Tema 3:

Implementación de sistemas combinacionales

Fundamentos de computadores

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

Contenidos

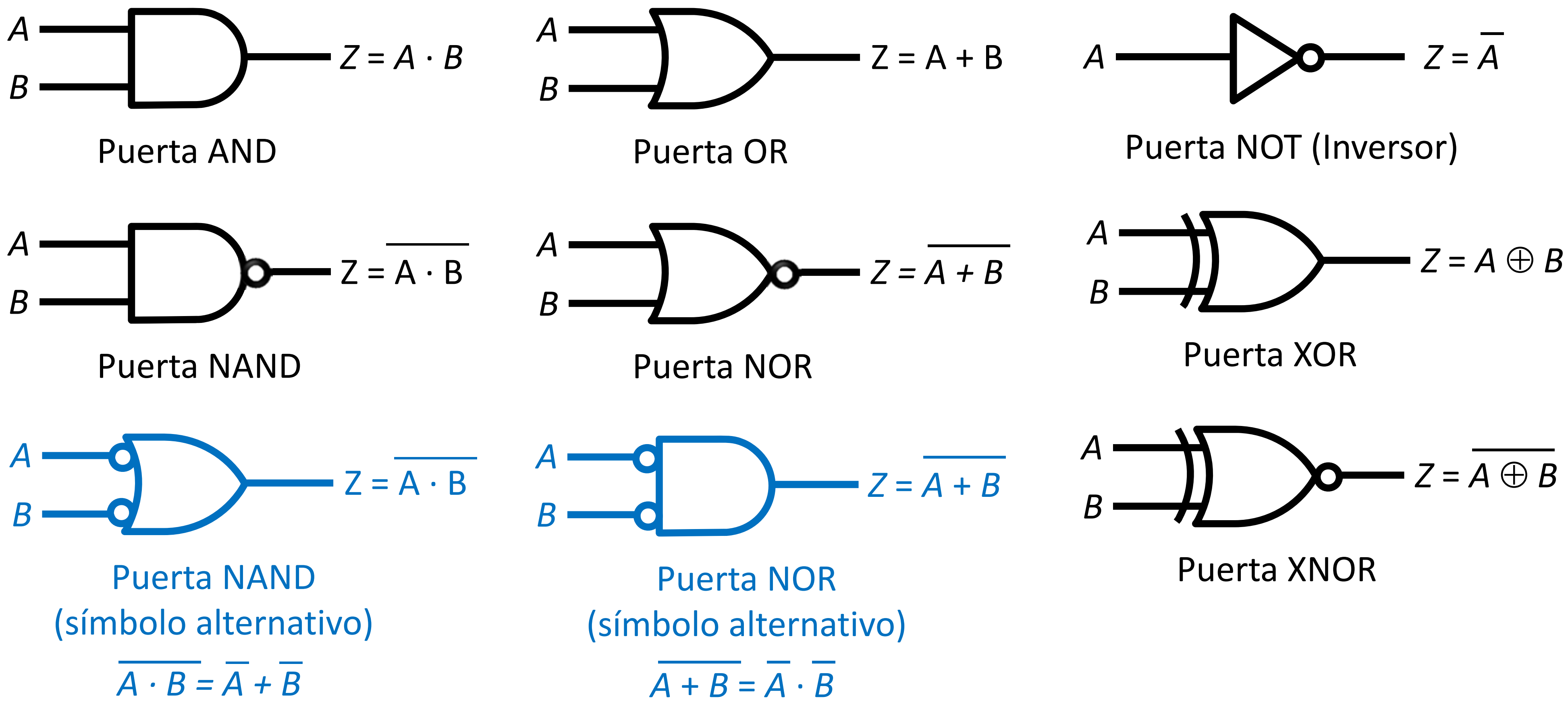
- ✓ Puertas lógicas.
- ✓ Conjuntos universales de puertas.
- ✓ Síntesis con puertas AND-OR-NOT.
- ✓ Síntesis con puertas NAND.
- ✓ Análisis de redes de puertas.
- ✓ Aspectos tecnológicos.
- ✓ Espacio de diseño y trade-offs.

Transparencias basadas en los libros:

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

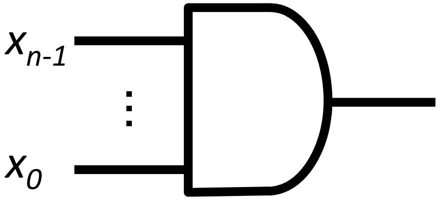
Puertas lógicas

- Dispositivo que realiza físicamente una función de conmutación sencilla.



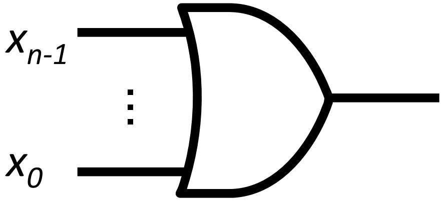
Puertas lógicas

- Existen puertas con mayor número de entradas:



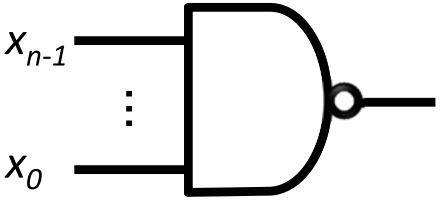
$z = x_{n-1} \cdot \ldots \cdot x_0$

Puerta AND de *n* entradas



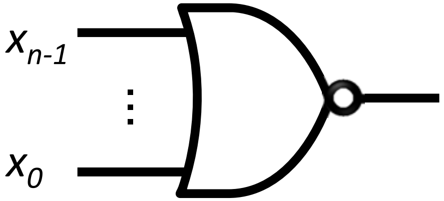
$z = x_{n-1} + \ldots + x_0$

Puerta OR de *n* entradas



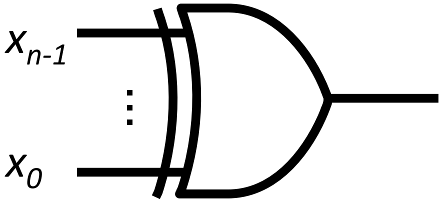
$z = \overline{x_{n-1} \cdot \ldots \cdot x_0}$

