Electrónica

Tema 5
Circuitos Combinacionales

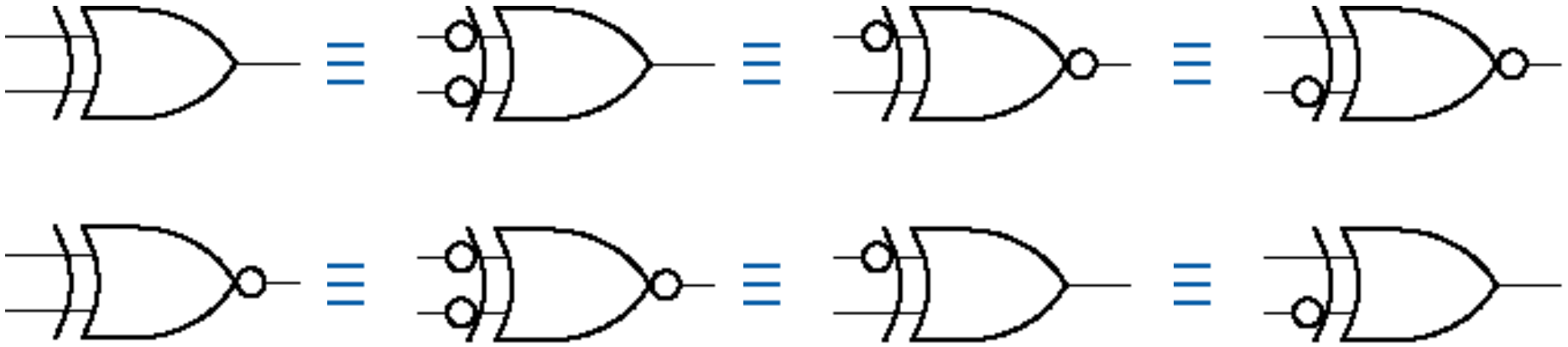
Puertas XOR

- Como una puerta OR, pero excluyendo el caso de que ambas entradas sean 1.

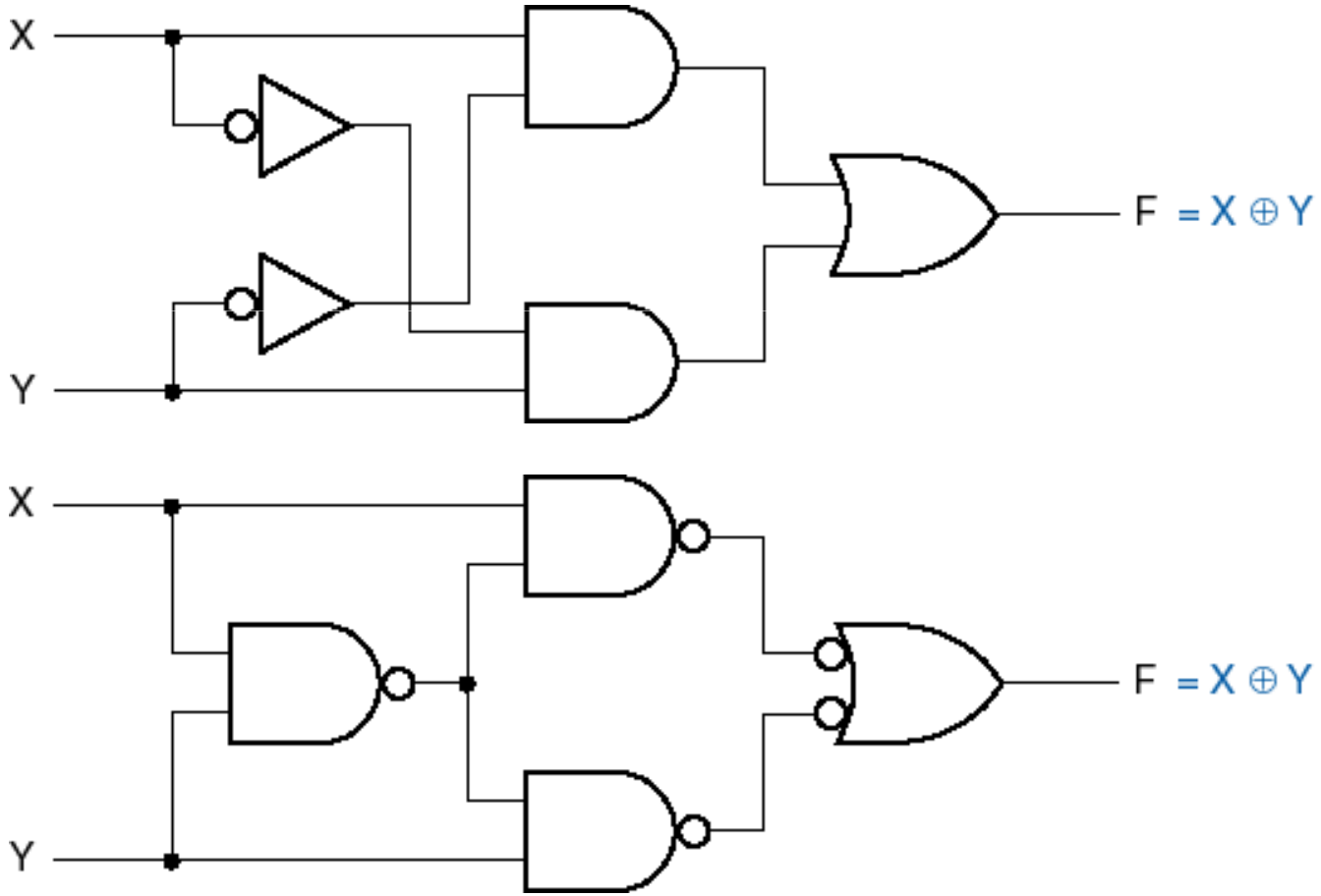
X	Y	X ⊕ Y (XOR)	(X ⊕ Y)' (XNOR)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	0	1

- XNOR: complemento de XOR

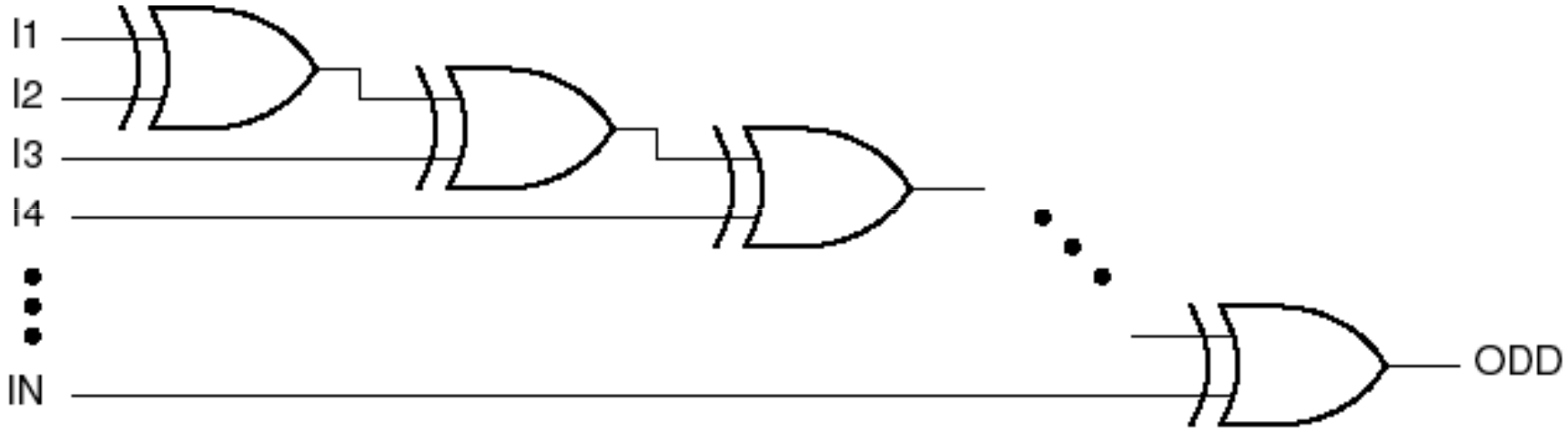
Símbolos XOR y XNOR



Circuitos XOR a nivel de puertas

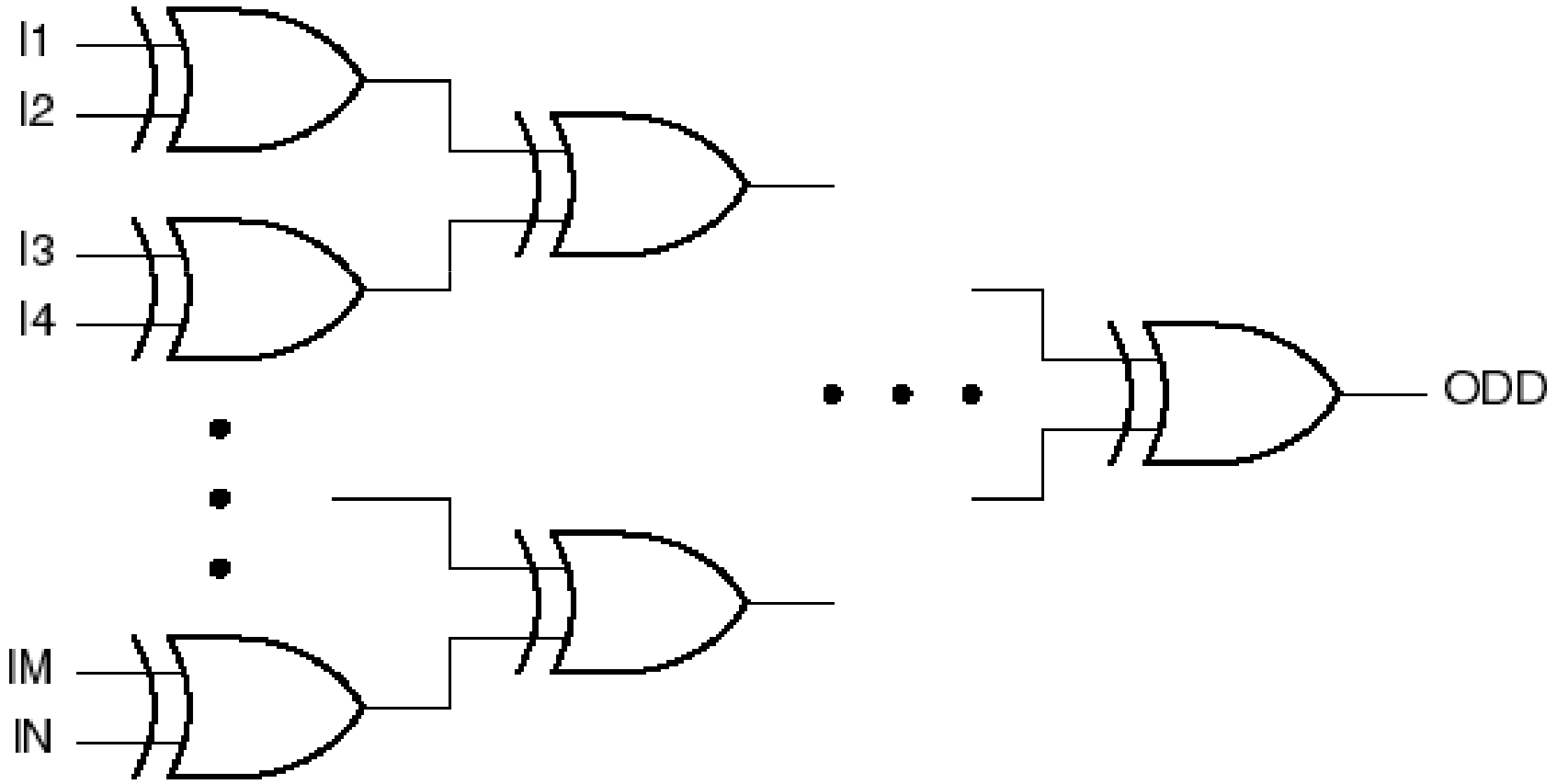


XOR con múltiples entradas



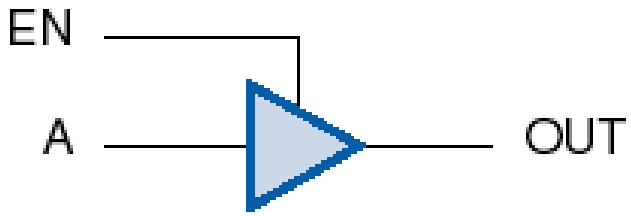
Se usa para generar bits de paridad en sistemas de computación

Árbol de paridad



Buffer Tri-estado

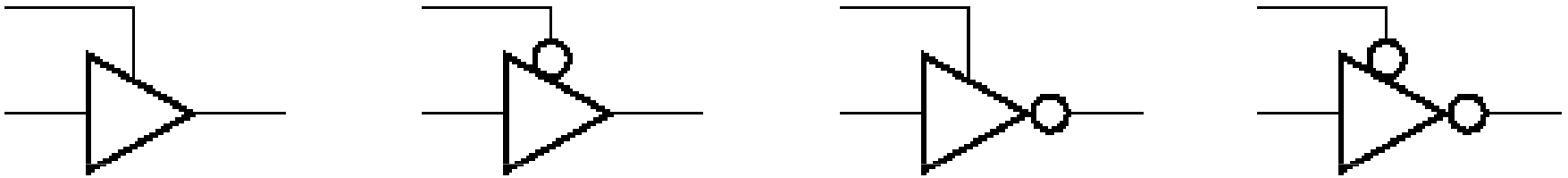
- Salida= LOW, HIGH, o Hi-Z.



