



electrónica general

Lógica Combinacional

Tema 6

## **circuitos combinacionales**

### **0. Introducción**

### **1. Circuitos combinacionales lógicos**

1.1. Multiplexor

1.2. Encoder

1.3. Decodificador, Demultiplexor

1.4. Convertidores de código

### **2. Circuitos combinacionales aritméticos**

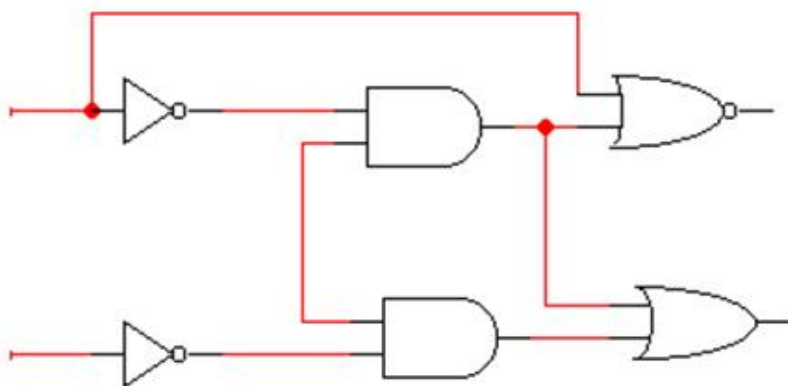
2.1. Sumador binario

2.2. Unidad Lógica Aritmética

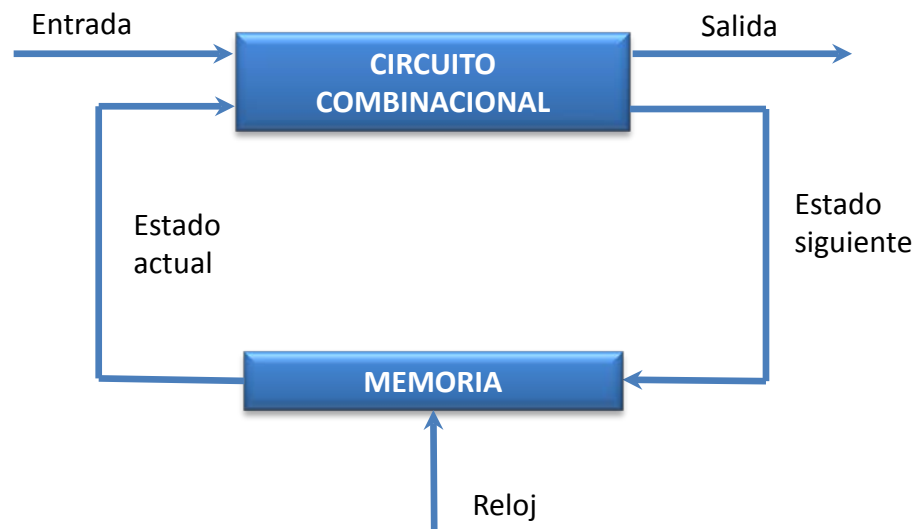
# 0. Introducción

Los circuitos digitales se dividen en dos grupos: **los circuitos combinacionales y circuitos secuenciales.**

**Circuito combinacional:** El valor de la salida depende únicamente del valor de las entradas, de tal forma que, para cada combinación de las variables lógicas de entrada, se tiene una salida.



**Circuito secuencial:** los valores de la salida en un instante dado no solamente depende de los valores de las entradas en ese instante sino también de valores que hayan tenido las entradas en instantes anteriores.



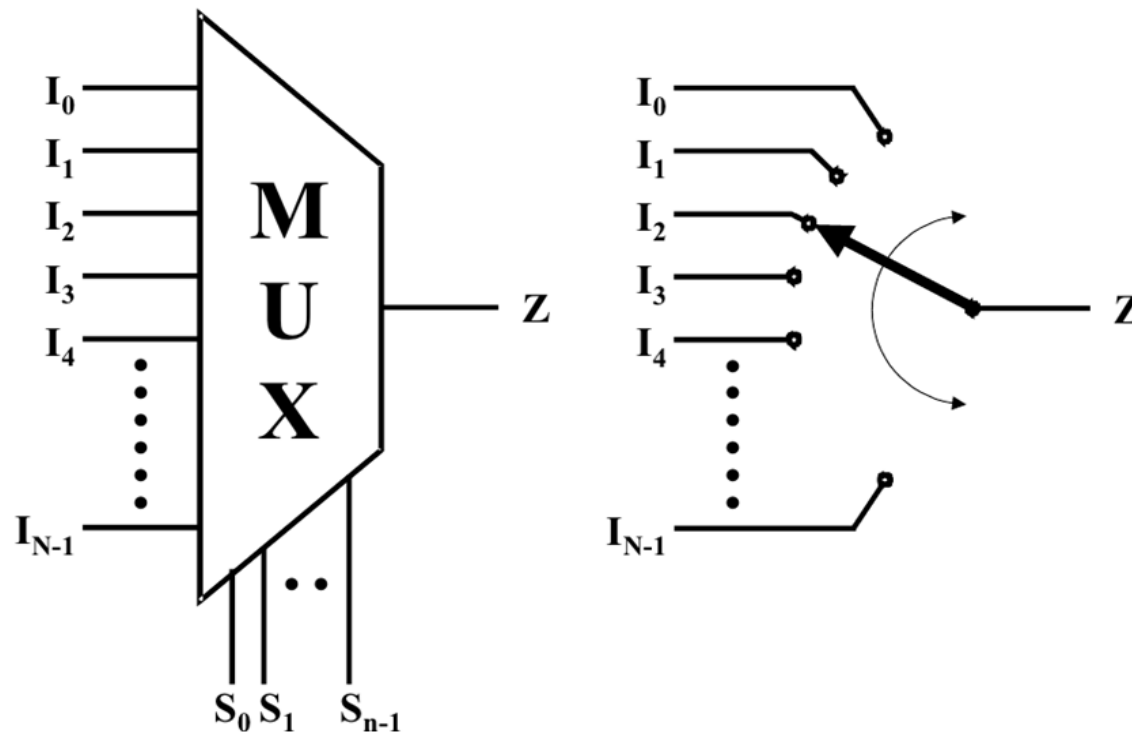
El procedimiento a seguir en el diseño de cualquier circuito combinacional:

- a. Realizar una tabla de verdad de cada una de las salidas del circuito en función de las variables lógicas de entrada.
- b. Obtener la función más simplificada posible de cada salida (usando métodos de simplificación que ya hemos visto).
- c. Elegir el tipo de puertas y la familia lógica con la que se va a diseñar el circuito combinacional.

# 1. Circuitos combinacionales lógicos

## 1.1. Multiplexor.

Es un circuito combinacional que realiza la función lógica de seleccionar un dato de entre varios procedentes de distintas fuentes.



Símbolo Lógico de un multiplexor (izquierda) y su similitud con un conmutador mecánico

# 1. Circuitos combinacionales lógicos

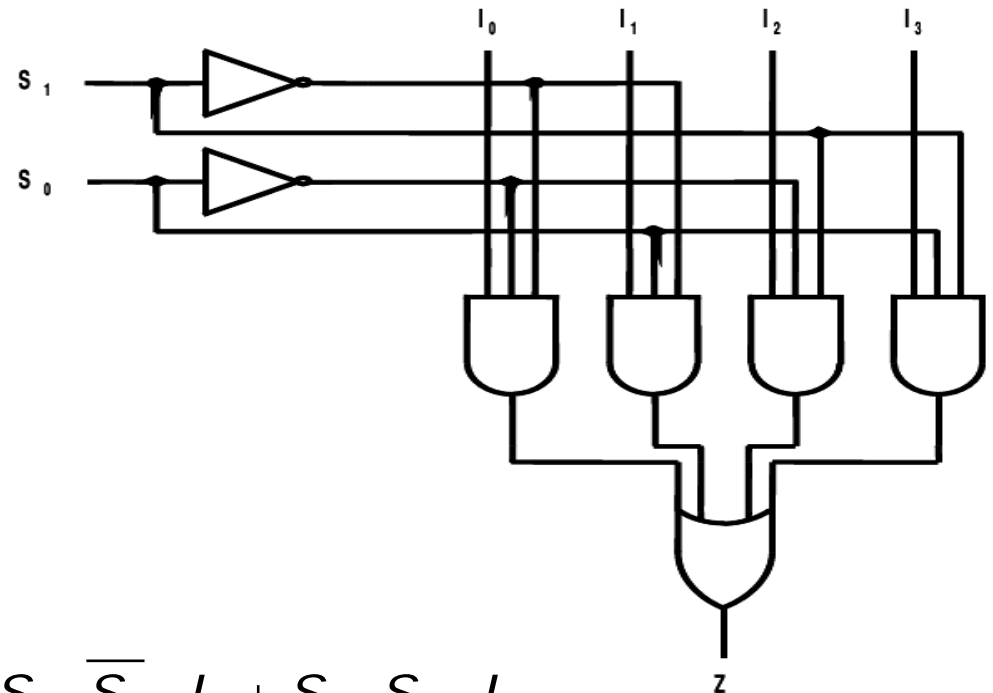
## 1.1. Multiplexor.

Para N entradas, se necesitan n entradas de control  $\Rightarrow N \leq 2^n$

Utilidad: Compartir una única línea de comunicación por más de un transmisor

Ejemplo: Multiplexor de 4 entradas

CONTROL		SALIDA
$S_1$	$S_0$	Z
0	0	$I_0$
0	1	$I_1$
1	0	$I_2$
1	1	$I_3$



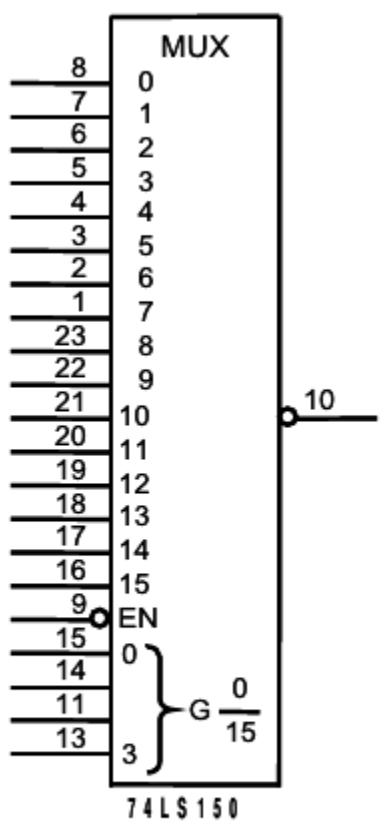
$$Z = \overline{S_1} \cdot \overline{S_0} \cdot I_0 + \overline{S_1} \cdot S_0 \cdot I_1 + S_1 \cdot \overline{S_0} \cdot I_2 + S_1 \cdot S_0 \cdot I_3$$

# 1. Circuitos combinacionales lógicos

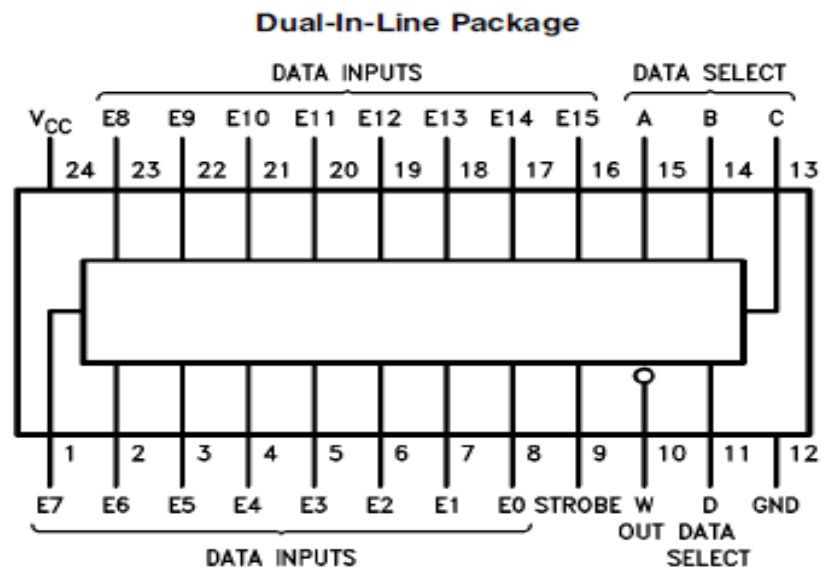
## 1.1. Multiplexor.