Puerta NAND de *n* entradas



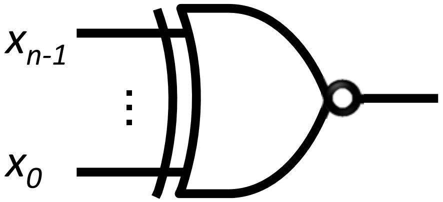
$z = \overline{x_{n-1} + \ldots + x_0}$

Puerta NOR de *n* entradas



$z = x_{n-1} \oplus \ldots \oplus x_0$

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

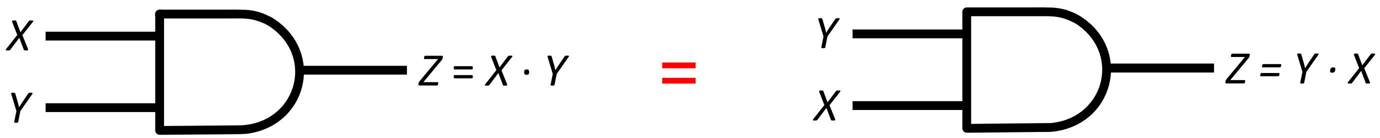


$z = \overline{x_{n-1} \oplus \ldots \oplus x_0}$

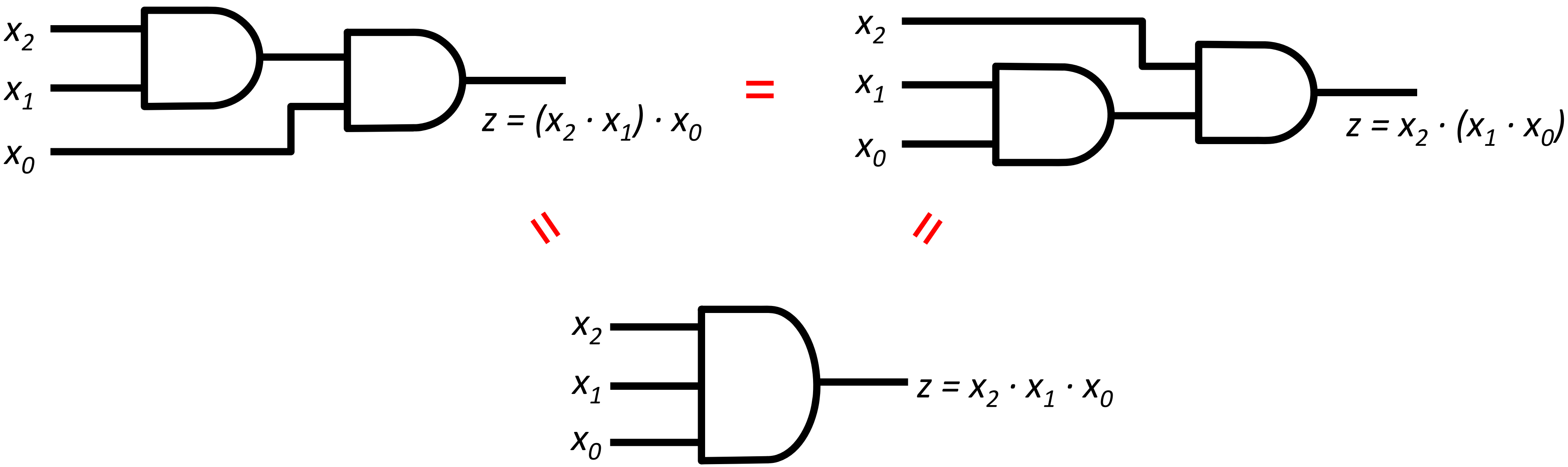
Puerta XNOR de *n* entradas
(z=1 si el número de x_i=1 es par)

Puertas lógicas

- Todas ellas son conmutativas:

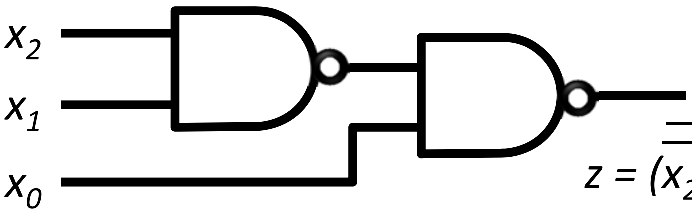


- AND, OR, XOR y XNOR son asociativas:



Puertas lógicas

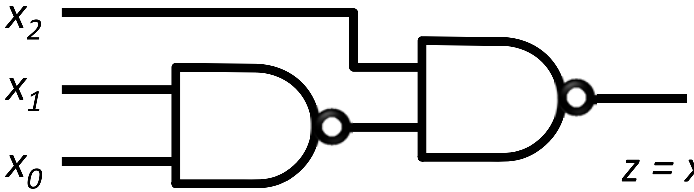
- Pero NAND y NOR no son asociativas.



$$z = \overline{\overline{(x_2 \cdot x_1)} \cdot x_0} = (x_2 \cdot x_1) + \overline{x_0} = \sum m(0,2,4,6,7)$$