EN	A	OUT
L	L	Hi-Z
L	H	Hi-Z
H	L	L
H	H	H

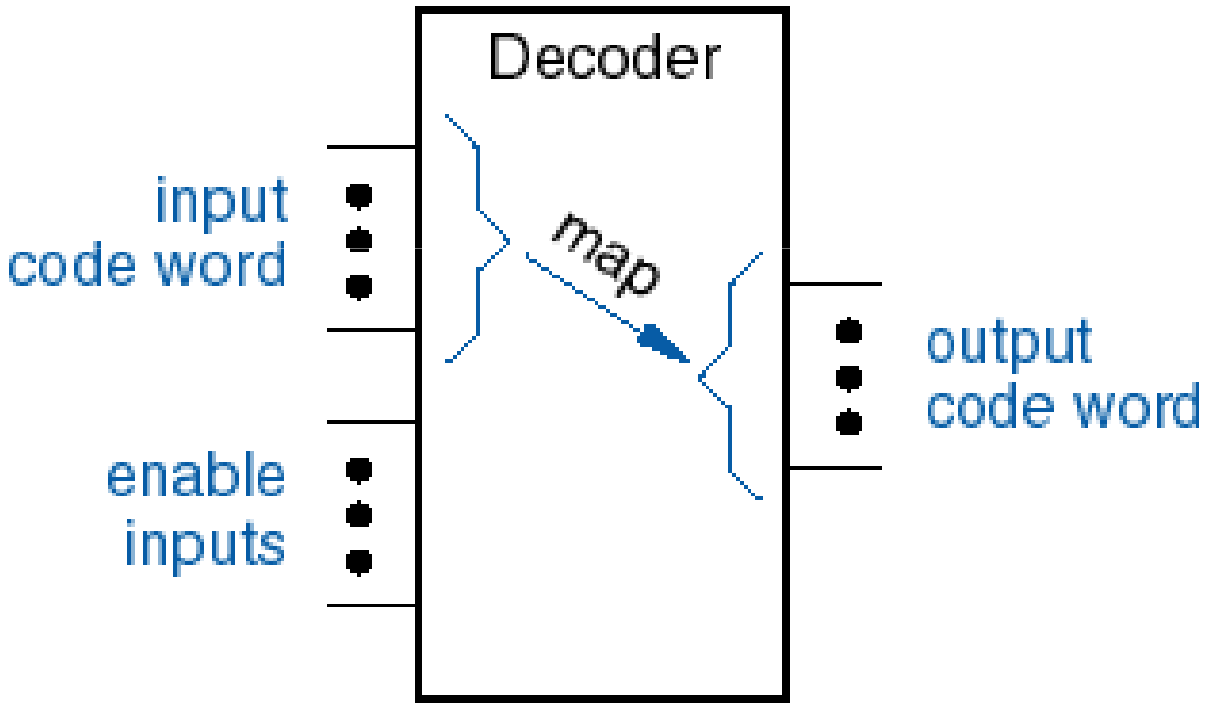
- Puede unir múltiples salidas simultáneamente.

Diferentes representaciones



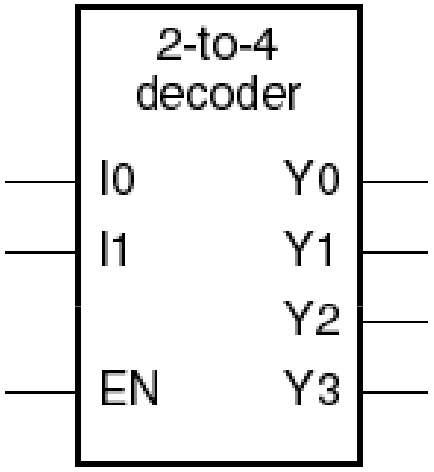
Decodificadores

- Estructura general de un decodificador



- Tiene n entradas y 2^n salidas (2-a-4, 3-a-8, …)

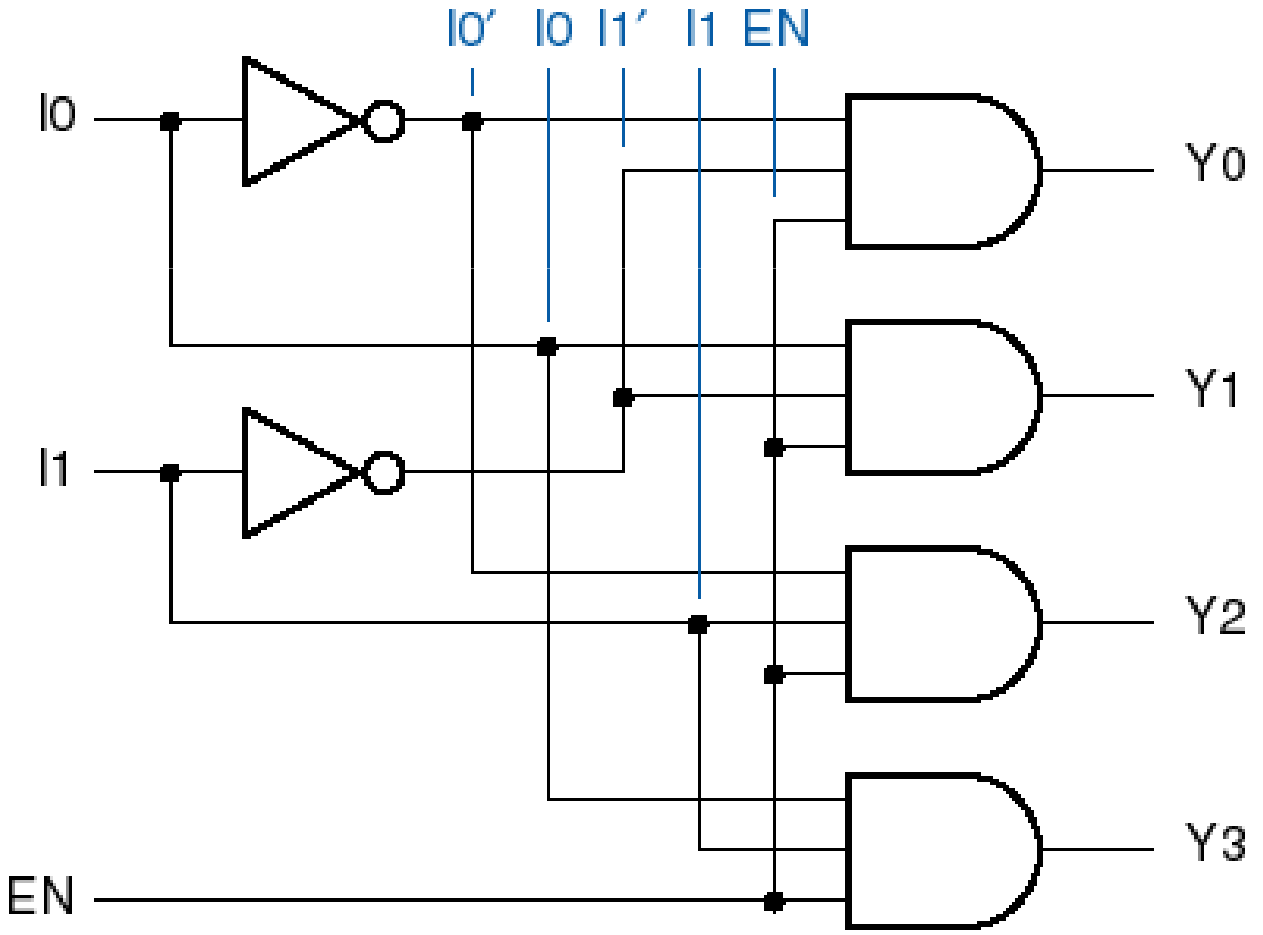
Decodificador 2-a-4



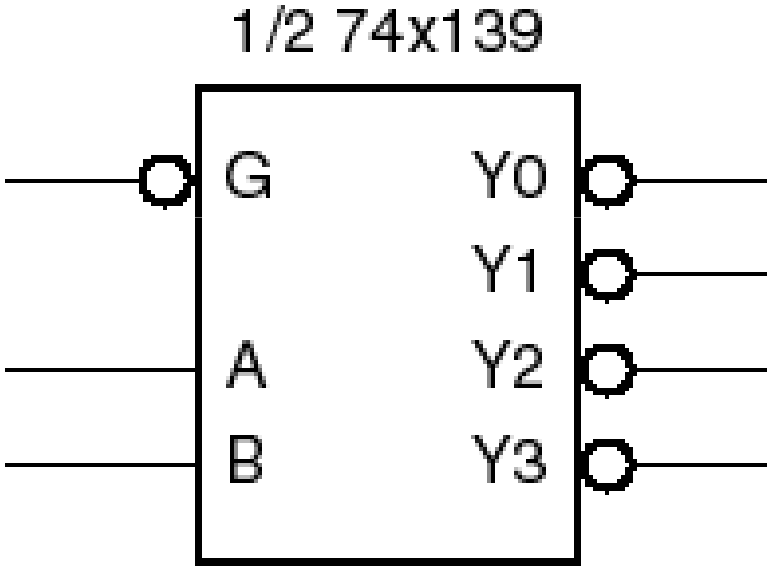
Inputs			*Outputs*			
EN	I1	I0	Y3	Y2	Y1	Y0
0	x	x	0	0	0	0
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0

“x”: Notación “don’t care”

