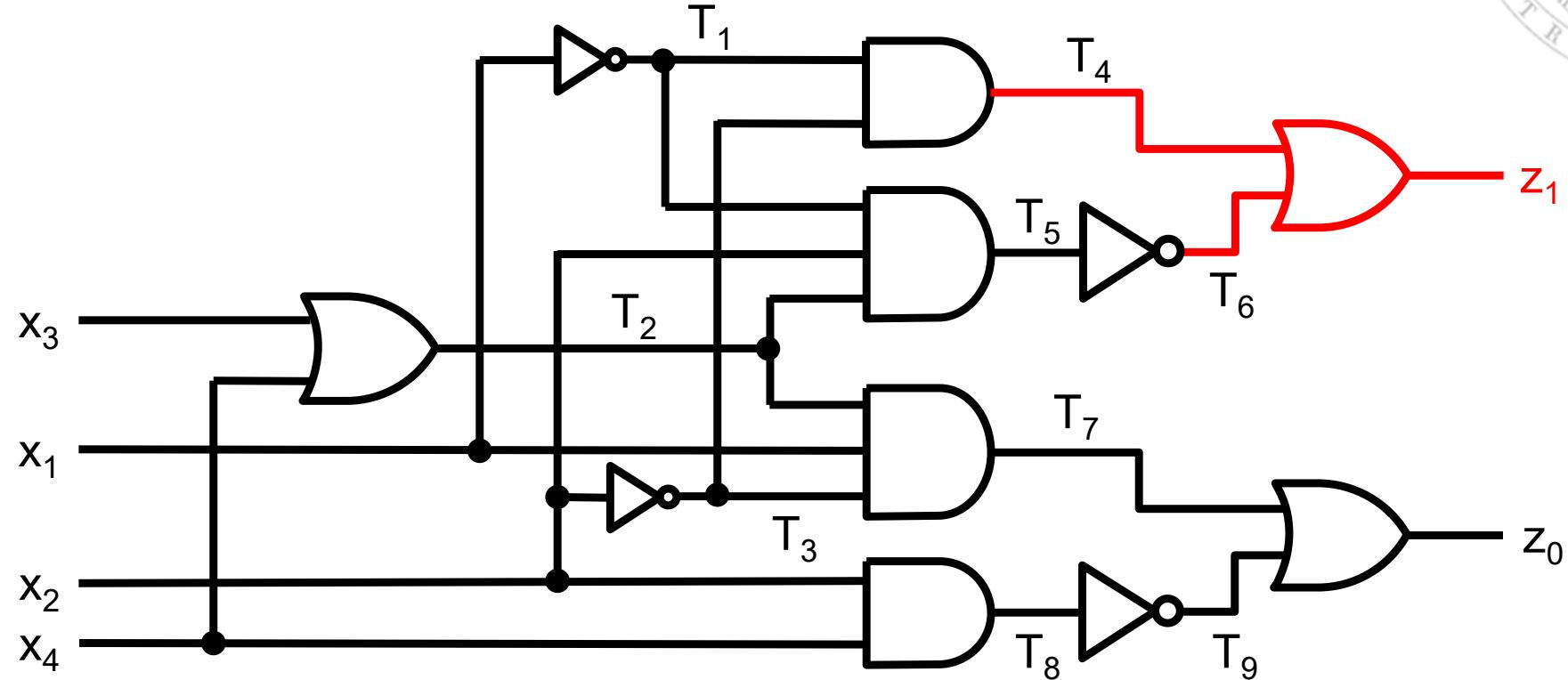
$$T_6 = \bar{T}_5 = \bar{x}_1 x_2 (x_3 + x_4)$$

$$T_9 = \bar{T}_8 = \bar{x}_2 x_4$$

$$z_0 = T_7 + T_9 = (x_3 + x_4) x_1 \bar{x}_2 + \bar{x}_2 x_4$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_2 = x_3 + x_4$$

$$T_3 = \overline{x_2}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$

$$T_8 = x_2 x_4$$

$$T_6 = \overline{T_5} = \overline{\overline{x_1} x_2 (x_3 + x_4)}$$

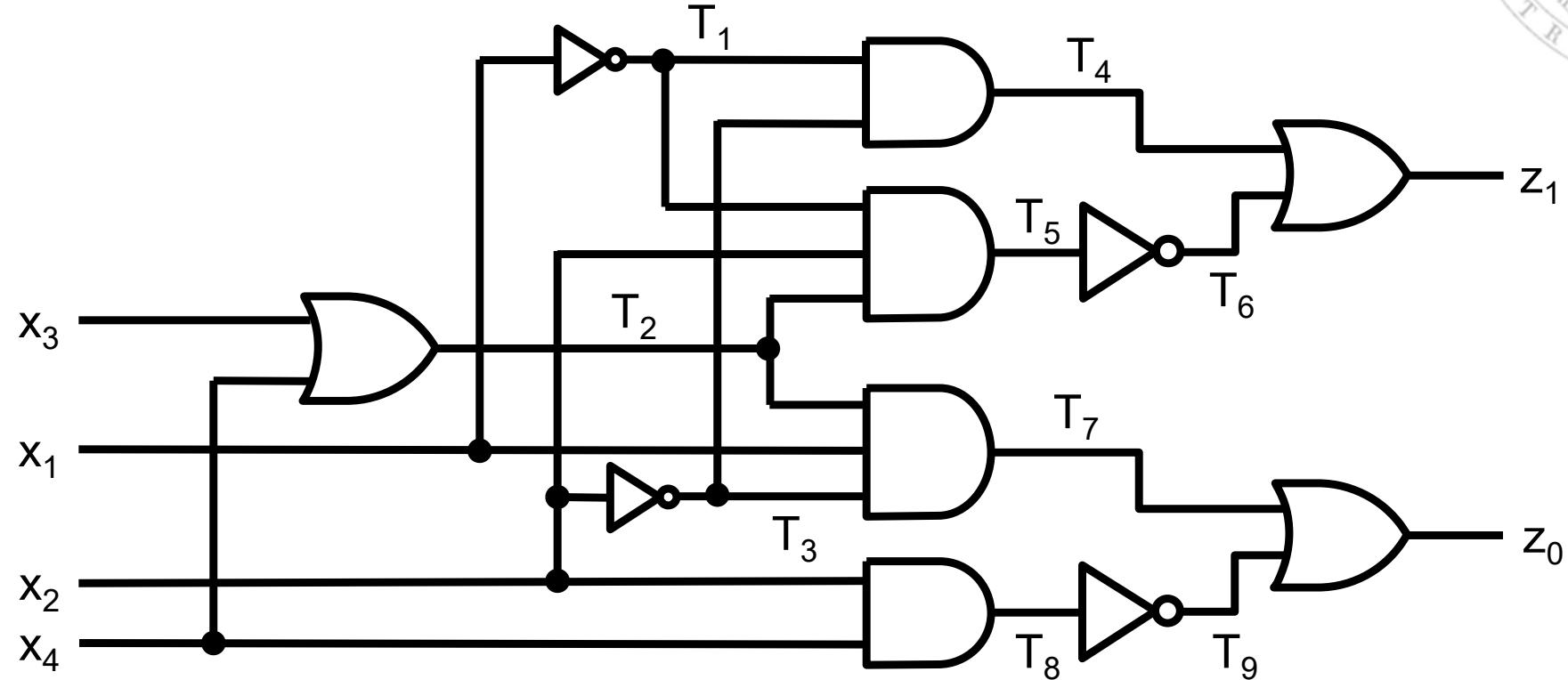
$$T_9 = \overline{T_8} = \overline{x_2 x_4}$$

$$z_0 = T_7 + T_9 = (x_3 + x_4) x_1 \overline{x_2} + \overline{x_2} x_4$$

$$z_1 = T_4 + T_6 = \overline{x_1} \overline{x_2} + \overline{x_1} x_2 (x_3 + x_4)$$



# Análisis de redes AND-OR



$$T_1 = \overline{x_1}$$

$$T_2 = x_3 + x_4$$

$$T_3 = \overline{x_2}$$

$$T_4 = T_1 T_3 = \overline{x_1} \overline{x_2}$$

$$T_5 = T_1 x_2 T_2 = \overline{x_1} x_2 (x_3 + x_4)$$

$$T_7 = T_2 x_1 T_3 = (x_3 + x_4) x_1 \overline{x_2}$$

$$T_8 = x_2 x_4$$

$$T_6 = \overline{T_5} = \overline{x_1} x_2 (x_3 + x_4)$$

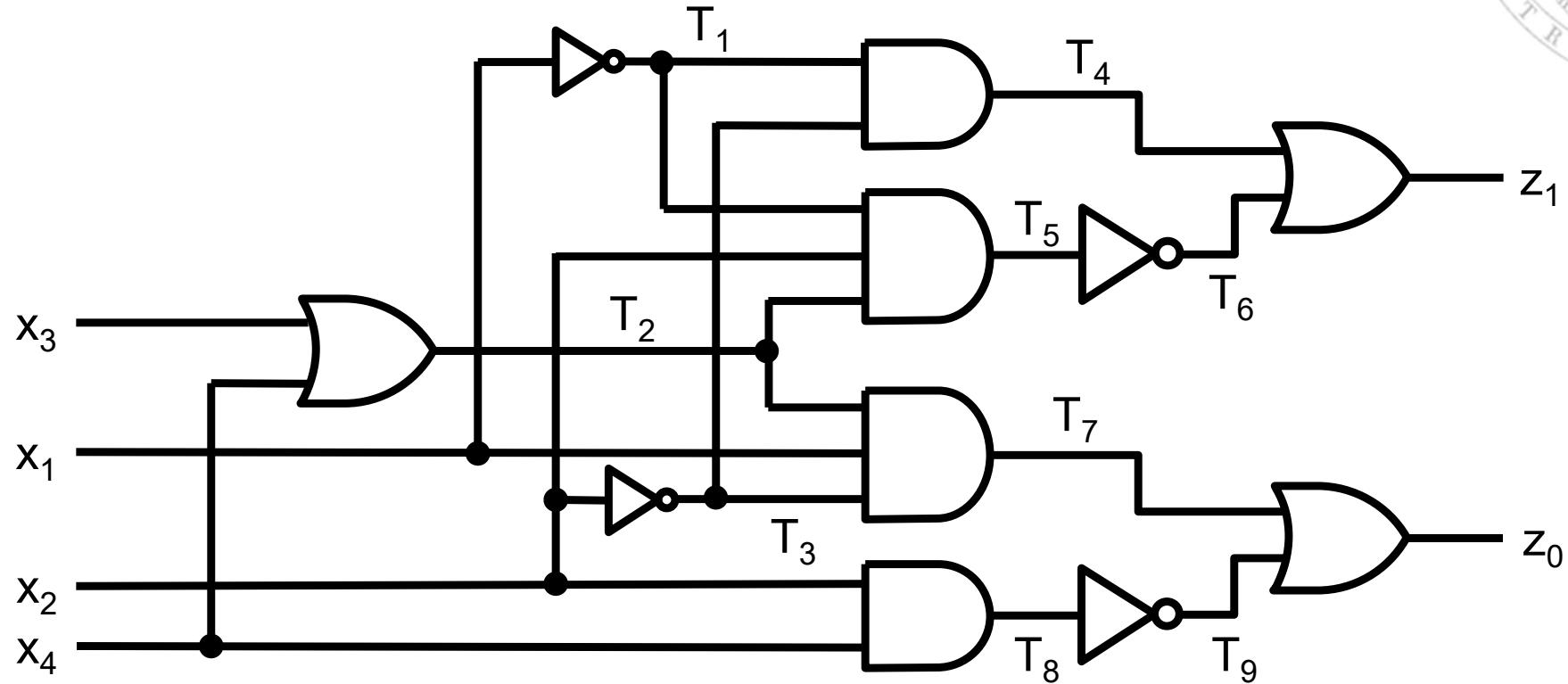
$$T_9 = \overline{T_8} = \overline{x_2 x_4}$$

$$z_0 = T_7 + T_9 = (x_3 + x_4) x_1 \overline{x_2} + \overline{x_2 x_4}$$

$$z_1 = T_4 + T_6 = \overline{x_1} \overline{x_2} + \overline{x_1} x_2 (x_3 + x_4)$$



# Análisis de redes AND-OR



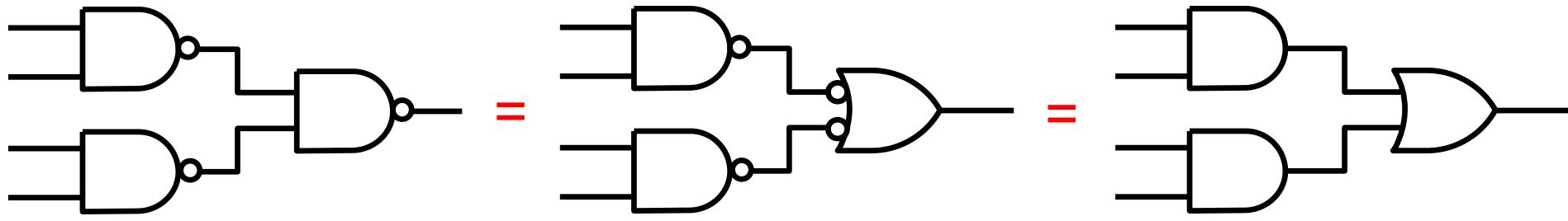
$$z_0 = \overline{x_2} + \overline{x_4}$$

$$z_1 = x_1 + \overline{x_2} + \overline{x_3} \overline{x_4}$$

# Análisis de redes NAND

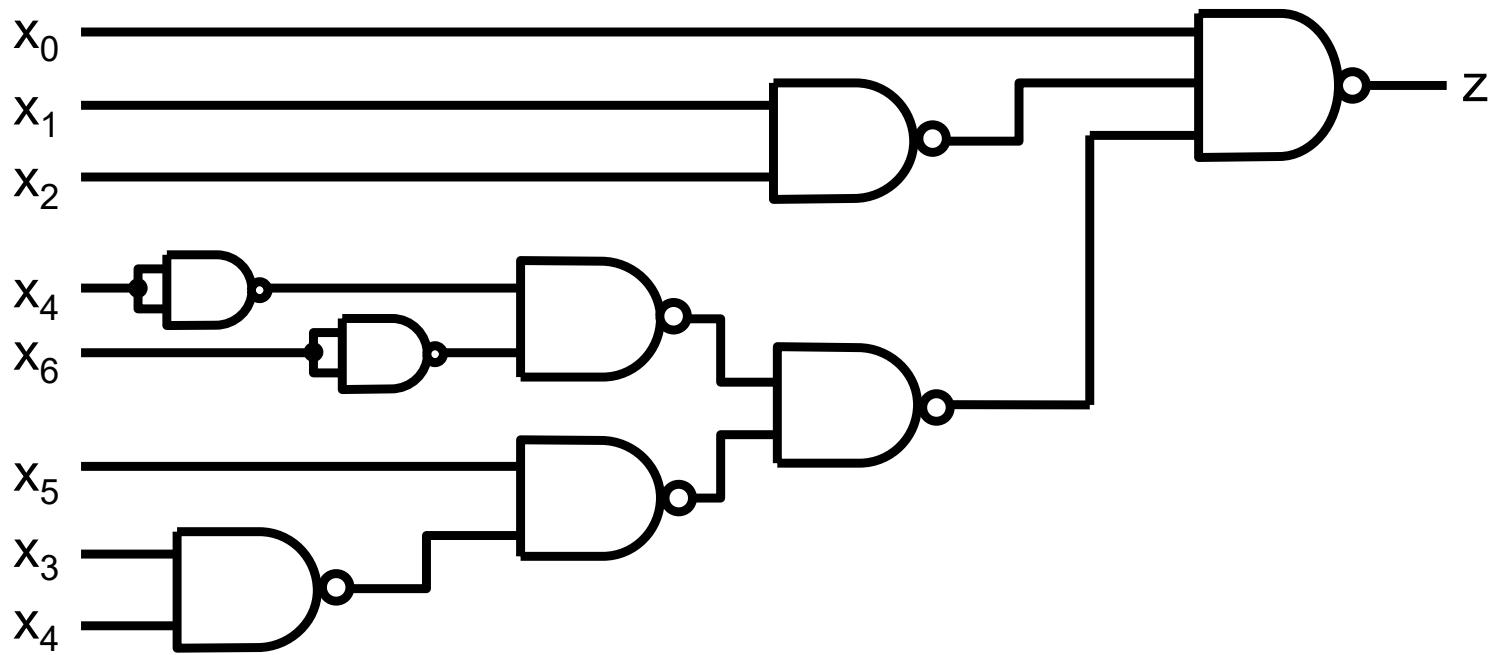


- 2 niveles NAND-NAND equivalen a 2 niveles AND-OR

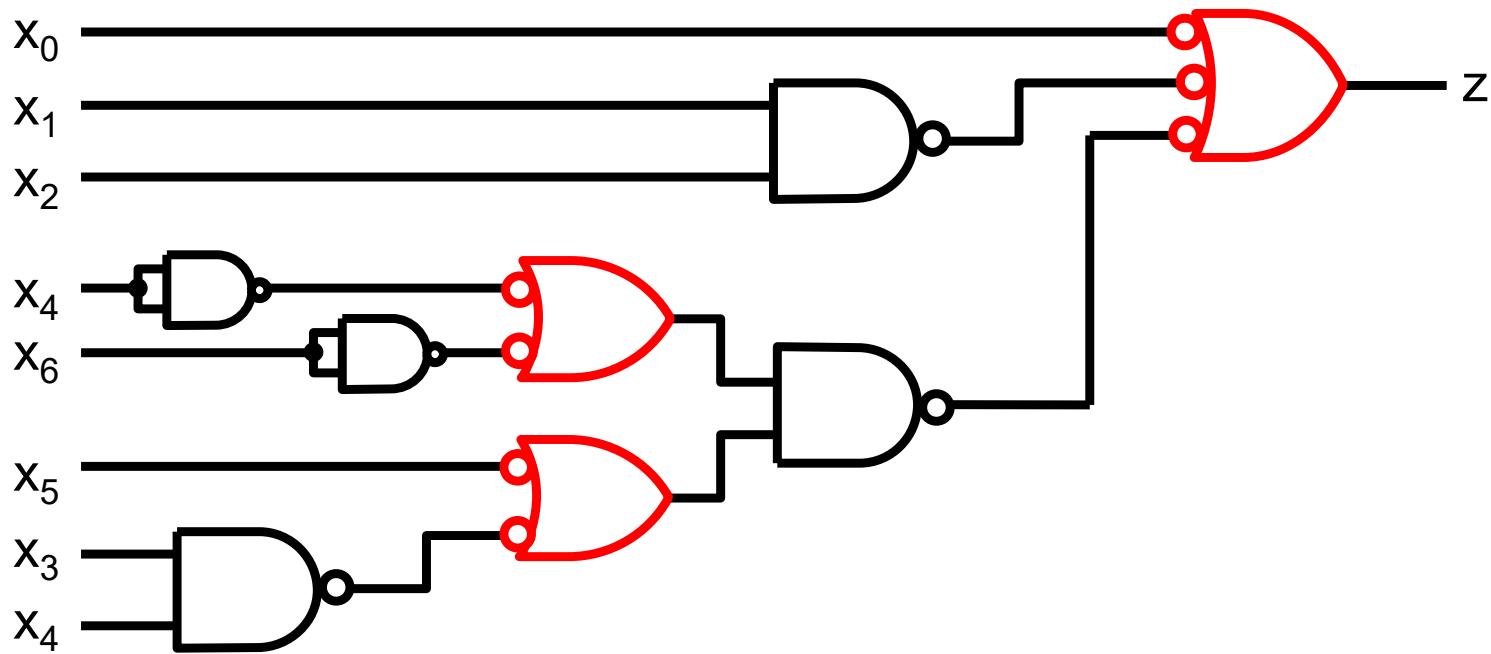


- Método:
  - Cambiar al símbolo alternativo las puertas NAND de los niveles pares de la red.
  - Eliminar dobles inversores donde sea posible.
  - Analizar la red AND-OR normalmente.