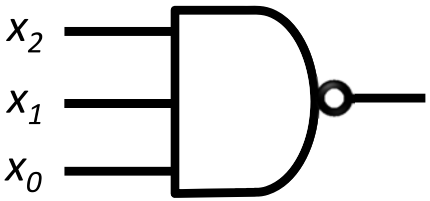
$\neq$

$\neq$



$$z = \overline{x_2 \cdot \overline{(x_1 \cdot x_0)}} = \overline{x_2} + (x_1 \cdot x_0) = \sum m(0,1,2,3,7)$$

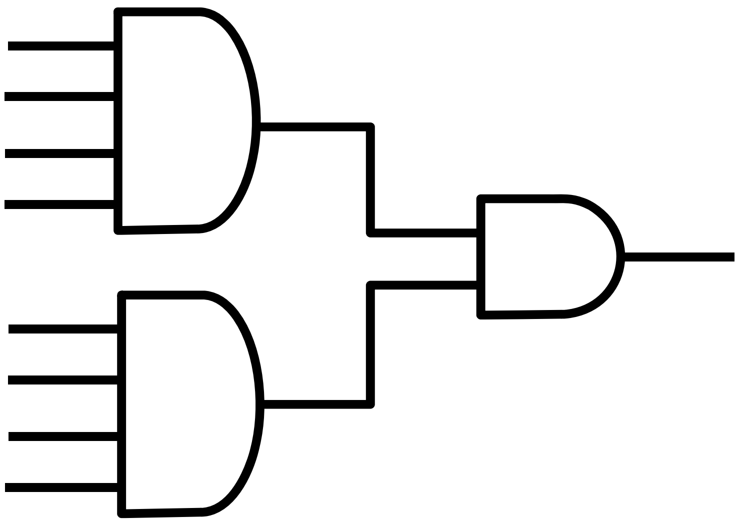
$\neq$



$$z = \overline{x_2 \cdot x_1 \cdot x_0} = \overline{x_2} + \overline{x_1} + \overline{x_0} = \sum m(0,1,2,3,4,5,6)$$

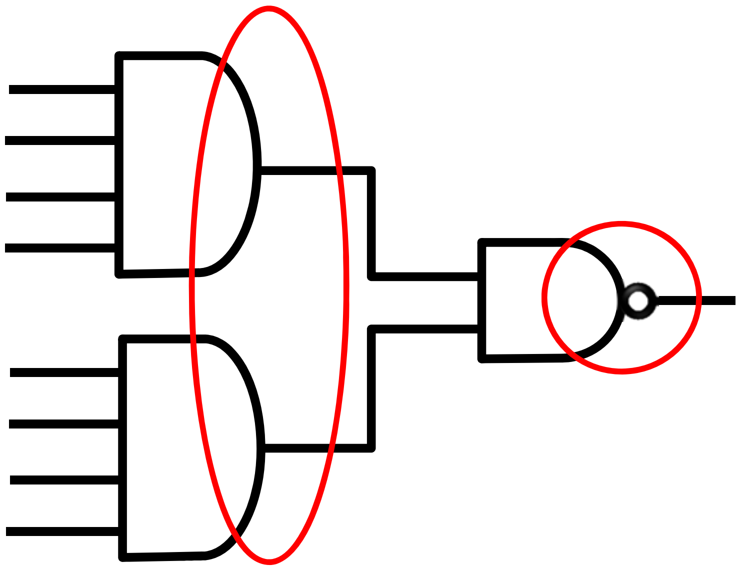
Puertas lógicas

- 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

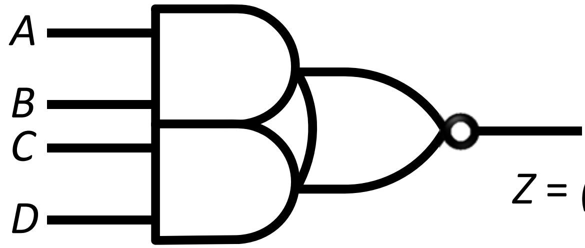


Implementación en árbol

Puerta NAND de 8 entradas

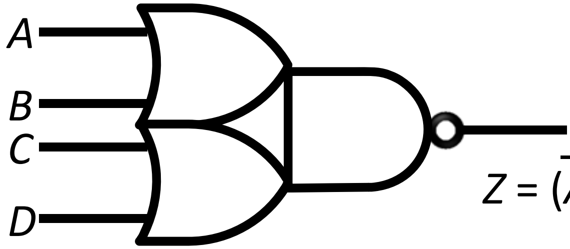
Puertas lógicas

- Existen puertas compuestas:



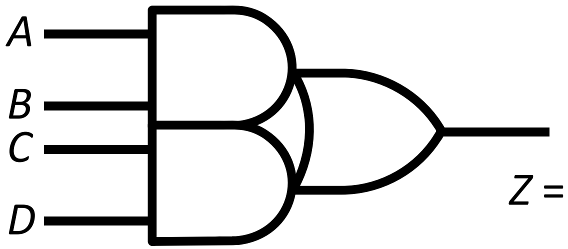
$Z = \overline{(A \cdot B) + (C \cdot D)}$