Existen diversos circuitos integrados multiplexores cuyas diferencias estriban, básicamente , en el número de entradas de datos y en las características eléctricas de la salida.

Algunos disponen de entrada de habilitación o *enable* (EN)



### Connection Diagrams



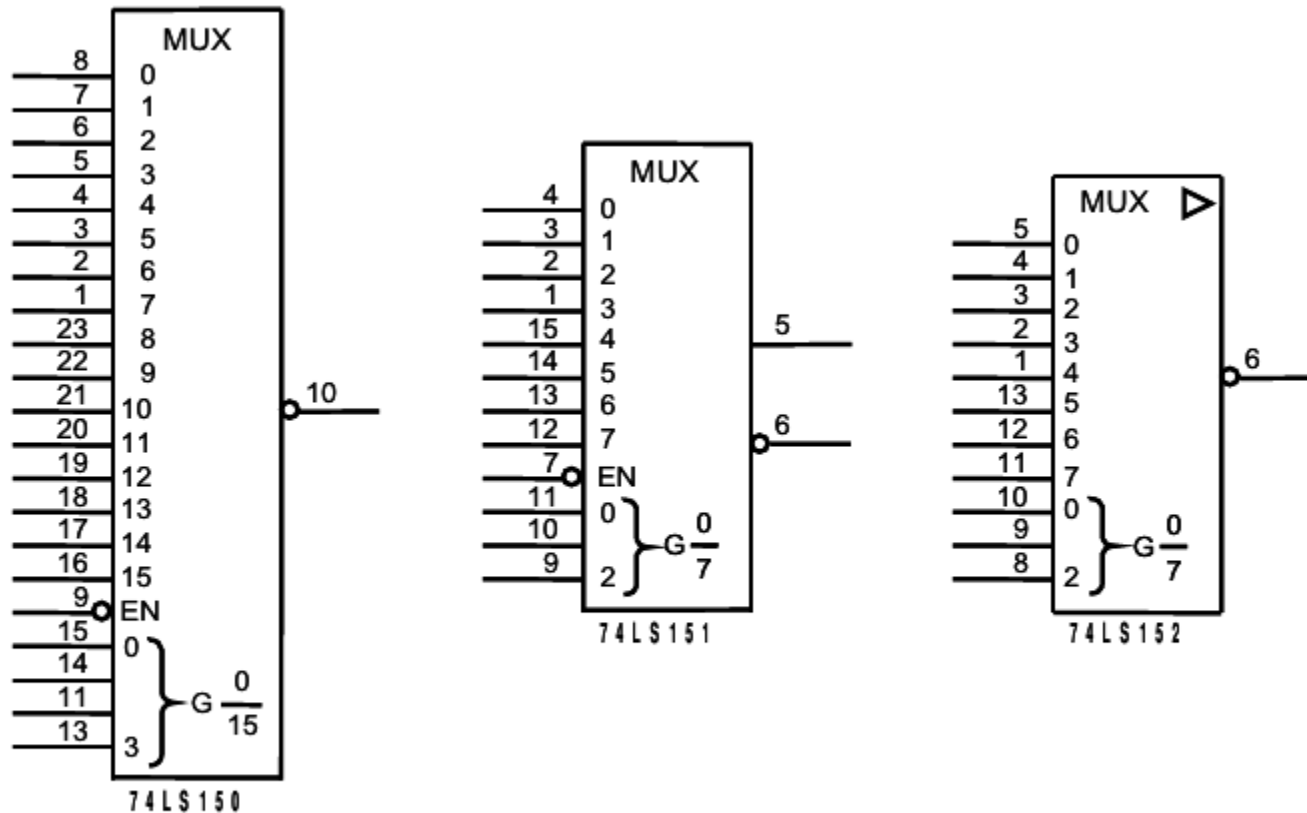
Order Number 54150DQMB, 54150FMQB,  
DM54150J or DM74150N  
See NS Package Number J24A, N24A or W24C

# 1. Circuitos combinacionales lógicos

## 1.1. Multiplexor.

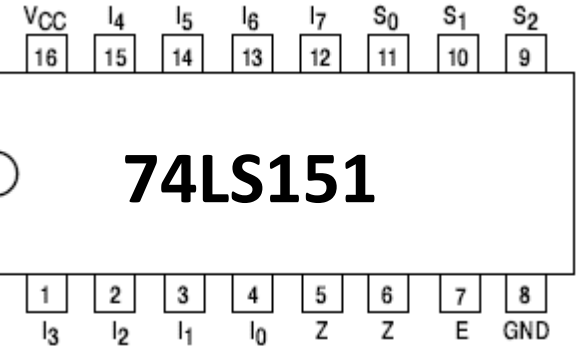
Existen diversos circuitos integrados multiplexores cuyas diferencias estriban, básicamente, en el número de entradas de datos y en las características eléctricas de la salida.

Algunos disponen de entrada de habilitación o *enable* (EN)





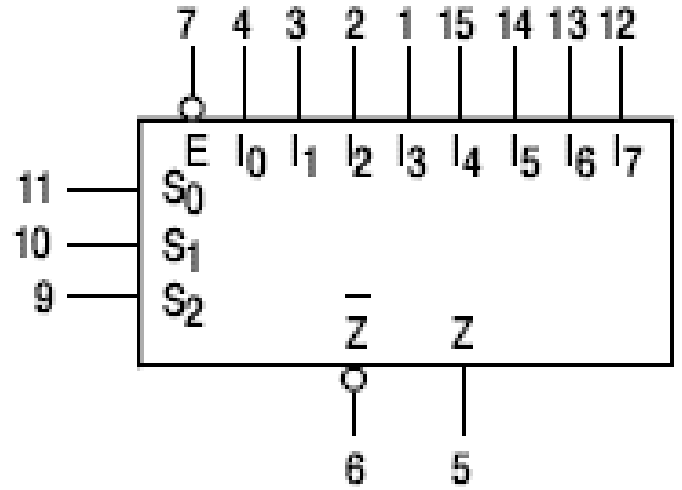
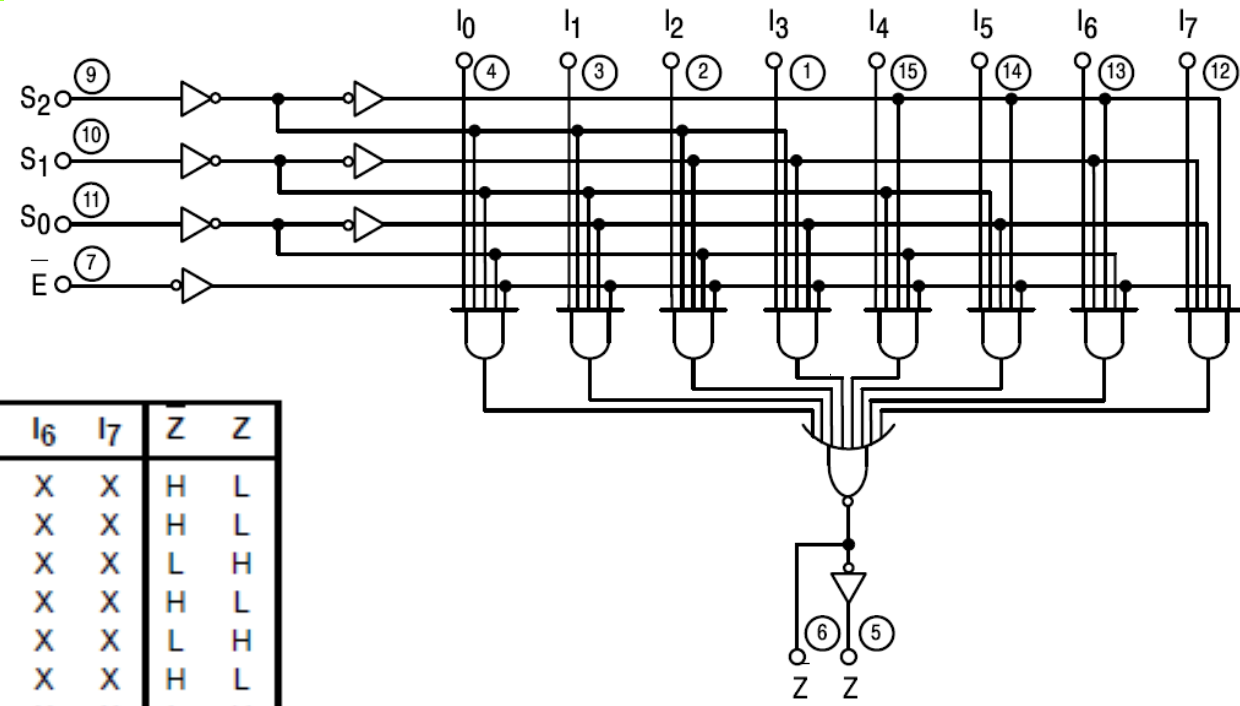
# 1. Circuitos combinacionales lógicos



TRUTH TABLE

E	S <sub>2</sub>	S <sub>1</sub>	S <sub>0</sub>	I <sub>0</sub>	I <sub>1</sub>	I <sub>2</sub>	I <sub>3</sub>	I <sub>4</sub>	I <sub>5</sub>	I <sub>6</sub>	I <sub>7</sub>	Z̄	Z
H	X	X	X	X	X	X	X	X	X	X	X	H	L
L	L	L	L	L	X	X	X	X	X	X	X	H	L
L	L	L	L	H	X	X	X	X	X	X	X	L	H
L	L	L	H	X	L	X	X	X	X	X	X	H	L
L	L	L	H	X	H	X	X	X	X	X	X	L	H
L	L	H	L	X	X	L	X	X	X	X	X	H	L
L	L	H	L	X	X	H	X	X	X	X	X	L	H
L	L	H	H	X	X	X	L	X	X	X	X	H	L
L	L	H	H	X	X	X	H	X	X	X	X	L	H
L	H	L	L	X	X	X	X	L	X	X	X	H	L
L	H	L	L	X	X	X	X	H	X	X	X	L	H
L	H	L	H	X	X	X	X	X	L	X	X	H	L
L	H	L	H	X	X	X	X	X	H	X	X	L	H
L	H	H	L	X	X	X	X	X	X	L	X	H	L
L	H	H	L	X	X	X	X	X	X	H	X	L	H
L	H	H	H	X	X	X	X	X	X	X	L	H	L
L	H	H	H	X	X	X	X	X	X	X	H	L	H