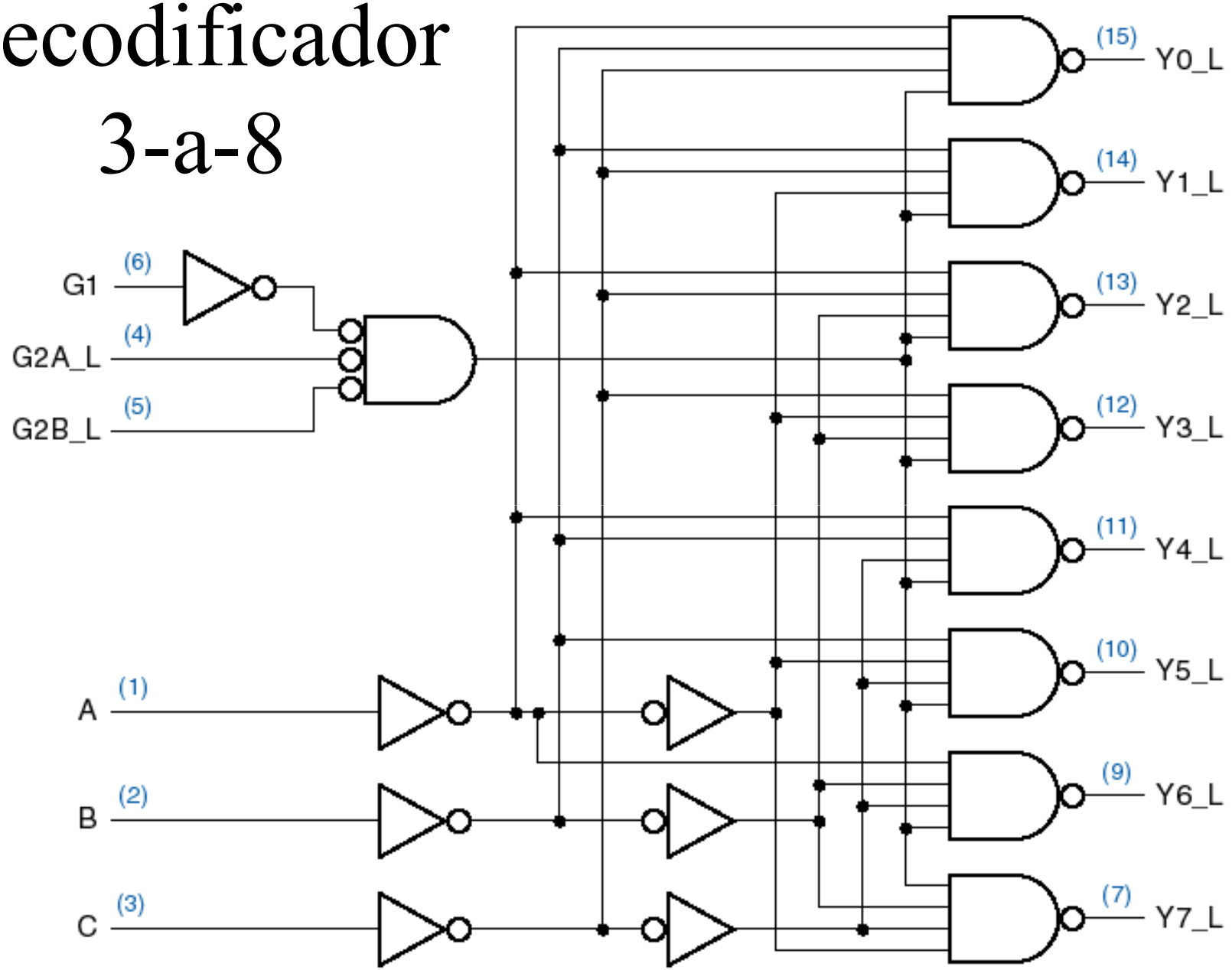
Diagrama lógico del Decodificador 2-a-4



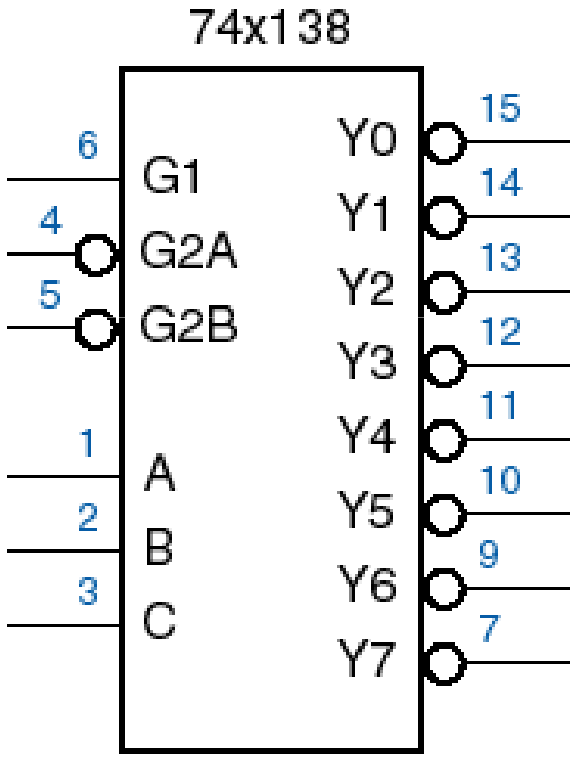
Símbolo del Decodificador 2-a-4



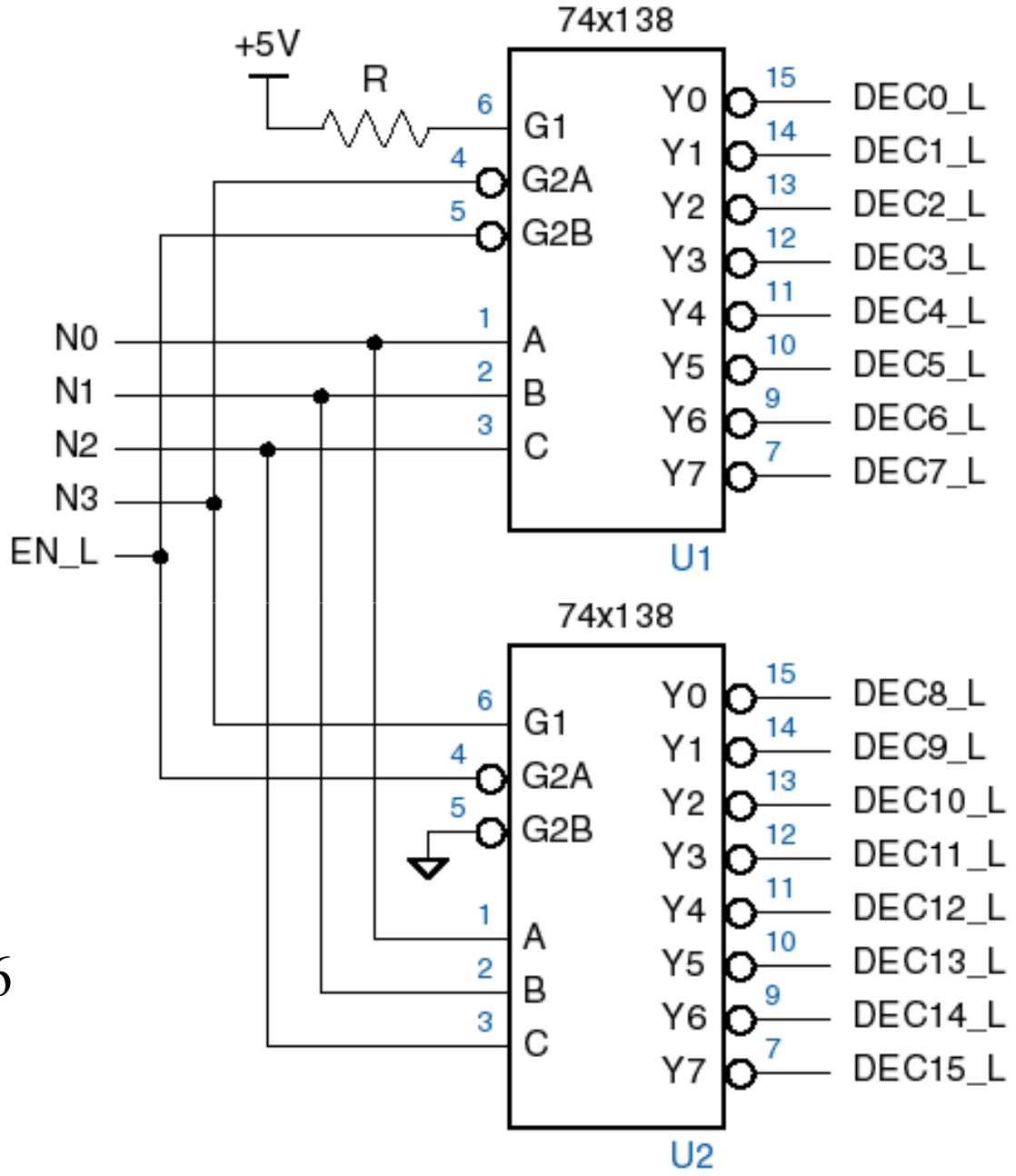
Decodificador 3-a-8



Símbolo del Decodificador 3-a-8

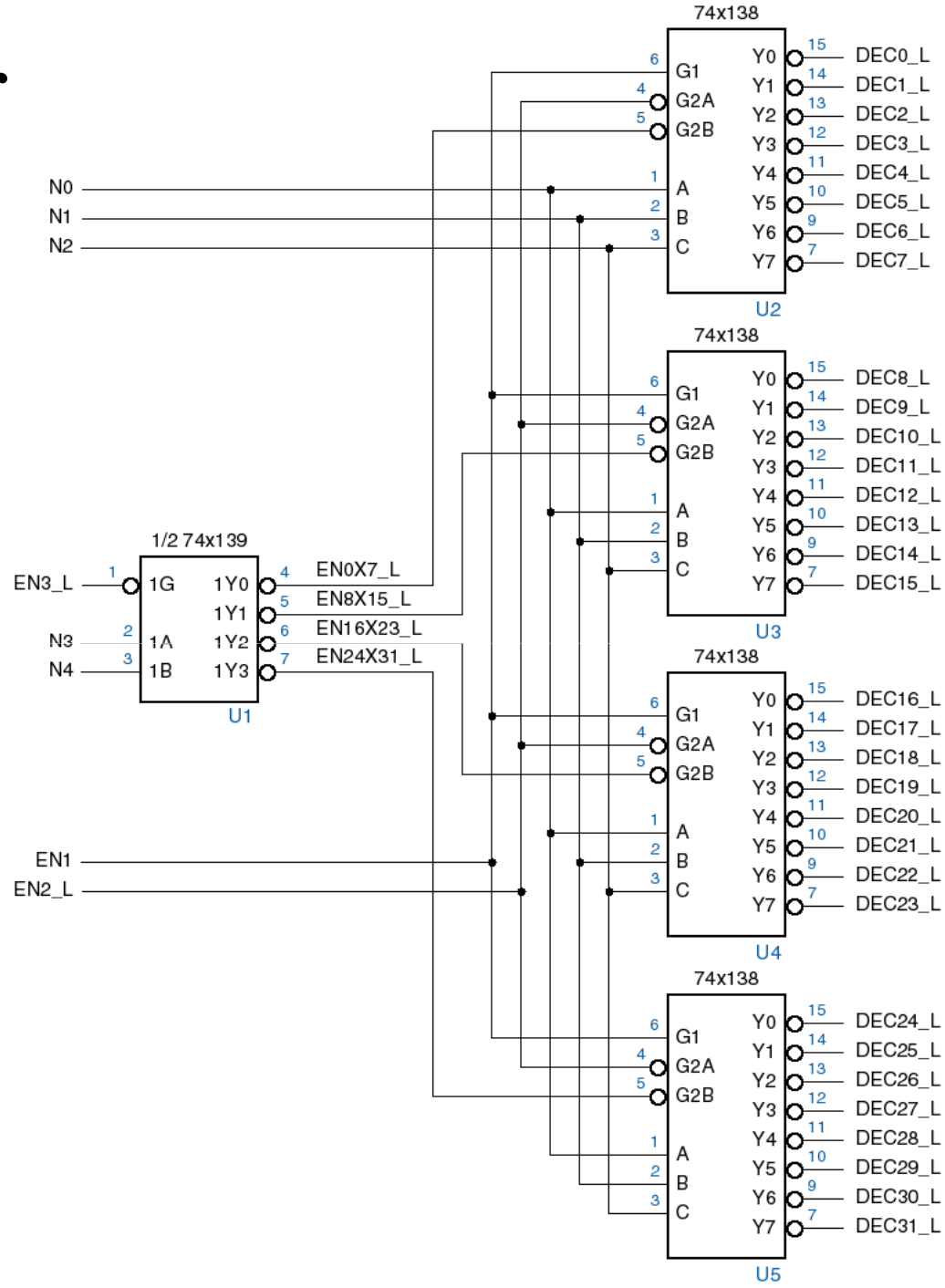


Decodificadores en cascada

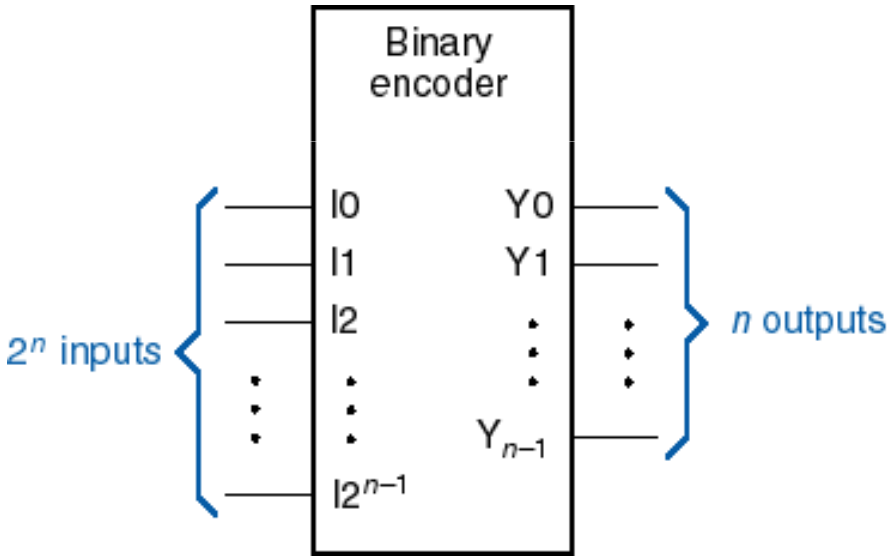


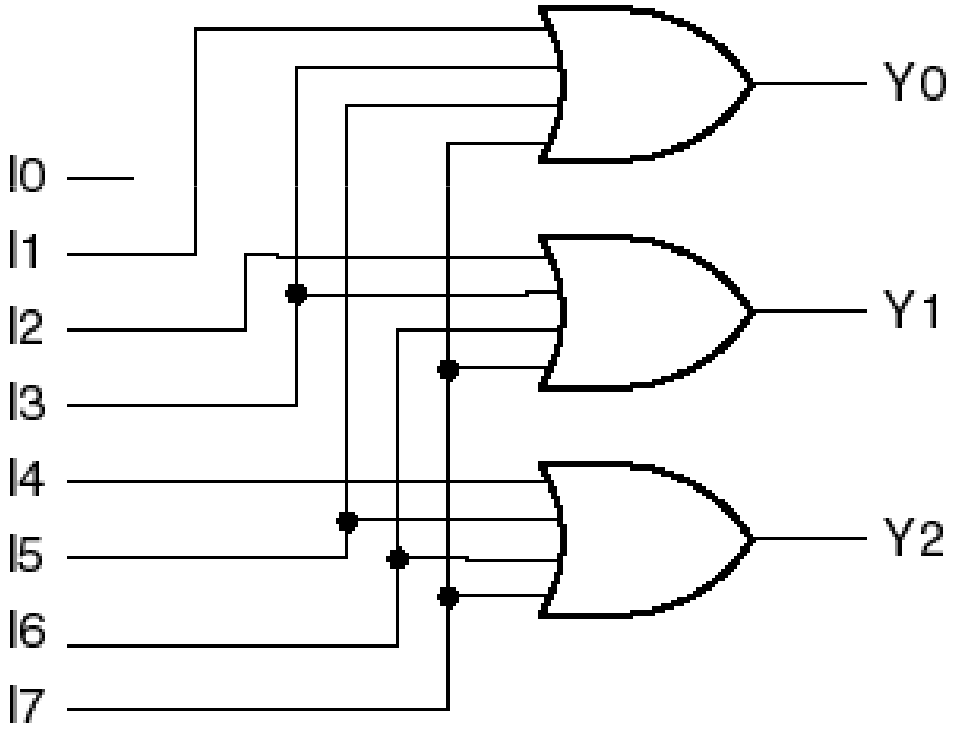
Decodificador 4-a-16

Decodificador 5-a-32

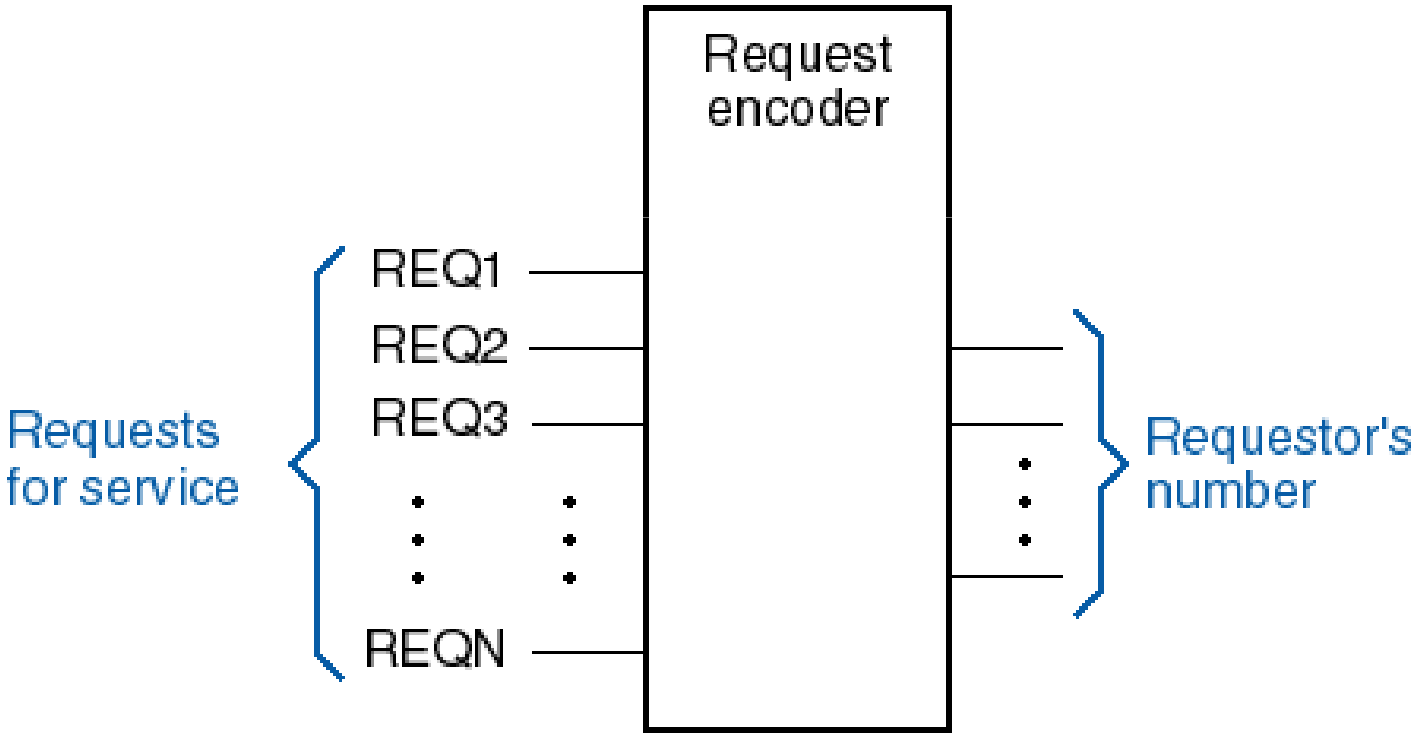


Codificadores

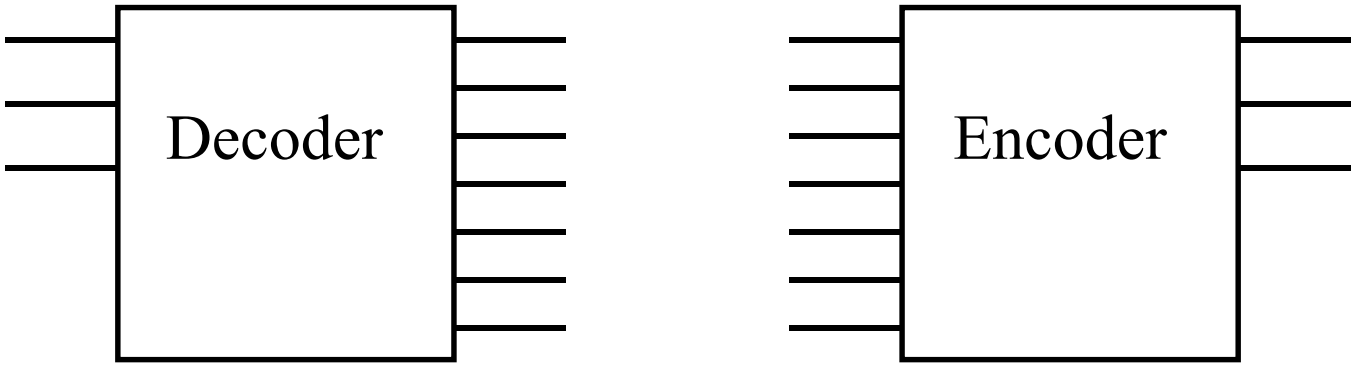