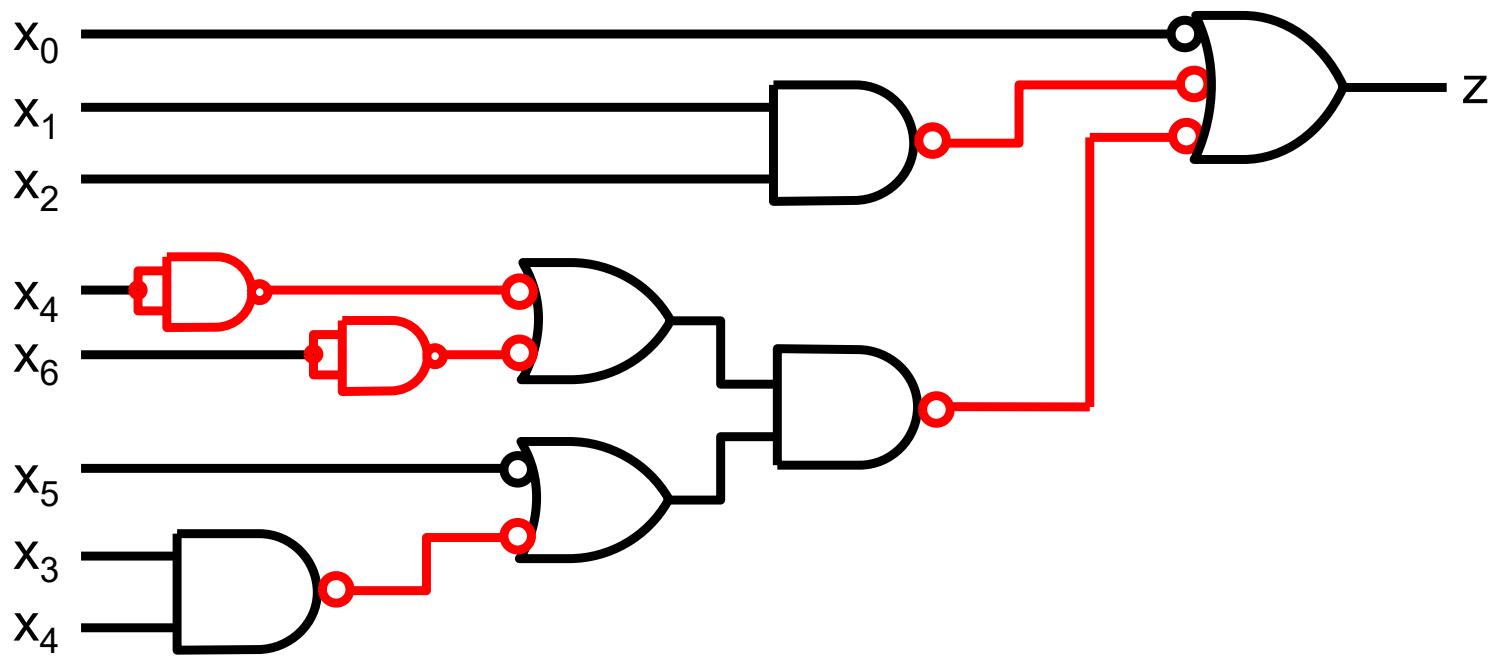
# Análisis de redes NAND



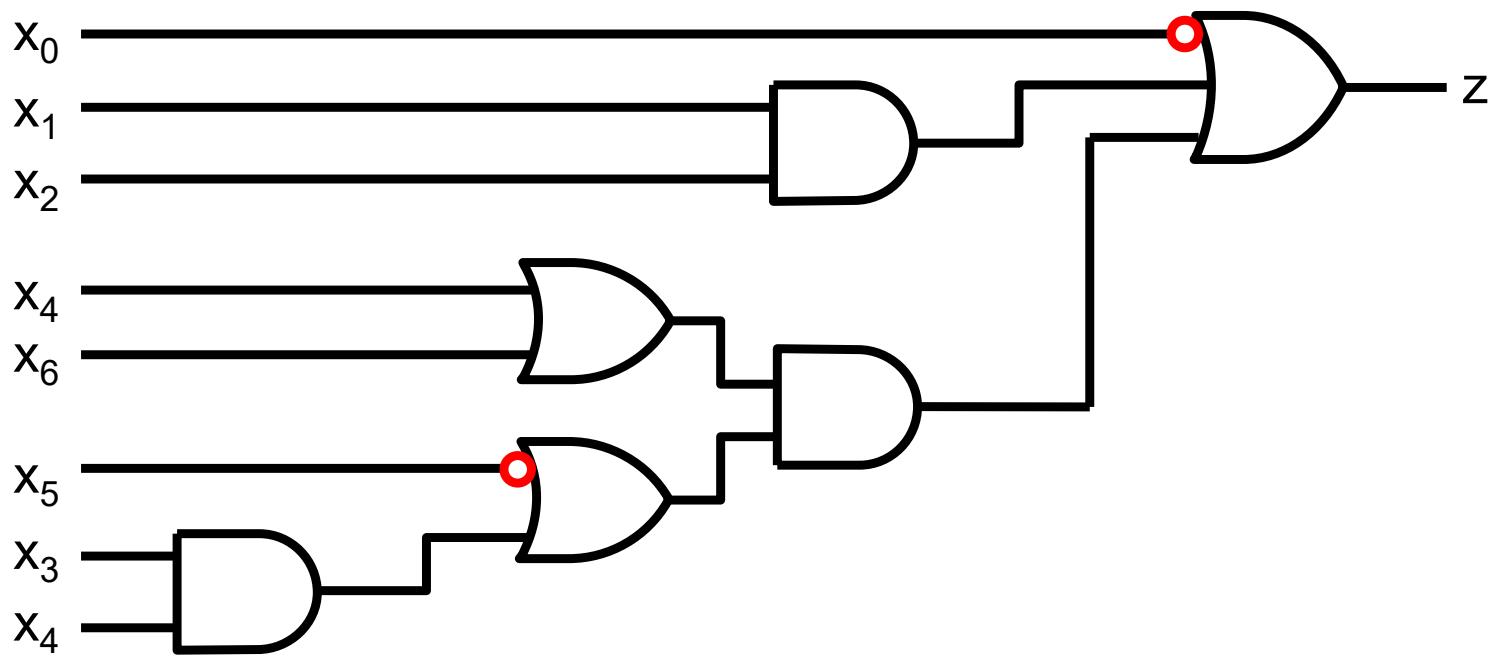
# Análisis de redes NAND



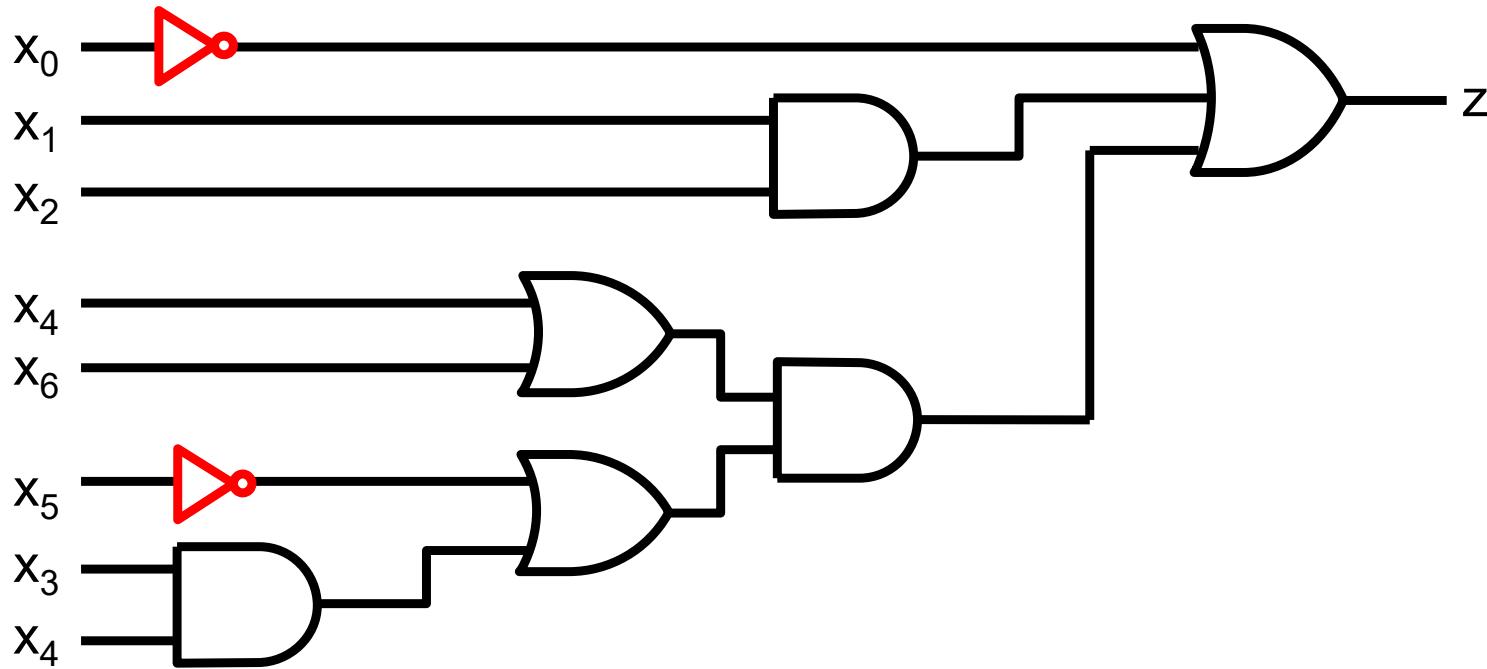
# Análisis de redes NAND



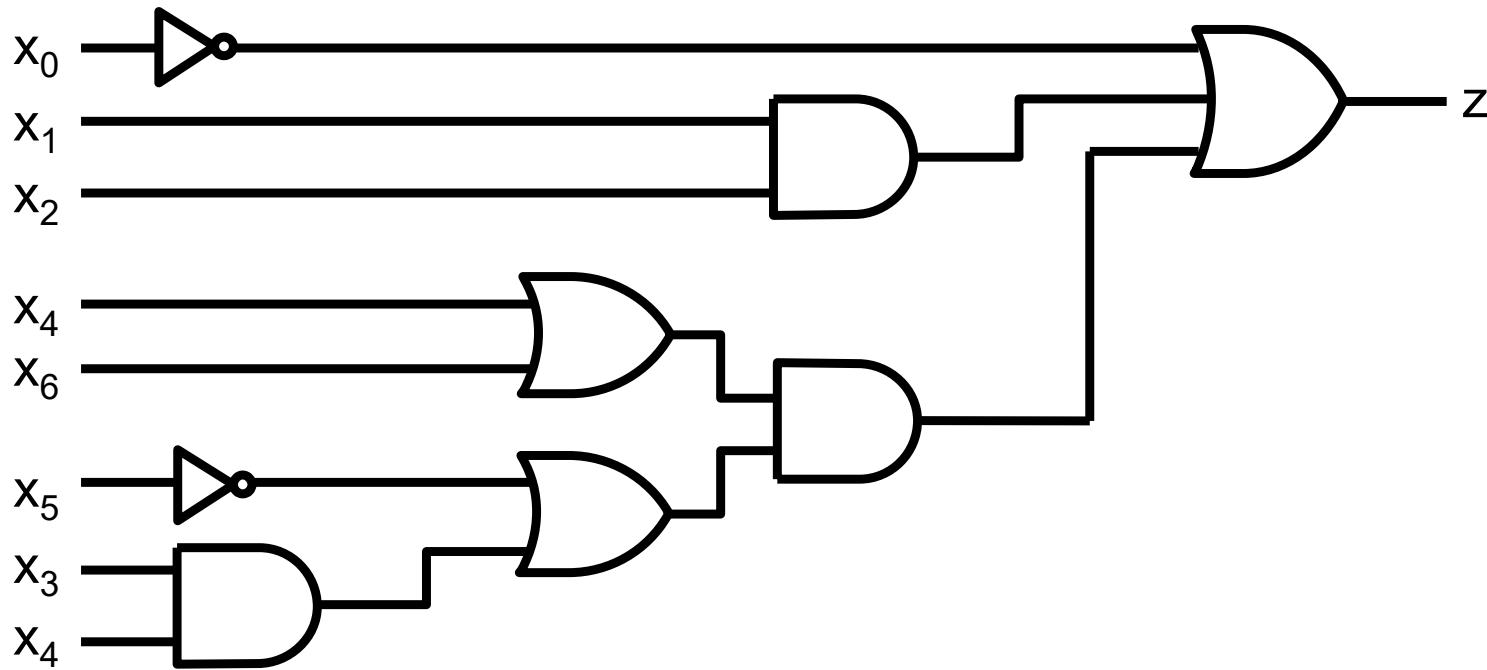
# Análisis de redes NAND



# Análisis de redes NAND



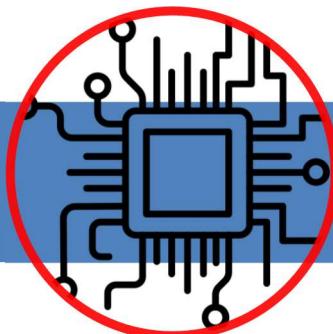
# Análisis de redes NAND



$$Z = \overline{x_0} + x_1 x_2 + (x_4 + x_6)(\overline{x_5} + x_3 x_4)$$

- Alimentación
- Señalización.
- Tecnología CMOS.
- Retardo, consumo y coste.
- Fabricación.
- Biblioteca de celdas.
- Evolución tecnológica.
- Entrada/salida elemental.
- Trade-offs.

## Apéndice tecnológico





# Alimentación

- Los circuitos digitales funcionan con **corriente continua**.
  - Los voltajes de alimentación más comunes son **+5V**, **+3.3V** y **+1.8V**
  - Dado que **todas y cada una** de las puertas deben conectarse a alimentación y tierra, dichas conexiones **suelen obviarse**

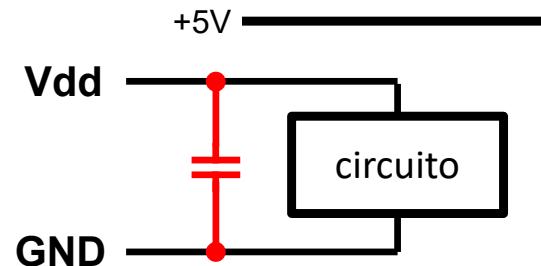
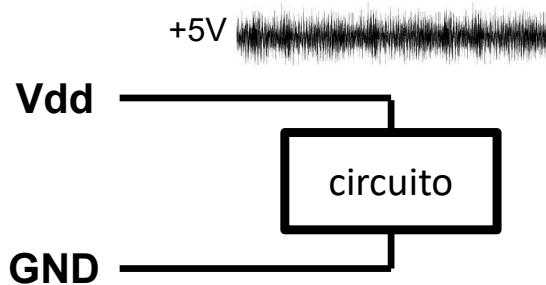


*símbolos de tierra*



*símbolos de alimentación*

- Se usan **condensadores de desacoplo** para asegurar la estabilidad del voltaje de alimentación en picos de consumo o en presencia de ruido



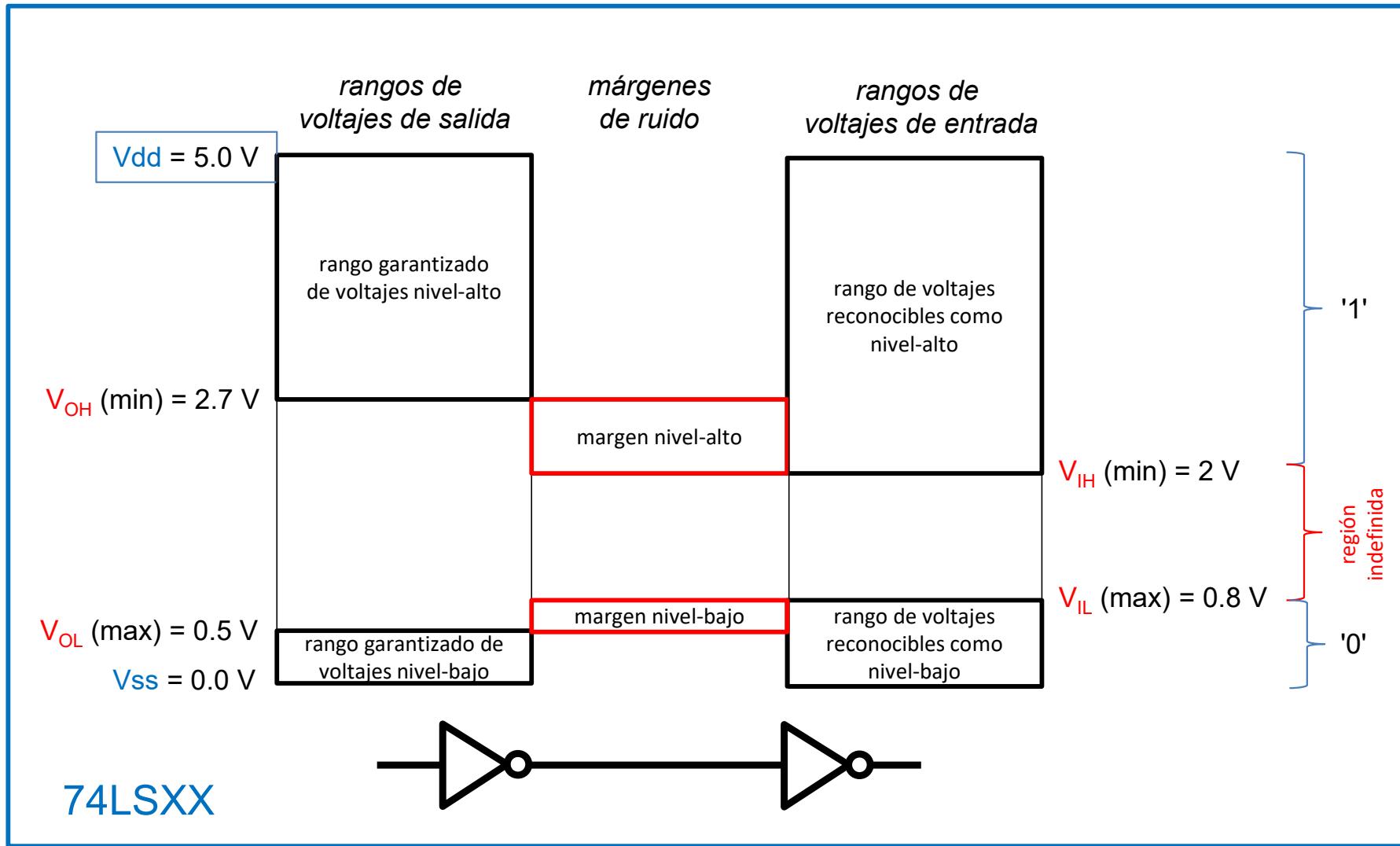


# Señalización

- **Niveles lógicos:** el comportamiento de un sistema digital electrónico se expresa en términos de niveles de tensión medidos en voltios.
  - Físicamente las puertas se implementan para que solo puedan generar 2 niveles de tensión: alta (H) y baja (L)
    - Los **sistemas lógicos positivos** asignan el valor '0' a 'L' y '1' a 'H'
    - Los **sistemas lógicos negativos** asignan el valor '1' a L y '0' a 'H'
  - CMOS y TTL son sistemas de lógica positiva.
- **Márgenes de ruido:** las puertas se diseñan para tolerar variaciones en los voltajes de las entradas/salidas.
  - El máximo nivel se denomina margen de ruido.



# Señalización TTL



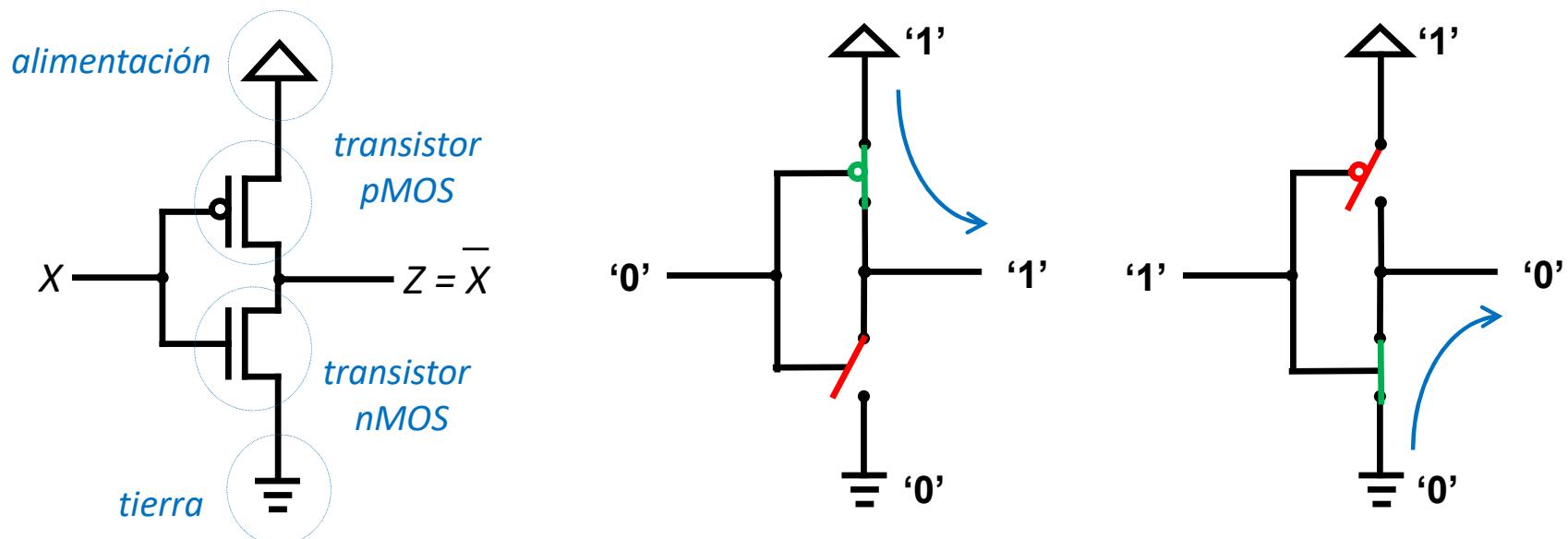


# Tecnología CMOS

## Inversor

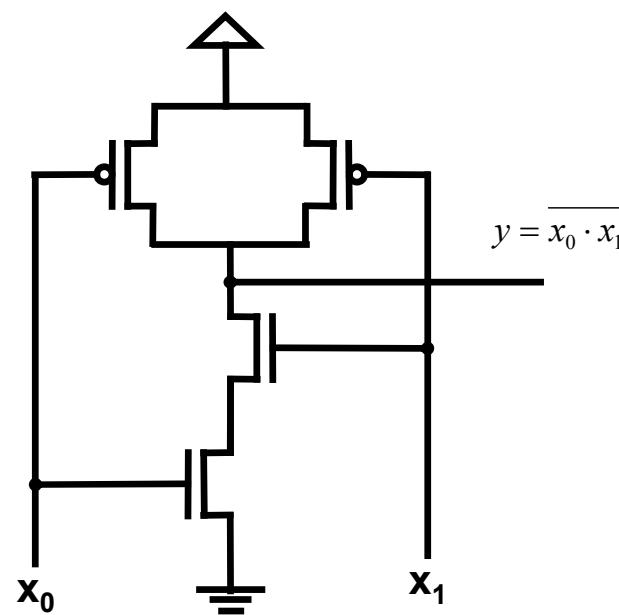
- Un **inversor CMOS** está formado por un transistor **pmos** en serie con un transistor **nmos** con sus puertas unidas:

  - la entrada del inversor es la puerta común y la salida, el punto de unión de los transistores.
  - el transistor **pmos** (pull-up) tiene un terminal conectado a Vdd y se encarga de poner la salida a '1' cuando conduce (cuando la entrada vale '0').
  - el transistor **nmos** (pull-down) tiene un terminal conectado a Vss y se encarga de poner la salida a '0' cuando conduce (cuando la entrada vale '1').

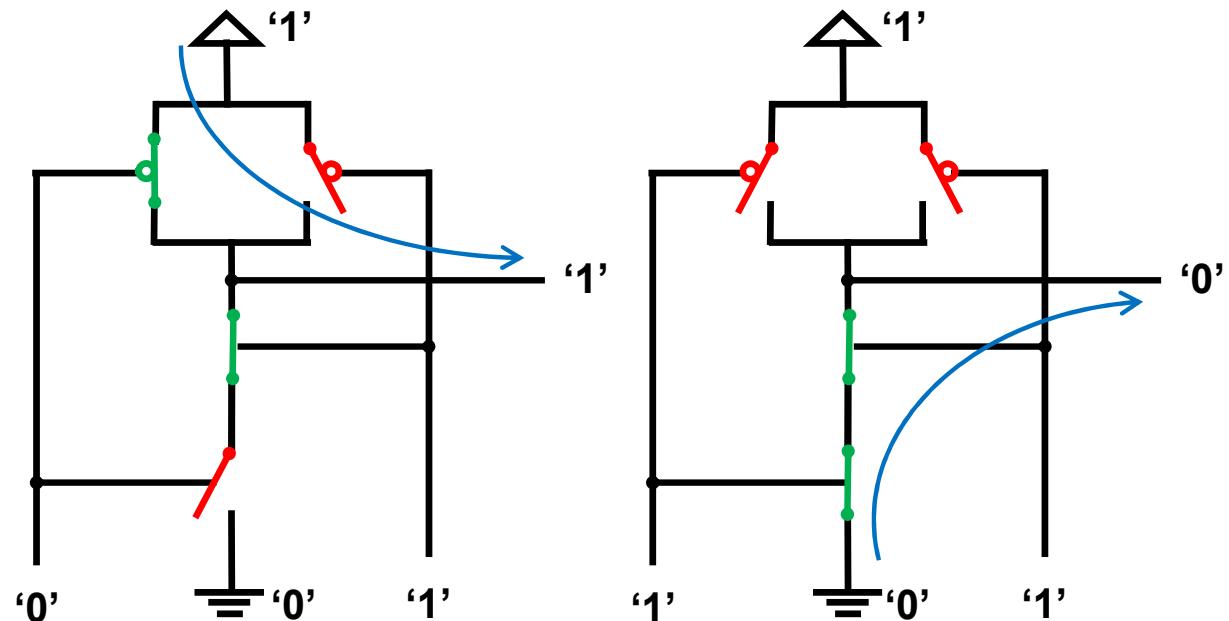


# Tecnología CMOS

## Puertas lógicas



Puerta NAND CMOS

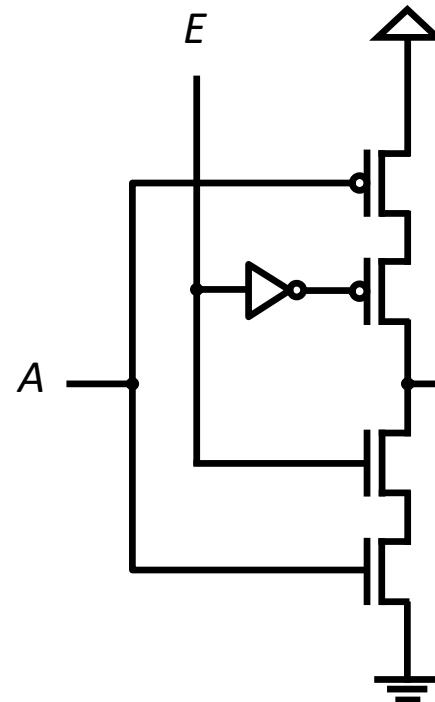




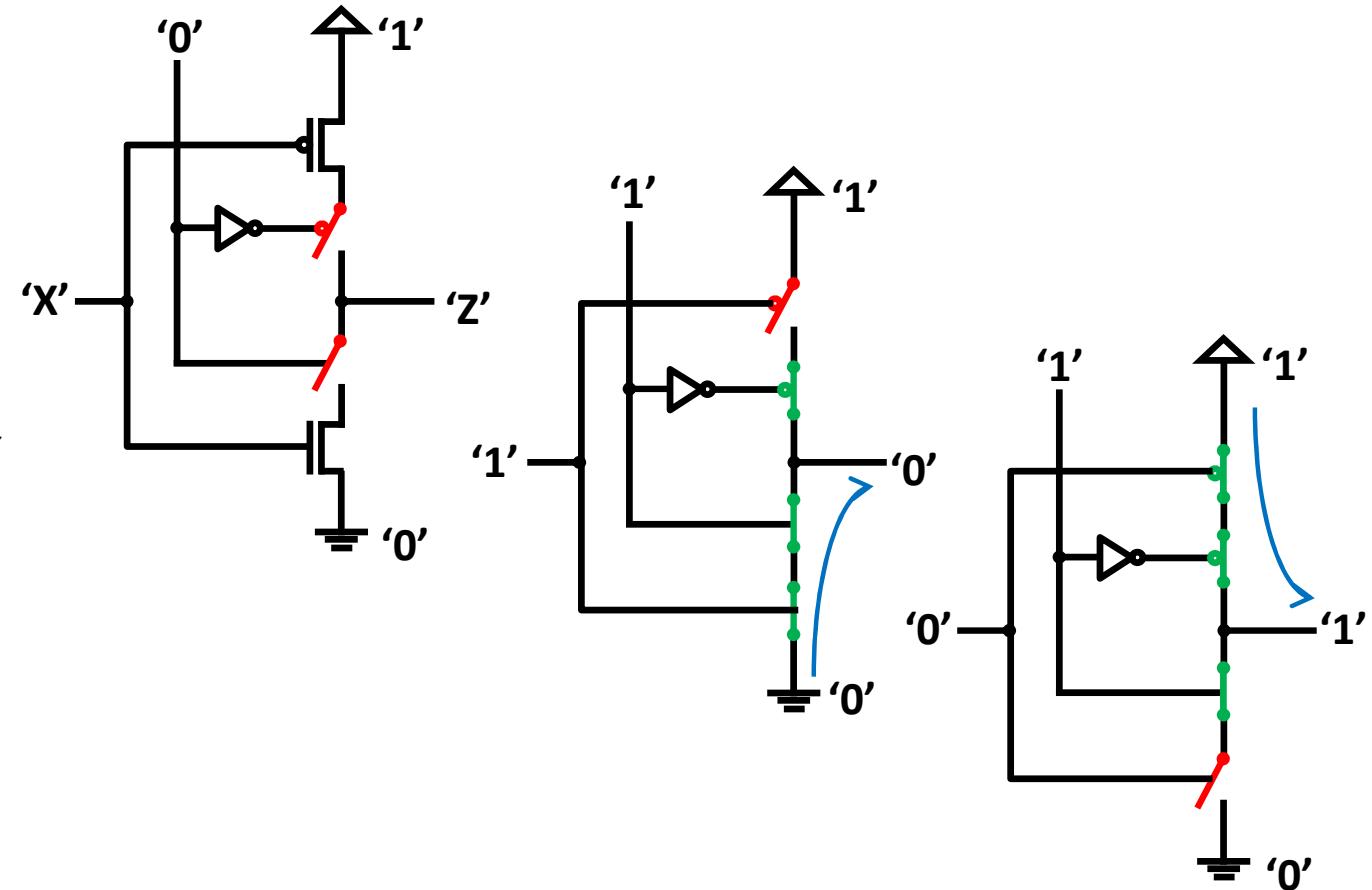
# Tecnología CMOS

## Buffer triestado

- Un **buffer triestado** se está formado dos transistores pmos en serie con dos transistores nmos y puertas:



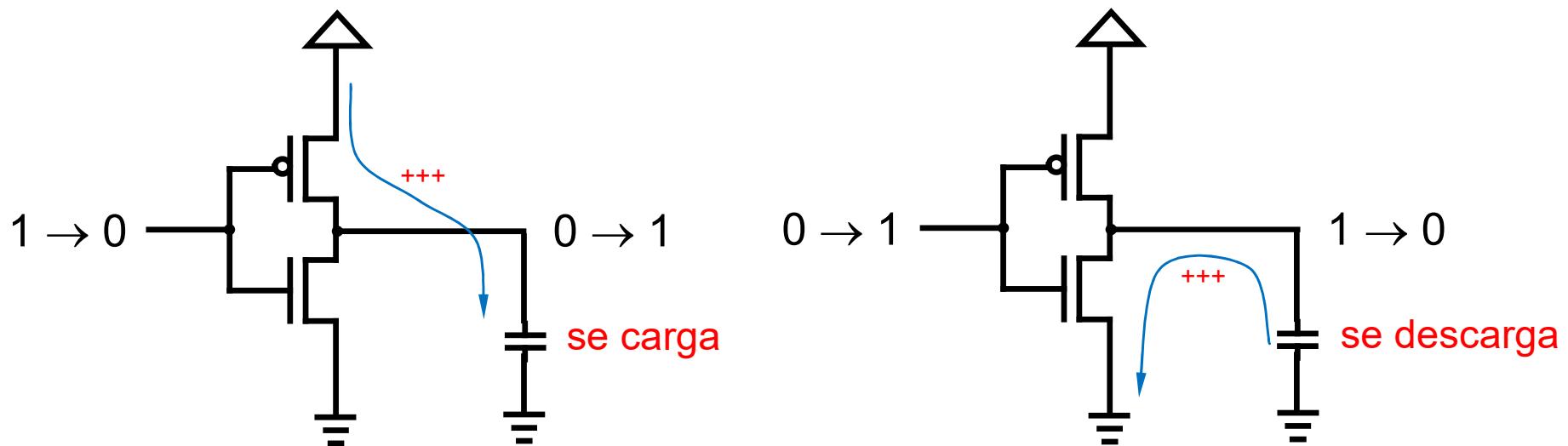
Buffer inversor  
triestado CMOS





# Retardo

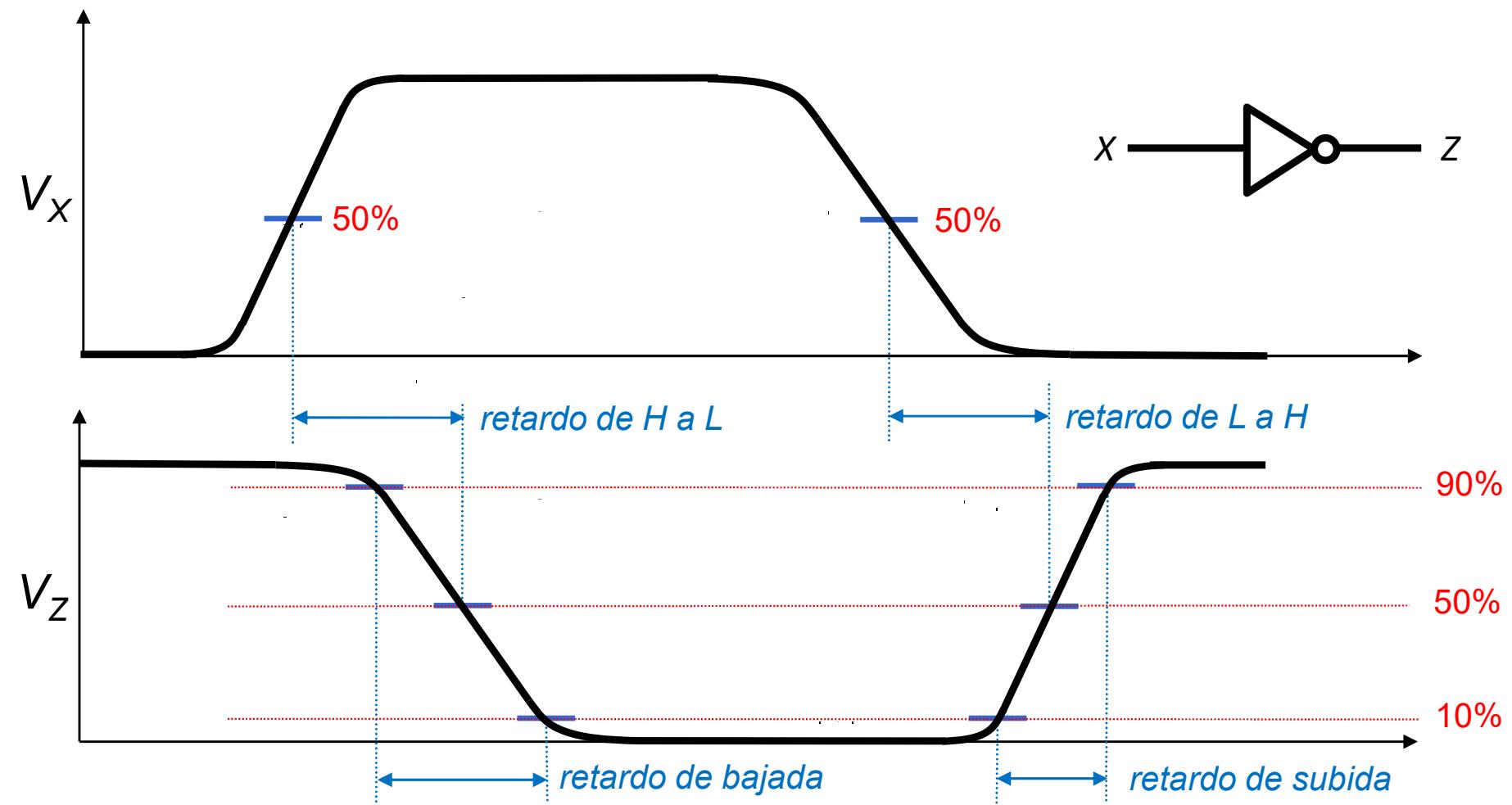
- El **retardo de conmutación** de una puerta es proporcional a su impedancia y a las capacidades parásitas de los elementos que estimula (otras puertas e interconexiones).
  - Mayor cuanto menos intensidad pueda circular por los transistores.
  - Mayor cuanto mayor sea la carga que tiene que circular.
    - más elementos conectados, interconexiones más largas, etc..





# Retardo

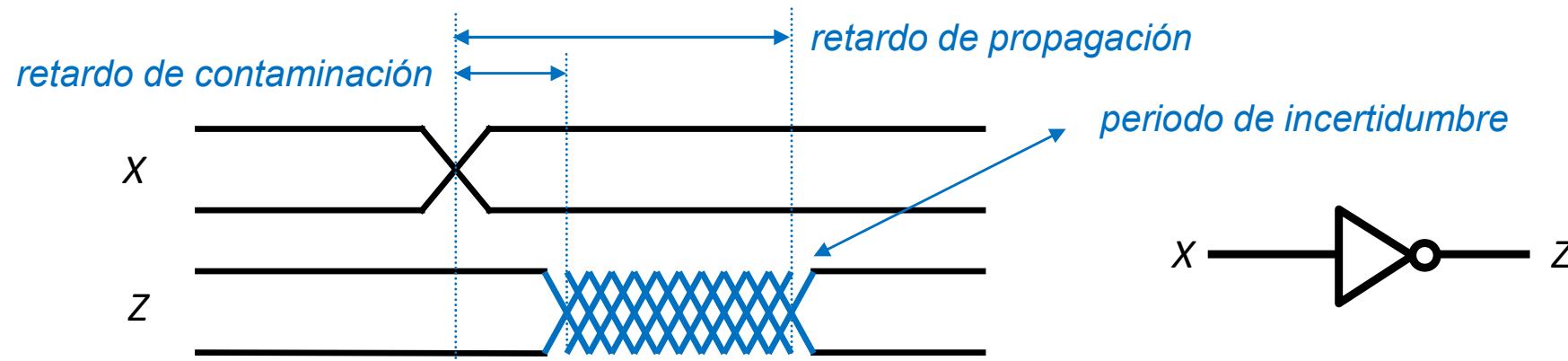
- En una conmutación, las señales no cambian de voltaje instantáneamente.





# Retardo

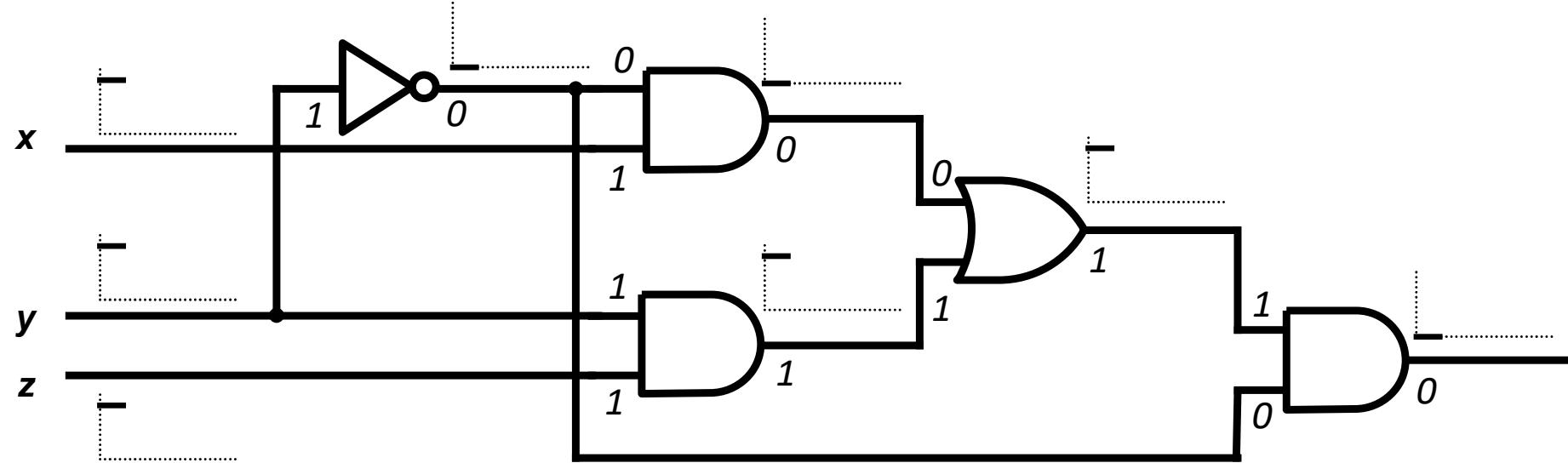
- El retardo es distinto según:
  - la entrada que provoca la conmutación de la salida.
  - la polaridad de la conmutación.
  - del valor del resto de entradas.
  - factores ambientales.
- El máximo se denomina **retardo de propagación** y el mínimo, **retardo de contaminación**.
  - Periodo de incertidumbre: intervalo comprendido entre ambos



# Retardo



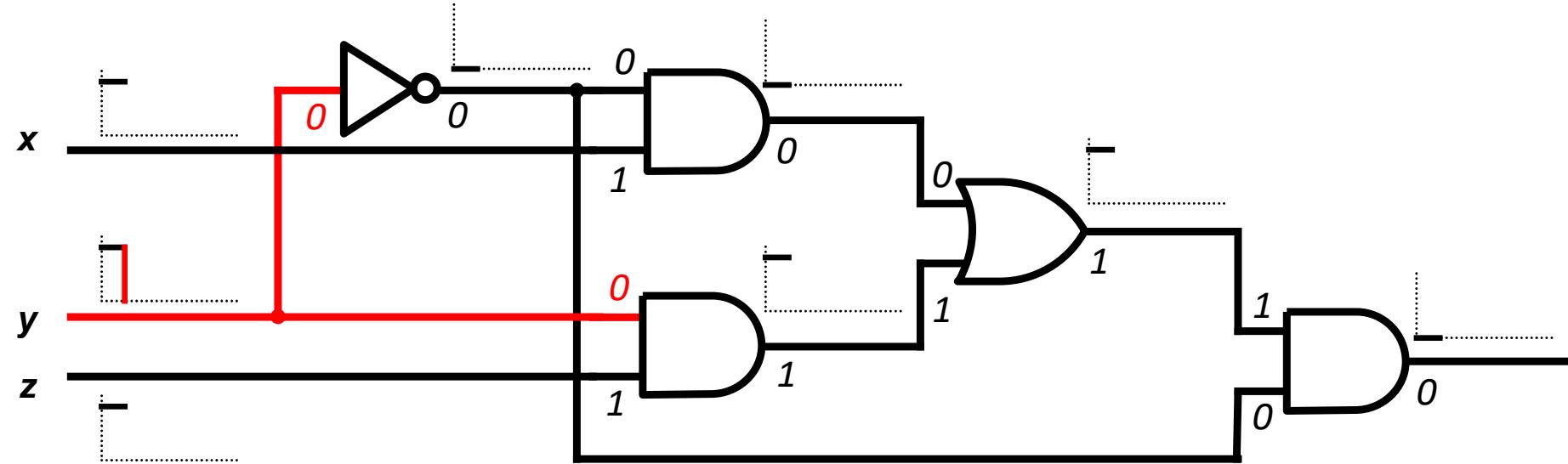
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.



# Retardo



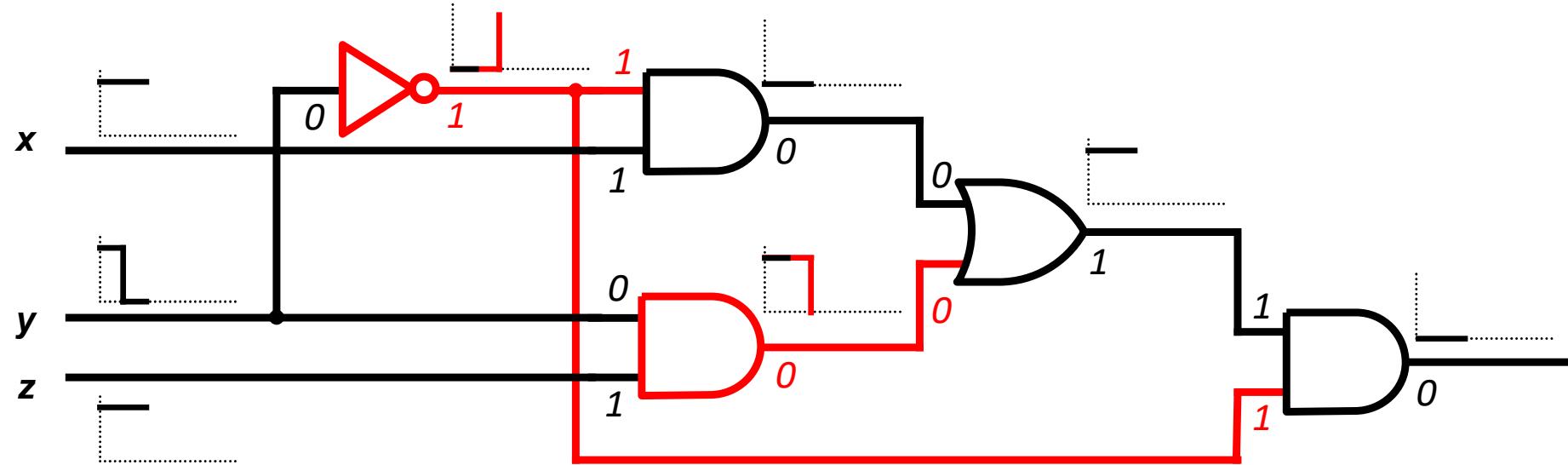
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.





# Retardo

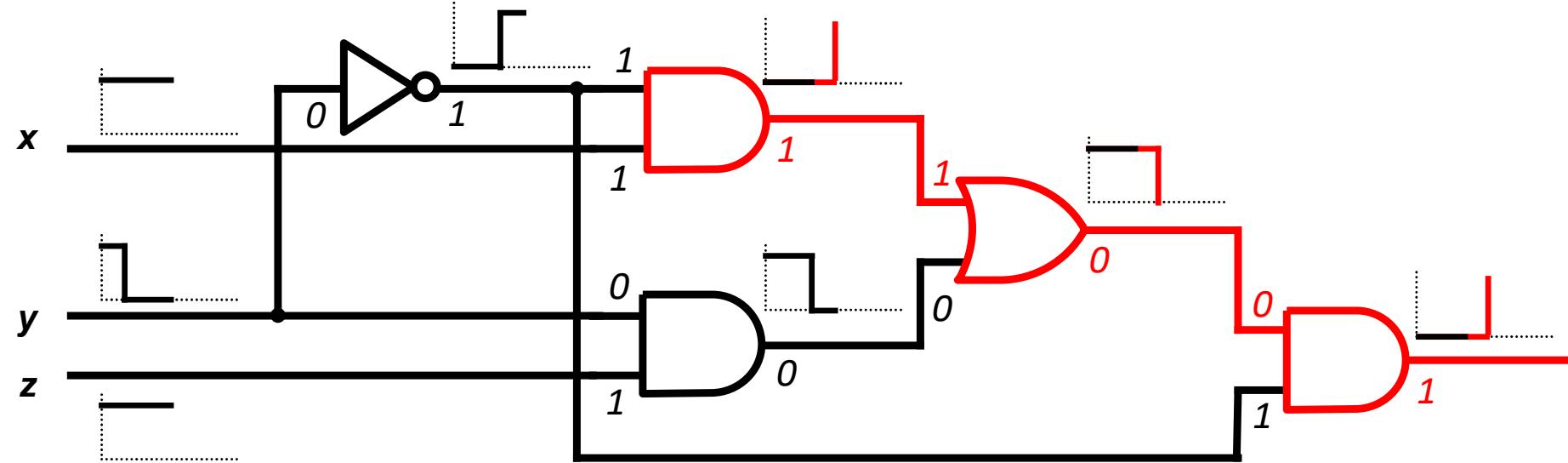
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.



# Retardo



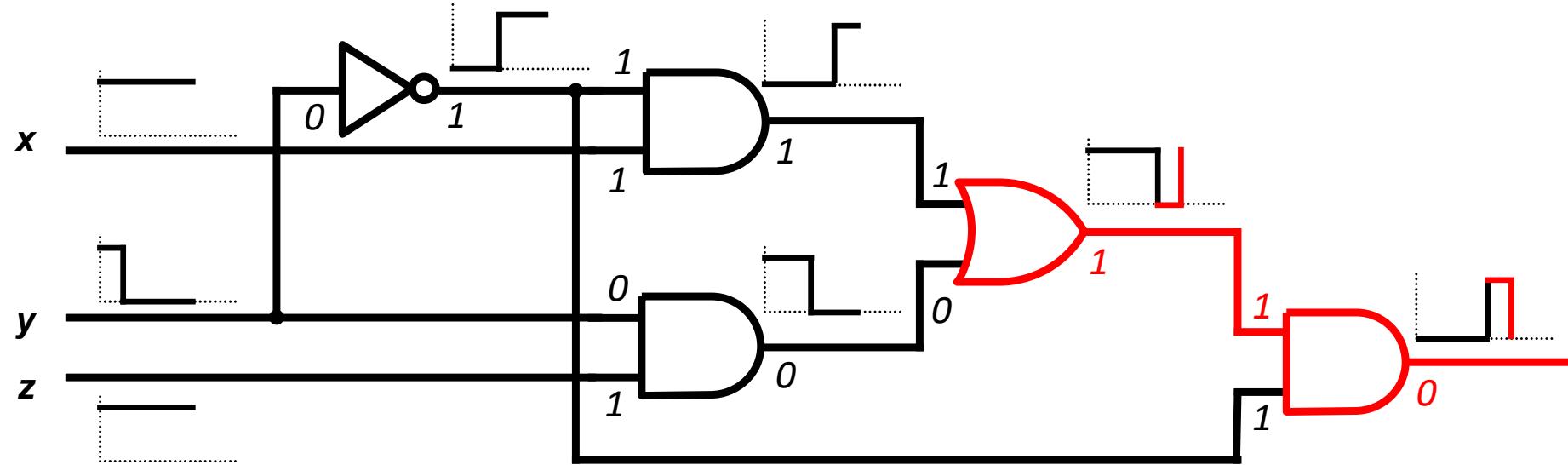
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.



# Retardo



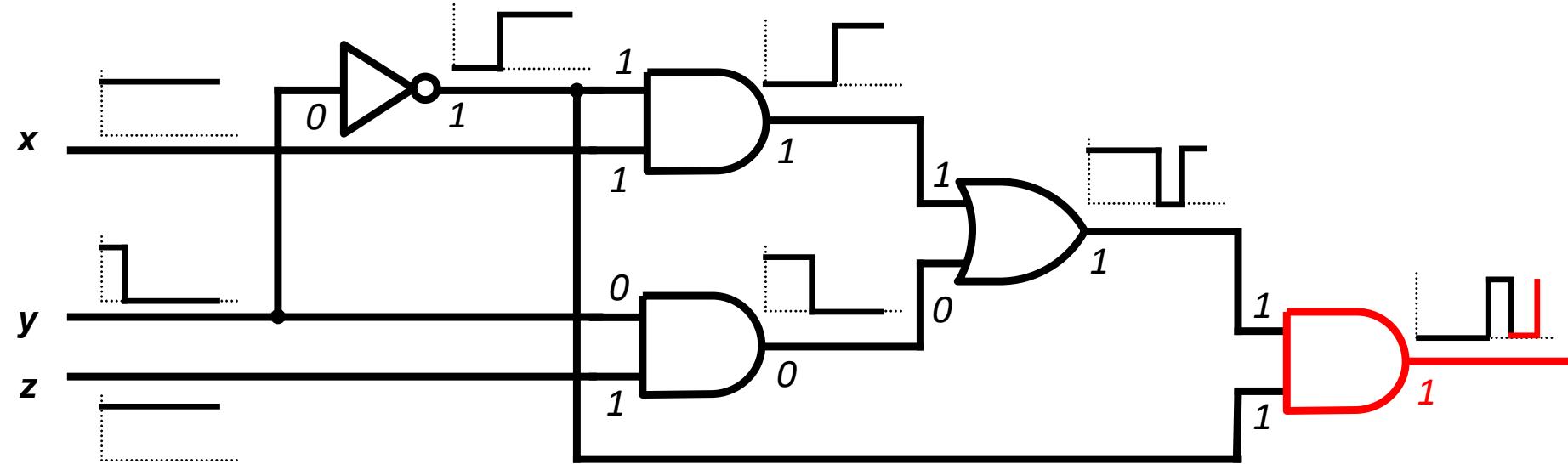
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.