Puerta AOI 2/2



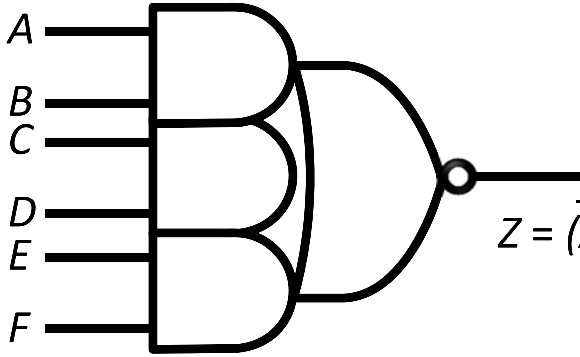
$Z = \overline{(A + B) \cdot (C + D)}$

Puerta OAI 2/2



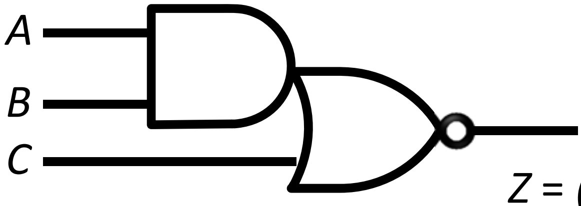
$Z = (A \cdot B) + (C \cdot D)$

Puerta AO 2/2



$Z = \overline{(A \cdot B) + (C \cdot D) + (E \cdot F)}$

Puerta AOI 2/2/2



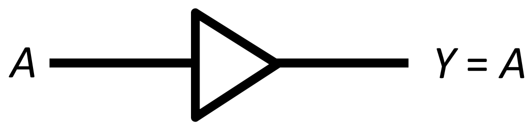
$Z = \overline{(A \cdot B) + C}$

Puerta AOI 2/1

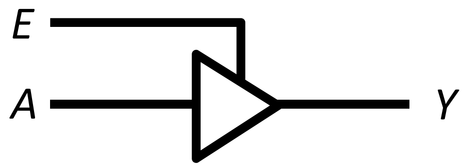
y algunas más...

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.

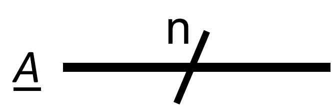
- Nivel de una red: número máximo de módulos que atraviesa cualquier camino que conecte una entrada con una salida
  - cuando la red es de puertas no se suelen contar los inversores.

Interconexiones

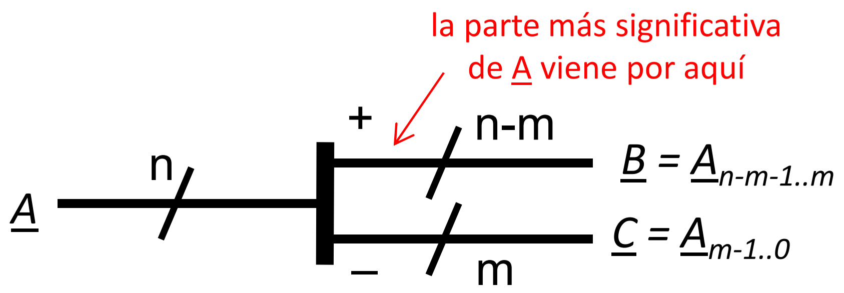
- Al dibujar el esquema de un circuito usaremos alguna notación adicional para las interconexiones:

interconexión de 1 bit

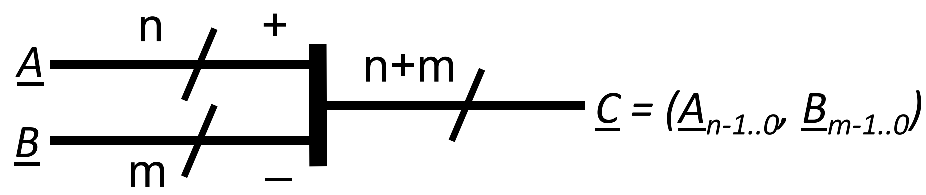
interconexión de 1 bit con un terminal desconectado



n interconexiones de 1 bit en paralelo



n interconexiones en paralelo se dividen en 2 ramas



n y m interconexiones en paralelo se unen en una única rama

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 algebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de puertas :
  - { AND, NOT }

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

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 algebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de puertas :
  - { AND, NOT }

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

  - { 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

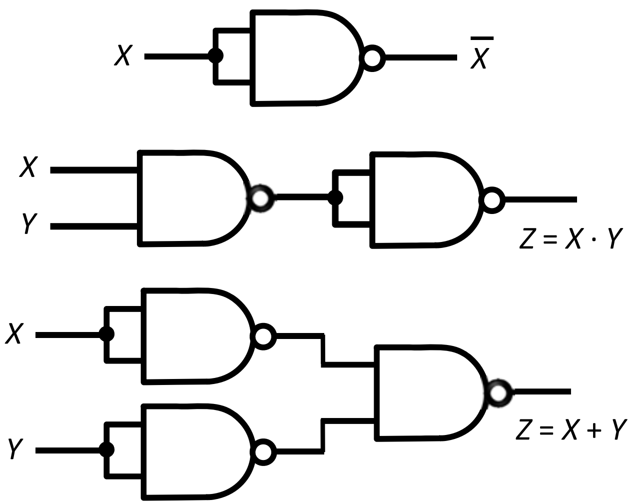
- Se dice que un conjunto de módulos combinacionales es universal si permite implementar cualquier FC
  - Un conjunto lo es, si con sus módulos pueden implementarse todos los operadores del algebra de conmutación.
  - El conjunto de puertas {AND, OR, NOT} es universal.
- Otros conjuntos universales de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