Aplicaciones de prioridad

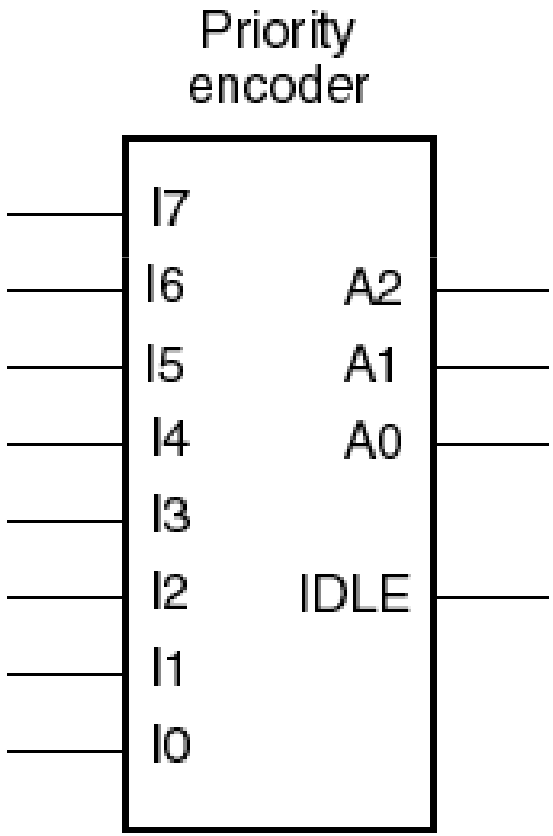


Decodificadores vs. Codificadores



Ambos módulos realizan operaciones opuestas

Codificador de prioridad de 8 entradas



Ecuaciones del codificador de prioridad

$$A2 = H4 + H5 + H6 + H7$$

$$A1 = H2 + H3 + H6 + H7$$

$$A0 = H1 + H3 + H5 + H7$$

$$H7 = I7$$

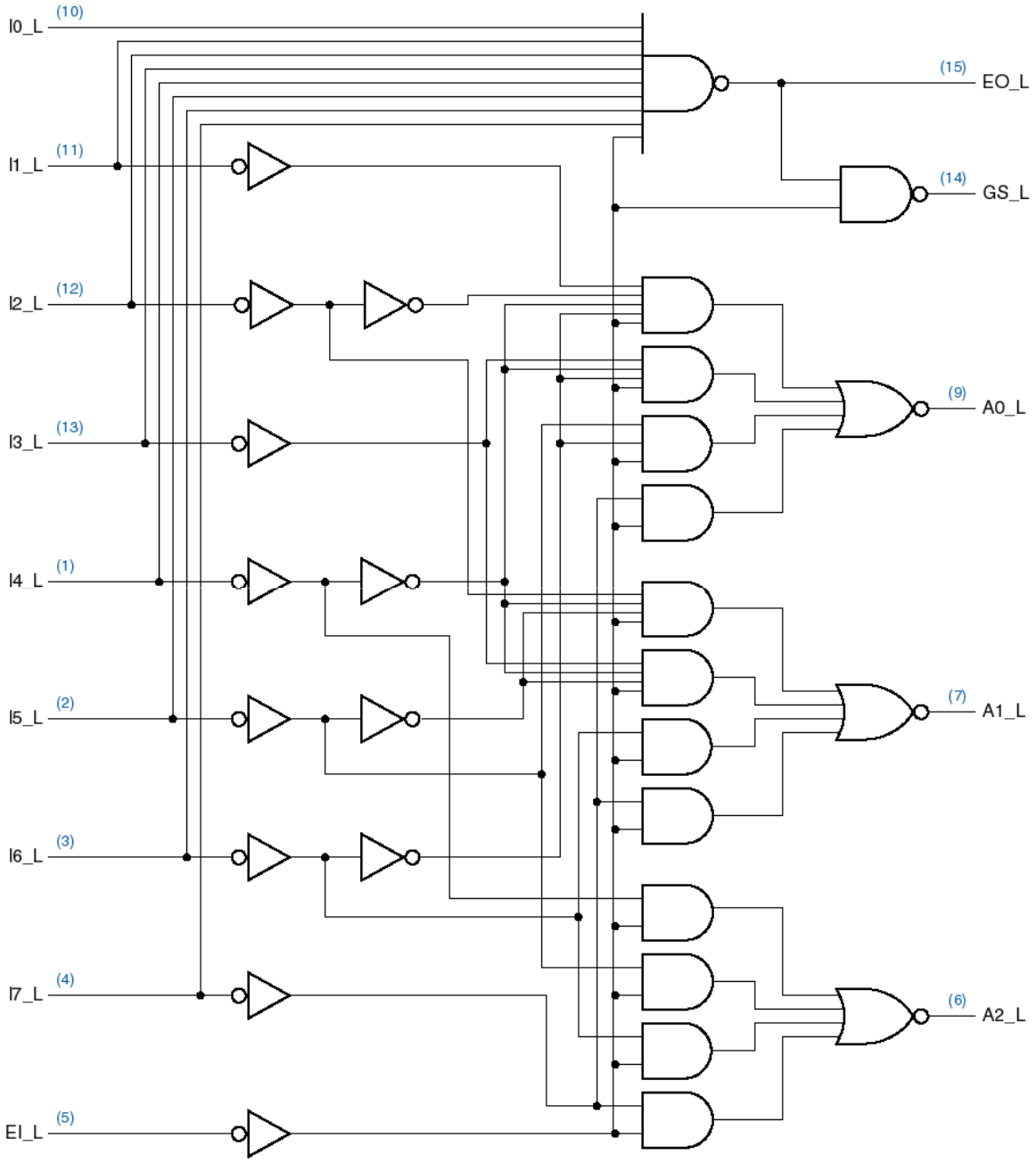
$$H6 = I6 \cdot I7'$$

$$H5 = I5 \cdot I6' \cdot I7'$$

$$IDLE = (I0 + I1 + I2 + I3 + I4 + I5 + I6 + I7)'$$

