

**CODIFICADORES  
CON  
PRIORIDAD**

# INDICE

- Definición de codificador
- Tipos de codificadores
  - Codificadores sin prioridad
  - Codificadores con prioridad
    - Circuito comercial de la familia 74
    - Descripción VHDL del dispositivo
    - Ejemplos de aplicaciones posibles del circuito
- Ejercicios
  - Resueltos
  - Trabajo personal
- Referencias

# DEFINICIÓN DE CODIFICADOR

Un codificador es un circuito con  $2^N$  entradas y N salidas, cuya misión es presentar en la salida el código binario correspondiente a la entrada activada.

Estos dispositivos MSI realizan la operación inversa a la realizada por los decodificadores.



# TIPOS DE CODIFICADORES

Existen dos tipos fundamentales de codificadores:

- Codificadores sin prioridad
- Codificadores con prioridad

# CODIFICADORES SIN PRIORIDAD

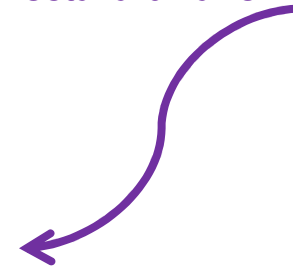
Su cometido es la generación de un número binario sobre sus  $n$  salidas que identifique cual de las entradas está activas.

Como no hay nada que impida el que se activen simultáneamente varias líneas de entrada de un codificador, los circuitos codificadores se diseñan normalmente para que respondan a una sola señal de entrada activa, dando lugar a los codificadores de prioridad.

# CODIFICADORES SIN PRIORIDAD

I0	I1	I2	I3	I4	I5	I6	I7		A2	A1	A0
1	0	0	0	0	0	0	0		0	0	0
0	1	0	0	0	0	0	0		0	0	1
0	0	1	0	0	0	0	0		0	1	0
0	0	0	1	0	0	0	0		0	1	1
0	0	0	0	1	0	0	0		1	0	0
0	0	0	0	0	1	0	0		1	0	1
0	0	0	0	0	0	1	0		1	1	0
0	0	0	0	0	0	0	1		1	1	1

Tabla de verdad de un codificador sin prioridad de octal a binario



Las funciones lógicas para las salidas, a partir de la tabla de verdad anterior y entrada ENABLE ( E ) para activar la entrada cuando vale 1 o inhibirla cuando vale 0, quedan:

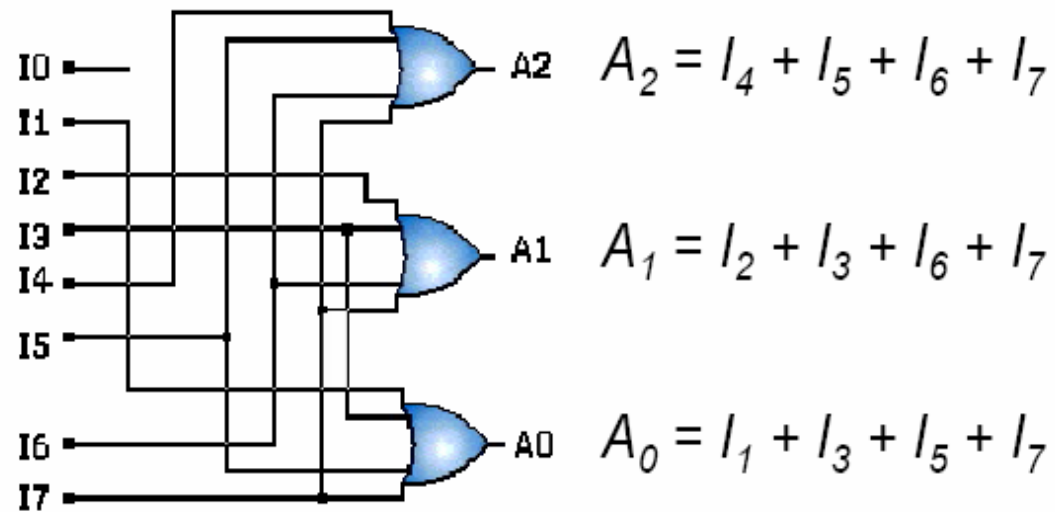
$$A_2 = ( I_4 + I_5 + I_6 + I_7 ) E$$

$$A_1 = ( I_2 + I_3 + I_6 + I_7 ) E$$

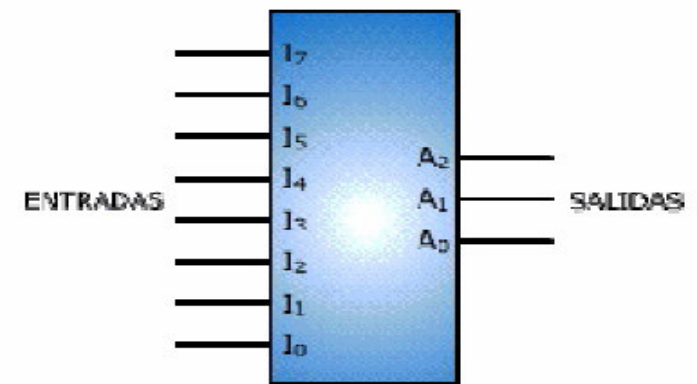
$$A_0 = ( I_1 + I_3 + I_5 + I_7 ) E$$

Siendo la función lógica :  $F = ( I_0 + I_1 + I_2 + I_3 + I_4 + I_5 + I_6 + I_7 ) E$  , una señal de salida especial que se activa cuando todas las señales de entrada son nulas y  $E=1$ .

# CODIFICADORES SIN PRIORIDAD



Ecuaciones de salida e implementación con puertas OR de un codificador de 8 entradas y 3 salidas



# CODIFICADORES CON PRIORIDAD

Los codificadores de prioridad son aquellos que seleccionan la entrada de mayor prioridad cuando se presentan varias entradas activas simultáneamente.

Para obtener codificadores que respondan a una sola señal de entrada activa, se le asigna valores fijos de prioridad a las líneas de entrada, de forma que en cada instante sólo se genera el código de salida de la entrada activa que tenga la máxima prioridad.



# CODIFICADORES CON PRIORIDAD

I0	I1	I2	I3	I4	I5	I6	I7		A2	A1	A0
1	0	0	0	0	0	0	0		0	0	0
X	1	0	0	0	0	0	0		0	0	1
X	X	1	0	0	0	0	0		0	1	0
X	X	X	1	0	0	0	0		0	1	1
X	X	X	X	1	0	0	0		1	0	0
X	X	X	X	X	1	0	0		1	0	1
X	X	X	X	X	X	1	0		1	1	0
X	X	X	X	X	X	X	1		1	1	1

Tabla de verdad de un codificador con prioridad de 8 a 3



**NOTA:** Las salidas se obtienen de la tabla como el OR de las filas en las que está a 1 sustituyendo expresiones y simplificando mediante el algebra de conmutación

# CODIFICADORES CON PRIORIDAD

Teniendo en cuenta que  $X_7 = I_7$ ,  $X_6 = I_7' I_6$ ,  $X_5 = I_7' I_6' I_5$ ,  $X_4 = I_7' I_6' I_5' I_4$ , ...,  $X_0 = I_7' I_6' I_5' I_4' I_3' I_2' I_1' I_0$ , las funciones lógicas para las salidas, a partir de la tabla de verdad anterior quedarían de la siguiente forma:

- $A_2 = X_4 + X_5 + X_6 + X_7 =$

$$I_7' I_6' I_5' I_4 + I_7' I_6' I_5 + I_7' I_6 + I_7 =$$

$$I_6' I_5' I_4 + I_6' I_5 + I_6 + I_7 = I_4 + I_5 + I_6 + I_7$$

- $A_1 = X_2 + X_3 + X_6 + X_7 =$

$$I_7' I_6' I_5' I_4' I_3' I_2 + I_7' I_6' I_5' I_4' I_3 + I_7' I_6 + I_7 =$$

$$I_5' I_4' (I_2 + I_3) + I_6 + I_7 =$$

$$I_5' I_4' I_2 + I_5' I_4' I_3 + I_6 + I_7$$

- $A_0 = X_1 + X_3 + X_5 + X_7 =$

$$I_7' I_6' I_5' I_4' I_3' I_2' I_1 + I_7' I_6' I_5' I_4' I_3 + I_7' I_6' I_5 + I_7 =$$

$$I_6' I_4' I_2' I_1 + I_6' I_4' I_3 + I_6' I_5 + I_7$$

# CODIFICADORES CON PRIORIDAD

Tabla de verdad de un codificador con prioridad de 4 a 2

$I_3 I_2$	00	01	11	10
00	X	0	0	0
01	1	1	1	1
11	1	1	1	1
10	1	1	1	1

$I_3 I_2$	00	01	11	10
00	X	0	1	1
01	0	0	0	0
11	1	1	1	1
10	1	1	1	1

ENTRADAS				SALIDAS	
X3	X2	X1	X0	A0	A1
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

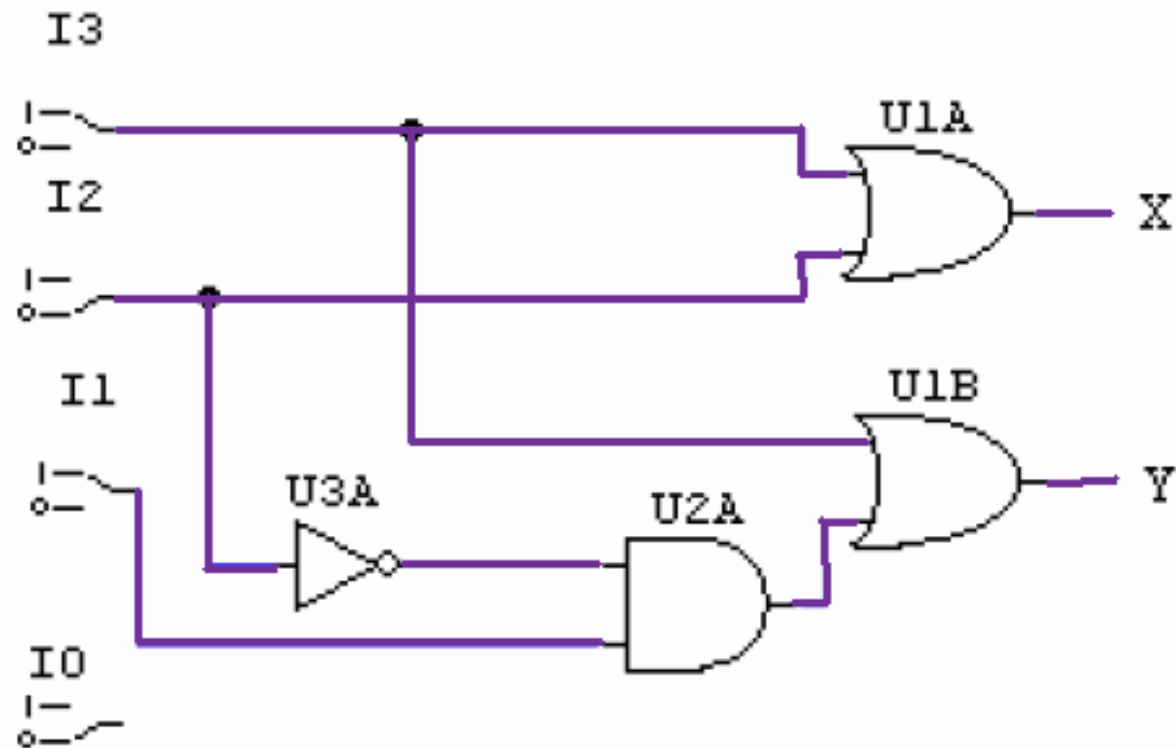
La función lógica que obtenemos a partir de los mapas Karnaugh sería la siguiente:

$$X = I_3 + I_2$$

$$Y = I_3 + \overline{I_2} * I_1$$

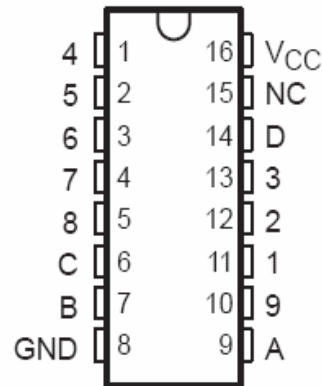
# CODIFICADORES CON PRIORIDAD

Implementación lógica:

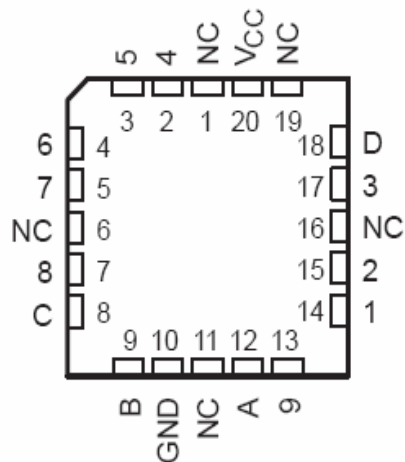


# CIRCUITO COMERCIAL DE LA FAMILIA 74

SN54147, SN54LS147 . . . J OR W PACKAGE  
 SN74147, SN74LS147 . . . D OR N PACKAGE  
 (TOP VIEW)



SN54LS147 . . . FK PACKAGE  
 (TOP VIEW)



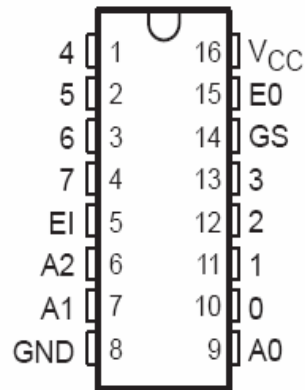
FUNCTION TABLE - '147, 'LS147

INPUTS									OUTPUTS			
1	2	3	4	5	6	7	8	9	D	C	B	A
H	H	H	H	H	H	H	H	H	H	H	H	H
X	X	X	X	X	X	X	X	L	L	H	H	L
X	X	X	X	X	X	X	L	H	L	H	H	H
X	X	X	X	X	X	L	H	H	H	L	L	L
X	X	X	X	X	L	H	H	H	H	L	L	H
X	X	X	X	L	H	H	H	H	H	L	H	L
X	X	X	L	H	H	H	H	H	H	H	L	H
X	L	H	H	H	H	H	H	H	H	H	L	H
L	H	H	H	H	H	H	H	H	H	H	H	L

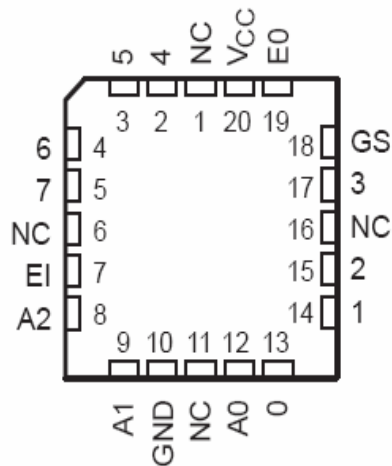
H = high logic level, L = low logic level, X = irrelevant

# CIRCUITO COMERCIAL DE LA FAMILIA 74

SN54148, SN54LS148 . . . J OR W PACKAGE  
 SN74148, SN74LS148 . . . D, N, OR NS PACKAGE  
 (TOP VIEW)



SN54LS148 . . . FK PACKAGE  
 (TOP VIEW)