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

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

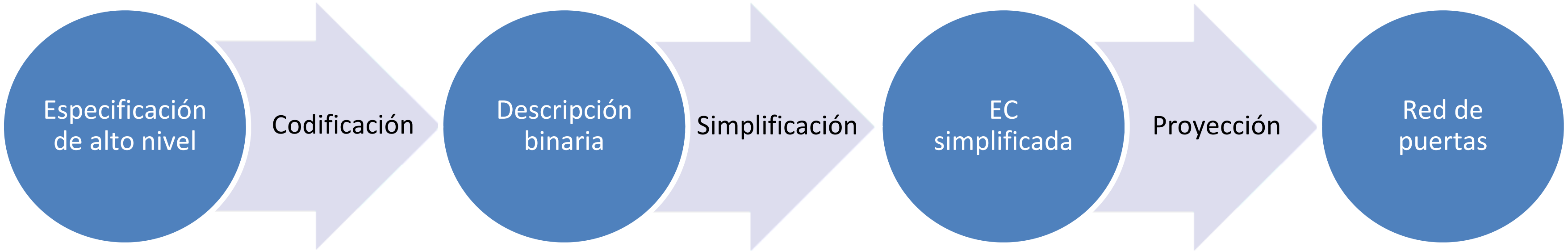
    $$a \cdot b = \overline{\overline{(a \cdot b)}} = \overline{a \uparrow b} = (a \uparrow b) \uparrow (a \uparrow b)$$

    $$a + b = \overline{\overline{(a + b)}} = \overline{(\bar{a} \cdot \bar{b})} = \bar{a} \uparrow \bar{b} = (a \uparrow a) \uparrow (b \uparrow b)$$
  - { OR, NOT }, { NOR }, { XOR, AND } ...



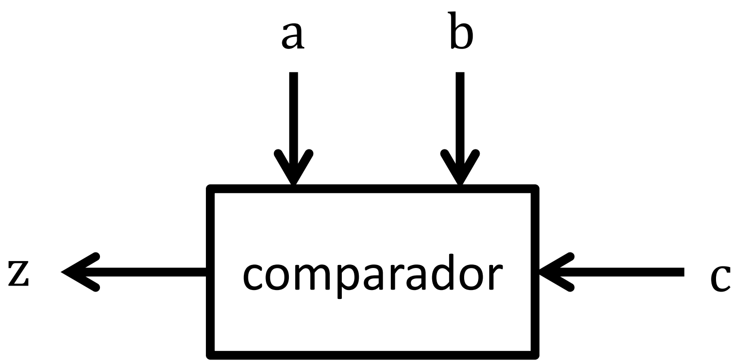
Síntesis de redes de puertas

- 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 multinivel
  - Tienen un número arbitrario de niveles y se reutilizan cálculos intermedios.
  - Para obtenerlas, se parte de un conjunto de SP y se factorizan heurísticamente .

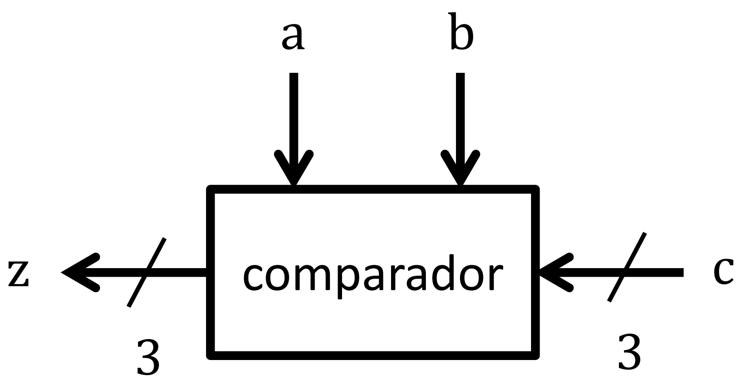
Síntesis de redes AND-OR



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

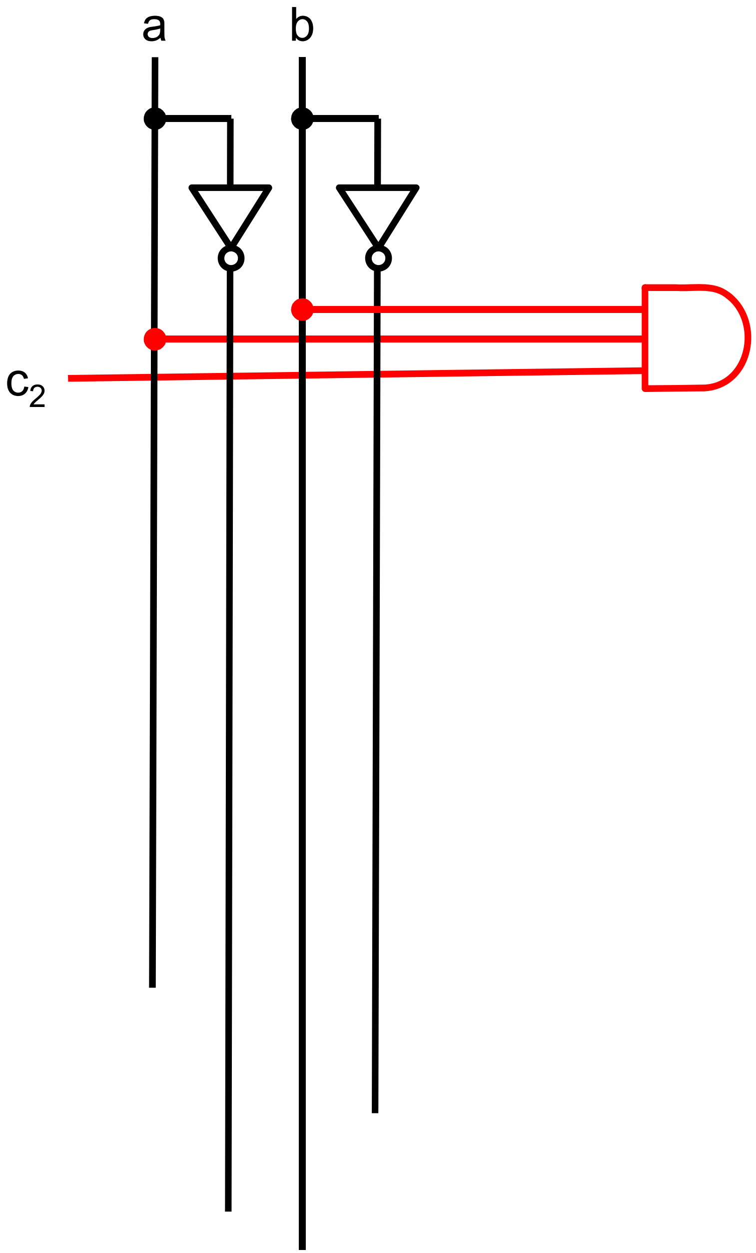
$$z = \begin{cases} aM & \text{si } (a>b) \text{ o } (a=b \text{ y } c=aM) \\ IG & \text{si } (a=b) \text{ y } (c=ig) \\ bM & \text{si } (a<b) \text{ o } (a=b \text{ y } c=bM) \end{cases}$$