# Retardo



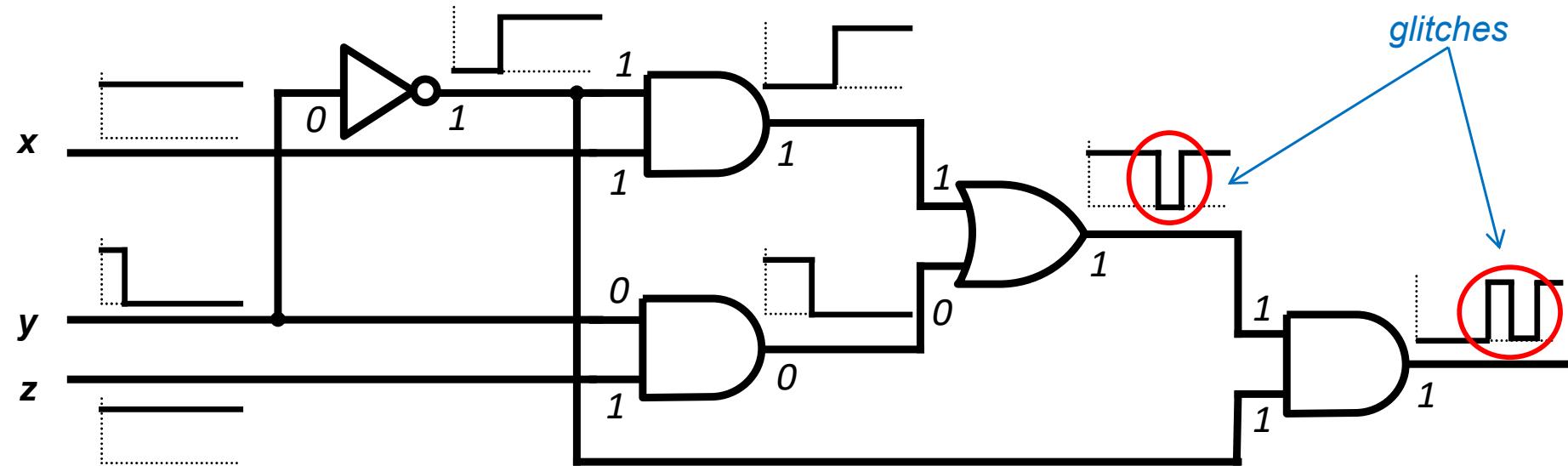
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.





# Retardo

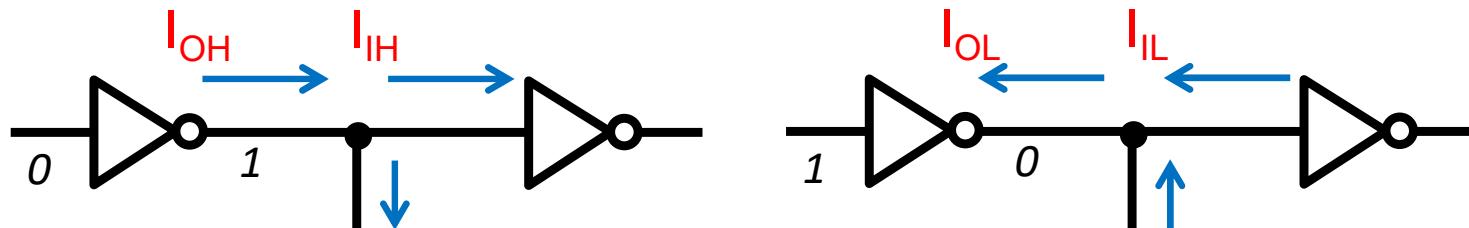
- La salida solo cambia dentro del periodo de incertidumbre y puede tomar valores inconsistentes con la FC/EC que implementa.
  - Fuera de ese periodo las salidas permanecen estables.





# Conectividad

- **Conectividad de salida (fan-out):** indica el número máximo de puertas que pueden conectarse a una salida sin degradar los niveles de tensión
  - cociente entre la intensidad máxima que puede suministrar/absorber un puerto de salida y la intensidad máxima que absorbe/suministra un puerto de entrada.
    - 74LSXX (TTL) = 20
    - CMOS teóricamente ilimitado (impedancia de entrada muy alta), pero mucha conectividad degrada el retardo de conmutación.





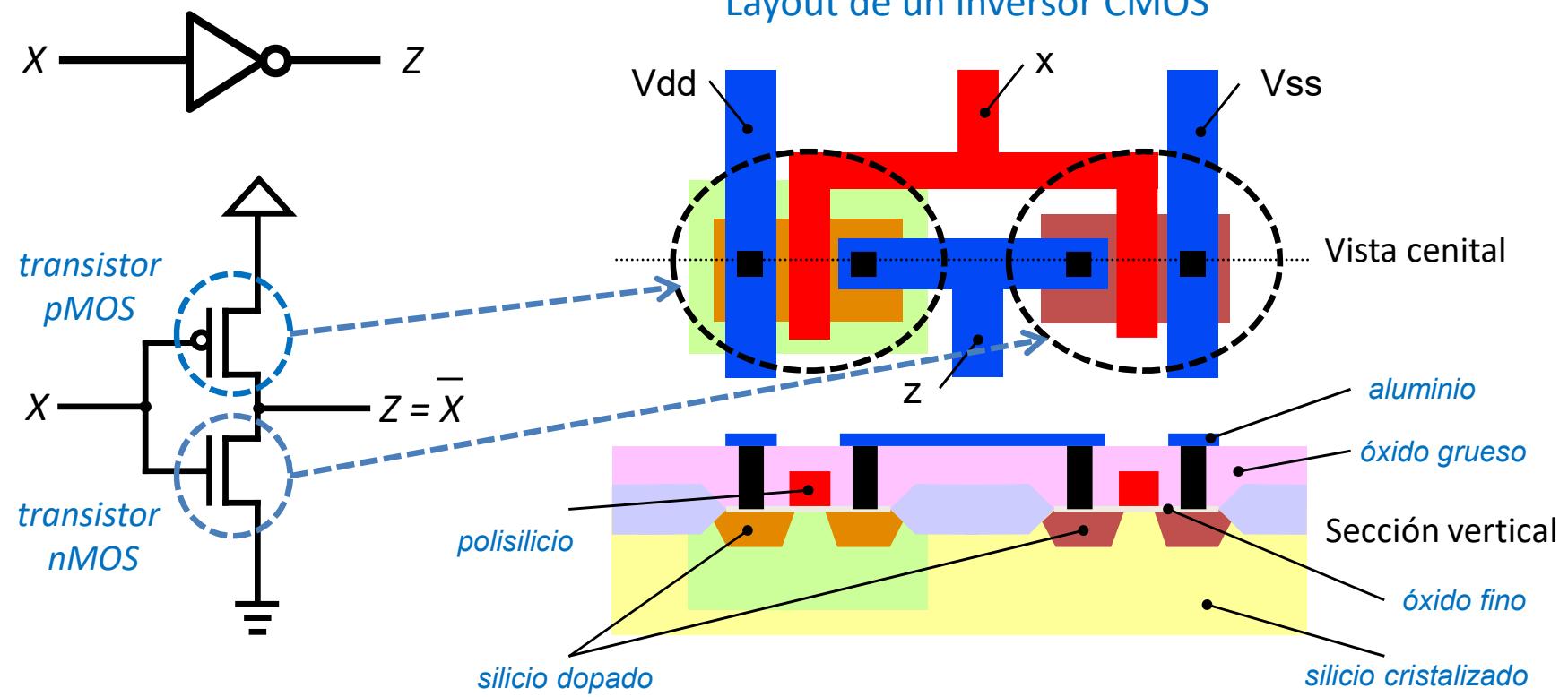
# Consumo

- El **consumo de una puerta** depende de
  - la cantidad de corriente que por ella circule
  - el voltaje de alimentación
- **Consumo estático:** ocurre con el dispositivo en régimen permanente (conectado pero en reposo)
  - En tecnología CMOS es muy pequeño
- **Consumo dinámico:** ocurre con el dispositivo en régimen transitorio (realizando cálculos)
  - **Por conmutación:** al cargar y descargar los condensadores de carga.
  - **Por cortocircuito:** a través del camino directo que momentáneamente existe entre alimentación y tierra cada vez que el dispositivo conmuta
    - los transistores no son conmutadores perfectos.
    - las señales no cambian instantáneamente.



# Coste

- El **coste** de un circuito depende del **área** que ocupe:
  - el área de las difusiones de los transistores.
  - el área del interconexiónado.

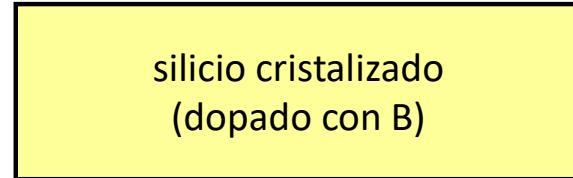




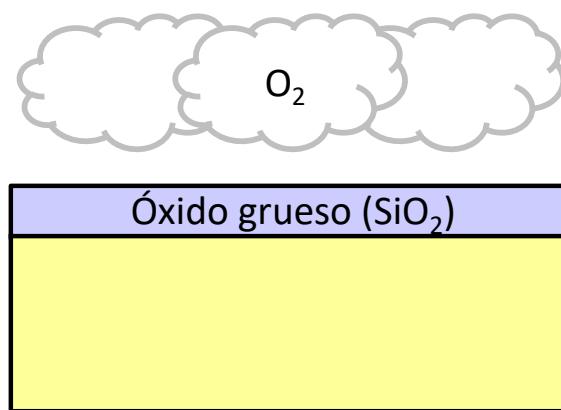
# Fabricación

## ■ El proceso de fabricación de un circuito VLSI:

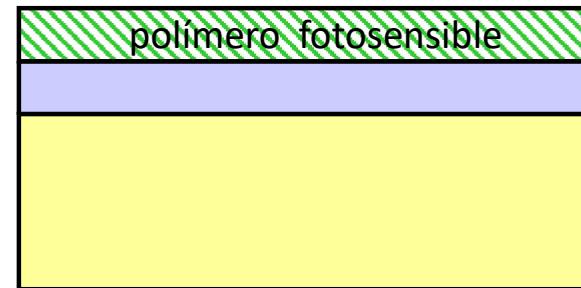
- Parte de un lingote (ingot) de silicio cristalizado.
- Que se corta en discos (wafer) que se pulen.
- Sobre el que se depositan distintos materiales mediante una serie de procesos **fotolitográficos**:
  - Difunden materiales sobre la oblea.
  - Protegen secciones de material **usando máscaras** trazadas por el diseñador.
  - Eliminan, dopan o metalizan las secciones no protegidas.



Wafer (tipo p)



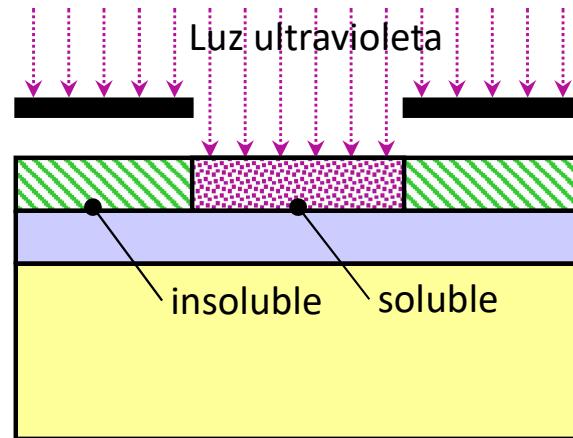
Oxide Growth



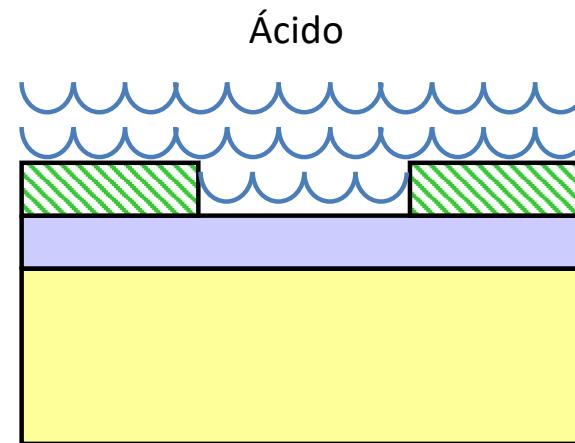
Photoresist Coating



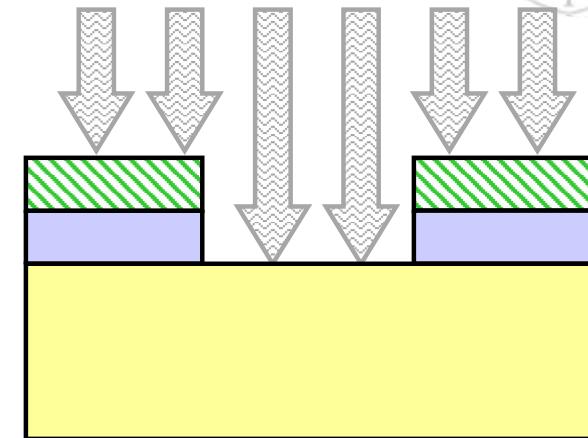
# Fabricación



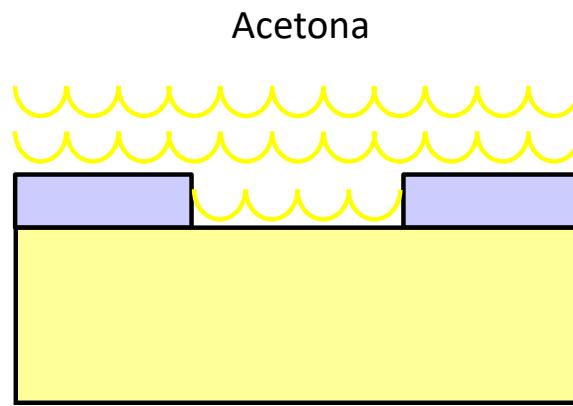
Photolithography (mask #1)



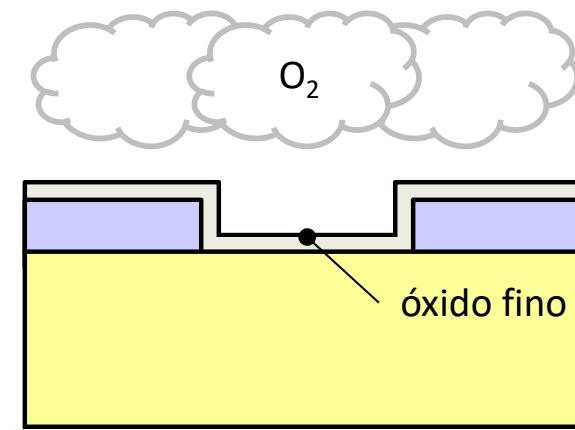
Soluble Photoresist Removing



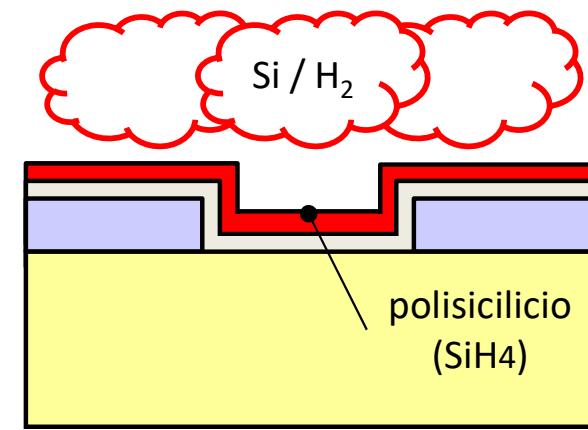
Chemical Etching



Photoresist Removing



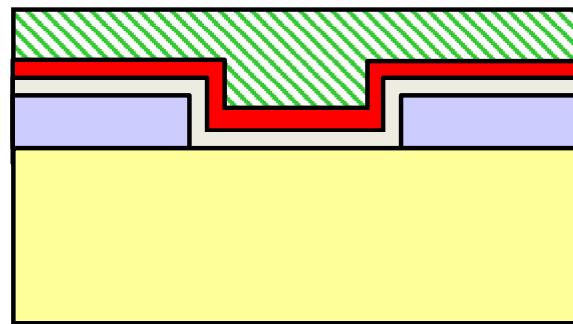
Oxide Growth



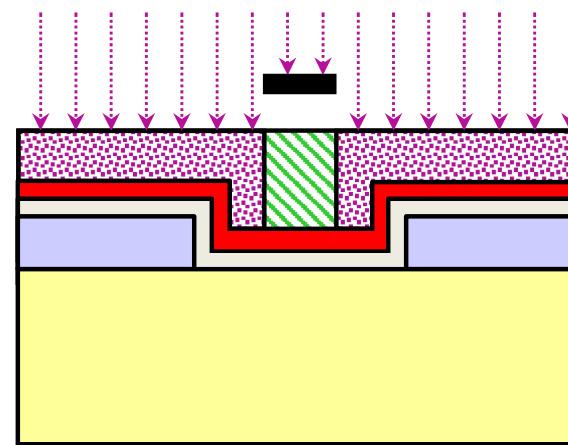
Chemical Vapor Deposition



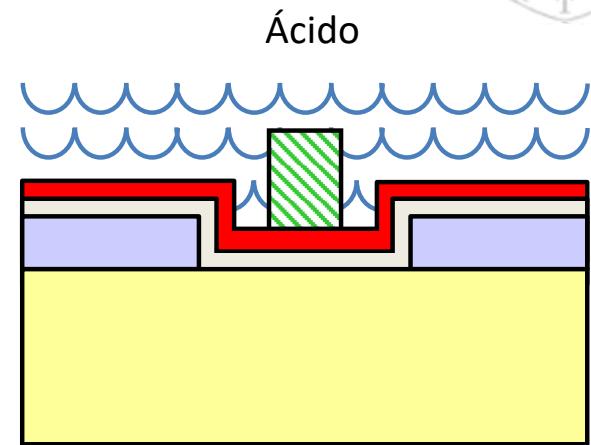
# Fabricación



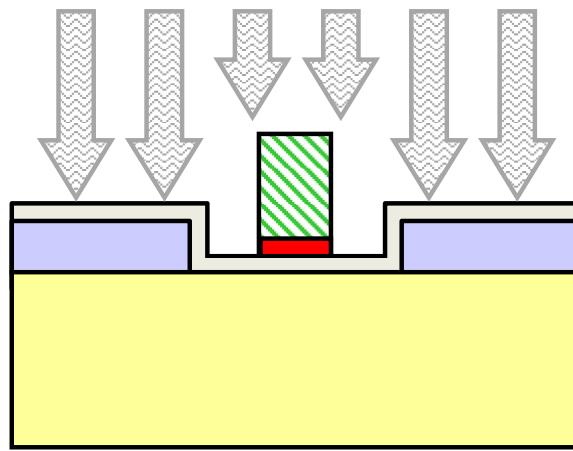
Photoresist Coating



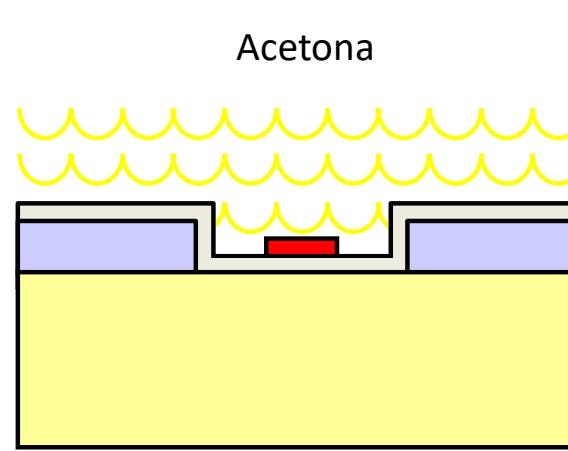
Photolithography (mask #2)



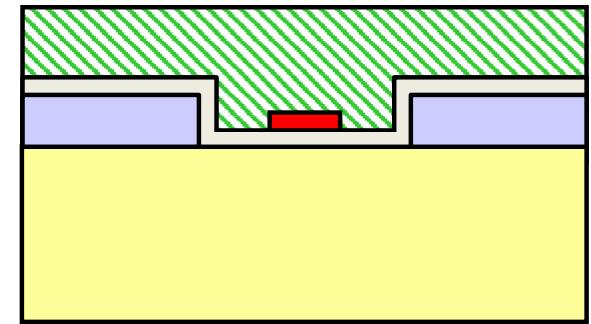
Soluble Photoresist Removing



Plasma Etching



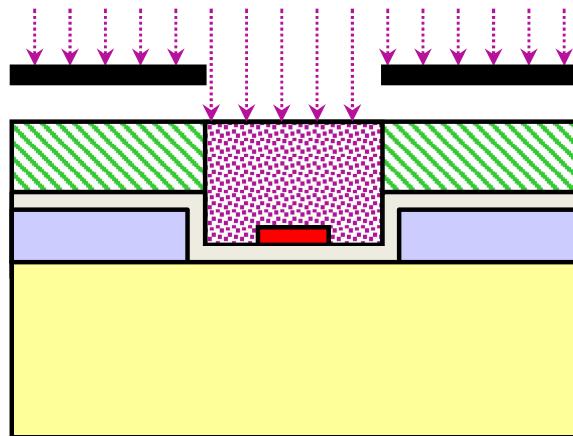
Photoresist Removing



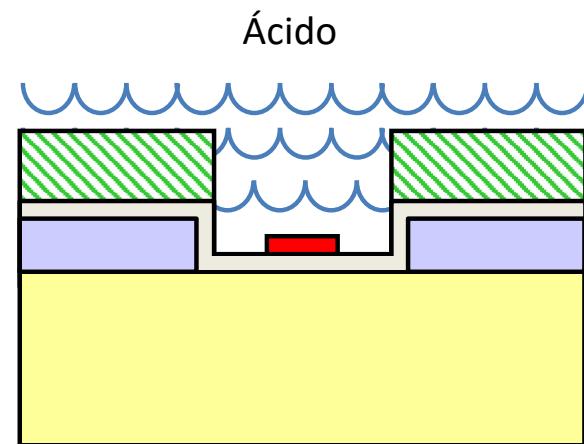
Photoresist Coating



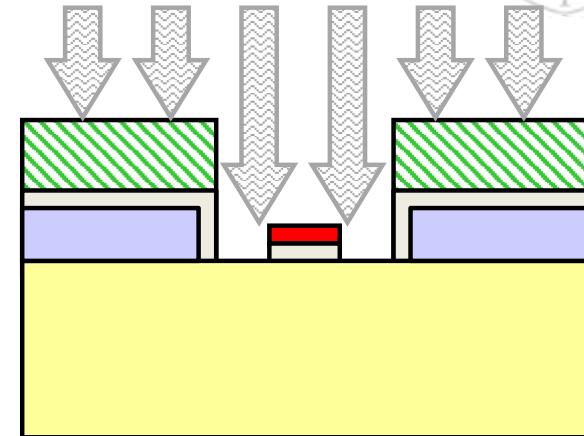
# Fabricación



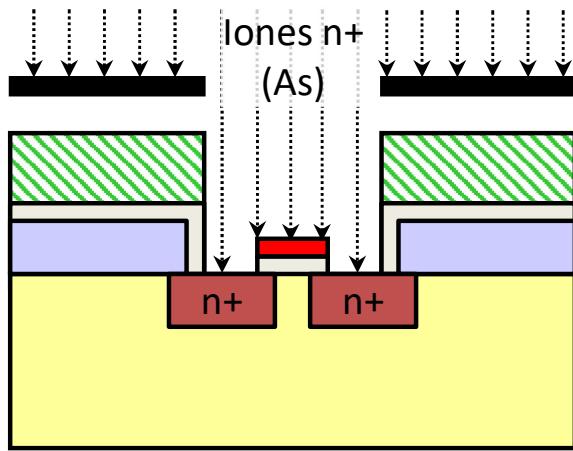
Photolithography (mask #3)