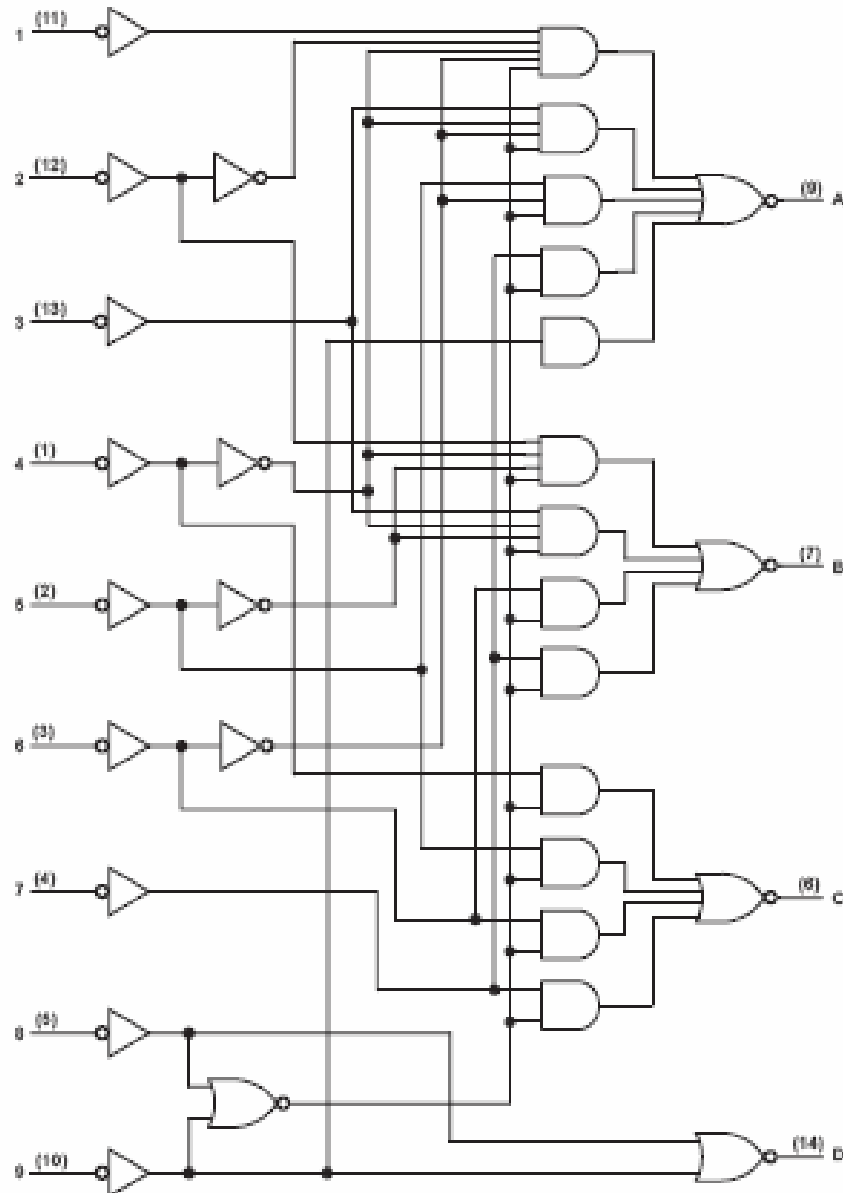
FUNCTION TABLE - '148, 'LS148

		INPUTS								OUTPUTS				
EI		0	1	2	3	4	5	6	7	A2	A1	A0	GS	EO
H		X	X	X	X	X	X	X	X	H	H	H	H	H
L		H	H	H	H	H	H	H	H	H	H	H	H	L
L		X	X	X	X	X	X	X	L	L	L	L	L	H
L		X	X	X	X	X	X	L	H	L	L	H	L	H
L		X	X	X	X	X	L	H	H	L	H	L	L	H
L		X	X	X	L	H	H	H	H	H	L	L	L	H
L		X	X	L	H	H	H	H	H	H	L	L	L	H
L		X	L	H	H	H	H	H	H	H	H	L	L	H
L		L	H	H	H	H	H	H	H	H	H	H	L	H