H = HIGH Voltage Level  
 L = LOW Voltage Level  
 X = Don't Care

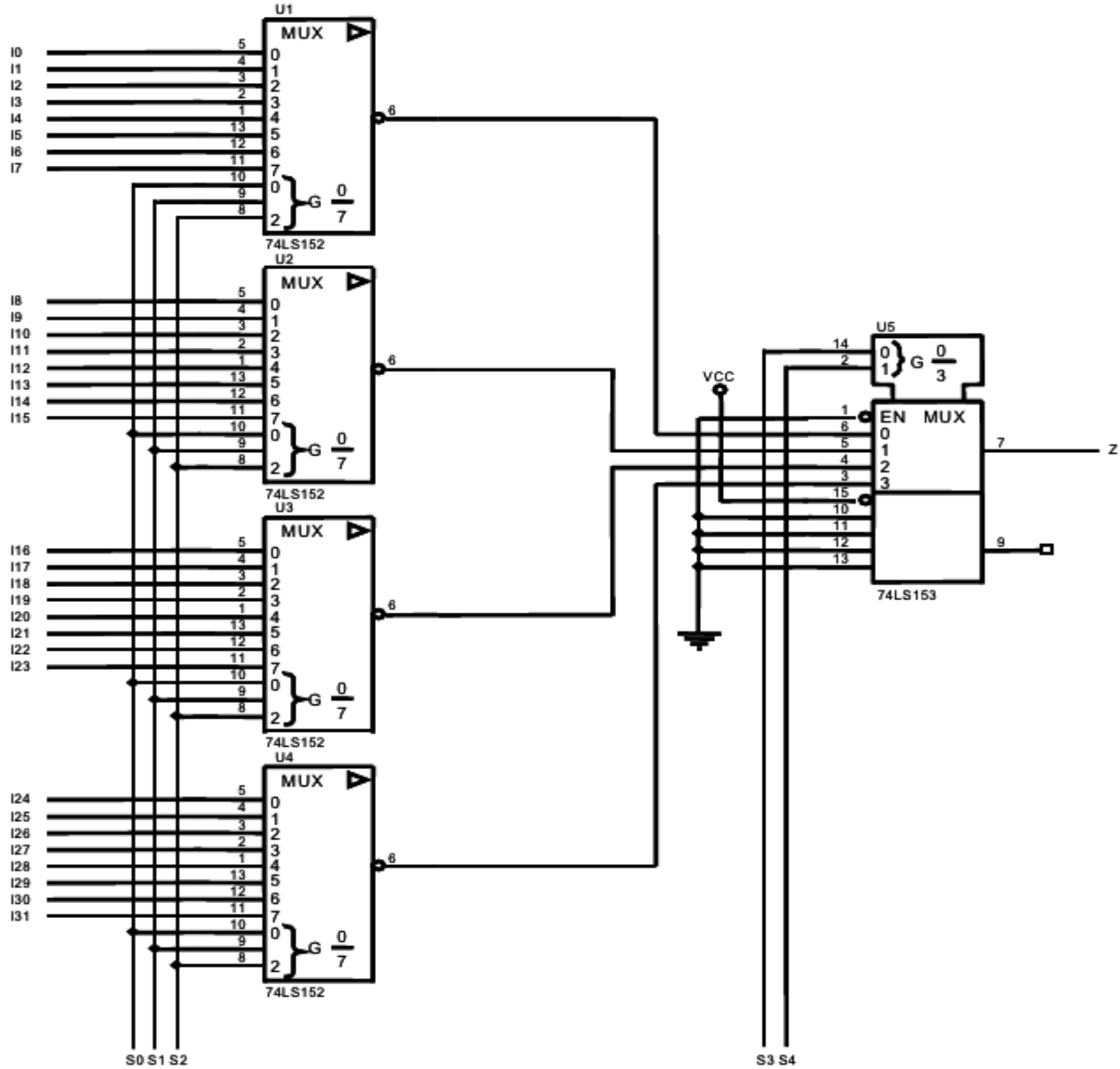


# 1. Circuitos combinacionales lógicos

## 1.1. Multiplexor.

Se pueden construir multiplexores de mayor capacidad a partir de multiplexores con un número reducido de entradas.

*Ejemplo:* multiplexor de 32 entradas a partir de multiplexores de 8 entradas



# 1. Circuitos combinacionales lógicos

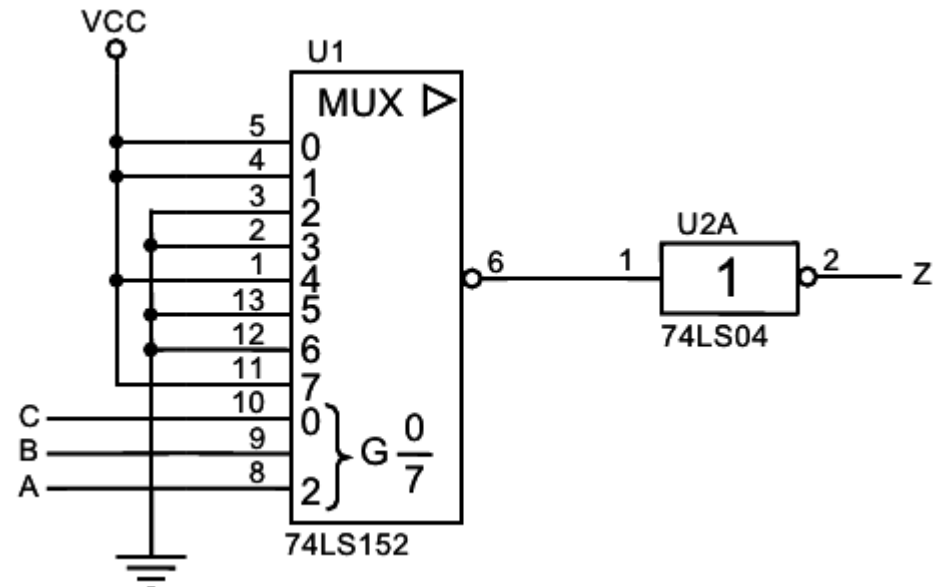
## 1.1. Multiplexor.

Debido a su estructura lógica formada por dos niveles de puertas AND-OR, el multiplexor se puede utilizar para crear funciones lógicas

Función Lógica de  $n$  variables  $\Rightarrow$  multiplexor de  $n-1$  o más entradas

$$Z = \bar{A} \cdot \bar{B} + A \cdot \bar{B} \cdot \bar{C} + A \cdot B \cdot C$$

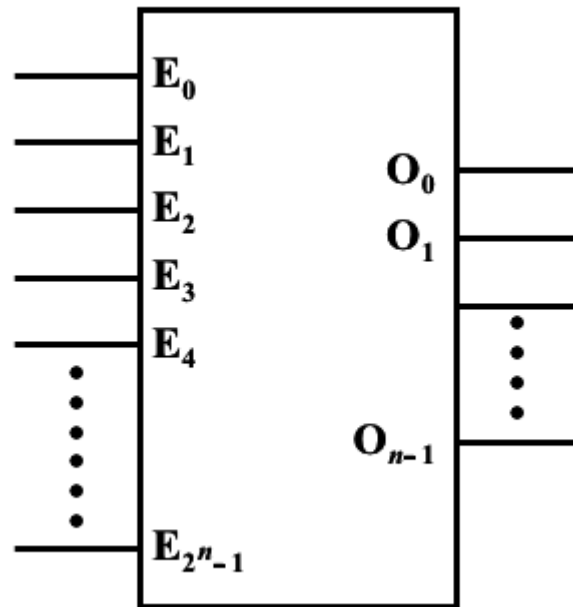
A	B	C	Z
0	0	0	1
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1



# 1. Circuitos combinacionales lógicos

## 1.2. Encoder.

Es un circuito combinacional lógico que acepta un máximo de  $2^n$  entradas de distintas características eléctricas (distintas fuentes) y genera  $n$  salidas (código binario) de tal forma que si hay una entrada activa, a la salida aparece la combinación binaria que representa a dicha entrada.



# 1. Circuitos combinacionales lógicos

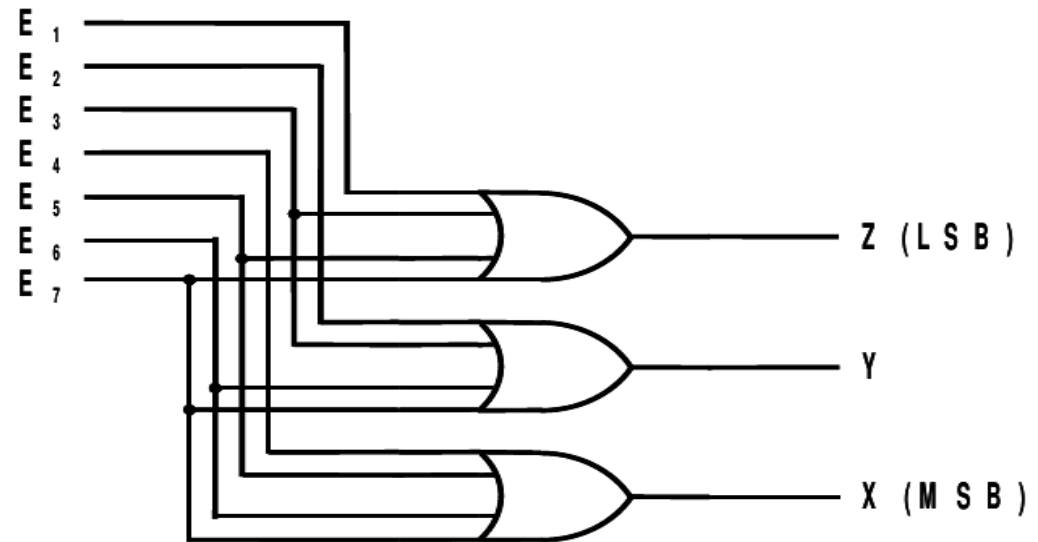
## 1.2. Encoder.

ENTRADAS	SALIDAS		
$E_i$	X	Y	Z
$E_0$	0	0	0
$E_1$	0	0	1
$E_2$	0	1	0
$E_3$	0	1	1
$E_4$	1	0	0
$E_5$	1	0	1
$E_6$	1	1	0
$E_7$	1	1	1

$$X = E_4 + E_5 + E_6 + E_7$$

$$Y = E_2 + E_3 + E_6 + E_7$$

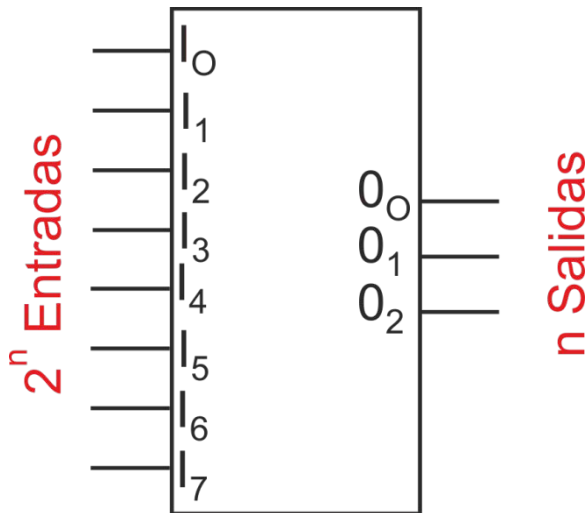
$$Z = E_1 + E_3 + E_5 + E_7$$



# 1. Circuitos combinacionales lógicos

## 1.2. Encoder.

### CODIFICADOR DECIMAL A BINARIO NATURAL



$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$O_2$	$O_1$	$O_0$
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	0	0	0	1
0	0	0	0	0	1	0	0	0	1	0
0	0	0	0	1	0	0	0	0	1	1
0	0	0	1	0	0	0	0	1	0	0
0	0	1	0	0	0	0	0	1	0	1
0	1	0	0	0	0	0	0	1	1	0
1	0	0	0	0	0	0	0	1	1	1

PROBLEMA: puede haber más de una entrada activa.