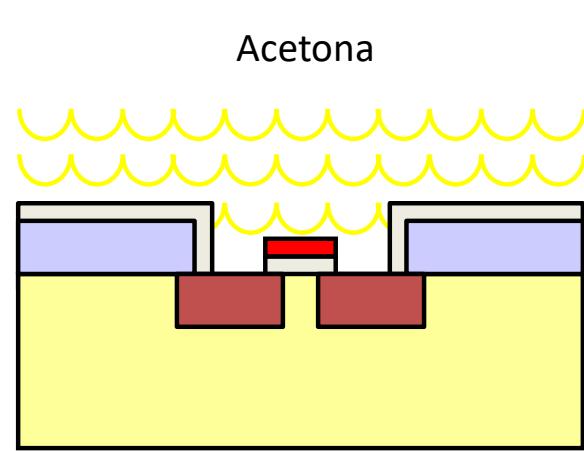
Soluble Photoresist Removing



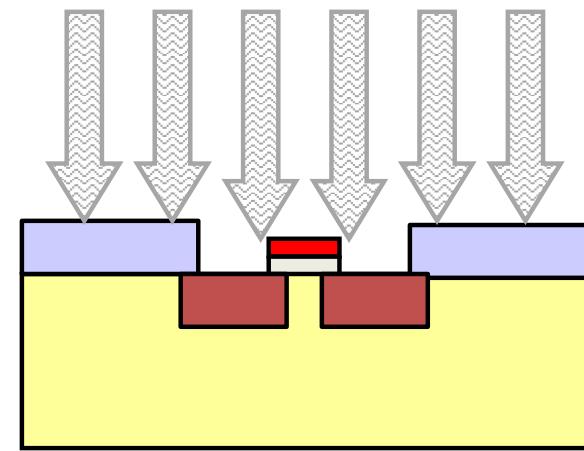
Chemical Etching



Ion Implantation (mask #4)



Photoresist Removing



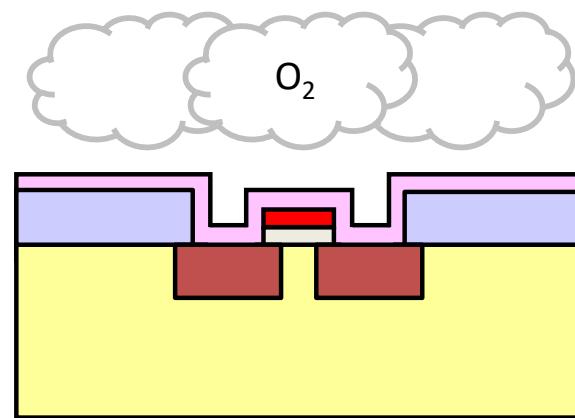
Chemical Etching



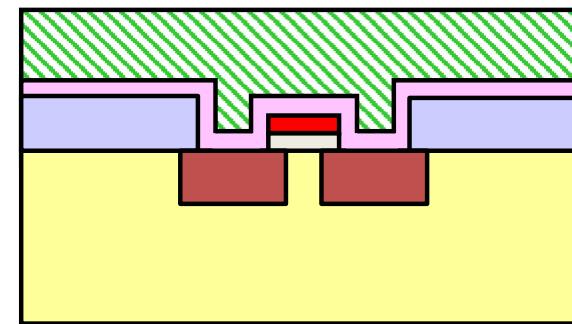
# Fabricación

vérion 14/07/23

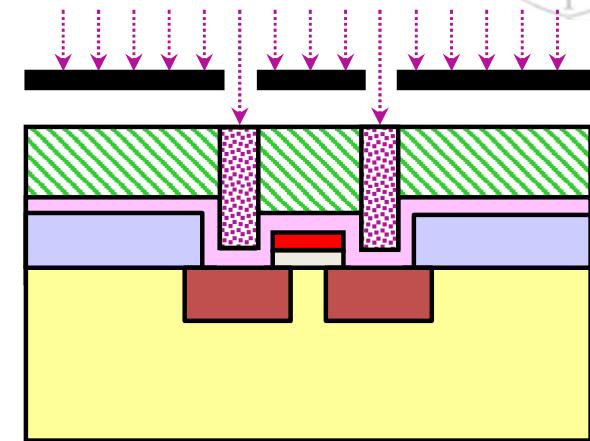
tema 3:  
Implementación de sistemas combinacionales



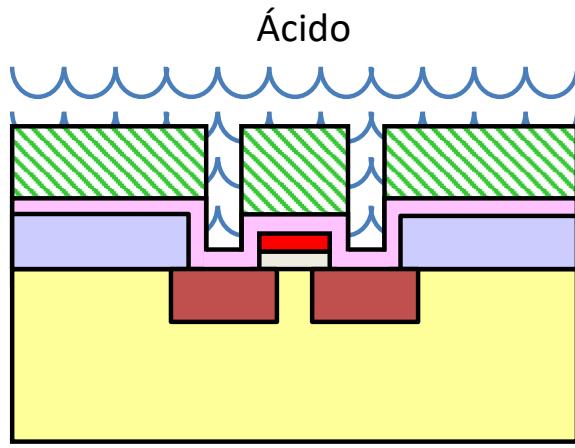
Oxide Growth



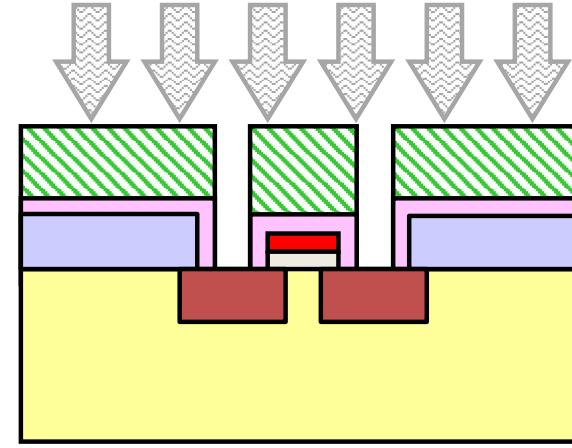
Photoresist Coating



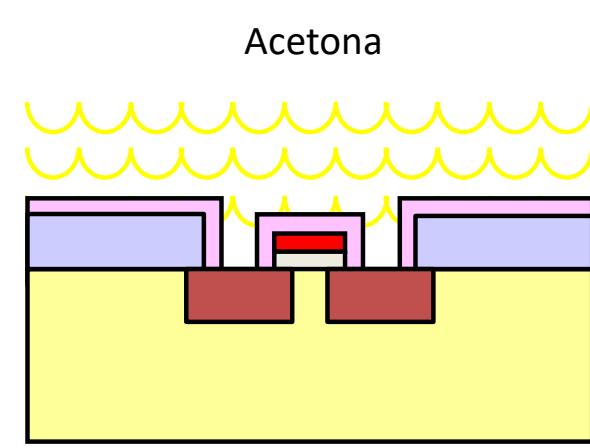
Photolithography (mask #5)



Soluble Photoresist Removing



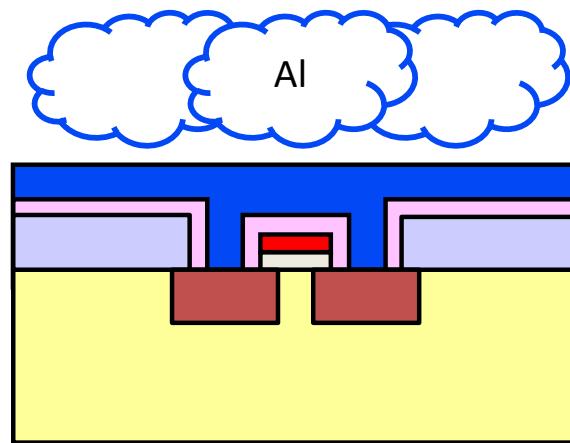
Chemical Etching



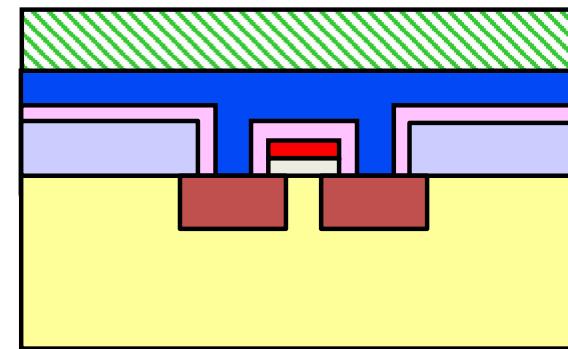
Photoresist Removing



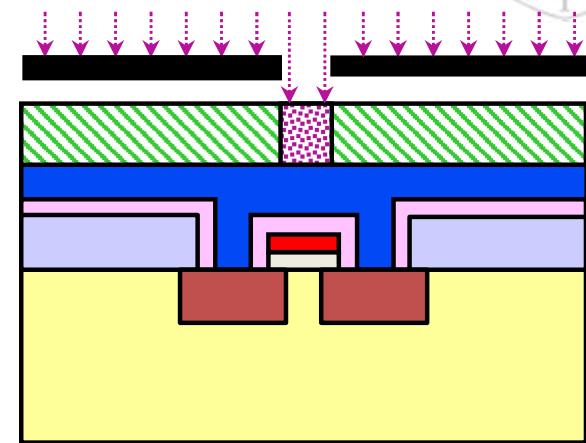
# Fabricación



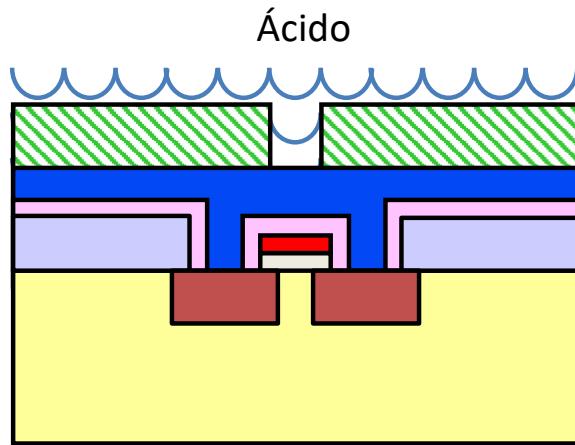
*Metal Vapor Deposition*



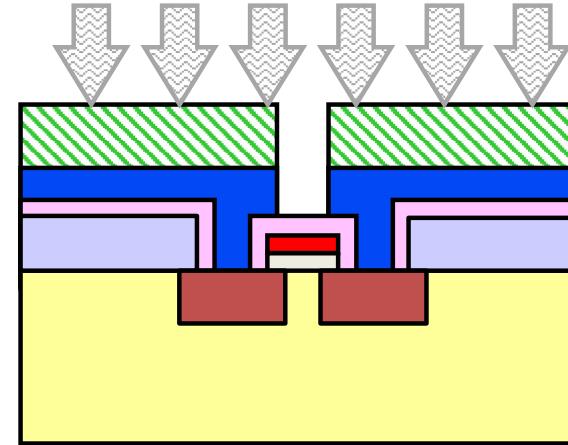
*Photoresist Coating*



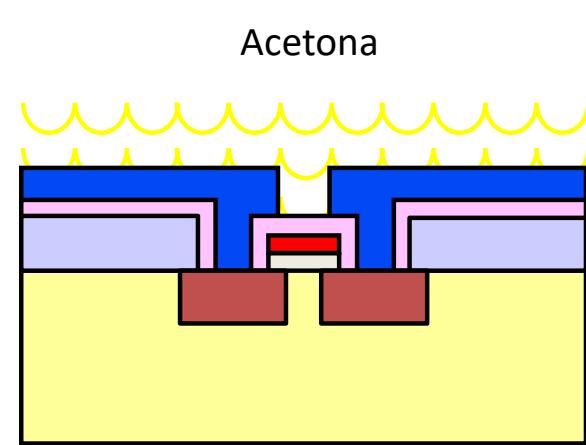
*Photolithography (mask #6)*



*Soluble Photoresist Removing*



*Chemical Etching*

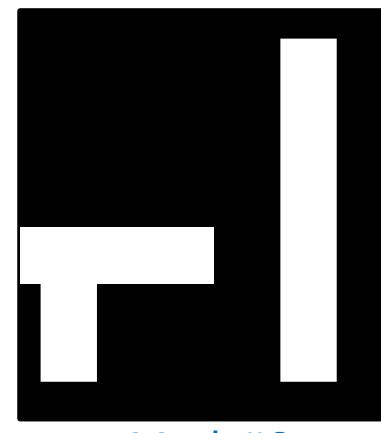
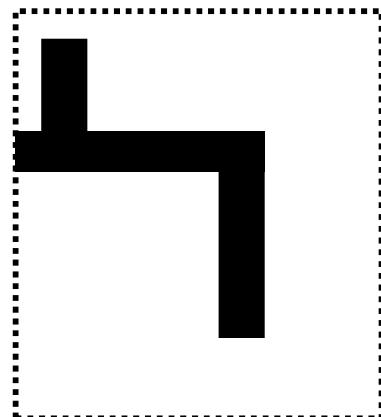
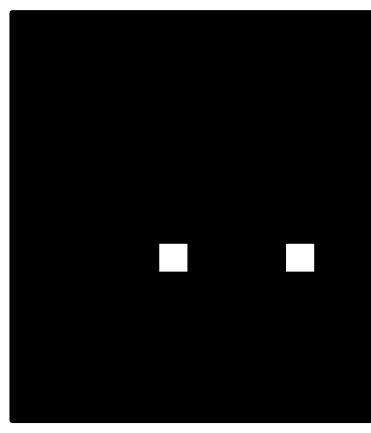
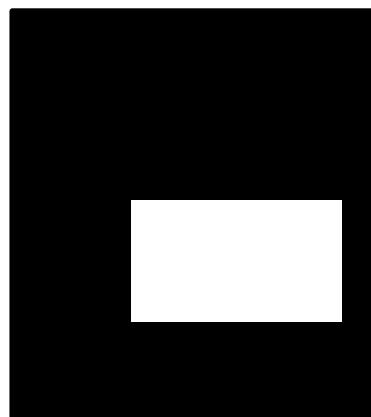
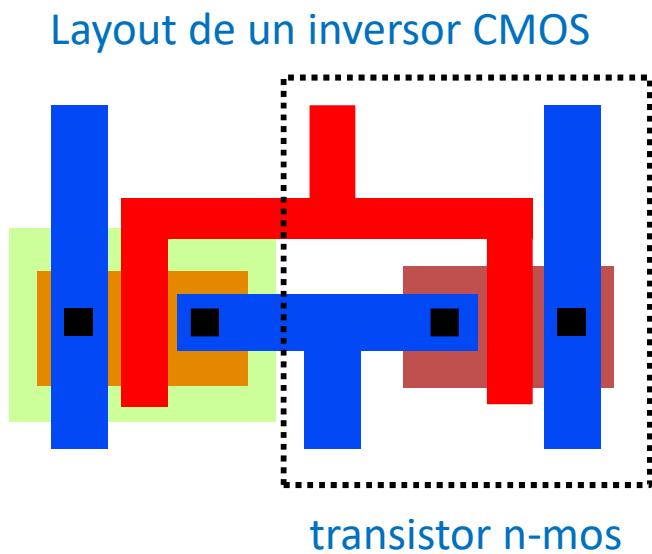


*Photoresist Removing*

# Fabricación



- El fabricante extrae las máscaras a usar durante el proceso de fabricación a partir del layout trazado por el diseñador.





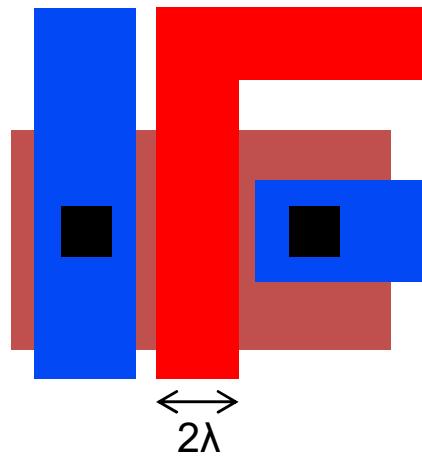
# Fabricación

- El **trazado del layout** (forma y ubicación de cada capa de material) es un proceso muy complejo que puede hacerse:
  - Desde cero: **diseño full-custom**
  - Ubicando y/o interconectando celdas prediseñadas a nivel físico o incluso ya prefabricadas: **diseño semi-custom**
- Existen distintas tecnologías semi-custom
  - Standard cells, Gate arrays, Sea of gates, FPGA, etc.
- Pero una muy popular es la basada en **standard cells**:
  - El fabricante facilita bibliotecas de celdas prediseñadas a nivel físico.
    - Cada celda realiza una función lógica simple.
    - Geométricamente es un rectángulo de altura fija y anchura variable (+ancha cuanto +compleja es su función lógica).
  - El diseñador implementa circuitos usando solo celdas de la biblioteca.
  - El layout se traza ubicando las celdas en filas e interconectándolas.



# Fabricación

- La geometría de un layout se define en términos de un único parámetro  $\lambda$ 
  - Representa la dimensión más pequeña definible en un proceso tecnológico, típicamente es la mitad de la longitud mínima de canal.
  - Todas las dimensiones se expresan como múltiplos enteros de  $\lambda$ , permitiendo el escalado proporcional de todo el circuito.
- Su valor absoluto disminuye en cada generación tecnológica:
  - 2004: 90nm , 2008: 45nm, 2012: 22nm, 2016: 10 nm, 2020: 5 nm
- Conforme  $\lambda$  disminuye, la densidad de integración aumenta.
  - Evolución de las NAND Flash (Samsung):
    - 2 Gb (90nm), 8 Gb (65nm), 16 Gb (50nm), 32Gb (32nm), 64 Gb (22nm)



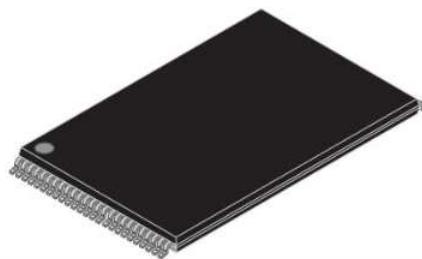


# Fabricación

- Fabricada la oblea, los circuitos se **recortan** y **encapsulan** en soportes plásticos o cerámicos.



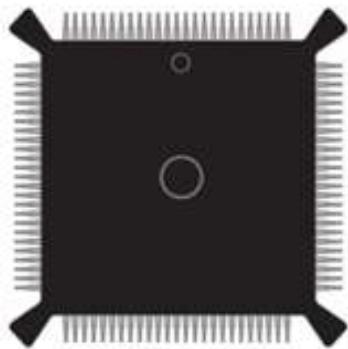
DIP  
(*Dual In-line Package*)



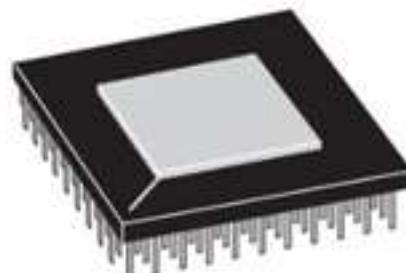
SOP  
(*Small Outline Package*)



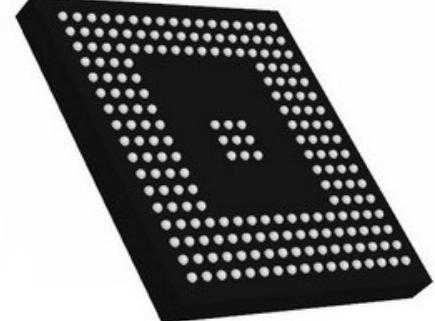
CC  
(*Chip Carrier*)



QFP  
(*Quad Flat Package*)



PGA  
(*Pin Grid Array*)

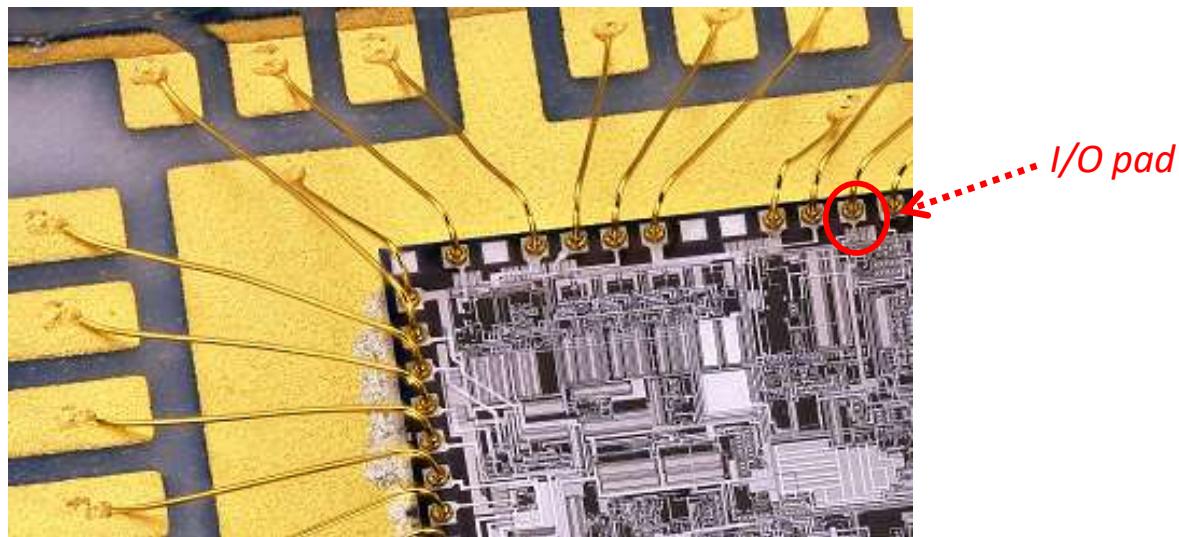


BGA  
(*Ball Grid Array*)

# Fabricación



- Los puertos de un circuito se conectan a los pines del encapsulado mediante **I/O pads** que incluyen:
  - Una superficie metálica sobre la soldar un cable
  - Diodos de protección ESD (electrostatic discharge)
  - Conversores de nivel de voltaje
  - Buffers para suministrar suficiente corriente al exterior





# Biblioteca de celdas

CMOS 90 nm

$V_{dd} = 1.2V$ ,  $T=25^\circ C$ ,  $C_L = 13 fF$

Puerta	# transistores	Área ( $\mu\text{m}^2$ )	Retardo (ps)	Consumo estático (nW)	Consumo dinámico (nW/MHz)
	2	6.4512	100	52	618
	6	7.3728	180	56	6545
	6	7.3728	171	58	6859
	4	5.5296	132	78	5208
	4	6.4512	132	49	6190
	14	13.8240	237	89	8702
	12	13.8240	136	82	16372



# Biblioteca de celdas

CMOS 90 nm

Puerta	# transistores	Área ( $\mu\text{m}^2$ )	Retardo (ps)	C. estático (nW)	C. dinámico (nW/MHz)
	8	8.2944	206	59	8222
	10	10.1376	214	63	8557
	8	9.2160	184	62	7396
	10	10.1376	199	64	7698
	6	11.9808	192	102	12200
	8	12.9024	178	161	15214
	6	11.9808	153	104	16521
	8	15.6672	126	119	14991



# Biblioteca de celdas

CMOS 90 nm

Puerta	# transistores	Área ( $\mu\text{m}^2$ )	Retardo (ps)	Consumo estático (nW)	Consumo dinámico (nW/MHz)
	6	11.9808	181	86	13912
	8	12.9024	204	89	13290
	12	15.6672	221	109	13982
	10	11.9808	238	63	10662
	8	12.9024	174	98	10666

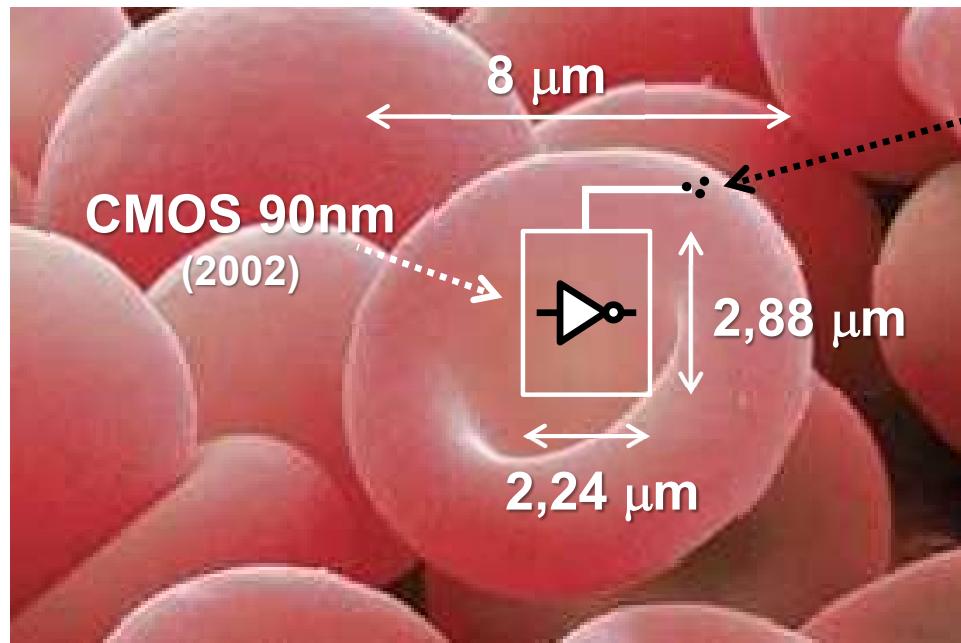
fuente: Synopsys (SAED EDK 90 nm)

# Biblioteca de celdas

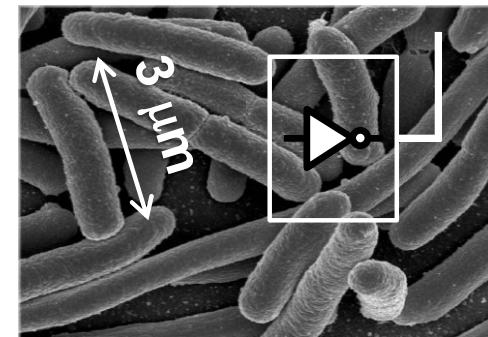
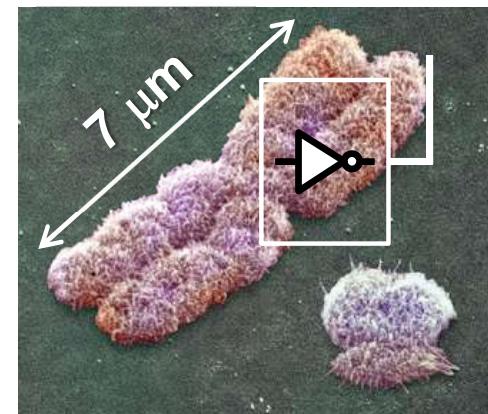
## CMOS 90 nm



fuente: [Learn\\_Genetics](#) (UTAH Univ.)