$$= I0' \cdot I1' \cdot I2' \cdot I3' \cdot I4' \cdot I5' \cdot I6' \cdot I7'$$

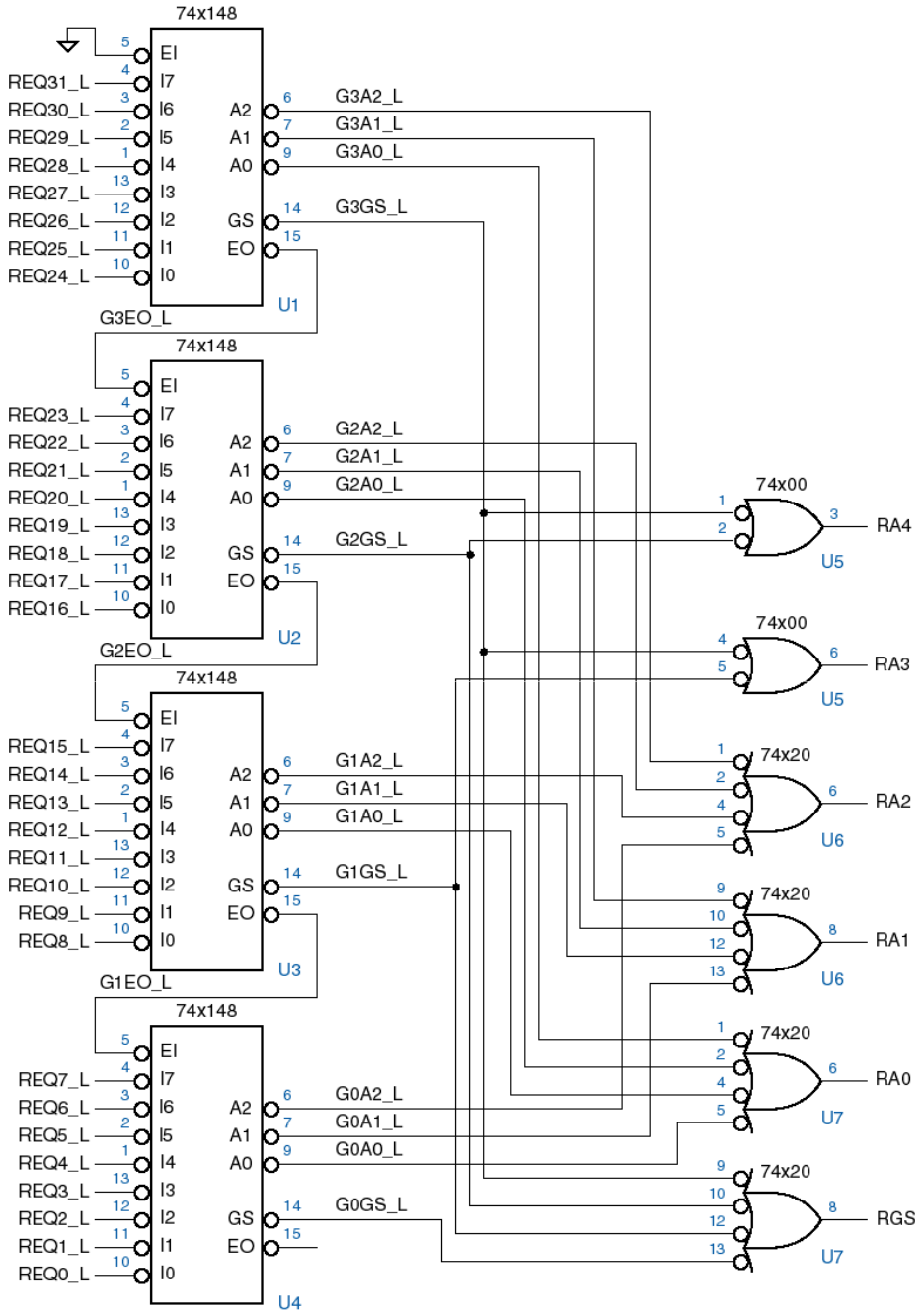
Codificador de 8 entradas



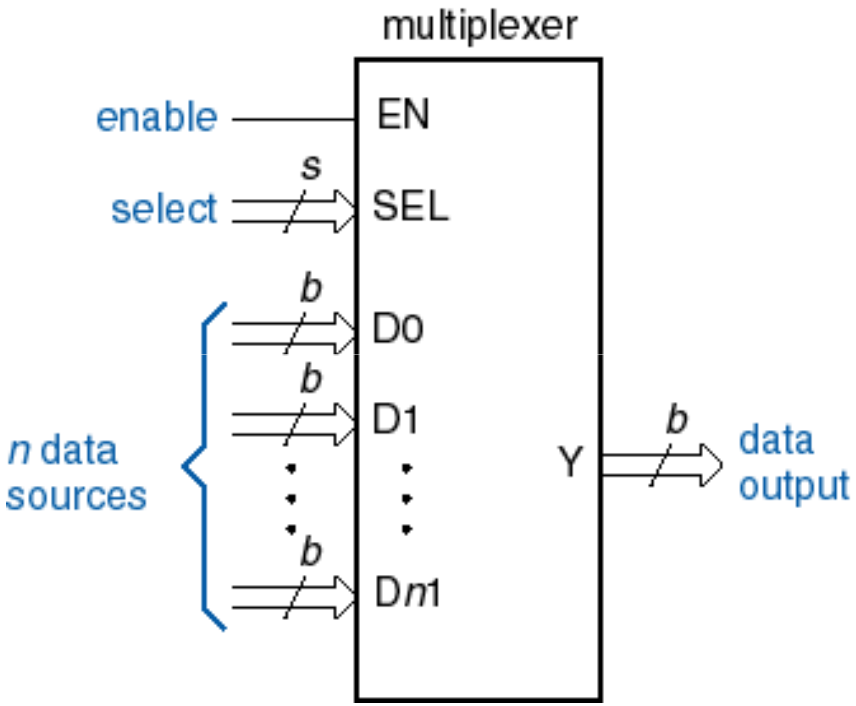
Inputs									*Outputs*				
EI_L	I0_L	I1_L	I2_L	I3_L	I4_L	I5_L	I6_L	I7_L	A2_L	A1_L	A0_L	GS_L	EO_L
1	x	x	x	x	x	x	x	x	1	1	1	1	1
0	x	x	x	x	x	x	x	0	0	0	0	0	1
0	x	x	x	x	x	x	0	1	0	0	1	0	1
0	x	x	x	x	x	0	1	1	0	1	0	0	1
0	x	x	x	x	0	1	1	1	0	1	1	0	1
0	x	x	x	0	1	1	1	1	1	0	0	0	1
0	x	x	0	1	1	1	1	1	1	0	1	0	1
0	x	0	1	1	1	1	1	1	1	1	0	0	1
0	0	1	1	1	1	1	1	1	1	1	1	0	1
0	1	1	1	1	1	1	1	1	1	1	1	1	0