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



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

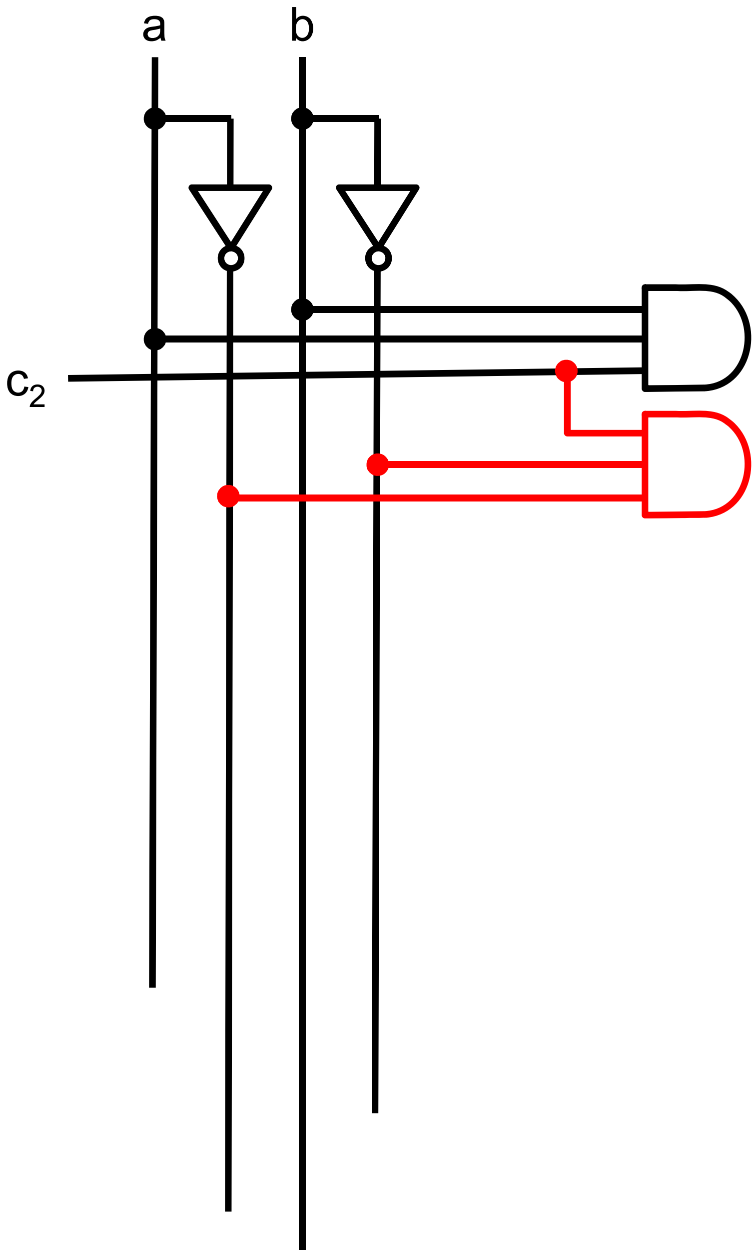
Síntesis de redes AND-OR



Implementación
a 2 niveles

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

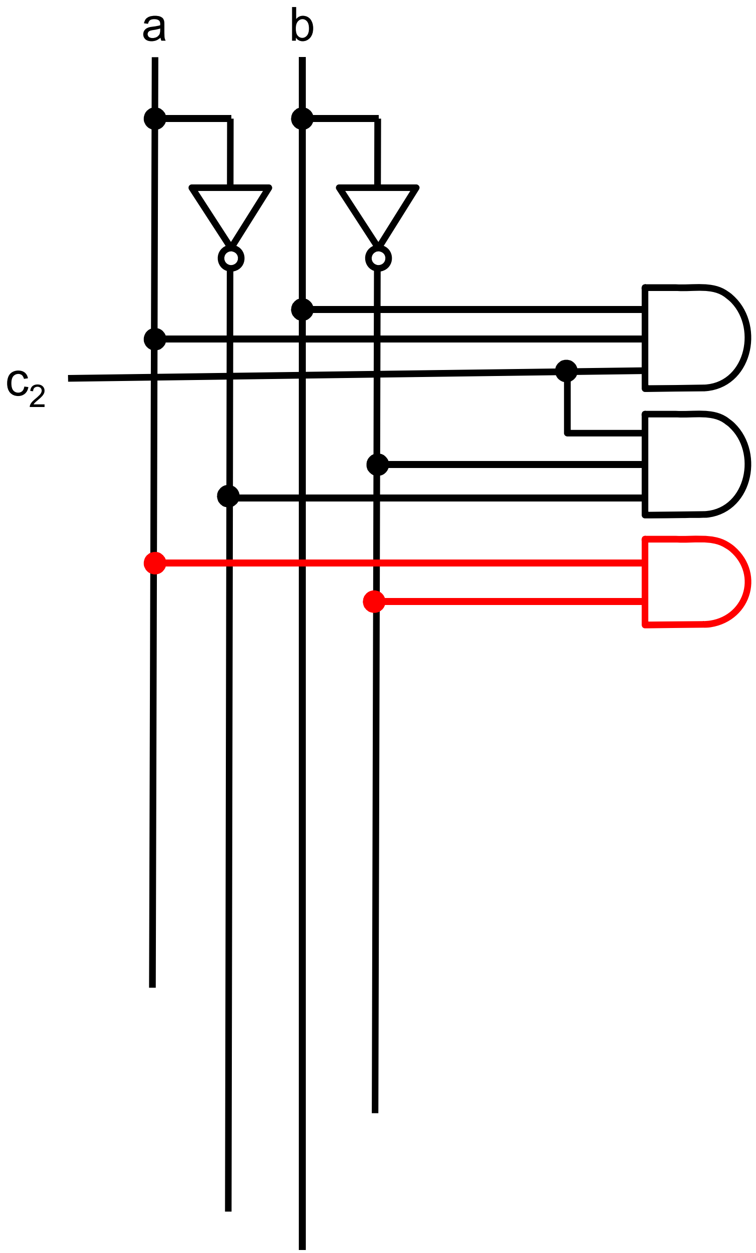
Síntesis de redes AND-OR



Implementación
a 2 niveles