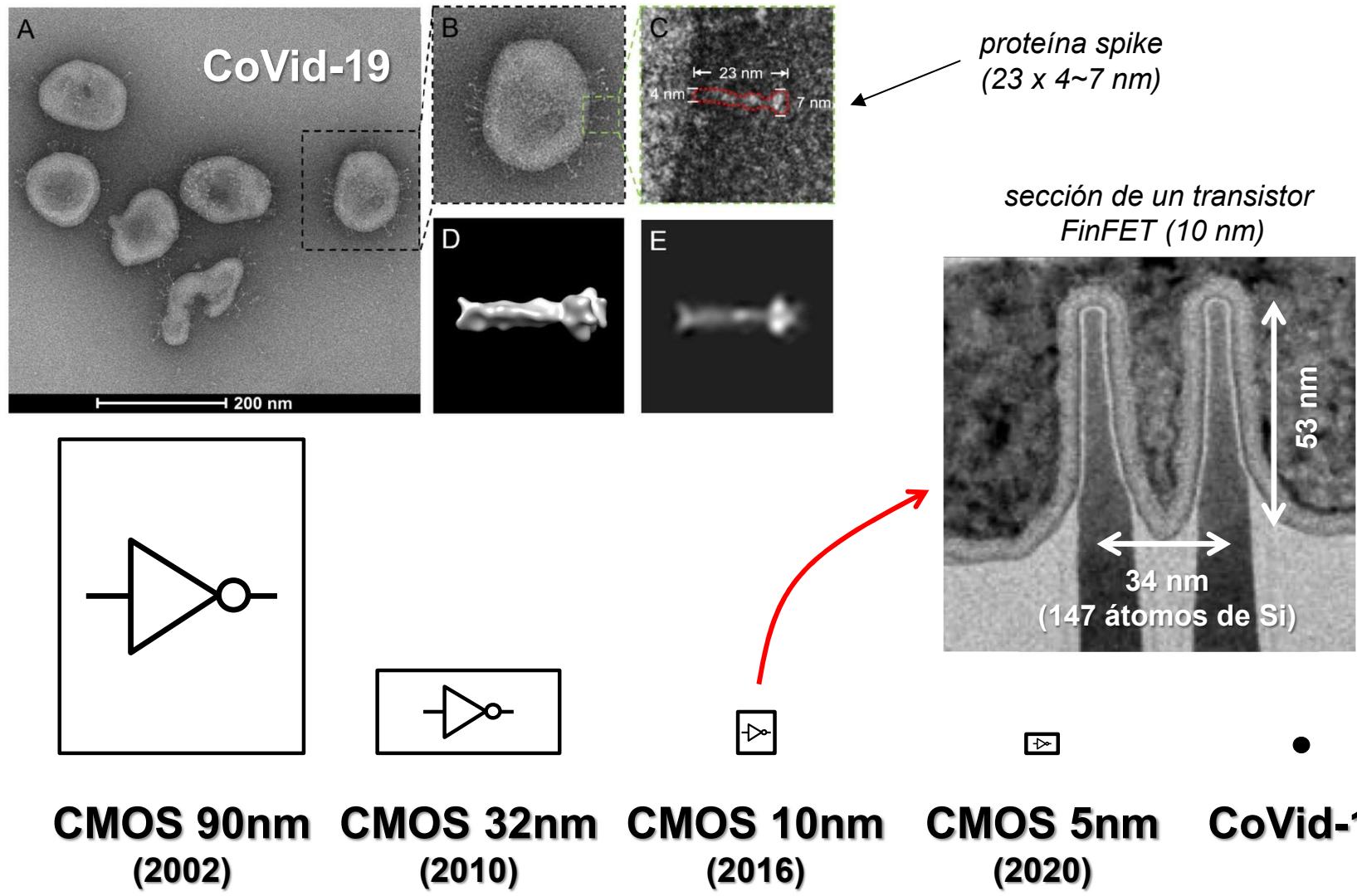
virus de COVID-19  
vs. interconexión



- Área de un inversor:  $6.45 \mu\text{m}^2$ 
  - Área de un glóbulo rojo ( $\varnothing 8\mu\text{m}$ ):  $50.26 \mu\text{m}^2$
- Retardo medio de un inversor: 100 ps
  - La luz en 100 ps recorre 3 cm
- Consumo de un inversor: 670 nW (comutando a 1 MHz)
  - Una pila alcalina AA (2890 mAh) alimentando al inversor duraría 740 años
- Anchura de una interconexión: 140 nm
  - Cromosoma X: 7000 nm, E. coli: 3000 nm, Covid-19: 140 nm

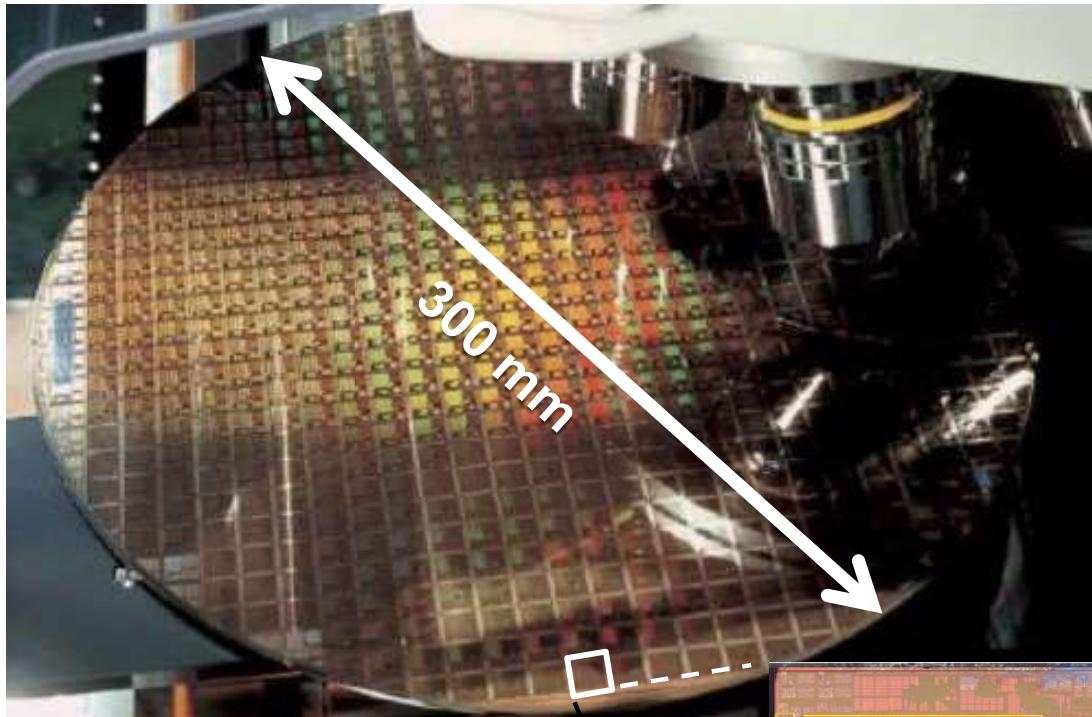


# Evolución tecnológica



fuentes: C.Liu et al., Viral Architecture of SARS-CoV-2 with Post-Fusion Spike Revealed by Cryo-EM, bioRxiv (2020)  
(adaptación) K. Mistry, 10 nm technology leadership, Technology and Manufacturing Day, Intel (2017)

# Evolución tecnológica



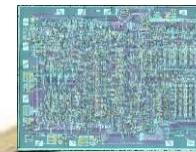
Apple A14 SoC (2020)

11.700.000.000 transistores  
(CMOS 5 nm, 88 mm<sup>2</sup>)



Intel 4004 (1971)

2.300 transistores  
(MOSFET 10 µm, 12 mm<sup>2</sup>)



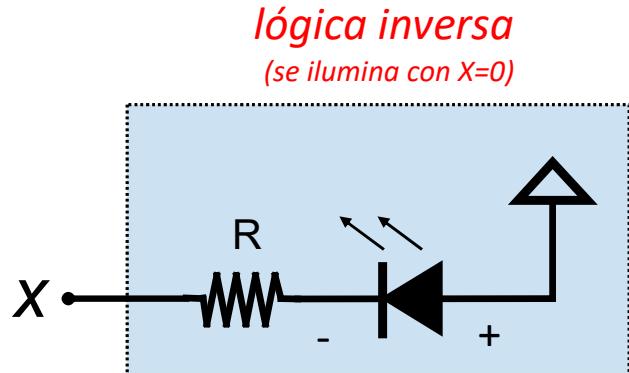
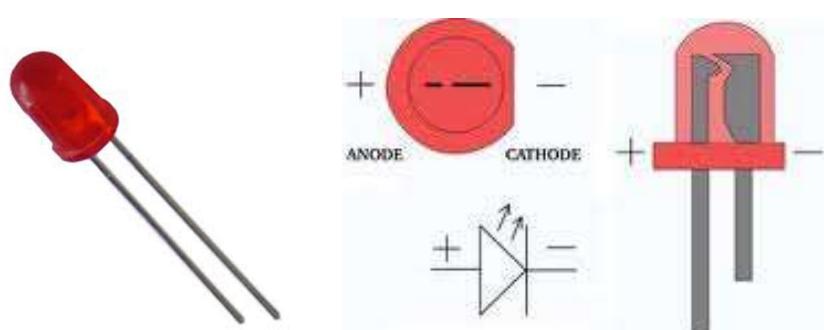
tr/mm<sup>2</sup>  
700.000x



# Entrada/salida elemental



- Un LED (Light-Emitting Diode) es el dispositivo más simple por el que un sistema digital puede mostrar su estado.
  - Típicamente se ilumina cuando entre cátodo (-) y ánodo (+) hay una diferencia de potencial superior a 1.7 V.
  - La máxima luminosidad se consigue cuando por él circula una intensidad de entre 10 y 20 mA.
  - Se conecta a un puerto de salida de un circuito a través de una **resistencia limitadora**.

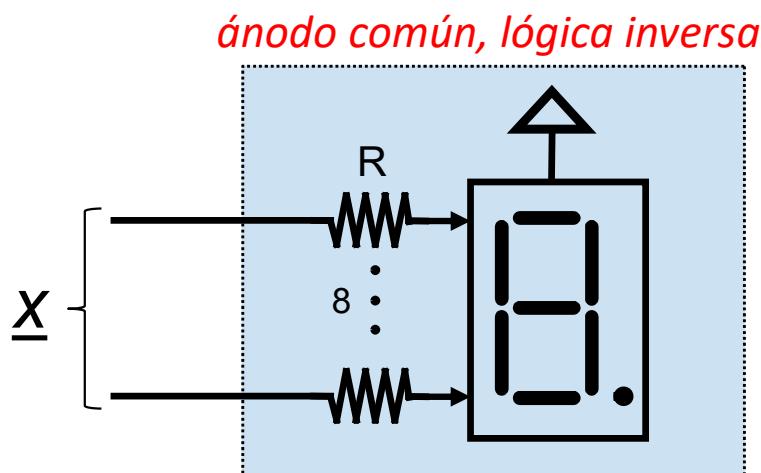
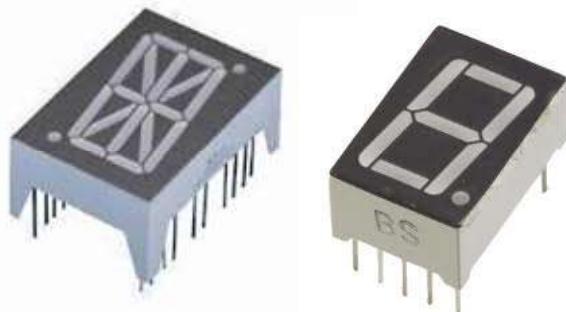


$$R \text{ (TTL)} = (5V - 1.7V) / 10 \text{ mA} = 330 \Omega$$

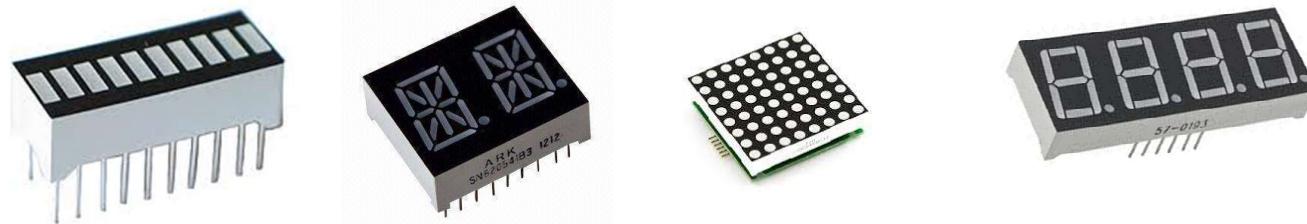


# Entrada/salida elemental

- Un **display 7-segmentos** es un banco de 7/8/9 leds que comparten uno de los terminales.
  - Cada led debe disponer de su propia **resistencia limitadora**.



- Existen un gran número de displays basados en leds

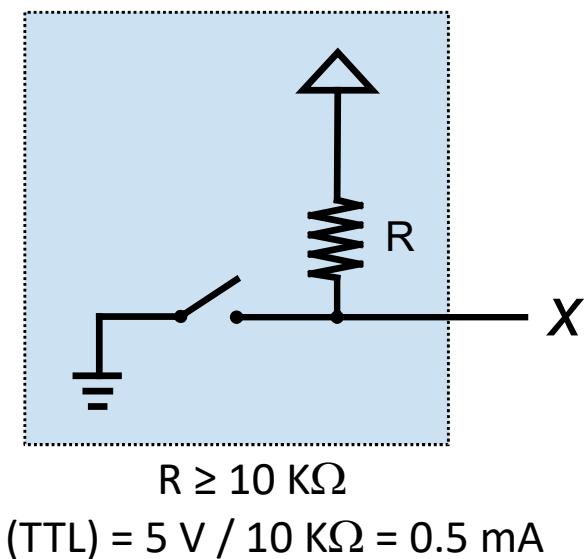




# Entrada/salida elemental

- Un **switch/pulsador** es el dispositivo más simple por el que introducir datos a un sistema digital.
  - Se conecta a un puerto de entrada de un circuito a través de una **resistencia de pull-up (o pull-down)**
    - Asegura un nivel lógico por defecto cuando no hay pulsación.
    - Evita el cortocircuito se crearía en su ausencia cuando hay pulsación.
    - Debe ser alta para limitar la corriente que circula por el pulsador (y así reducir el consumo) cuando este se cierra.

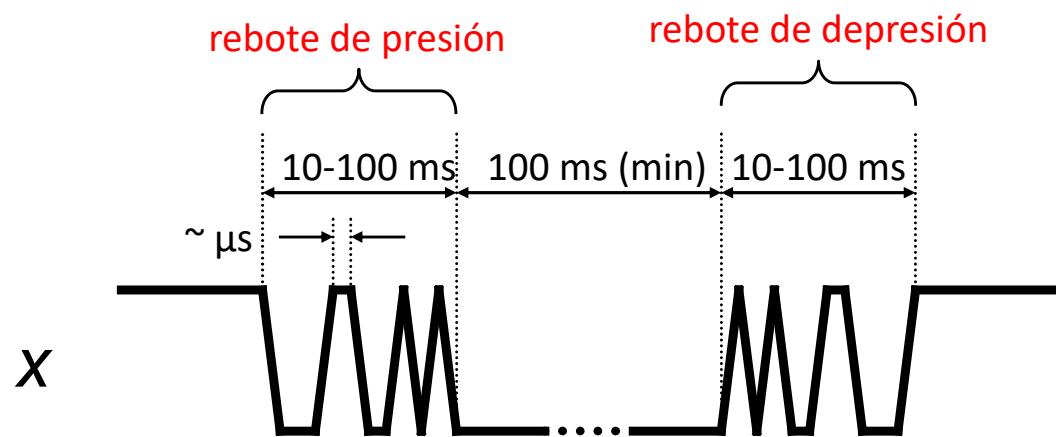
***Lógica inversa**  
(al pulsar, X=0)*





# Entrada/salida elemental

- No obstante, cuando un sistema digital **lee el estado de un interruptor** encuentra los siguientes problemas:
  - Cada cambio de estado del interruptor, genera un vaivén transitorio en la señal de entrada (**rebote**).
    - Una pulsación puede interpretarse erróneamente como una serie de ellas
  - Los cambios de estado del interruptor son **asíncronos**.



- Por ello, los sistemas digitales incorporan mecanismos de **sincronización** y **filtrado de rebotes** en sus entradas externas.



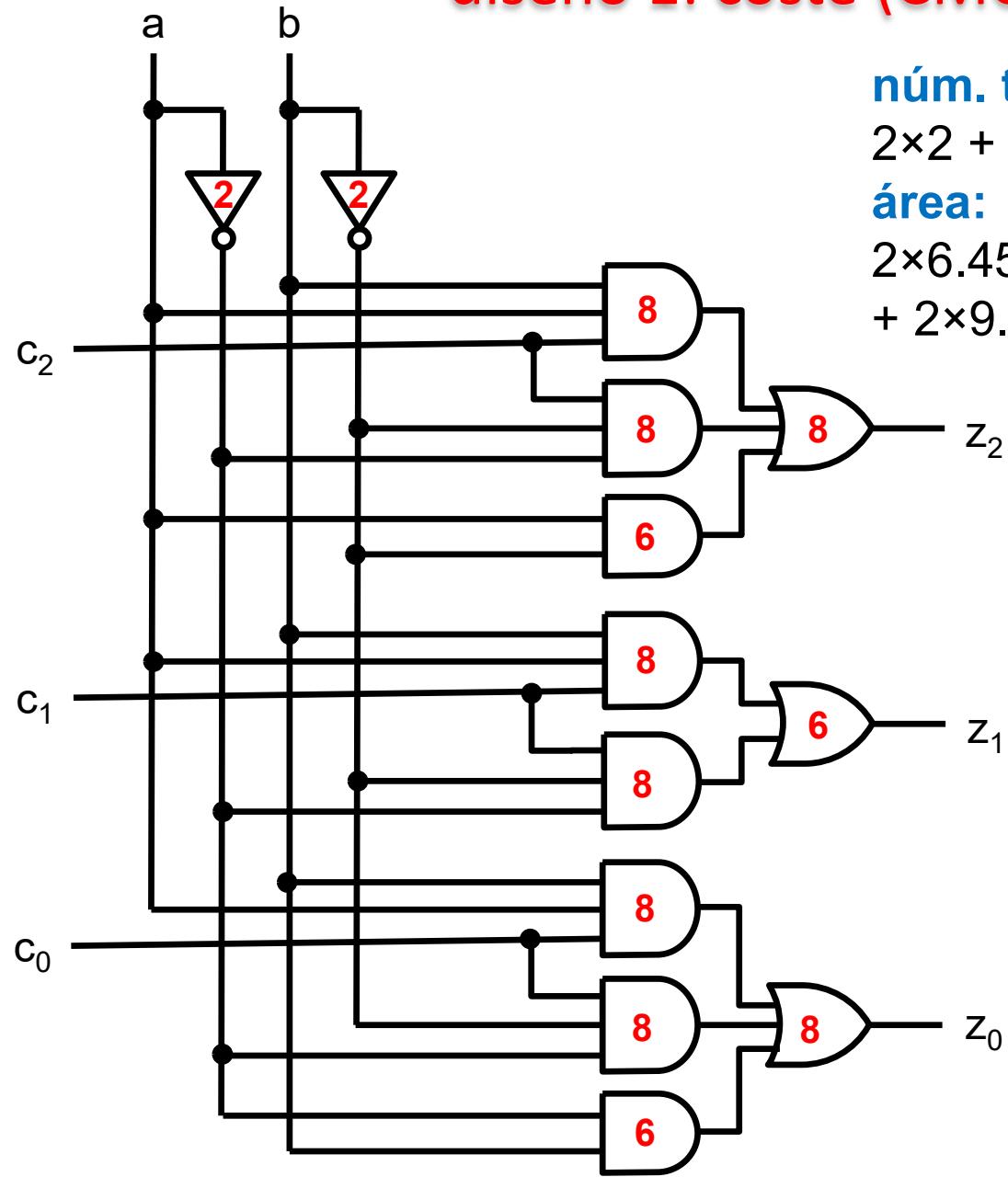
# Trade-offs

- El diseño digital es un **proceso de ingeniería**:
  - Debe elegirse el **mejor circuito posible** que realice una función.
  - Pero ... ¿cuál es el mejor cuando infinidad hacen lo mismo?
- Las **métricas de calidad** permiten tomar la decisión:
  - Coste, velocidad, consumo, robustez, reusabilidad, time-to-market...
  - Dado que la mejora en una métrica afecta al resto, hay que llegar a un compromiso (**trade-off**).
    - por ejemplo, típicamente una mejora en velocidad implica un aumento del coste
  - La importancia relativa de las mismas varía en cada producto.
    - por ejemplo, una misma funcionalidad se implementará de forma distinta si se integra en una lavadora o si lo hace en un satélite
- El conjunto de implementaciones posibles de una misma especificación se denomina **espacio de diseño**
  - el diseño digital es un problema de **optimización multi-objetivo**