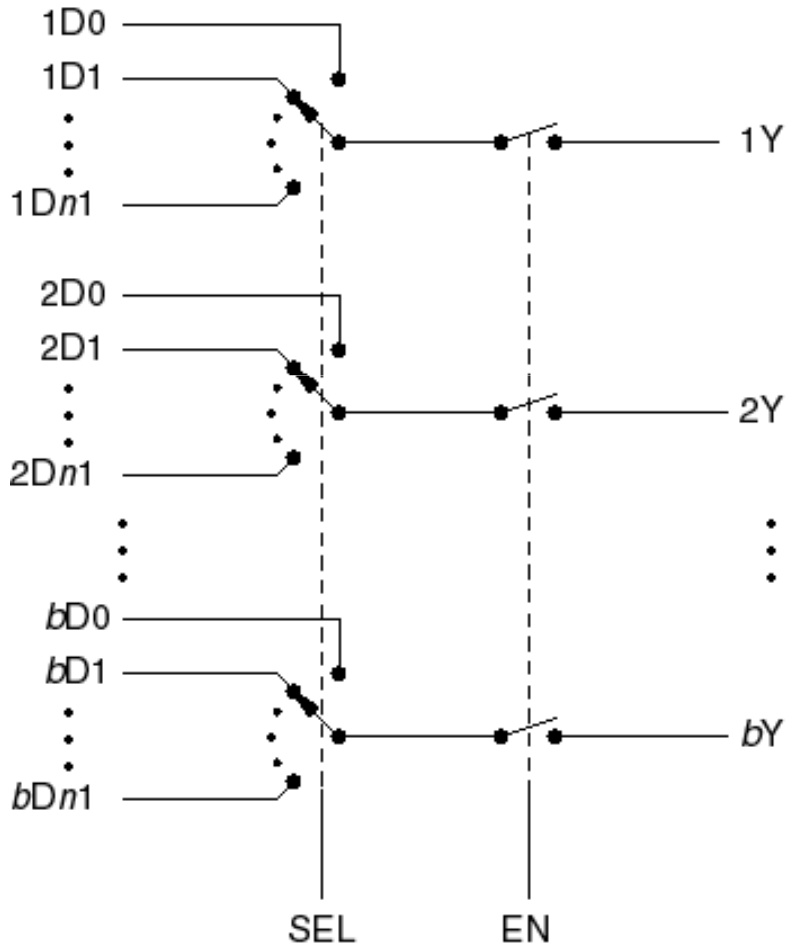
Codificadores de prioridad en cascada

- Codificador de prioridad de 32 entradas



Multiplexores





Multiplexor de 8 entradas

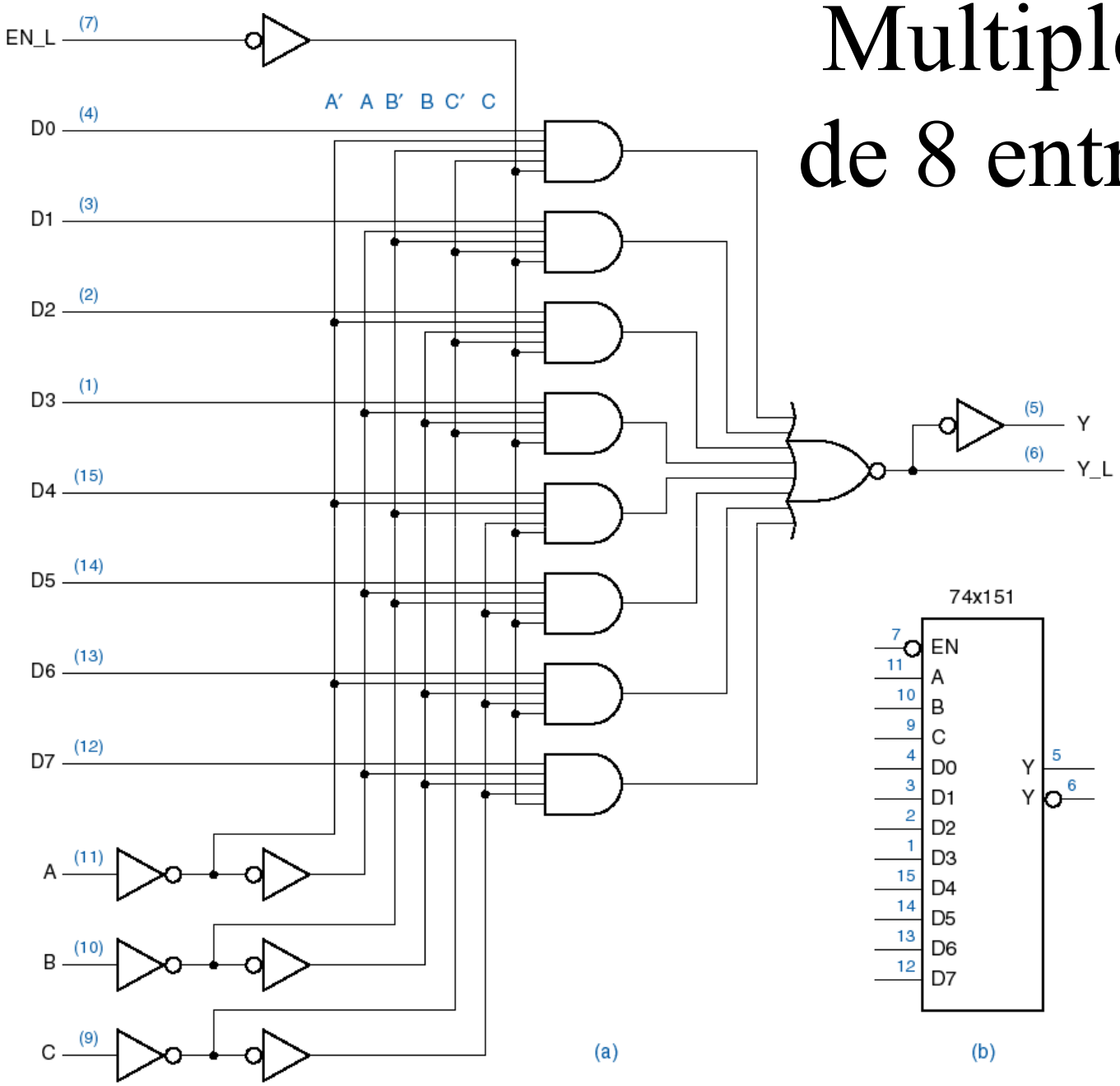
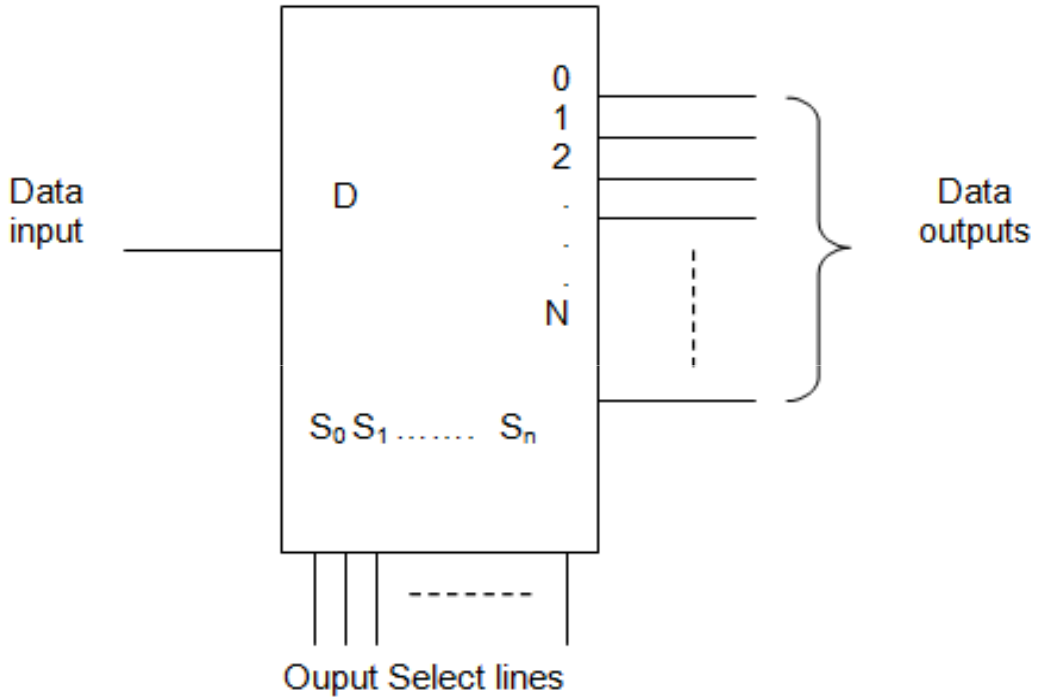


Tabla de verdad

Inputs				*Outputs*	
EN_L	C	B	A	Y	Y_L
1	x	x	x	0	1
0	0	0	0	D0	D0′
0	0	0	1	D1	D1′
0	0	1	0	D2	D2′
0	0	1	1	D3	D3′
0	1	0	0	D4	D4′
0	1	0	1	D5	D5′
0	1	1	0	D6	D6′
0	1	1	1	D7	D7′

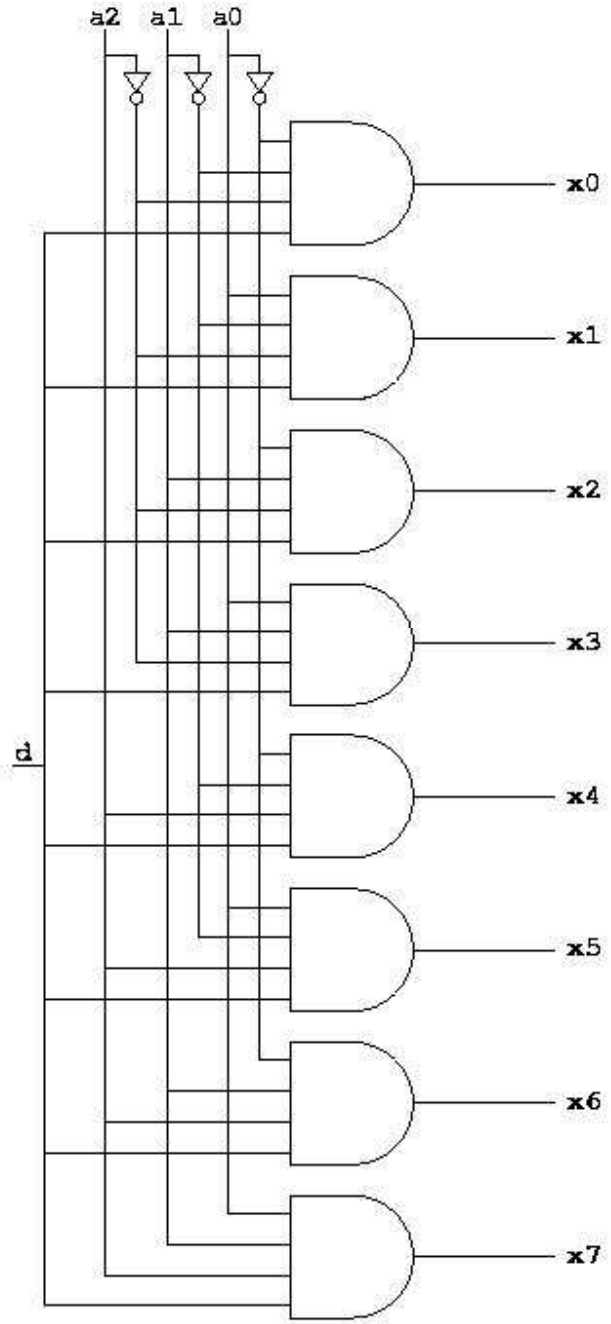
Demultiplexores



Operación opuesta a la del multiplexor:

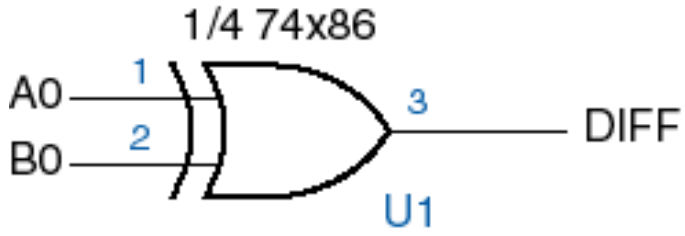
- 1 entrada de datos
- n entradas de selección
- $N=2^n$ salidas de datos