SOLUCIÓN: encoder de prioridad

# 1. Circuitos combinacionales lógicos

## 1.2. Encoder.

### Encoder de prioridad

La salida responde a la entrada de mayor valor

$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$O_2$	$O_1$	$O_0$
0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1	0	0	0
0	0	0	0	0	0	1	X	0	0	1
0	0	0	0	0	1	X	X	0	1	0
0	0	0	0	1	X	X	X	0	1	1
0	0	0	1	X	X	X	X	1	0	0
0	0	1	X	X	X	X	X	1	0	1
0	1	X	X	X	X	X	X	1	1	0
1	X	X	X	X	X	X	X	1	1	1

# 1. Circuitos combinacionales lógicos

## 1.2. Encoder.

### Encoder de prioridad

$E_{in}$	$I_7$	$I_6$	$I_5$	$I_4$	$I_3$	$I_2$	$I_1$	$I_0$	$O_2$	$O_1$	$O_0$	GS	$E_{out}$
0	X	X	X	X	X	X	X	X	0	0	0	0	0
1	0	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	0	0	0	0	0	1	0	0	0	1	0
1	0	0	0	0	0	0	1	X	0	0	1	1	0
1	0	0	0	0	0	1	X	X	0	1	0	1	0
1	0	0	0	0	1	X	X	X	0	1	1	1	0
1	0	0	0	1	X	X	X	X	1	0	0	1	0
1	0	0	1	X	X	X	X	X	1	0	1	1	0
1	0	1	X	X	X	X	X	X	1	1	0	1	0
1	1	X	X	X	X	X	X	X	1	1	1	1	0

$$O_2 = E_{in} (I_4 + I_5 + I_6 + I_7)$$

$$O_1 = E_{in} (I_2 \cdot \bar{I}_4 \cdot \bar{I}_5 + I_3 \cdot \bar{I}_4 \cdot \bar{I}_5 + I_6 + I_7)$$

$$O_0 = E_{in} (I_1 \cdot \bar{I}_2 \cdot \bar{I}_4 \cdot \bar{I}_6 + I_3 \cdot \bar{I}_4 \cdot \bar{I}_6 + I_5 \cdot \bar{I}_6 + I_7)$$

$$E_{out} = E_{in} \cdot \bar{I}_0 \cdot \bar{I}_1 \cdot \bar{I}_2 \cdot \bar{I}_3 \cdot \bar{I}_4 \cdot \bar{I}_5 \cdot \bar{I}_6 \cdot \bar{I}_7$$

$$GS = E_{in} (I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7)$$



# 1. Circuitos combinacionales lógicos

## 1.2. Encoder.

### Encoder de prioridad

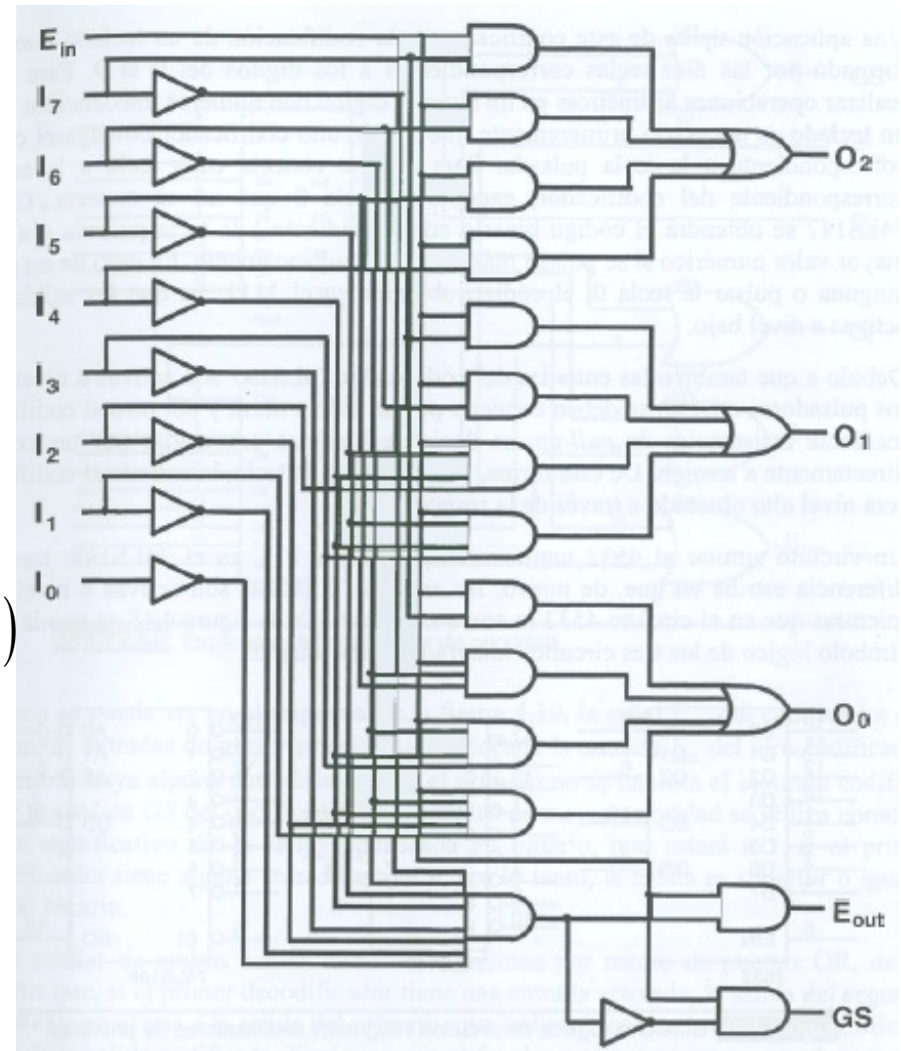
$$O_2 = E_{in} (I_4 + I_5 + I_6 + I_7)$$

$$O_1 = E_{in} (I_2 \cdot \overline{I_4} \cdot \overline{I_5} + I_3 \cdot \overline{I_4} \cdot \overline{I_5} + I_6 + I_7)$$

$$O_0 = E_{in} (I_1 \cdot \overline{I_2} \cdot \overline{I_4} \cdot \overline{I_6} + I_3 \cdot \overline{I_4} \cdot \overline{I_6} + I_5 \cdot \overline{I_6} + I_7)$$

$$E_{out} = E_{in} \cdot \overline{I_0} \cdot \overline{I_1} \cdot \overline{I_2} \cdot \overline{I_3} \cdot \overline{I_4} \cdot \overline{I_5} \cdot \overline{I_6} \cdot \overline{I_7}$$

$$GS = E_{in} (I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7)$$



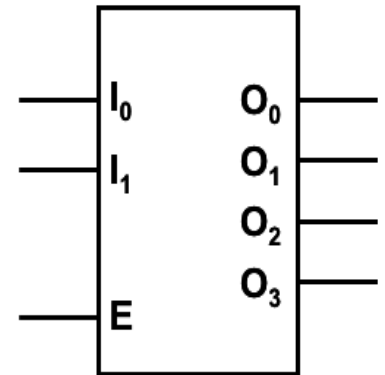
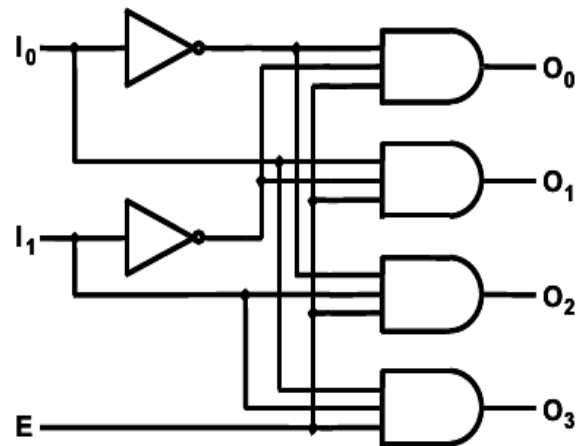
# 1. Circuitos combinacionales lógicos

## 1.3. Decodificador, demultiplexor.

Realiza la operación inversa al encoder: para cada combinación de entradas, sólo una de las salidas tiene un nivel lógico diferente a las demás (a la entrada se tiene una información codificada y a la salida se obtiene decodificada).

Los decodificadores tienen  $n$  entradas y  $m$  salidas, de forma que  $m \leq 2^n$ .

E	I <sub>1</sub>	I <sub>0</sub>	O <sub>3</sub>	O <sub>2</sub>	O <sub>1</sub>	O <sub>0</sub>
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
0	X	X	0	0	0	0



$$O_0 = E \cdot \overline{I_0} \cdot \overline{I_1} \quad O_1 = E \cdot \overline{I_0} \cdot I_1$$

$$O_2 = E \cdot I_0 \cdot \overline{I_1} \quad O_3 = E \cdot I_0 \cdot I_1$$

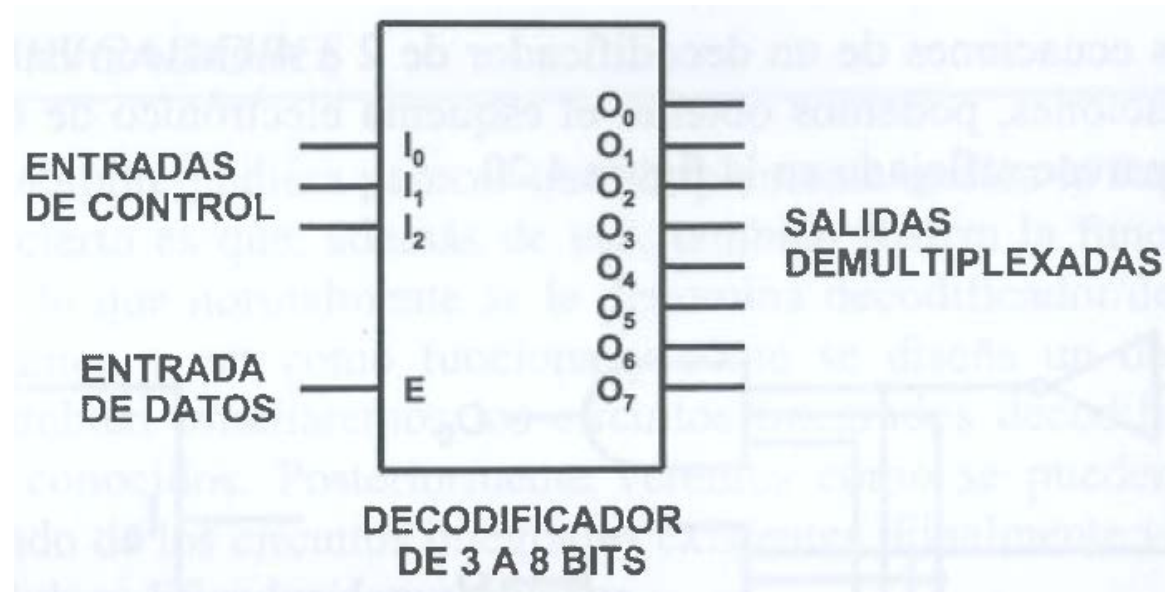
# 1. Circuitos combinacionales lógicos

## 1.3. Decodificador, demultiplexor.

Demultiplexor: acción opuesta al multiplexor, tiene una única entrada cuyo valor se trasfiere a una de las múltiples salidas, seleccionada por medio de las entradas de control.