# Trade-offs

diseño 1: coste (CMOS 90 nm)



núm. transistores:

$$2 \times 2 + 6 \times 8 + 2 \times 6 + 2 \times 8 + 1 \times 6 = 86$$

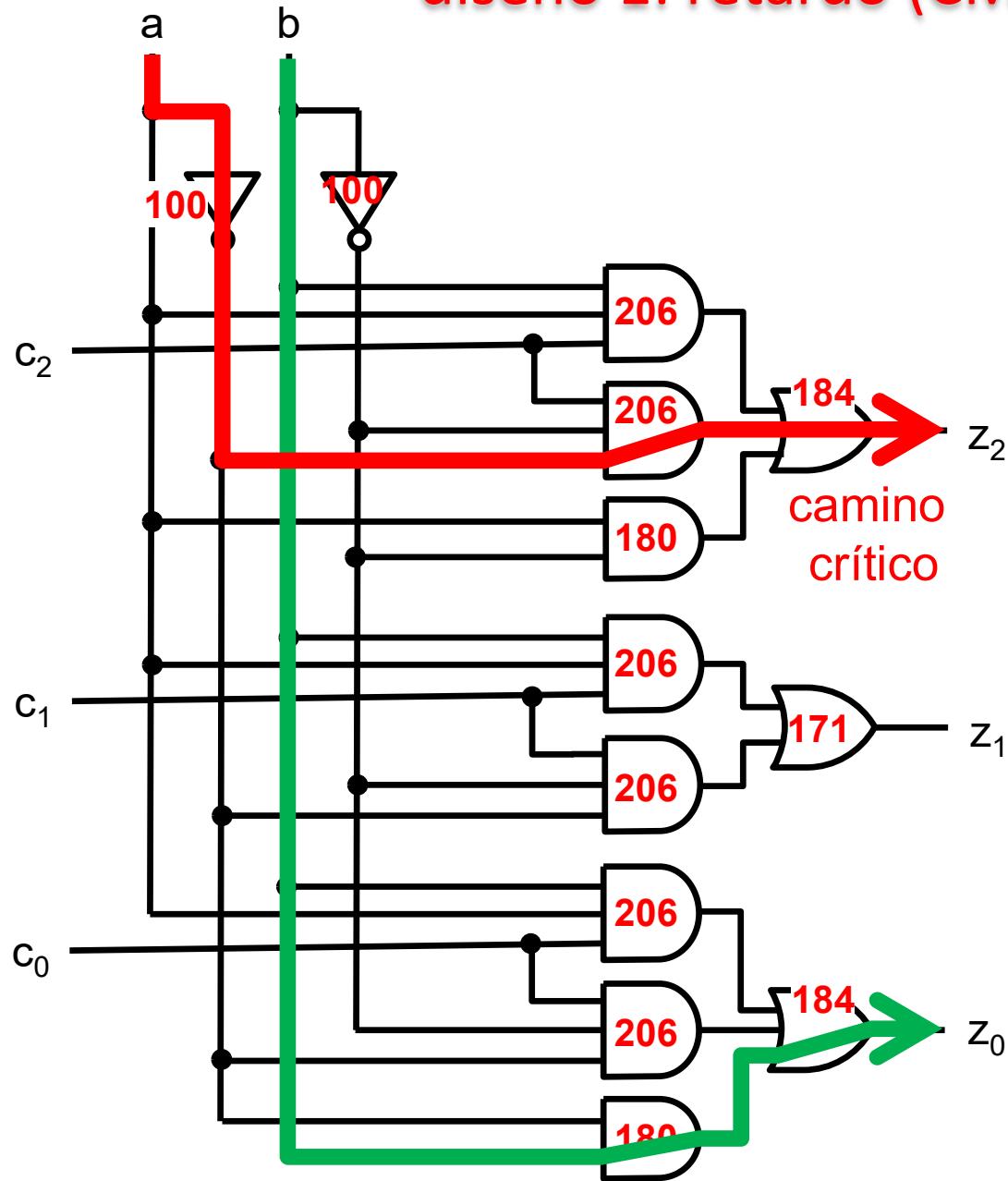
área:

$$\begin{aligned} & 2 \times 6.4512 + 6 \times 8.2944 + 2 \times 7.3728 + \\ & + 2 \times 9.2160 + 1 \times 7.3728 = 103.2 \mu\text{m}^2 \end{aligned}$$



# Trade-offs

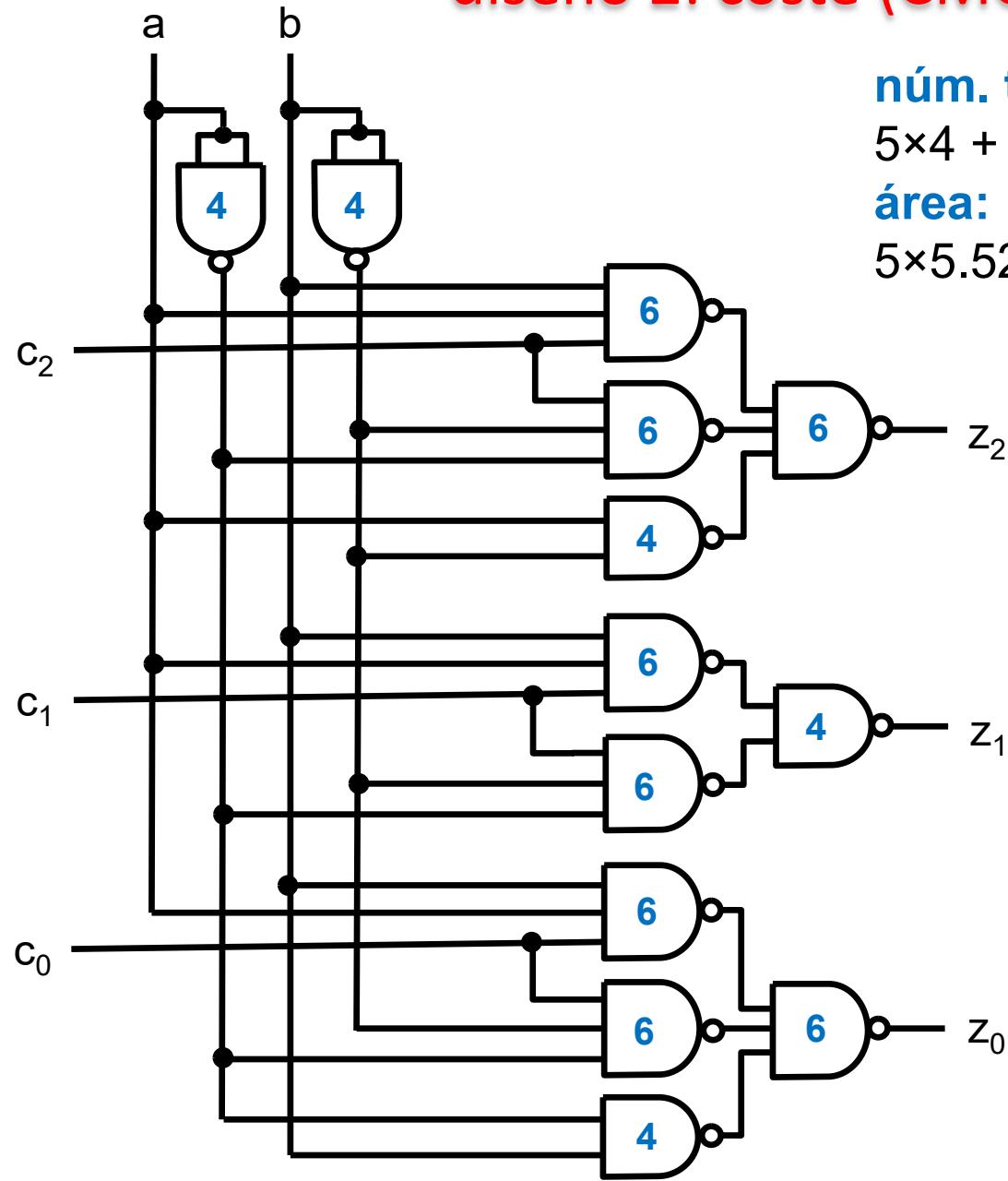
diseño 1: retardo (CMOS 90 nm)





# Trade-offs

diseño 2: coste (CMOS 90 nm)



núm. transistores:

$$5 \times 4 + 8 \times 6 = 68$$

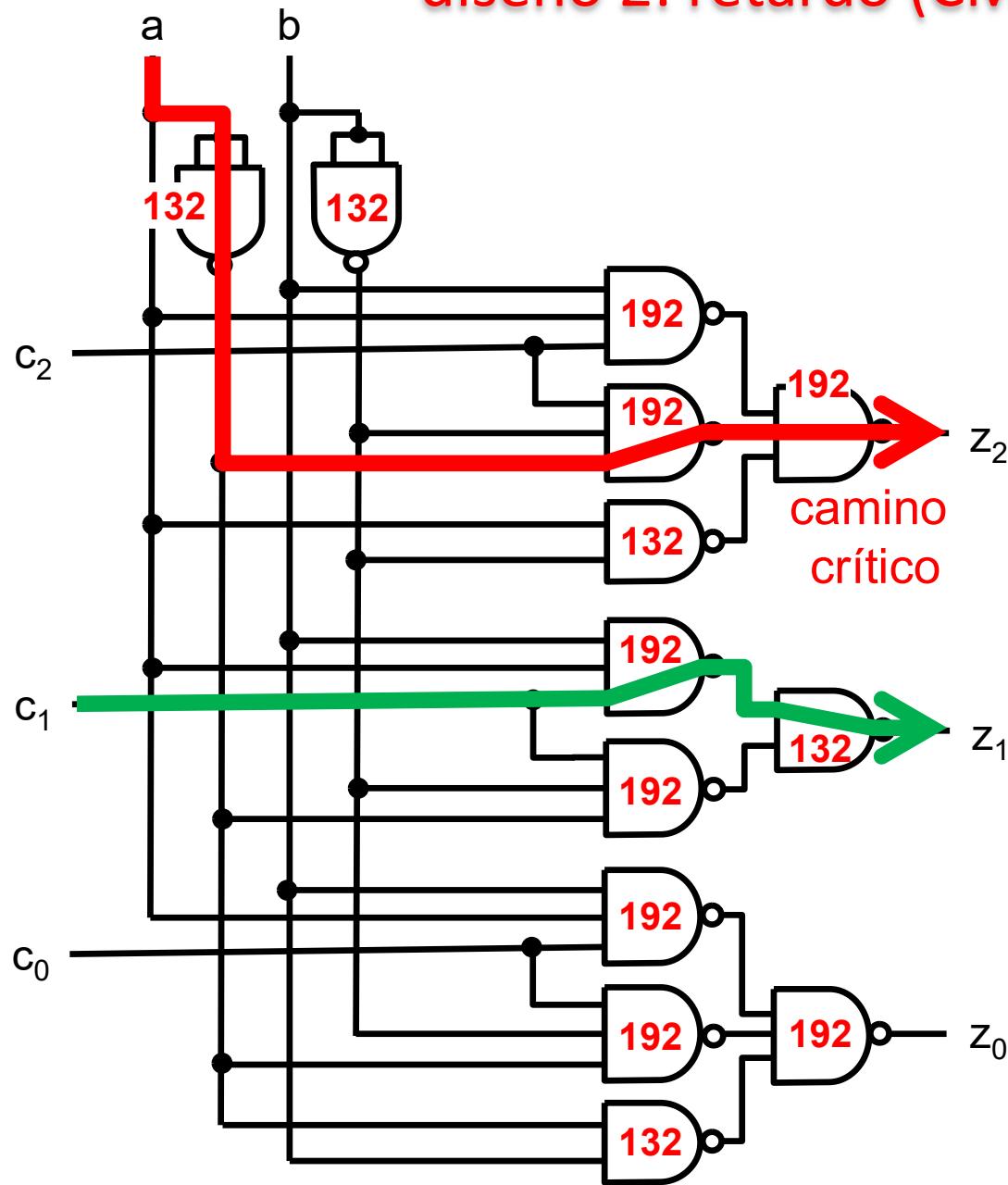
área:

$$5 \times 5.5296 + 8 \times 11.9808 = 123.3 \mu\text{m}^2$$



# Trade-offs

diseño 2: retardo (CMOS 90 nm)



$$t_d = 132 + 192 + 192 = 516 \text{ ps}$$
$$t_c = 132 + 192 = 324 \text{ ps}$$

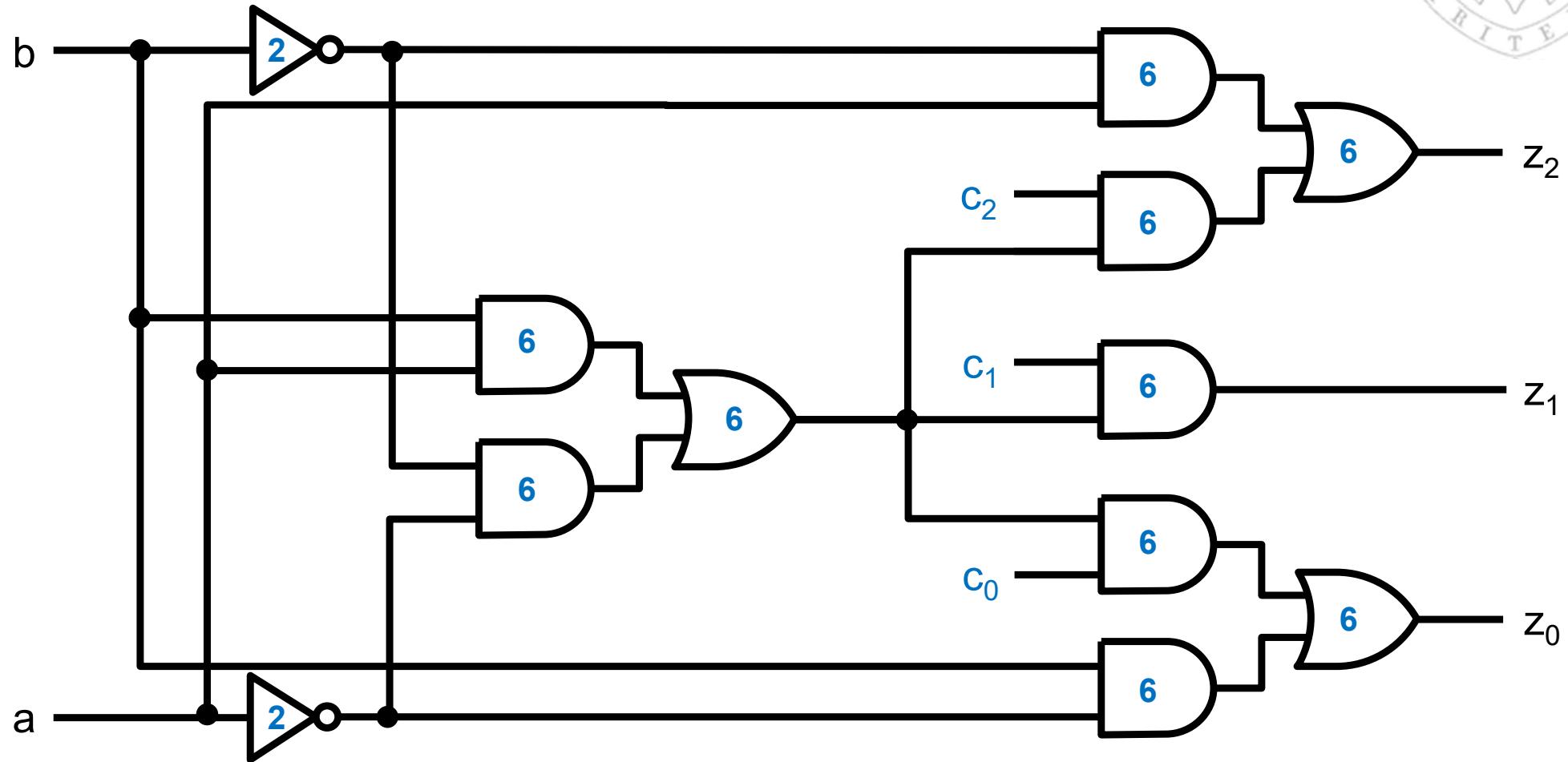
$$t_d = 132 + 192 + 132 = 456 \text{ ps}$$
$$t_c = 192 + 132 = 324 \text{ ps}$$

$$t_d = 132 + 192 + 192 = 516 \text{ ps}$$
$$t_c = 132 + 192 = 324 \text{ ps}$$



# Trade-offs

diseño 3: coste (CMOS 90 nm)



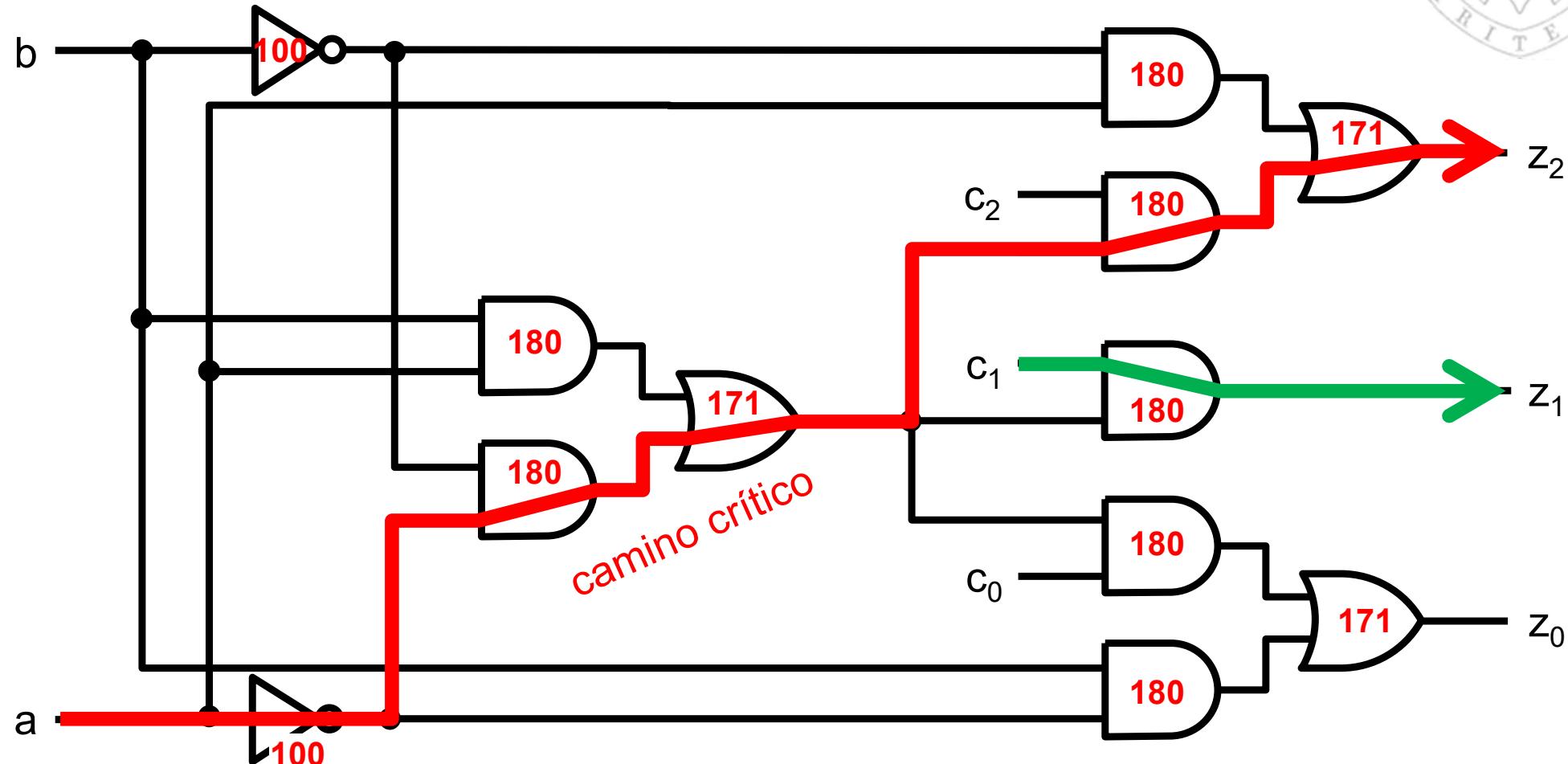
núm. transistores:  $7 \times 6 + 3 \times 6 + 2 \times 2 = 64$

área:  $7 \times 7.3728 + 3 \times 7.3728 + 2 \times 6.4512 = 86.63 \mu\text{m}^2$



# Trade-offs

diseño 3: retardo (CMOS 90 nm)



$$z_2: t_d = 100 + 180 + 171 + 180 + 171 = 802 \text{ ps}$$

$$t_c = 180 + 171 = 315 \text{ ps}$$

$$z_1: t_d = 100 + 180 + 171 + 180 = 631 \text{ ps}$$

$$t_c = 180 \text{ ps}$$

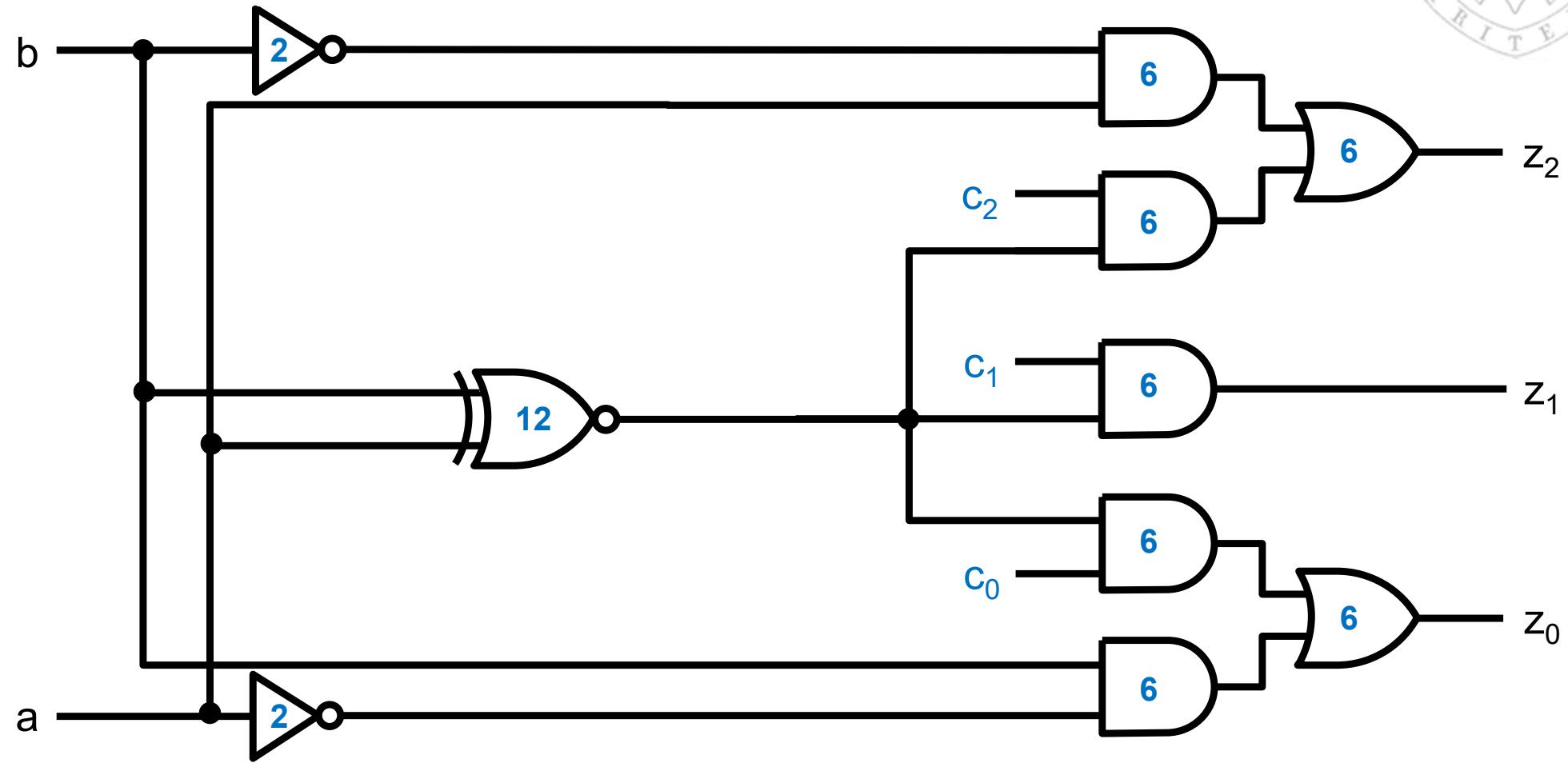
$$z_0: t_d = 100 + 180 + 171 + 180 + 171 = 802 \text{ ps}$$

$$t_c = 180 + 171 = 315 \text{ ps}$$



# Trade-offs

diseño 4: coste (CMOS 90 nm)