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

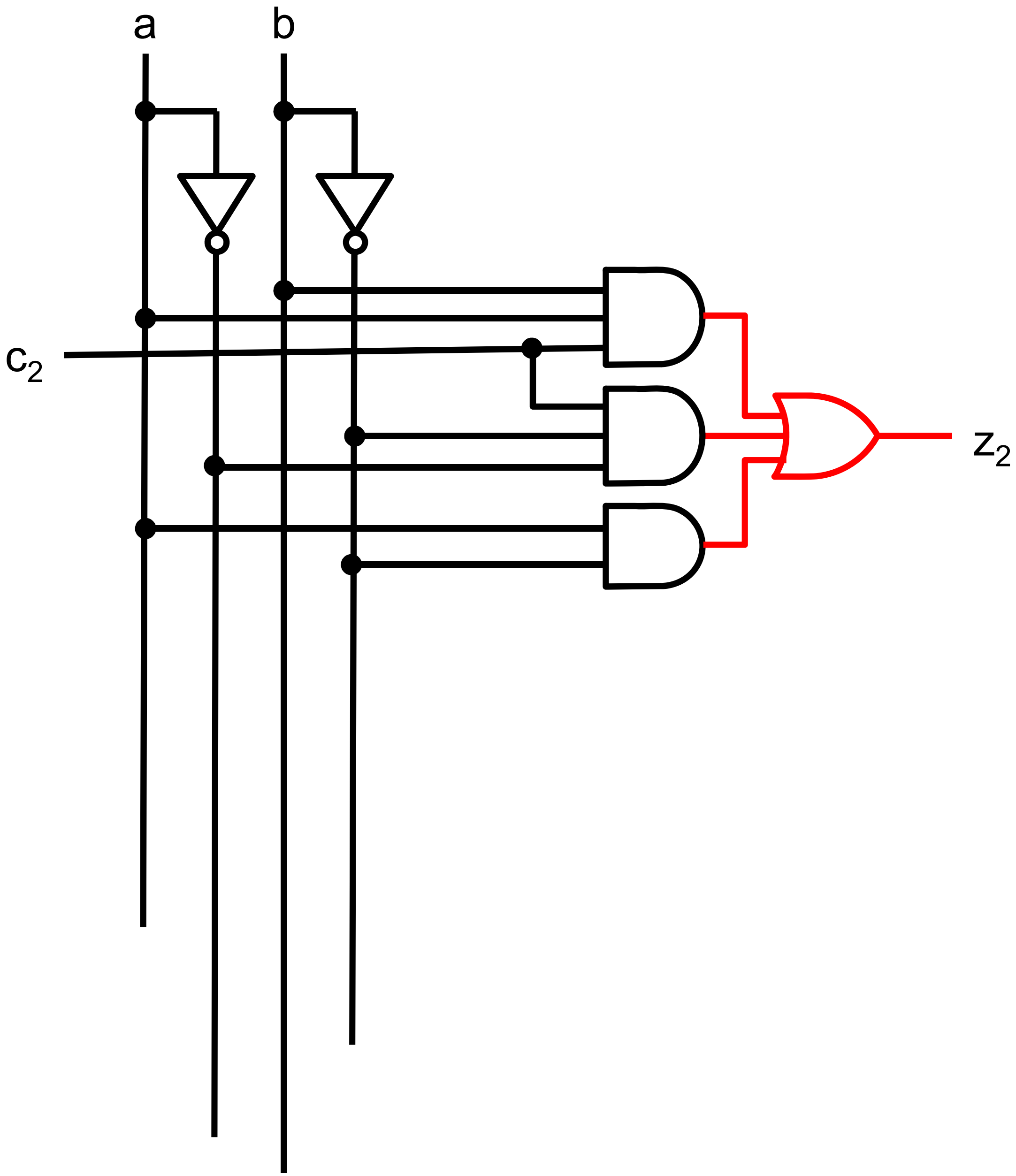
Síntesis de redes AND-OR



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



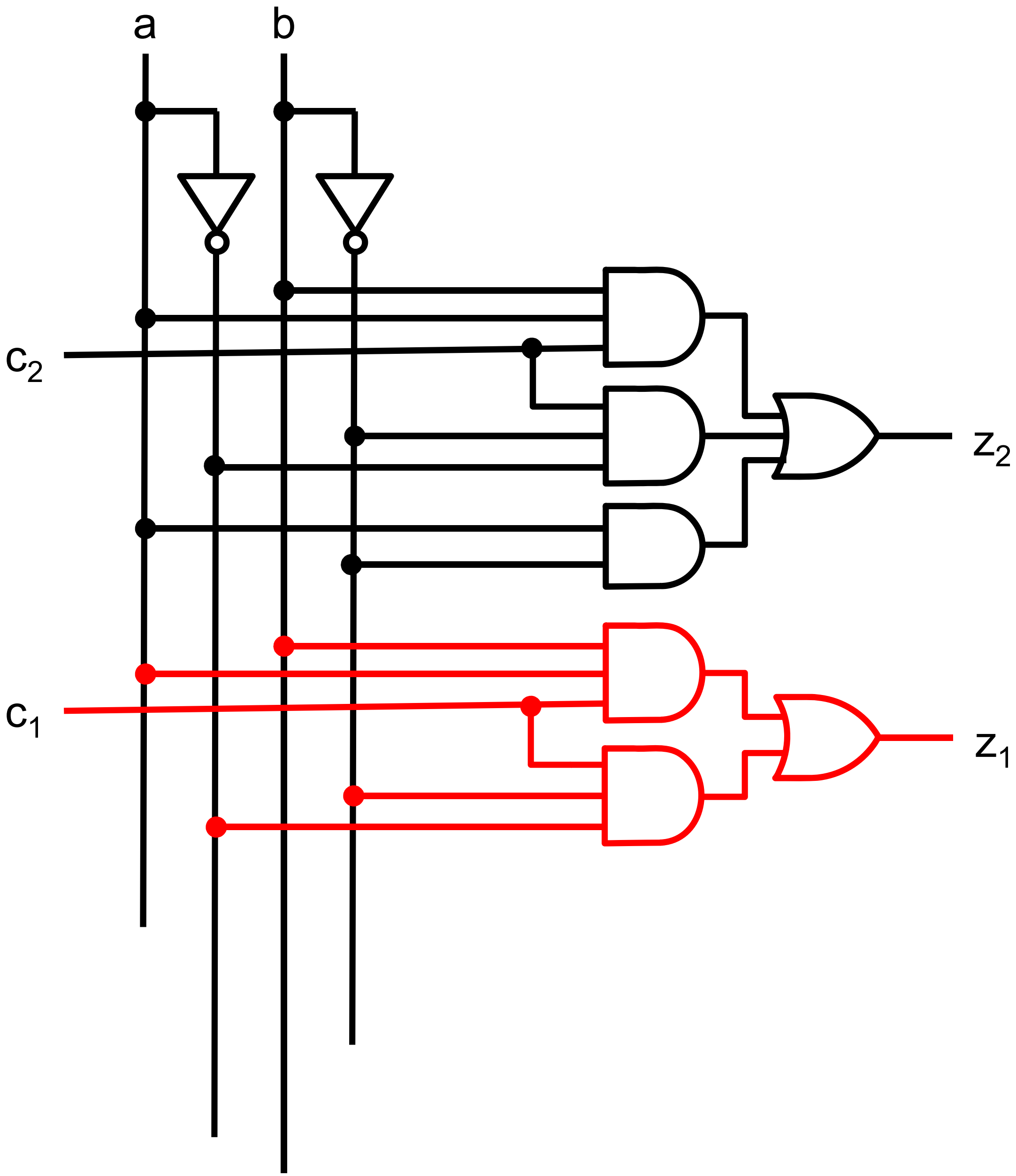
Implementación a 2 niveles

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

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