Demultiplexor de 8 salidas

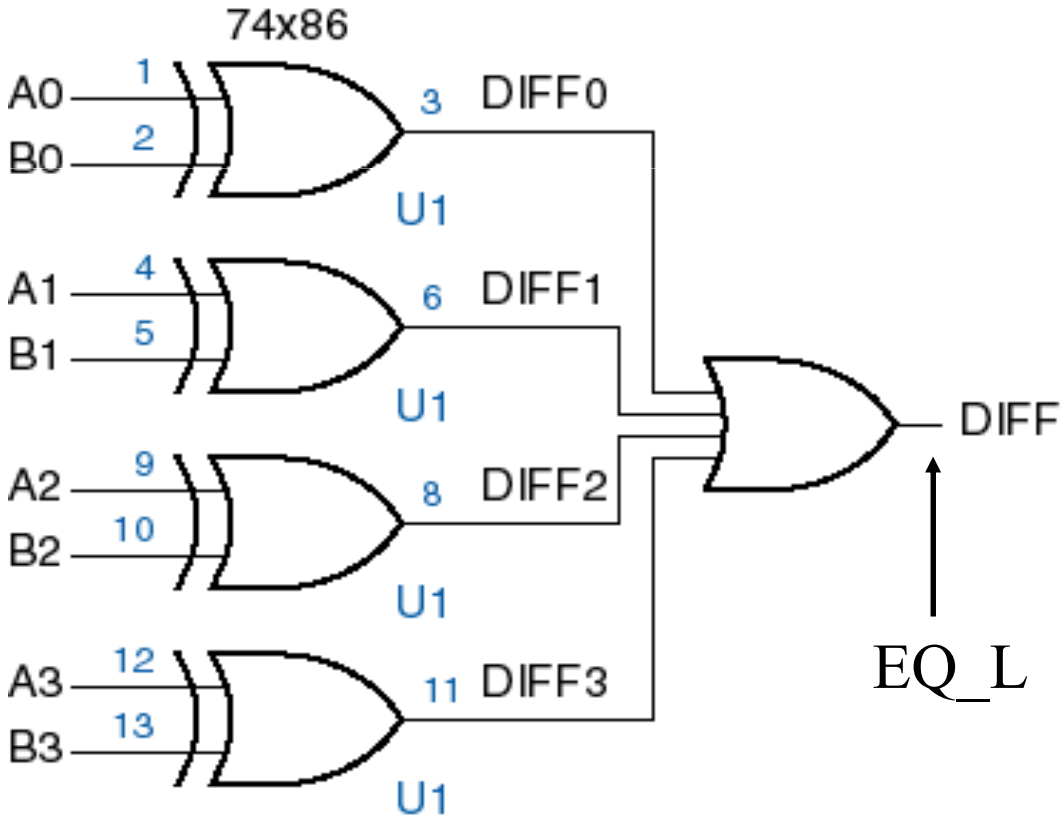


Comparadores

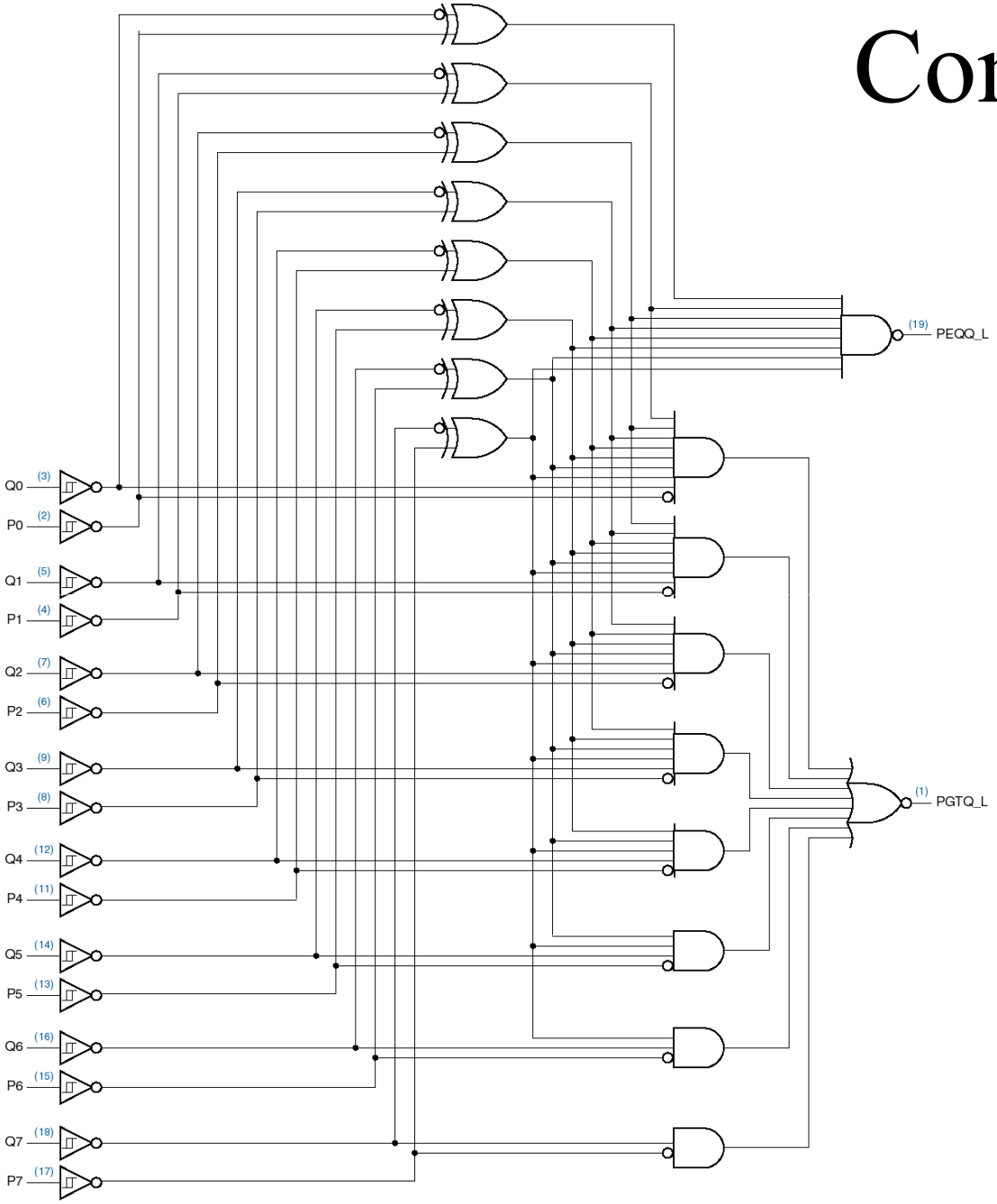
- Comparador de 1 bit

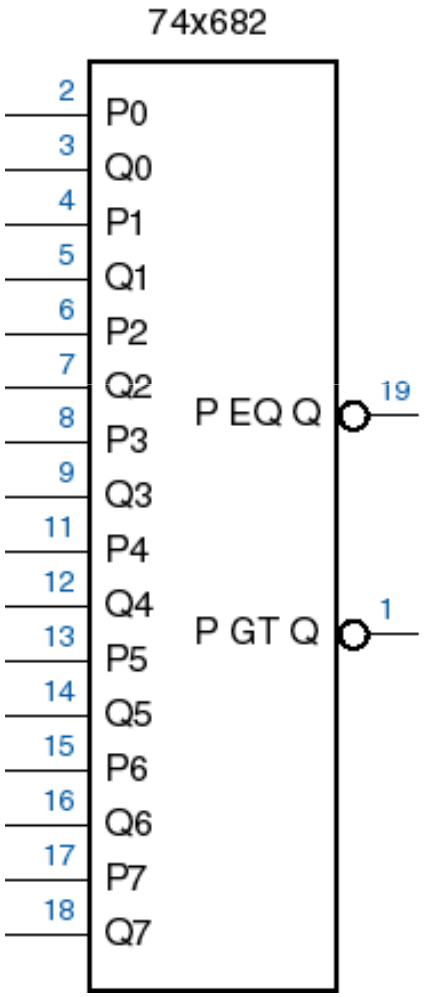


- Comparador de 4 bits

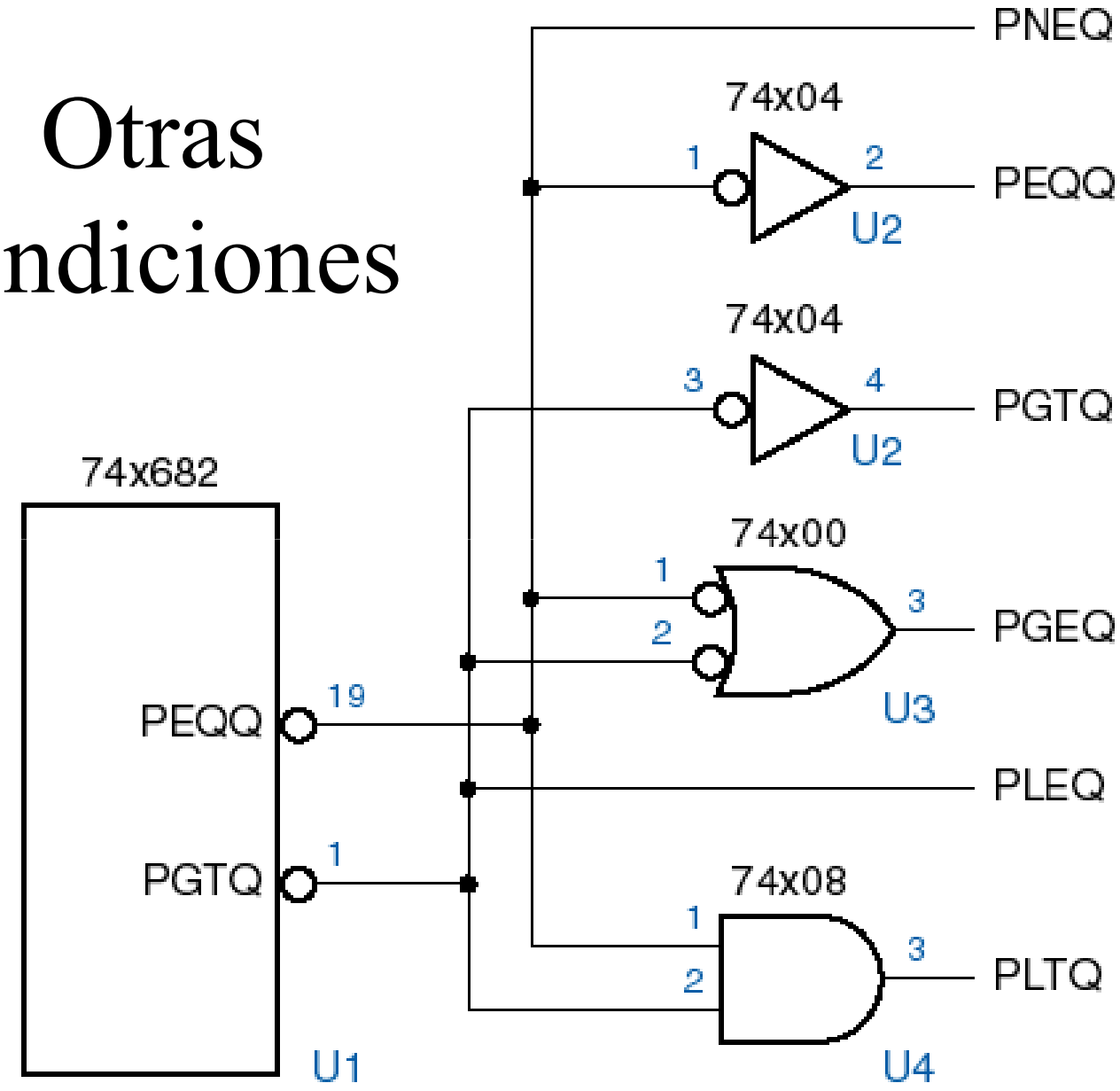


Comparador de 8 bits





Otras condiciones

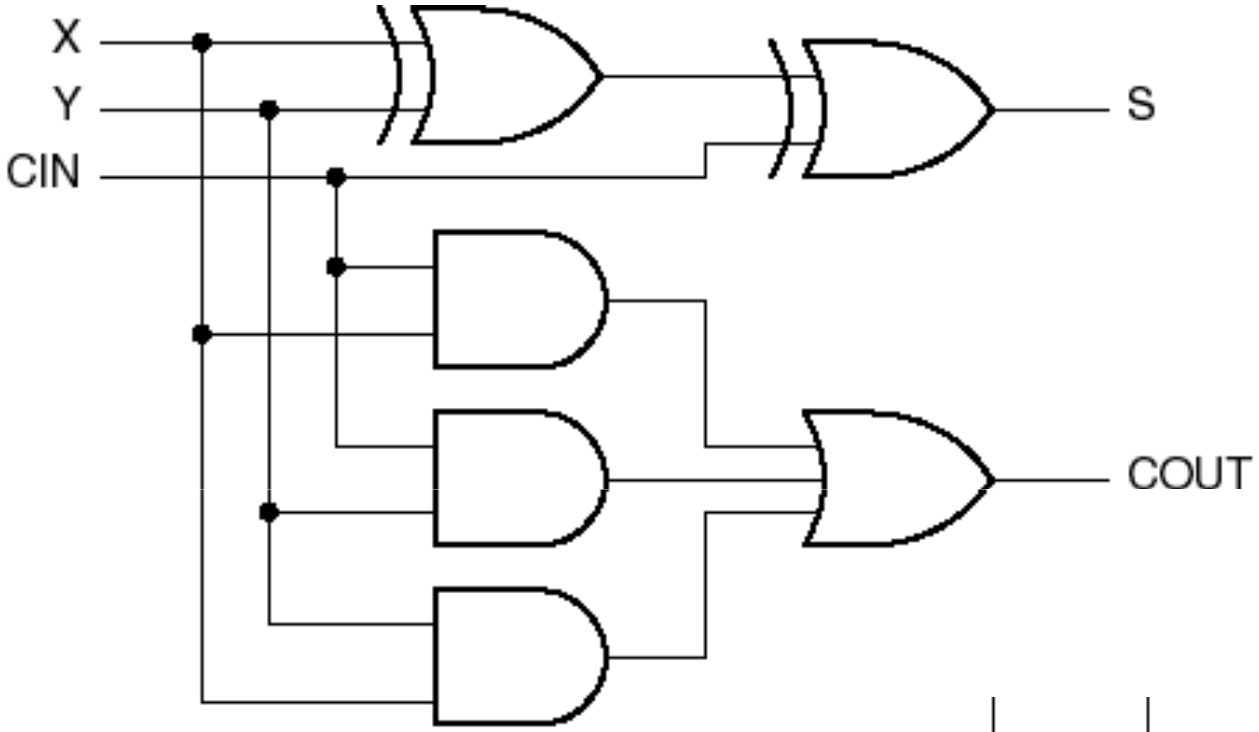


Sumadores

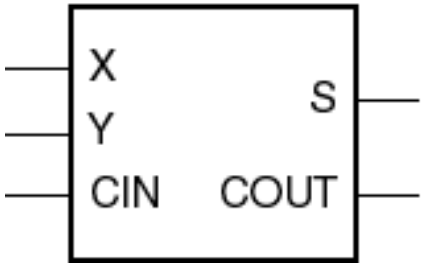
Tabla de verdad

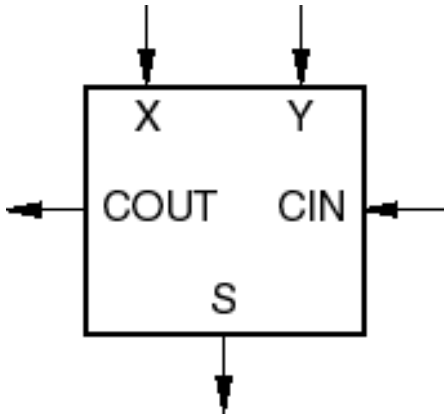
X	Y	Cin	S	Cout
0	0	0	0	0
0	0	1	1	0
0	1	0	1	0
0	1	1	0	1
1	0	0	1	0
1	0	1	0	1
1	1	0	0	1
1	1	1	1	1