$S_1$	$S_0$	$O_3$	$O_2$	$O_1$	$O_0$
0	0	0	0	0	E
0	1	0	0	E	0
1	0	0	E	0	0
1	1	E	0	0	0

Demultiplexor de 1 a 4

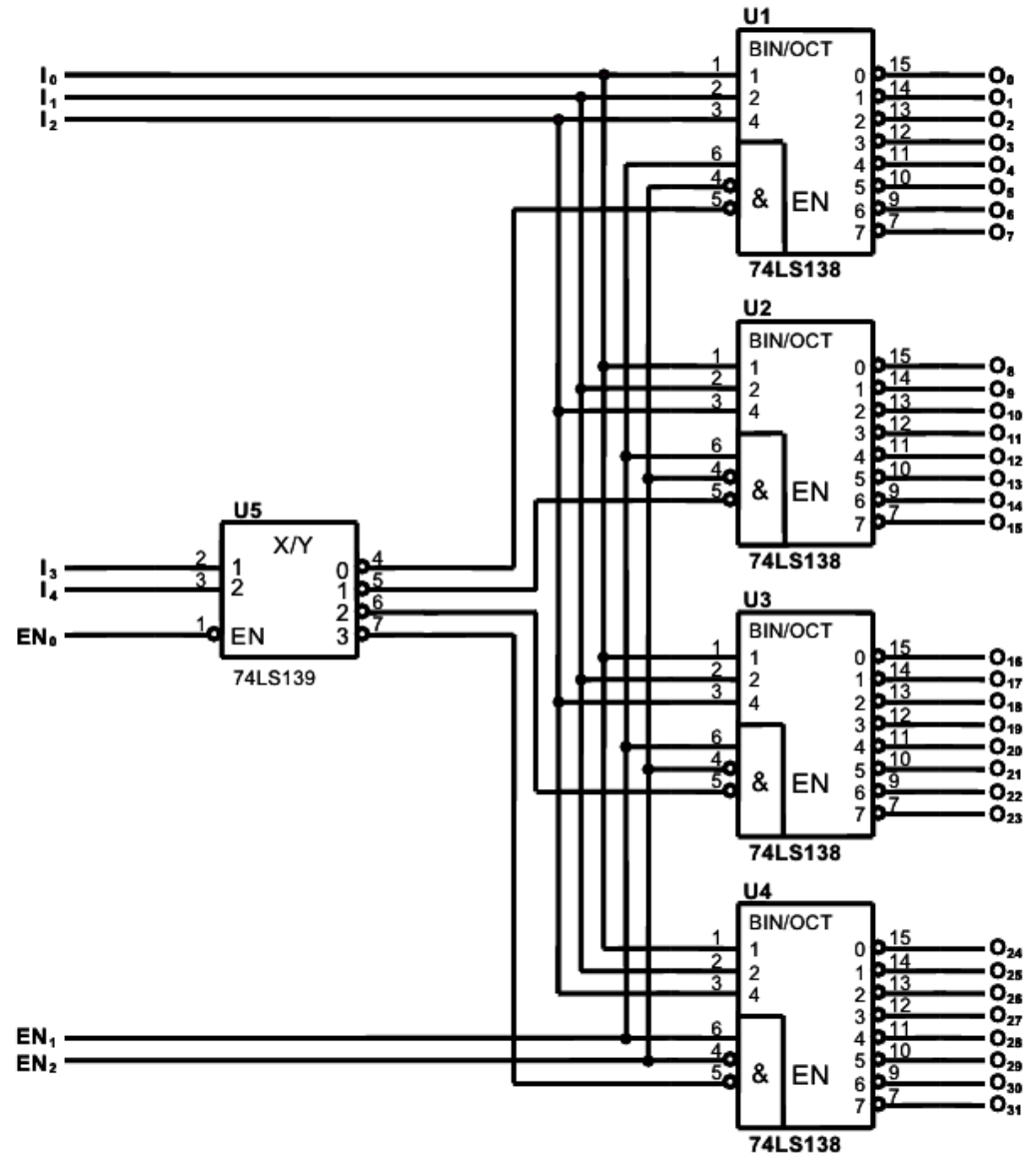


Demultiplexor de 1 a 8

# 1. Circuitos combinacionales lógicos

## 1.3. Decodificador, demultiplexor.

Mediante CI disponibles se puede aumentar la capacidad del decodificador/demultiplexor.



# 1. Circuitos combinacionales lógicos

## 1.4. Convertidores de código.

Circuito combinacional que parte de un código de  $N$  bits y lo transforma en un código de  $M$  bits.

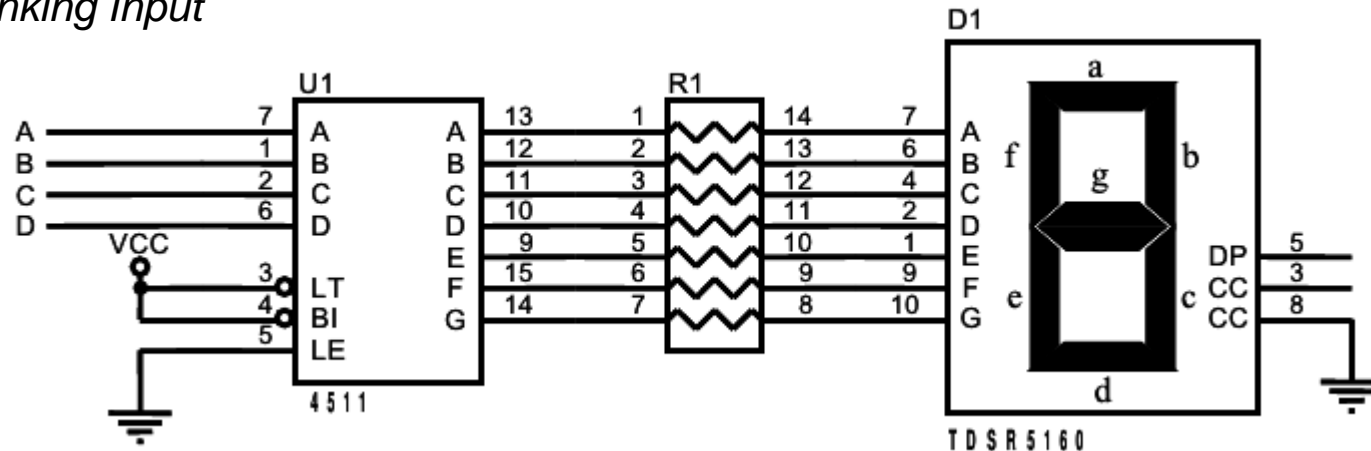
Ejemplo: convertidor de código BCD a 7 segmentos.

Tabla de la verdad de un convertor de BCD a 7 segmentos

Nº	D	C	B	A	a	b	c	d	e	f	g
0	0	0	0	0	1	1	1	1	1	1	0
1	0	0	0	1	0	1	1	0	0	0	0
2	0	0	1	0	1	1	0	1	1	0	1
3	0	0	1	1	1	1	1	1	0	0	1
4	0	1	0	0	0	1	1	0	0	1	1
5	0	1	0	1	1	0	1	1	0	1	1
6	0	1	1	0	1	0	1	1	1	1	1
7	0	1	1	1	1	1	1	0	0	0	0
8	1	0	0	0	1	1	1	1	1	1	1
9	1	0	0	1	1	1	1	1	0	1	1

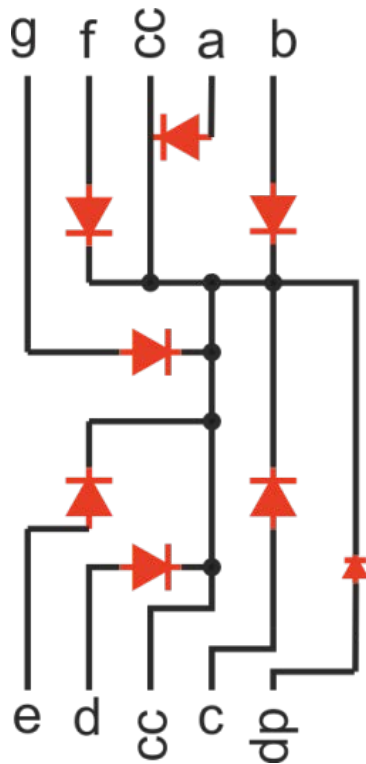
LT  $\Rightarrow$  *Test Lamp*

BI  $\Rightarrow$  *Blanking Input*

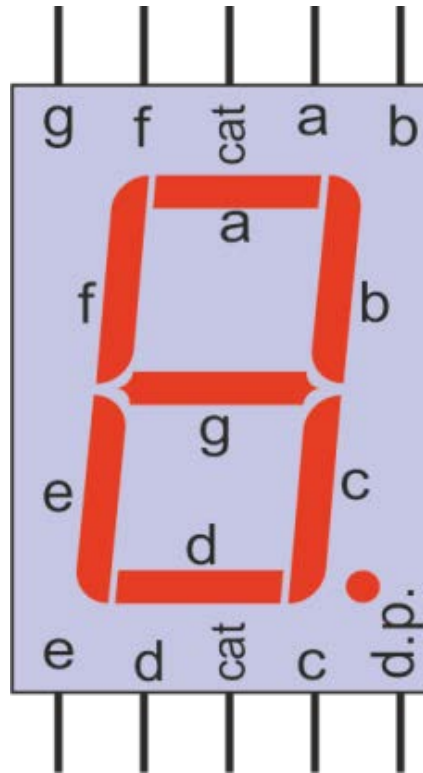


# 1. Circuitos combinacionales lógicos

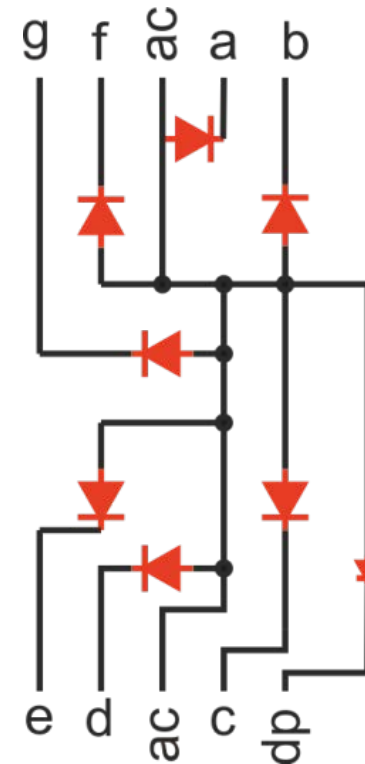
## 1.4. Convertidores de código.



**Cátodo Común**



**Display 7 segmentos**

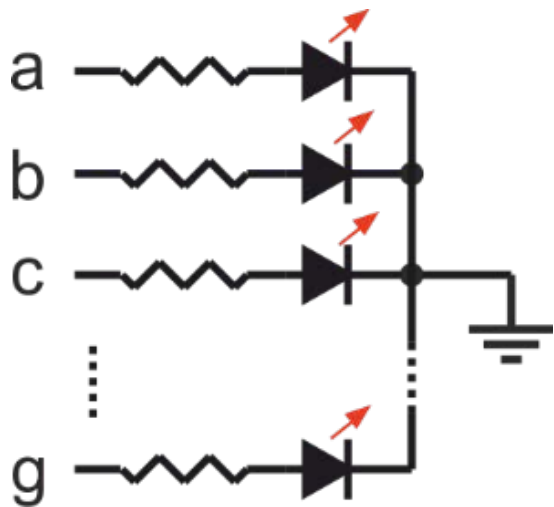


**Ánodo Común**

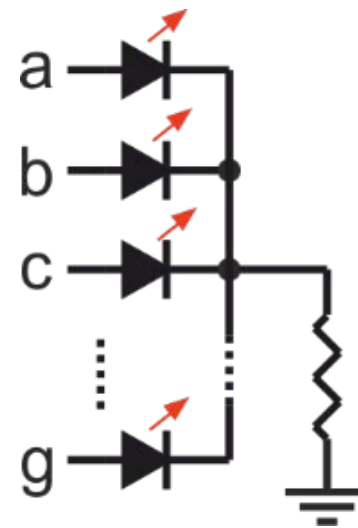
# 1. Circuitos combinacionales lógicos

## 1.4. Convertidores de código.

Es imprescindible poner resistencias para limitar la corriente por los diodos



Todos los diodos lucen con igual intensidad siempre

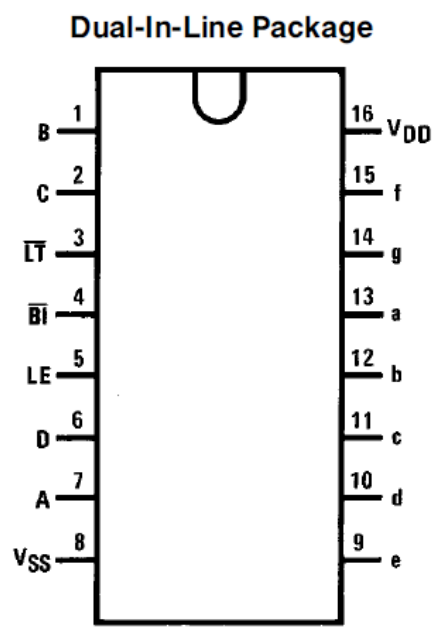


Los diodos lucen con distinta intensidad, dependiendo de cuantos estén en ON

# 1. Circuitos combinacionales lógicos

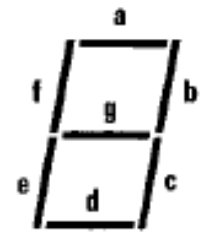
## 1.4. Convertidores de código.