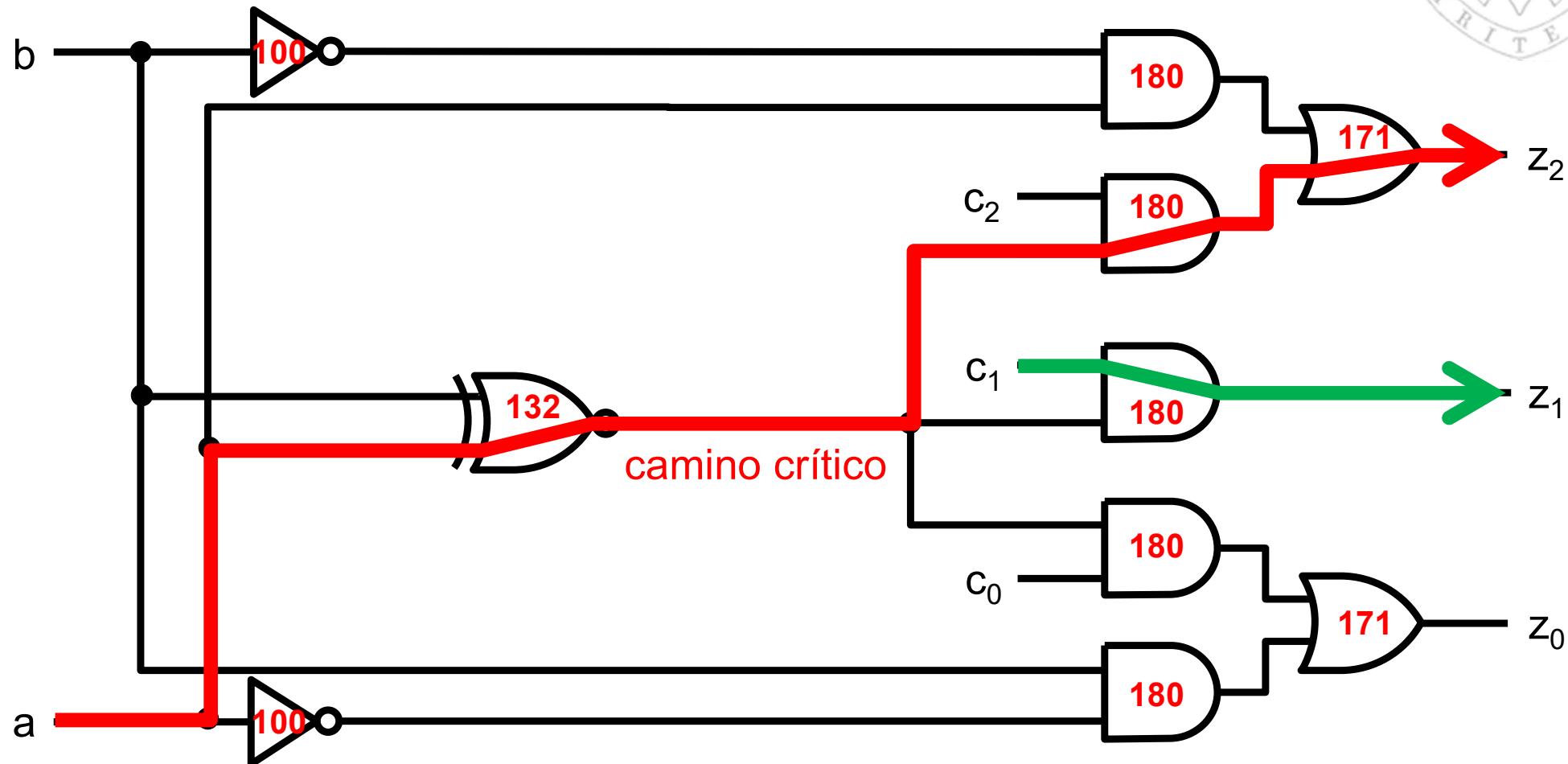
núm. transistores:  $1 \times 12 + 5 \times 6 + 2 \times 6 + 2 \times 2 = 58$

área:  $1 \times 13.824 + 5 \times 7.3728 + 2 \times 7.3728 + 2 \times 6.4512 = 78.35 \mu\text{m}^2$



# Trade-offs

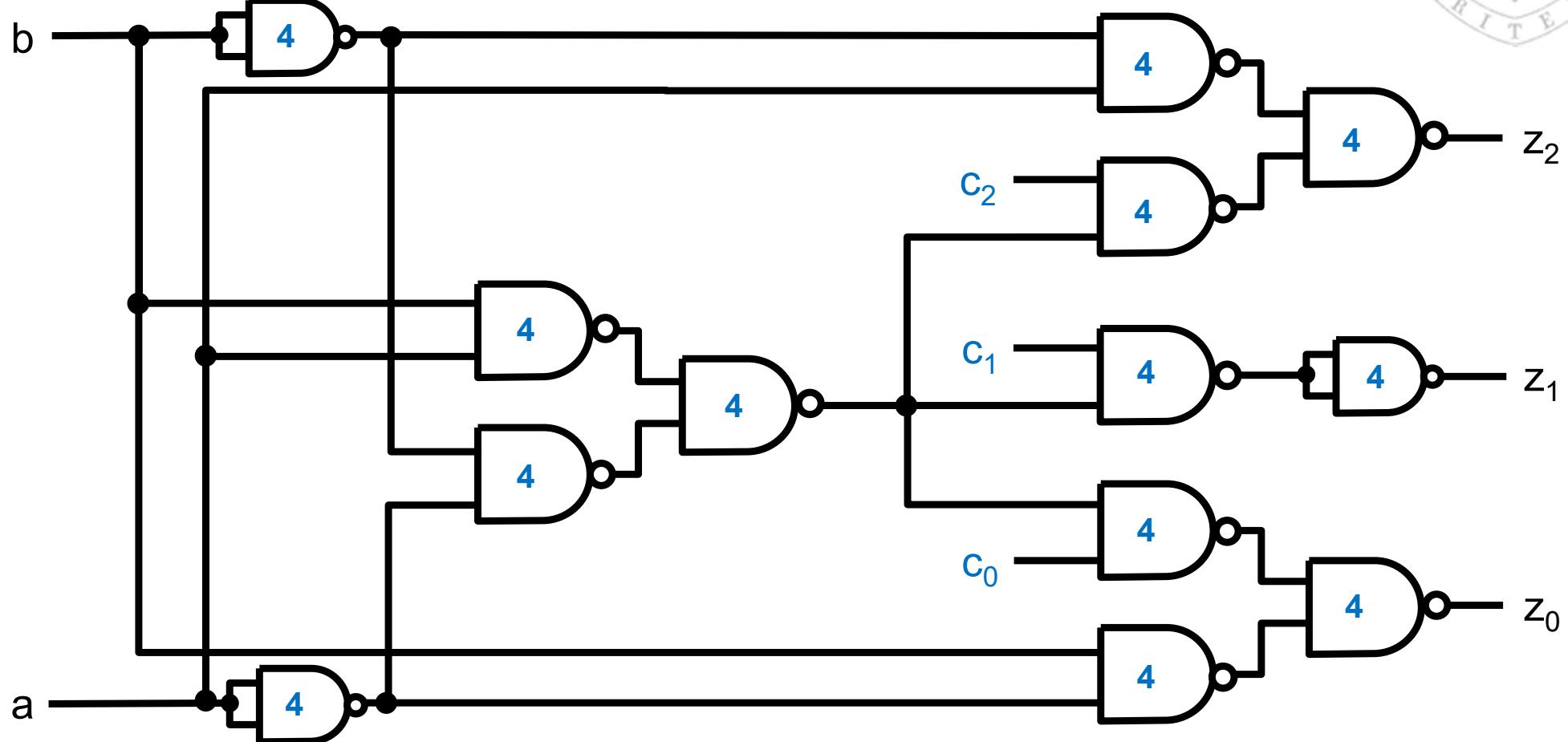
diseño 4: retardo (CMOS 90 nm)





# Trade-offs

diseño 5: coste (CMOS 90 nm)



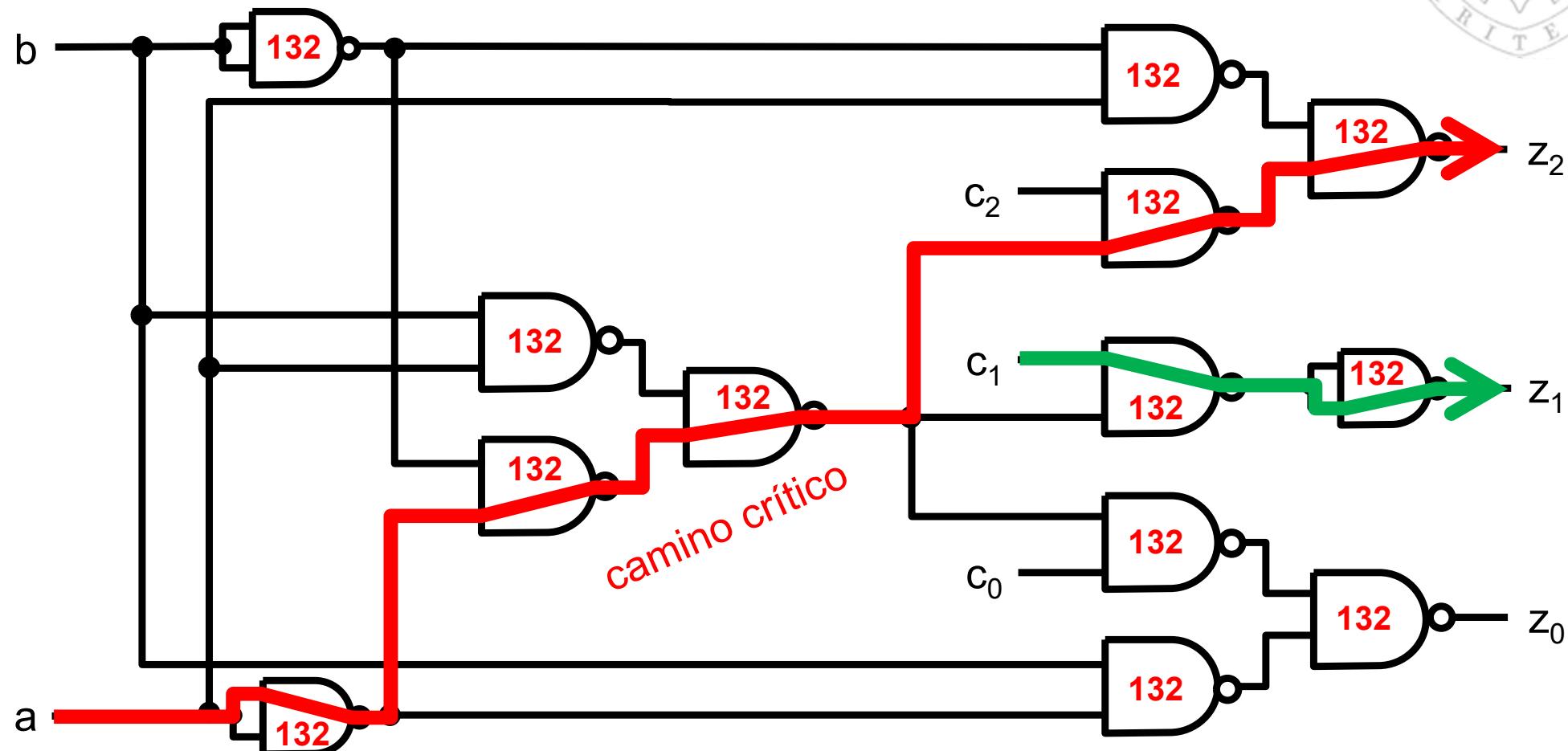
núm. transistores:  $13 \times 4 = 52$

área:  $13 \times 5.5296 = 71.88 \mu\text{m}^2$



# Trade-offs

diseño 5: retardo (CMOS 90 nm)



$$z_2: \quad t_d = 5 \times 132 = 660 \text{ ps}$$

$$z_1: \quad t_d = 5 \times 132 = 660 \text{ ps}$$

$$z_0: \quad t_d = 5 \times 132 = 660 \text{ ps}$$

$$t_c = 2 \times 132 = 264 \text{ ps}$$

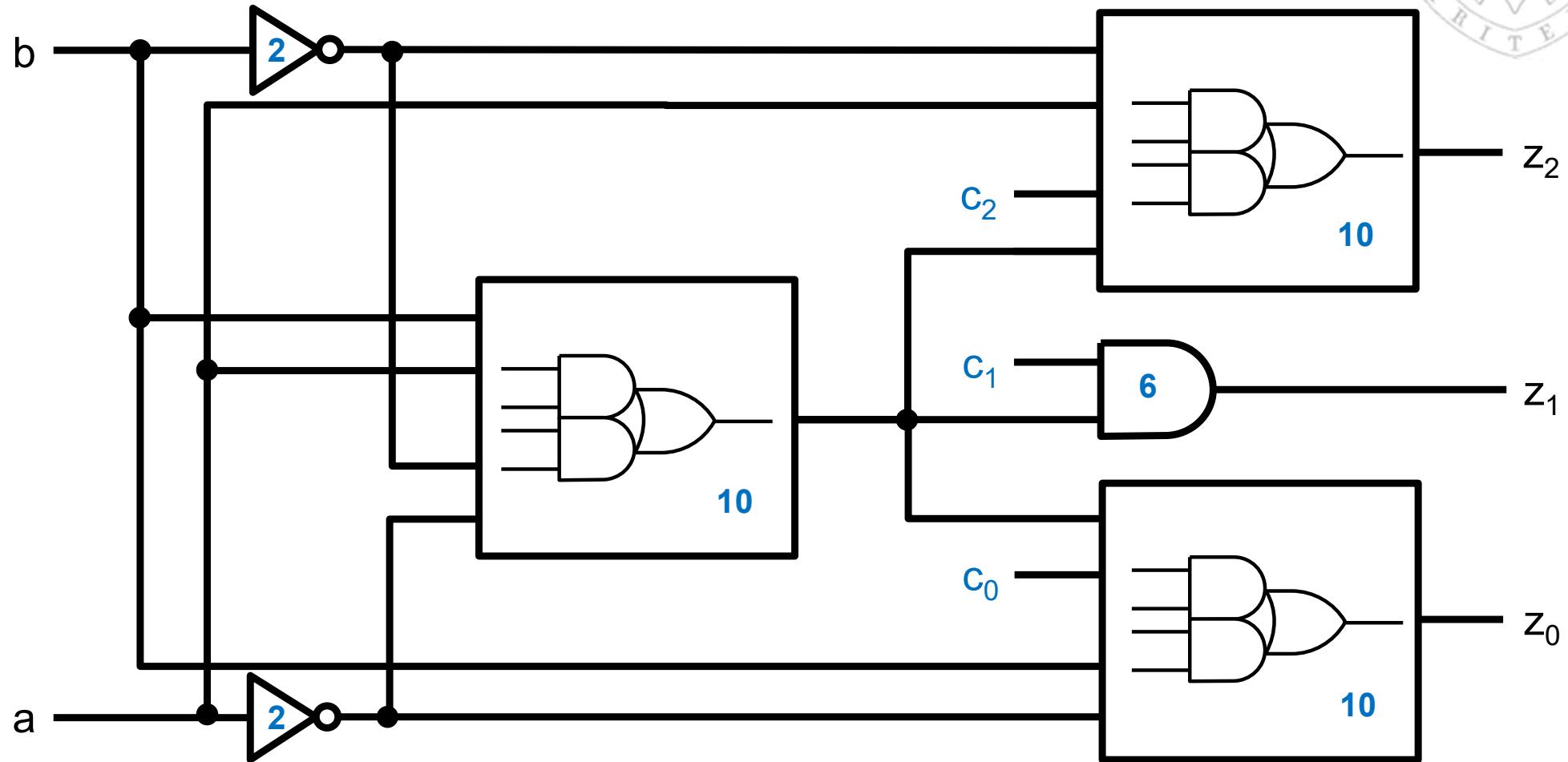
$$t_c = 2 \times 132 = 264 \text{ ps}$$

$$t_c = 2 \times 132 = 264 \text{ ps}$$



# Trade-offs

diseño 6: coste (CMOS 90 nm)



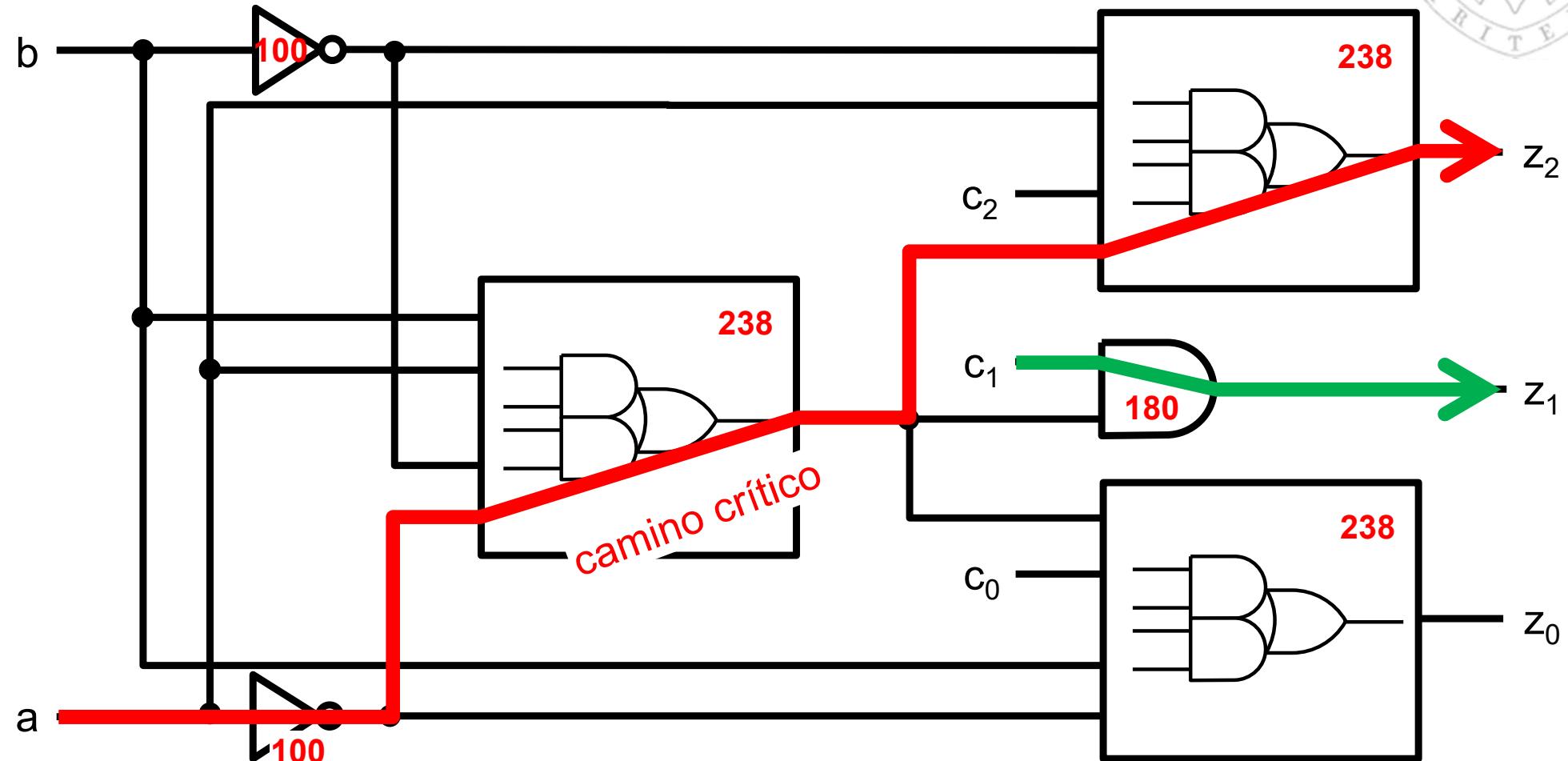
núm. transistores:  $3 \times 10 + 6 + 2 \times 2 = 40$

área:  $3 \times 11.9808 + 7.3728 + 2 \times 6.4512 = 56.22 \mu\text{m}^2$



# Trade-offs

diseño 6: retardo (CMOS 90 nm)



$$z_2: \quad t_d = 100 + 238 + 238 = 576 \text{ ps}$$

$$t_c = 238 \text{ ps}$$

$$z_1: \quad t_d = 100 + 238 + 180 = 518 \text{ ps}$$

$$t_c = 180 \text{ ps}$$

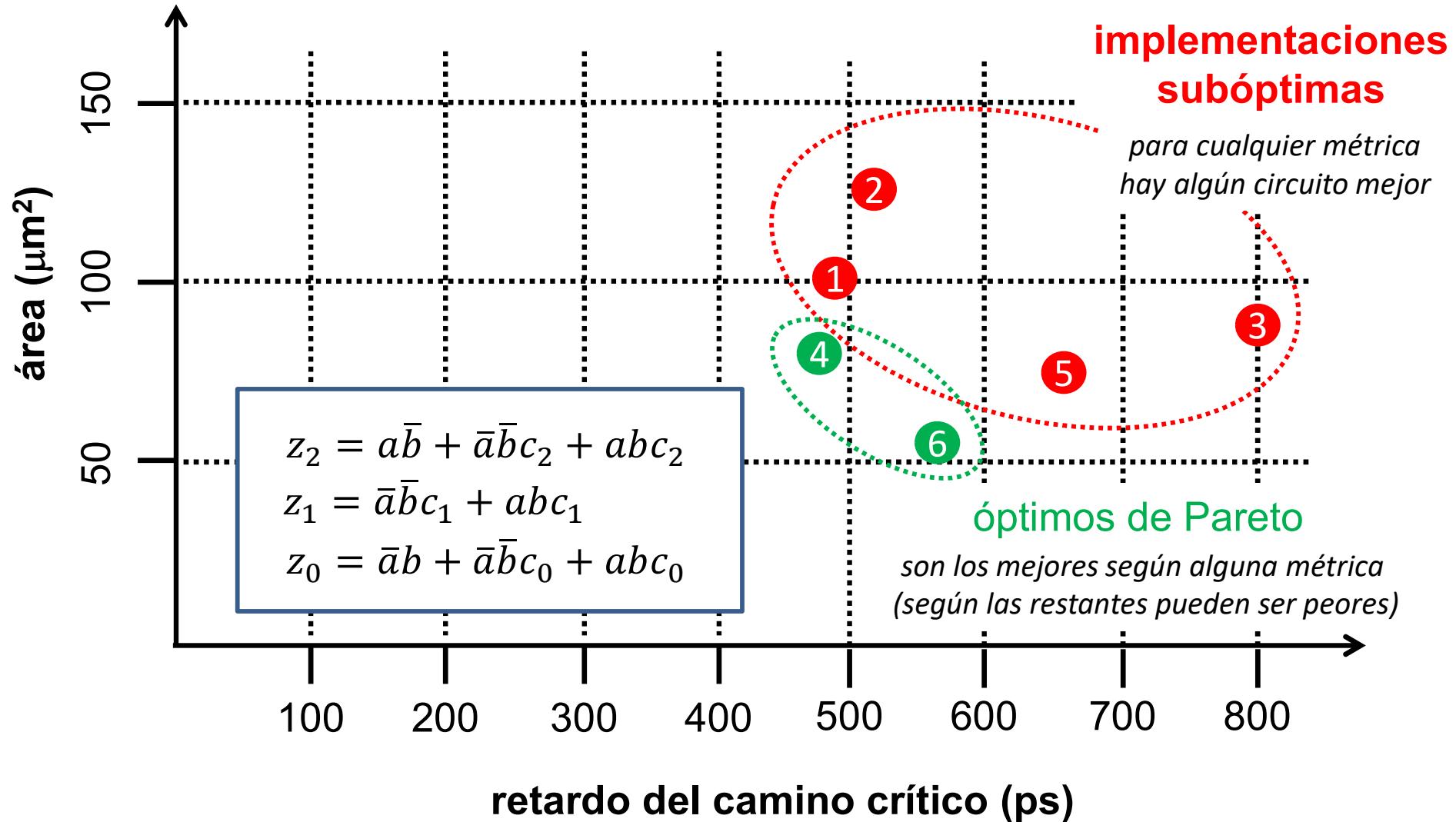
$$z_0: \quad t_d = 100 + 238 + 238 = 576 \text{ ps}$$

$$t_c = 238 \text{ ps}$$



# Trade-offs

Espacio de diseño (CMOS 90 nm)



# Acerca de *Creative Commons*



## ■ Licencia CC (*Creative Commons*)



- Ofrece algunos derechos a terceras personas bajo ciertas condiciones. Este documento tiene establecidas las siguientes:



### **Reconocimiento (Attribution):**

En cualquier explotación de la obra autorizada por la licencia hará falta reconocer la autoría.



### **No comercial (Non commercial):**

La explotación de la obra queda limitada a usos no comerciales.



### **Compartir igual (Share alike):**

La explotación autorizada incluye la creación de obras derivadas siempre que mantengan la misma licencia al ser divulgadas.

Más información: <https://creativecommons.org/licenses/by-nc-sa/4.0/>