Sumador de 1 bit con acarreo

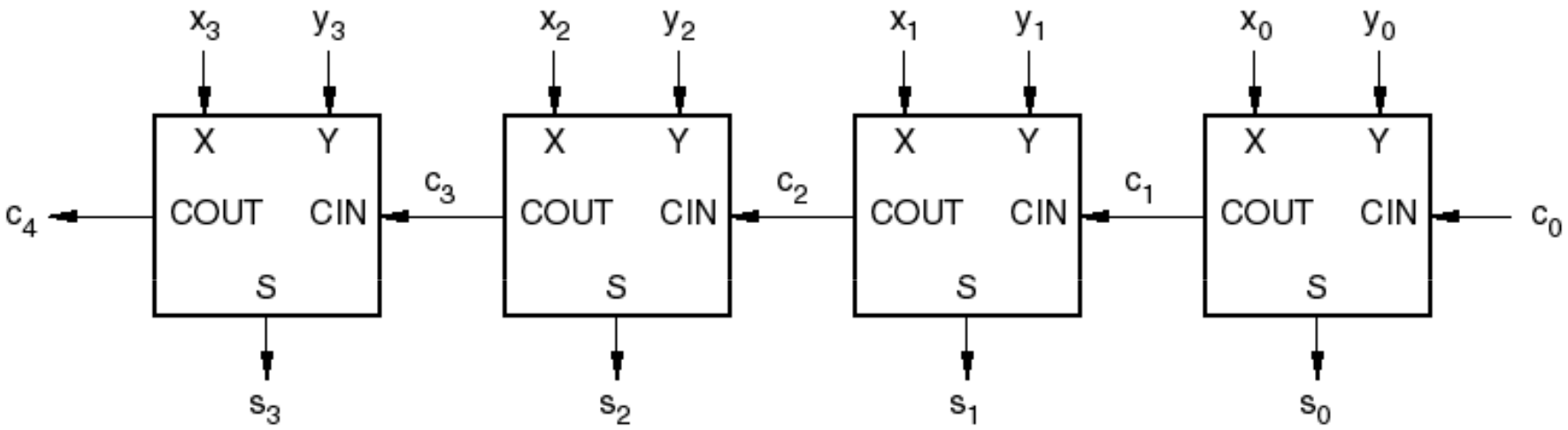


full adder





Sumadores encadenados

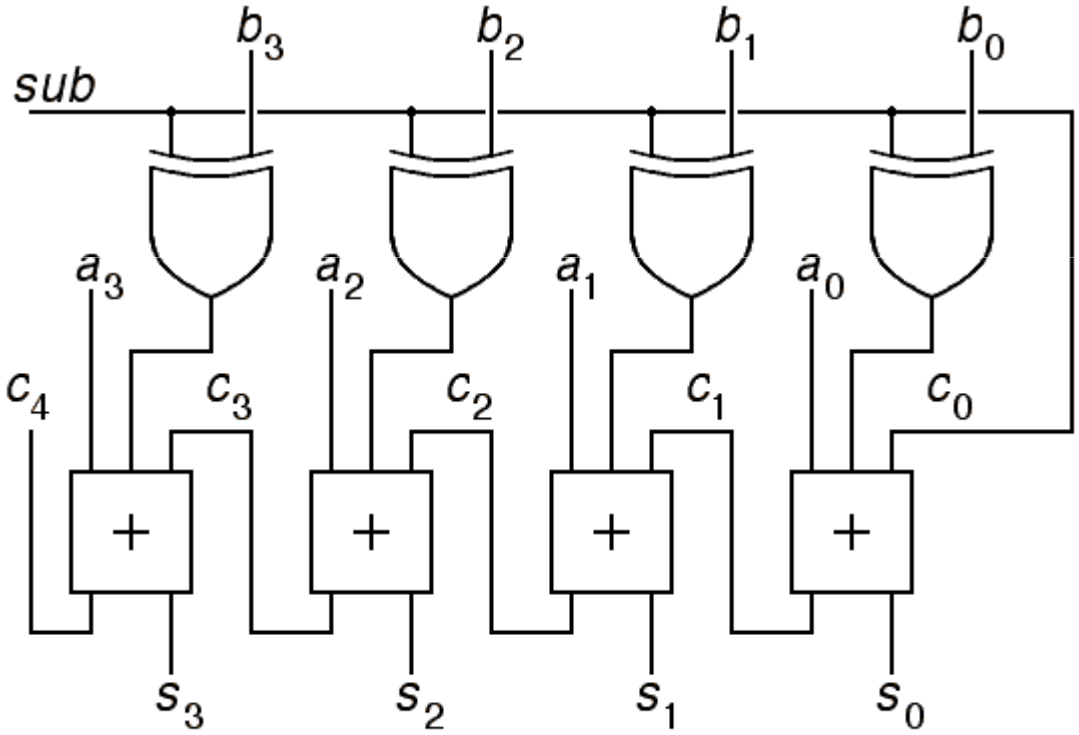


- La velocidad está limitada por la trasmisión del acarreo.

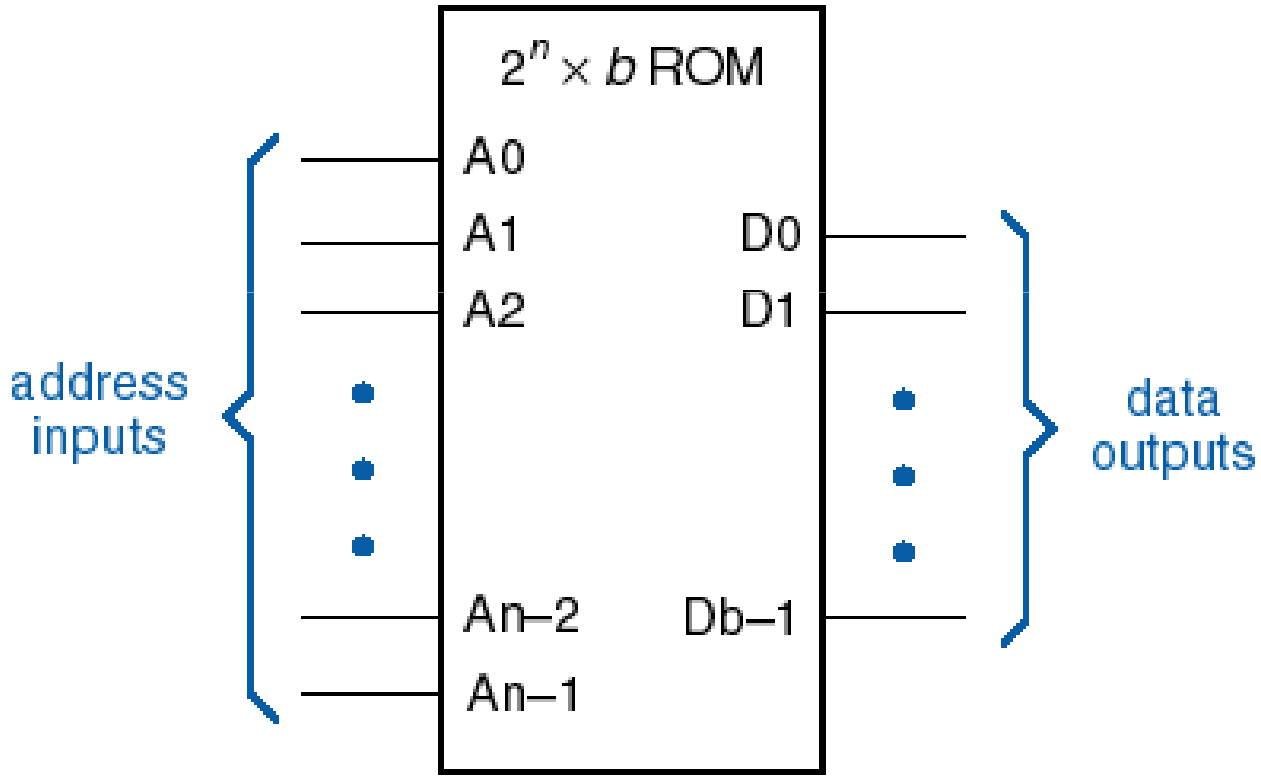
Restadores

- No existe un circuito específico para realizar la resta, sino que se lleva a cabo sumando el valor opuesto.
- El valor opuesto se calcula usando la representación de “Complemento a 2”.

Restador de 4 bits



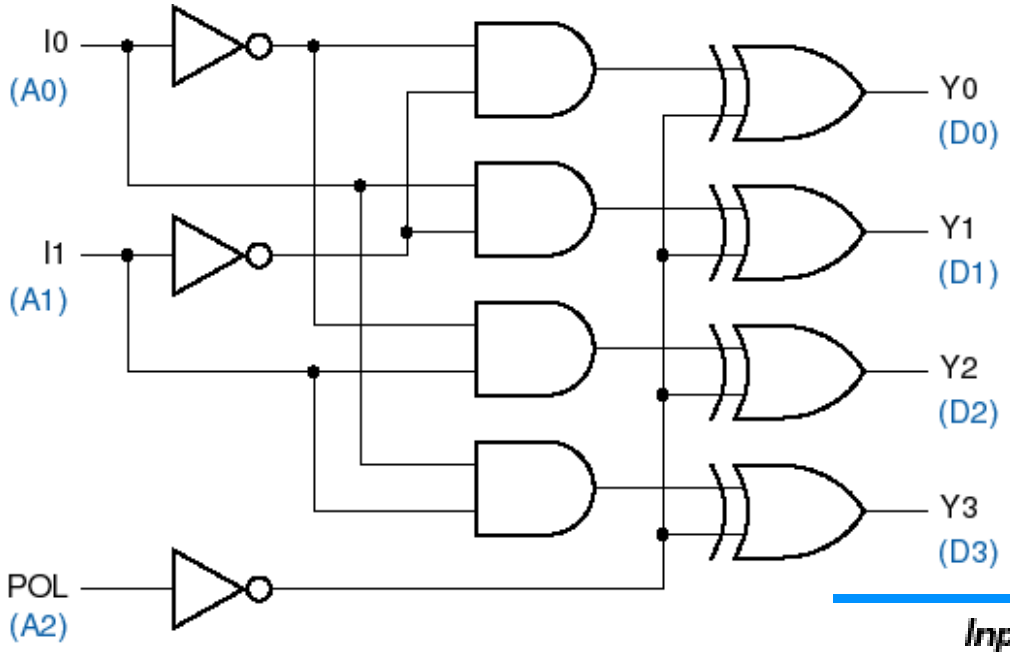
Memorias ROM

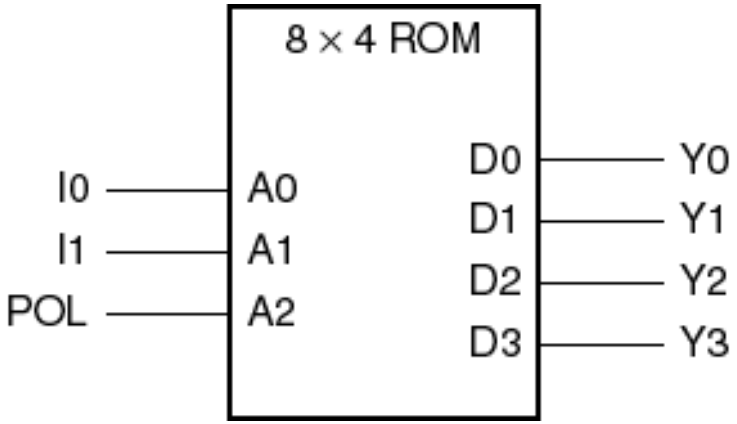


Ventajas de la ROM

- Una ROM se puede ver como una imagen física de una tabla de verdad:
  - Puede realizar cualquier función lógica
  - Direcciones (entradas) = Entradas de la función
  - Datos (salidas) = Salidas de la función

Implementación de funciones lógicas con ROM





Inputs			*Outputs*			
A2	*A1*	*A0*	*D3*	*D2*	*D1*	*D0*
0	0	0	1	1	1	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	1
1	0	0	0	0	0	1
1	0	1	0	0	1	0
1	1	0	0	1	0	0
1	1	1	1	0	0	0