### Connection Diagram



**CD4511**

### Segment Identification



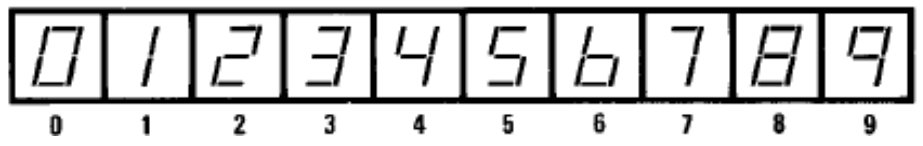
### Truth Table

Inputs							Outputs							
LE	$\overline{BI}$	$\overline{LT}$	D	C	B	A	a	b	c	d	e	f	g	Display
X	X	0	X	X	X	X	1	1	1	1	1	1	1	B
X	0	1	X	X	X	X	0	0	0	0	0	0	0	
0	1	1	0	0	0	0	1	1	1	1	1	1	0	0
0	1	1	0	0	0	1	0	1	1	0	0	0	0	1
0	1	1	0	0	1	0	1	1	0	1	1	0	1	2
0	1	1	0	0	1	1	1	1	1	1	0	0	1	3
0	1	1	0	1	0	0	0	1	1	0	0	1	1	4
0	1	1	0	1	0	1	1	0	1	1	0	1	1	5
0	1	1	0	1	1	0	0	0	1	1	1	1	1	6
0	1	1	0	1	1	1	1	1	1	0	0	0	0	7
0	1	1	1	0	0	0	1	1	1	1	1	1	1	8
0	1	1	1	0	0	1	1	1	1	0	0	1	1	9
0	1	1	1	0	1	0	0	0	0	0	0	0	0	
0	1	1	1	0	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	0	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	0	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
0	1	1	1	1	1	1	0	0	0	0	0	0	0	
1	1	1	X	X	X	X				*				*

X = Don't Care

\*Depends upon the BCD code applied during the 0 to 1 transition of LE.

### Display





## 2. Circuitos combinacionales aritméticos

### 2.1. Sumador binario.

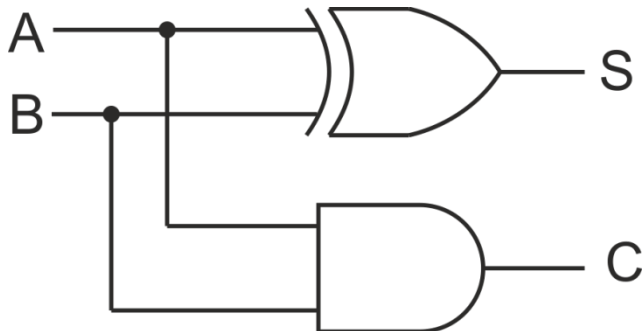
#### Semisumador (Half-Adder)

Circuito que realiza la suma de dos bits sin tener en cuenta la llevada

A	B	S	C
0	0	0	0
0	1	1	0
1	0	1	0
1	1	0	1

$$S = A \oplus B$$

$$C = A \cdot B$$





# 2. Circuitos combinacionales aritméticos

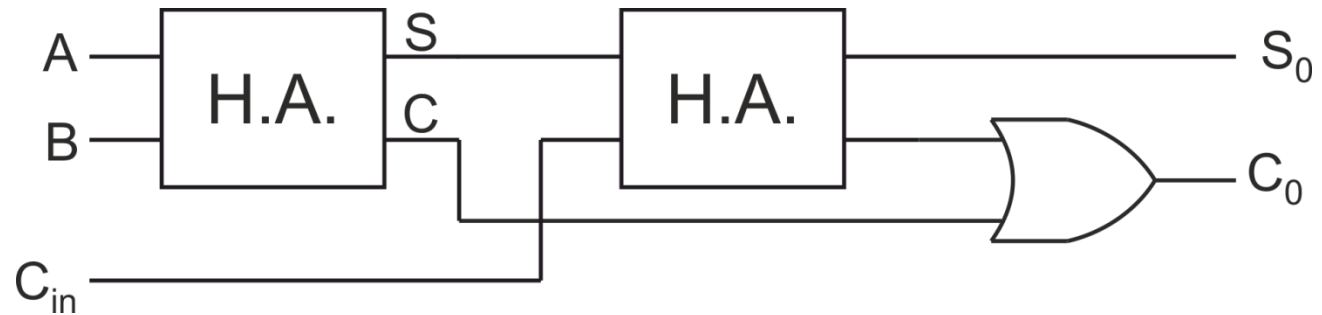
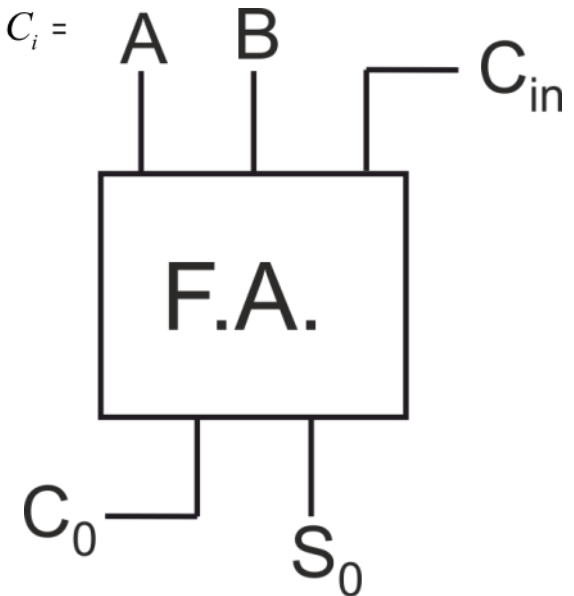
## 2.1. Sumador binario.

### Sumador completo (Full-Adder)

A	B	C <sub>in</sub>	S <sub>0</sub>	C <sub>0</sub>
0	0	0	0	0
0	1	0	1	0
1	0	0	1	0
1	1	0	0	1
0	0	1	1	0
0	1	1	0	1
1	0	1	0	1
1	1	1	1	1

$$\begin{aligned} S_i &= \bar{A}_i \cdot B_i \cdot \bar{C}_i + A_i \cdot \bar{B}_i \cdot \bar{C}_i + \bar{A}_i \cdot \bar{B}_i \cdot C_i + A_i \cdot B_i \cdot C_i = \\ &= (\bar{A}_i \cdot B_i + A_i \cdot \bar{B}_i) \cdot \bar{C}_i + (\bar{A}_i \cdot \bar{B}_i + A_i \cdot B_i) \cdot C_i = \\ &= A_i \oplus B_i \cdot \bar{C}_i + \overline{A_i \oplus B_i} \cdot C_i = \\ &= A_i \oplus B_i \oplus C_i \end{aligned}$$