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

Síntesis de redes AND-OR



Implementación
a 2 niveles

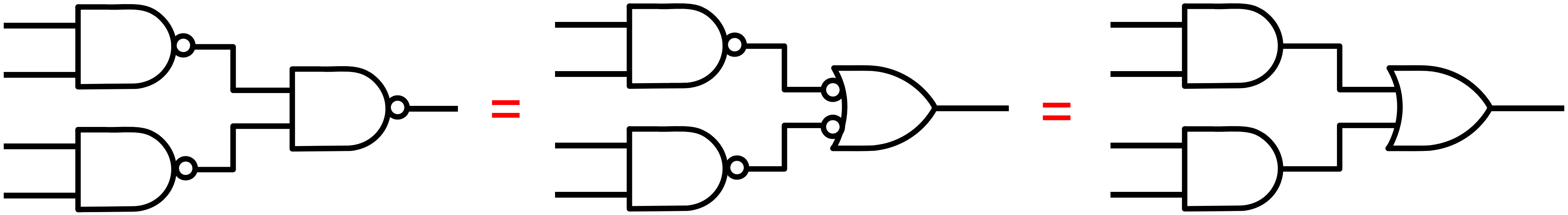
$$z_2 = a\bar{b} + \bar{a}\bar{b}c_2 + abc_2$$

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

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

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.