$$C_{i+1} = A_i \cdot B_i + A_i \cdot C_i + B_i \cdot C_i$$

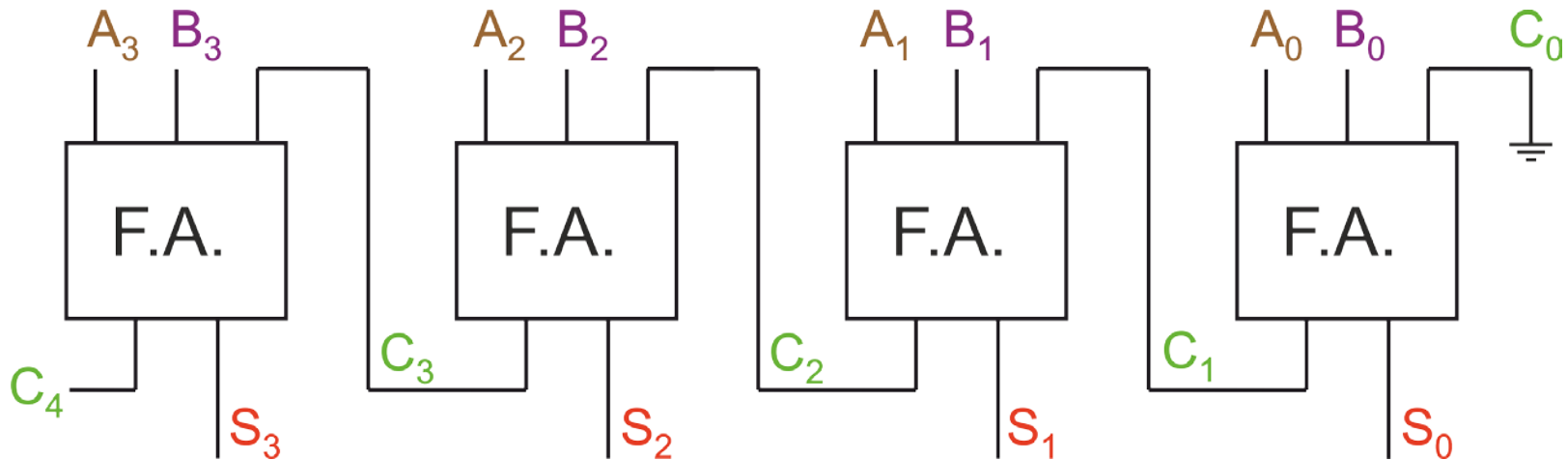


# 2. Circuitos combinacionales aritméticos

## 2.1. Sumador binario.

Sumador de 4 bits

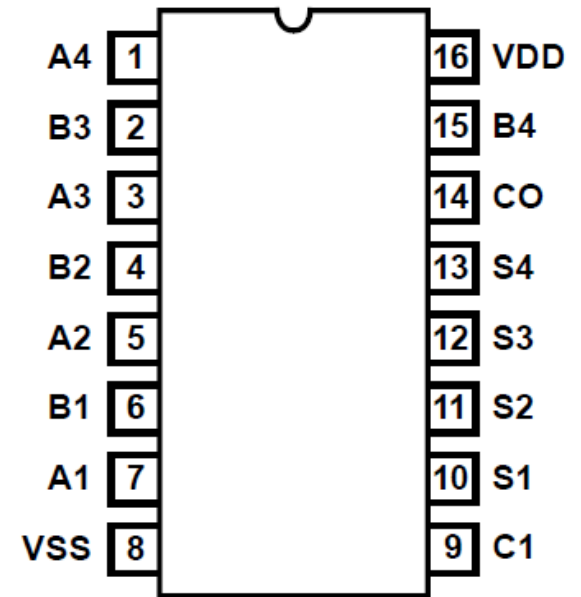
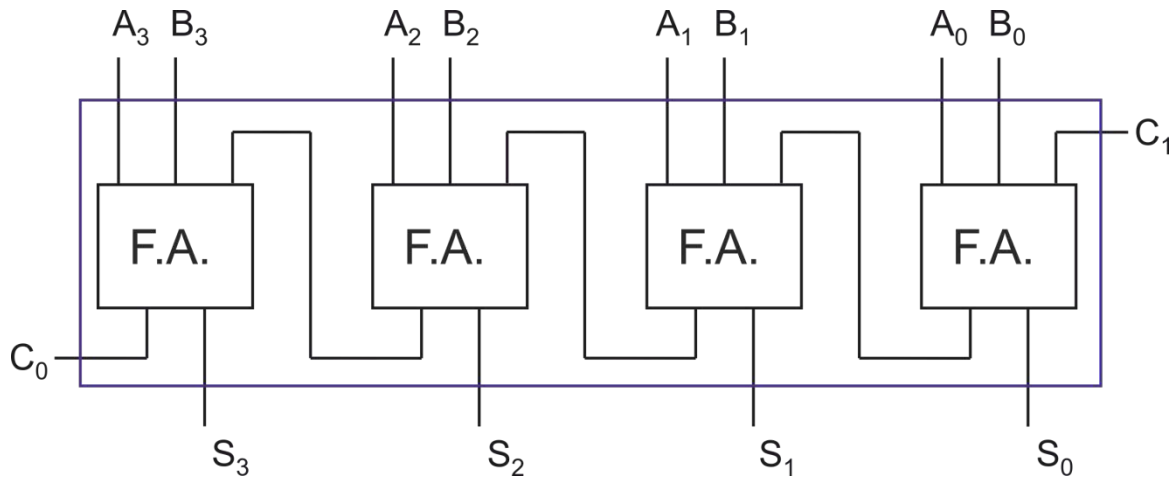
	$C_3$	$C_2$	$C_1$	$C_0$	
+	<b>A</b>	<b>1</b>	<b>1</b>	<b>1</b>	<b>0</b>
	$A_3$	$A_2$	$A_1$	$A_0$	
	<b>B</b>	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>
	$B_3$	$B_2$	$B_1$	$B_0$	
<hr/>					
	<b>1</b>	<b>1</b>	<b>0</b>	<b>1</b>	<b>1</b>
	$C_4$	$S_3$	$S_2$	$S_1$	$S_0$



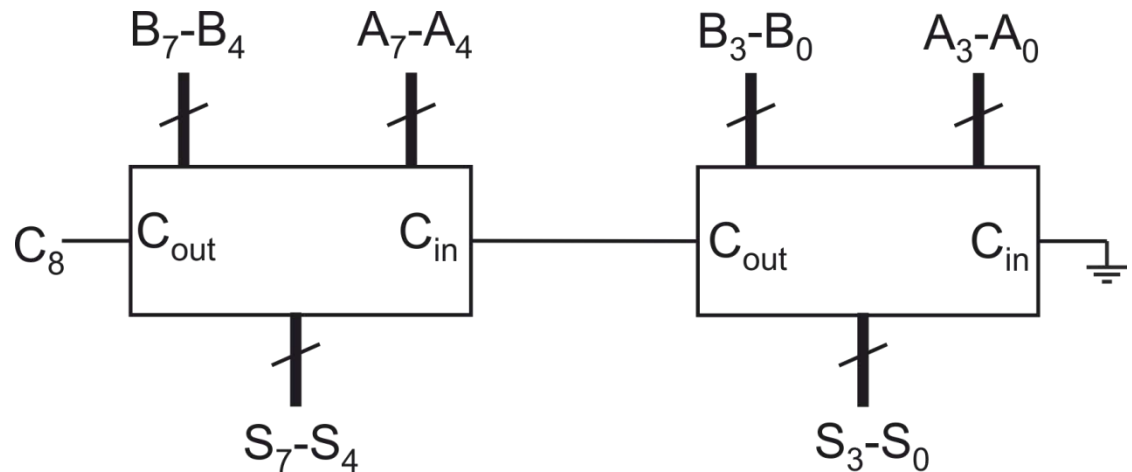
# 2. Circuitos combinacionales aritméticos

## 2.1. Sumador binario.

### Sumador de 4 bits **CD4008**



Se pueden acoplar en cascada para sumar números de más bits

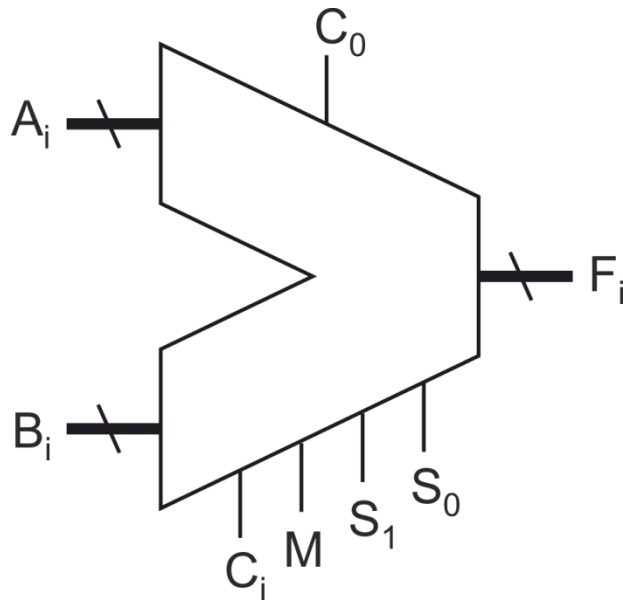


# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).

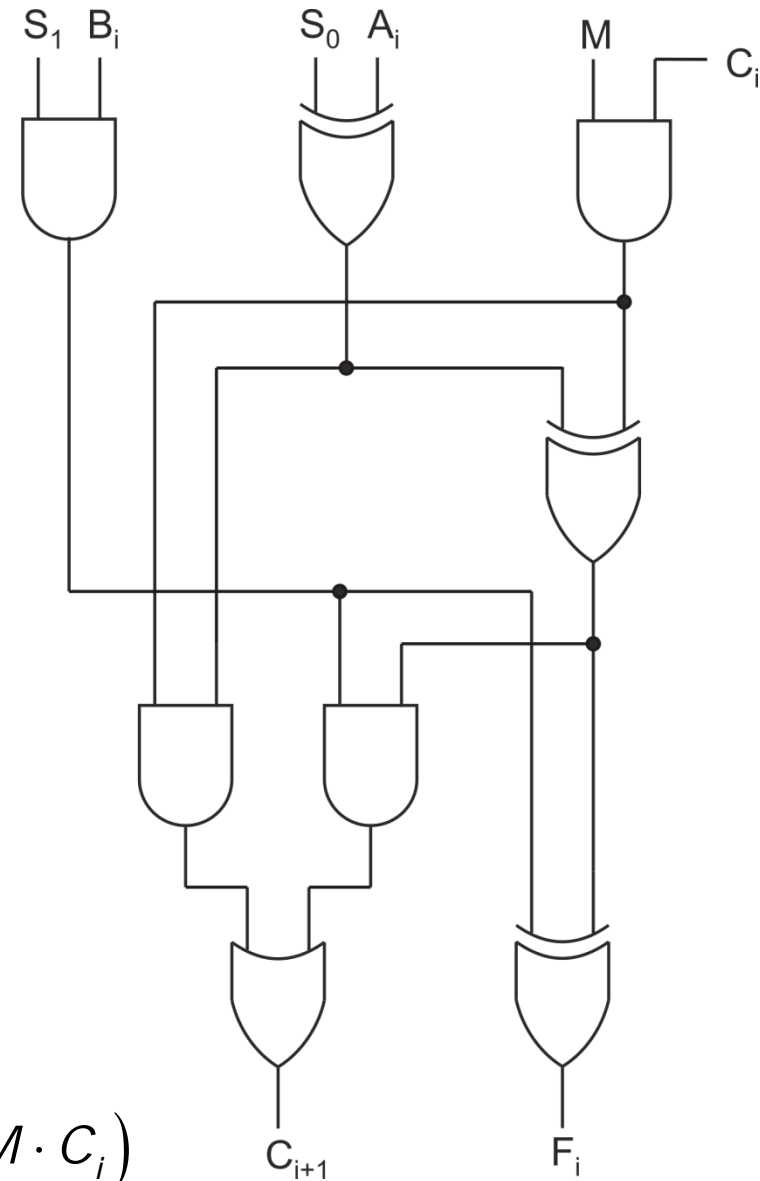
### Arithmetic Logic Unit

Circuito combinacional que realiza operaciones aritméticas y/o lógicas dependiendo de unas variables de control



$$F_i = ((S_0 \oplus A_i) \oplus M \cdot C_i) \oplus S_1 \cdot B_i$$

$$C_{i+1} = M \cdot C_i (S_0 \oplus A_i) + S_1 \cdot B_i ((S_0 \oplus A_i) \oplus M \cdot C_i)$$

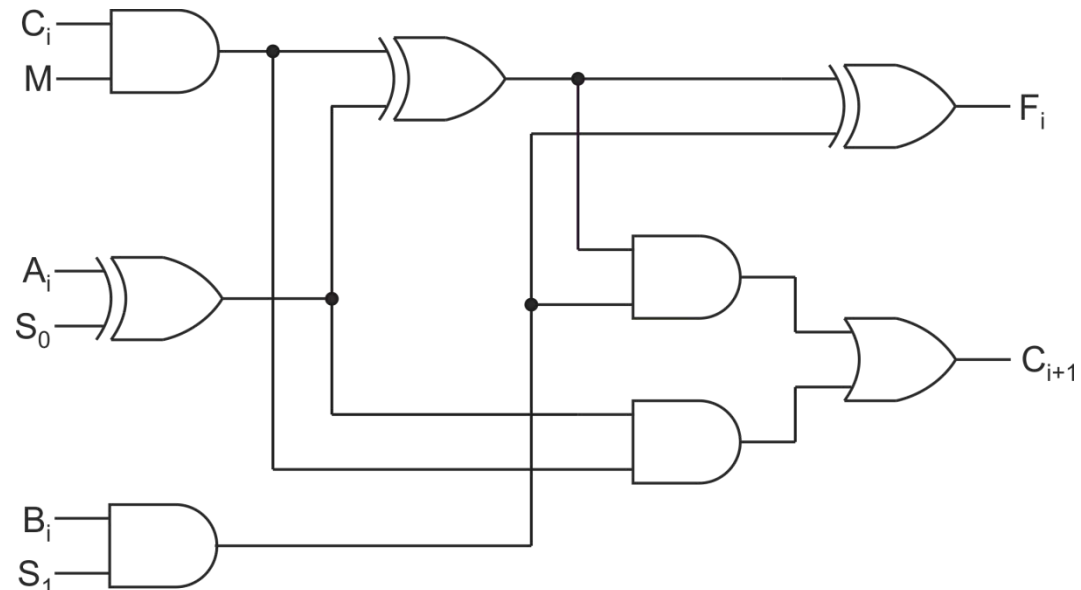


# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).

M=0 (Función Lógica)			
$S_1$	$S_0$	$F_i$ (F. Lógica)	Comentario
0	0	$F_i = A_i$	Entrada A transferida a la salida
0	1	$F_i = \bar{A}_i$	Entrada A complementada y transferida
1	0	$F_i = A_i \oplus B_i$	XOR
1	1	$F_i = \overline{A_i \oplus B_i}$	XNOR

$$F_i = S_0 \oplus A_i \oplus S_1 \cdot B_i$$
$$C_{i+1} = S_1 \cdot B_i (S_0 \oplus A_i)$$

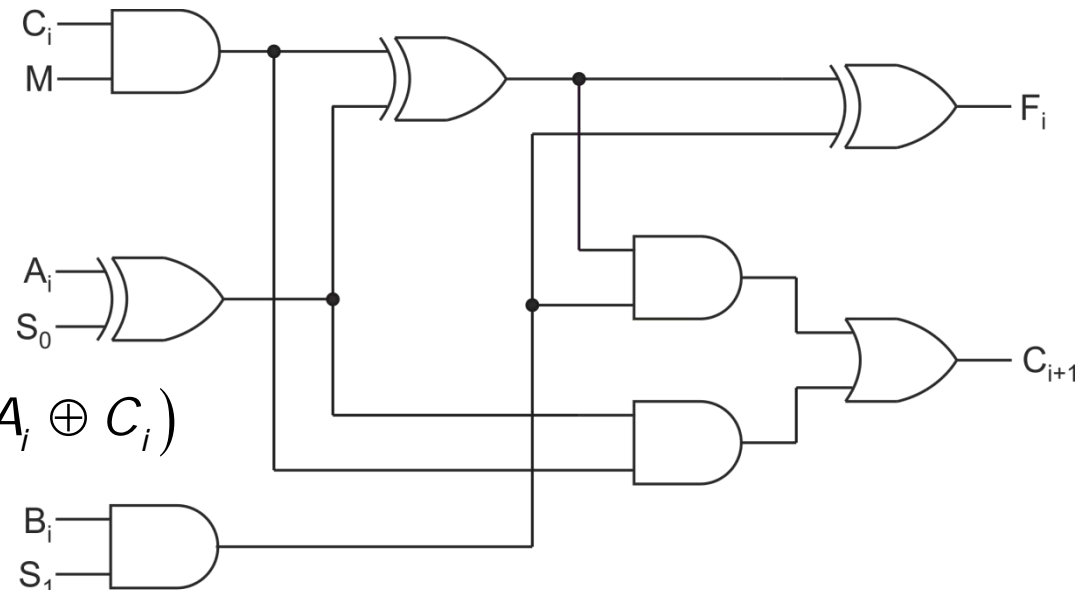


# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).

**M=1 (Función Aritmética) y  $C_i = 0$**

$S_1$	$S_0$	$F_i$ (F. aritmética)	Comentario
0	0	$F_i = A_i$	Entrada A transferida a la salida
0	1	$F_i = \bar{A}_i$	Complemento a uno de A
1	0	$F_i = A_i + B_i$	Suma de A y B
1	1	$F_i = \bar{A}_i + B_i$	Suma de B y el complemento a uno de A



$$F_i = S_0 \oplus A_i \oplus C_i \oplus S_1 \cdot B_i$$

$$C_{i+1} = C_i(S_0 \oplus A_i) + S_1 \cdot B_i(S_0 \oplus A_i \oplus C_i)$$

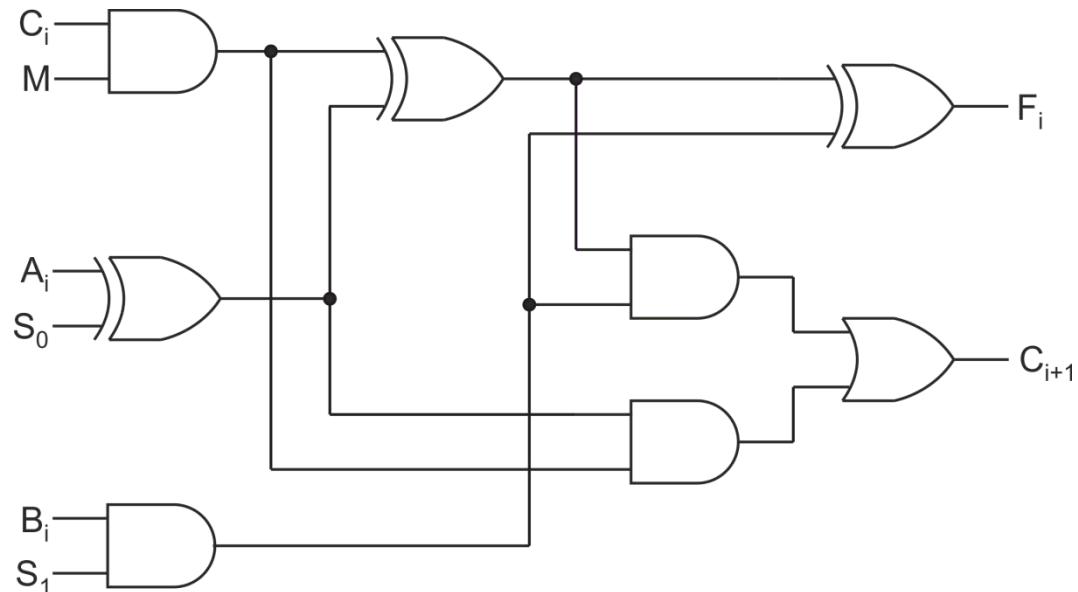


# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).

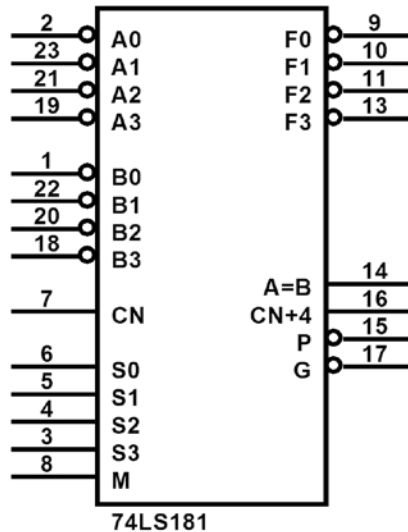
**M=1 (Función Aritmética) y  $C_i = 1$**

$S_1$	$S_0$	$F_i$ (F. aritmética)	Comentario
0	0	$F_i = A_i + 1$	Incrementa A
0	1	$F_i = \bar{A}_i + 1$	Complemento a dos de A
1	0	$F_i = A_i + B_i + 1$	Incrementa la suma de A y B
1	1	$F_i = \bar{A}_i + B_i + 1$	B menos A



# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).



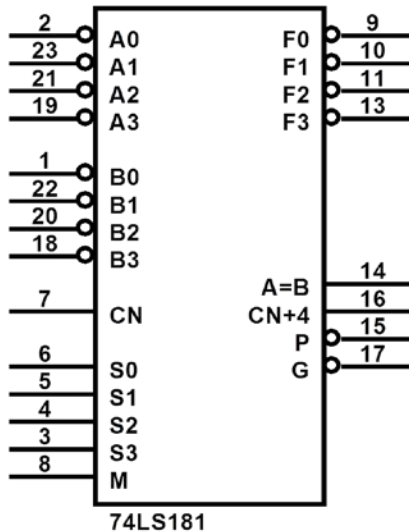
La ALU 4-bit 74LS181

### Pin Descriptions

Pin Names	Description
$\overline{A0}-\overline{A3}$	Operand Inputs (Active LOW)
$\overline{B0}-\overline{B3}$	Operand Inputs (Active LOW)
S0-S3	Function Select Inputs
M	Mode Control Input
$C_n$	Carry Input
$\overline{F0}-\overline{F3}$	Function Outputs (Active LOW)
A = B	Comparator Output
$\overline{G}$	Carry Generate Output (Active LOW)
$\overline{P}$	Carry Propagate Output (Active LOW)
$C_{n+4}$	Carry Output

# 2. Circuitos combinacionales aritméticos

## 2.2. Unidad Lógica Aritmética (ALU).



La ALU 4-bit 74LS181

Function Table

Mode Select Inputs				Active LOW Operands & F <sub>n</sub> Outputs		Active HIGH Operands & F <sub>n</sub> Outputs	
S3	S2	S1	S0	Logic	Arithmetic (Note 2)	Logic	Arithmetic (Note 2)
				(M = H)	(M = L) (C <sub>n</sub> = L)	(M = H)	(M = L) (C <sub>n</sub> = H)
L	L	L	L	$\bar{A}$	A minus 1	$\bar{A}$	A
L	L	L	H	$\overline{AB}$	AB minus 1	$\bar{A} + \bar{B}$	A + B
L	L	H	L	$\bar{A} + \bar{B}$	$A\bar{B}$ minus 1	$\bar{A} B$	A + $\bar{B}$
L	L	H	H	Logic 1	minus 1	Logic 0	minus 1
L	H	L	L	$\bar{A} + \bar{B}$	A plus (A + $\bar{B}$ )	$\overline{AB}$	A plus $A\bar{B}$
L	H	L	H	$\bar{B}$	AB plus (A + $\bar{B}$ )	$\bar{B}$	(A + B) plus $A\bar{B}$
L	H	H	L	$\bar{A} \oplus \bar{B}$	A minus B minus 1	A $\oplus$ B	A minus B minus 1
L	H	H	H	A + $\bar{B}$	A + $\bar{B}$	$A\bar{B}$	AB minus 1
H	L	L	L	$\bar{A} B$	A plus (A + B)	$\bar{A} + B$	A plus AB
H	L	L	H	A $\oplus$ B	A plus B	$\bar{A} \oplus \bar{B}$	A plus B
H	L	H	L	B	$A\bar{B}$ plus (A + B)	B	(A + $\bar{B}$ ) plus AB
H	L	H	H	A + B	A + B	AB	AB minus 1
H	H	L	L	Logic 0	A plus A (Note 1)	Logic 1	A plus A (Note 1)
H	H	L	H	$A\bar{B}$	AB plus A	A + $\bar{B}$	(A + B) plus A
H	H	H	L	AB	$A\bar{B}$ minus A	A + B	(A + $\bar{B}$ ) plus A
H	H	H	H	A	A	A	A minus 1

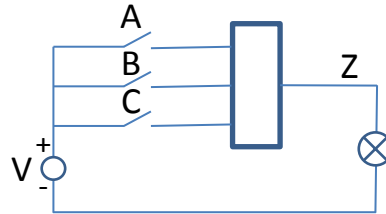
**Note 1:** Each bit is shifted to the next most significant position.

**Note 2:** Arithmetic operations expressed in 2s complement notation.

## 2. Circuitos combinacionales aritméticos

### Problemas.

1.- Resolver el problema de controlar una lámpara con 3 interruptores de manera que al variar uno de ellos la lámpara varíe. a) Utilizar puertas lógicas, b) multiplexor de 8:1, c) multiplexor de 4:1, d) decodificador de 3 entradas.



2.- Un proceso tiene 3 indicadores de temperatura del punto p cuyas salidas T1, T2 y T3 adoptan dos niveles de tensión bien diferenciadas, según que la temperatura sea menor o mayor e igual que  $t_1$ ,  $t_2$ ,  $t_3$  respectivamente,  $t_1 < t_2 < t_3$  (  $\text{temp} < t \rightarrow T = 0$ , y si  $\text{temp} \geq t \rightarrow T = 1$ ).

Se desea generar una señal que adopte un nivel 1 lógico si la temperatura está comprendida entre  $t_1$  y  $t_2$ , o es igual o superior a  $t_3$ , y nivel 0 lógico en caso contrario.

2.1.- Realizar el circuito eléctrico con puertas lógicas.

2.2.- Realizar el circuito eléctrico con un decodificador de 4 entradas

2.3.- Realizar el circuito eléctrico con un multiplexor de 4 a 1.

3.- Realizar un circuito combinacional que realice la suma aritmética de dos números binarios, uno de un bit, y otro de dos bits, y cuyo resultado también esté dado en binario. Representar el circuito mediante puertas lógicas.

### Problemas.

- 4.- Realizar un multiplexor de 4 entradas de datos en función de multiplexores de 2 entradas de datos.
- 5.- Realizar un decodificador de 4 entradas y 16 salidas a partir de decodificadores de 2 entradas y 4 salidas
- 6.- Diseñar un codificador que convierta una información en binario natural de 3 bits en un código de 7 segmentos (utilizar un display de 7 segmentos de cátodo común como el mostrado en la diapositiva 21)