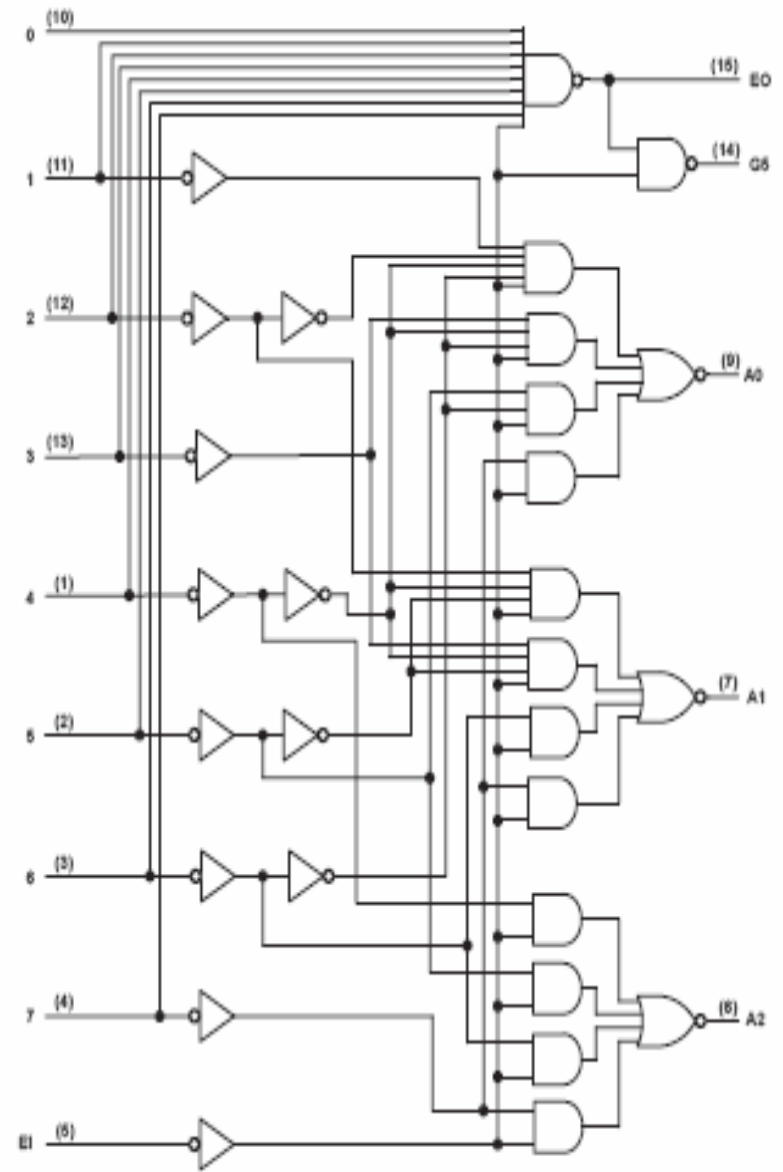
H = high logic level, L = low logic level, X = irrelevant

# CIRCUITO COMERCIAL DE LA FAMILIA 74

'147, 'LS147 logic diagram (positive logic)



'148, 'LS148 logic diagram (positive logic)




# DESCRIPCIÓN VHDL DEL DISPOSITIVO

## Codificador de prioridad de 8 a 3 en VHDL

```
library IEEE;
use IEEE.std_logic_1164.all; -- Libreria estandar

entity CodificadorPrioridad_8_a_4 is
port(
  I : in std_logic_vector(7 downto 0); -- Entradas
  Y : out std_logic; -- Salida de activacion
  B : out std_logic_vector(2 downto 0) -- Salida codificada
);
end CodificadorPrioridad_8_a_4;
```

```
architecture Codificador of CodificadorPrioridad_8_a_4 is
begin
  process(I)
  begin
    if I(7)='1' then
      Y <= '1';
      B <= "111";
    elsif I(6)='1' then
      Y <= '1';
      B <= "110";
    elsif I(5)='1' then
      Y <= '1';
      B <= "101";
    elsif I(4)='1' then
      Y <= '1';
      B <= "100";
    elsif I(3)='1' then
      Y <= '1';
      B <= "011";
    elsif I(2)='1' then
      Y <= '1';
      B <= "010";
    elsif I(1)='1' then
      Y <= '1';
      B <= "001";
    elsif I(0)='1' then
      Y <= '1';
      B <= "000";
    else
      Y <= '0';
      B <= "000";
    end if;
  end process;
end Codificador;
```



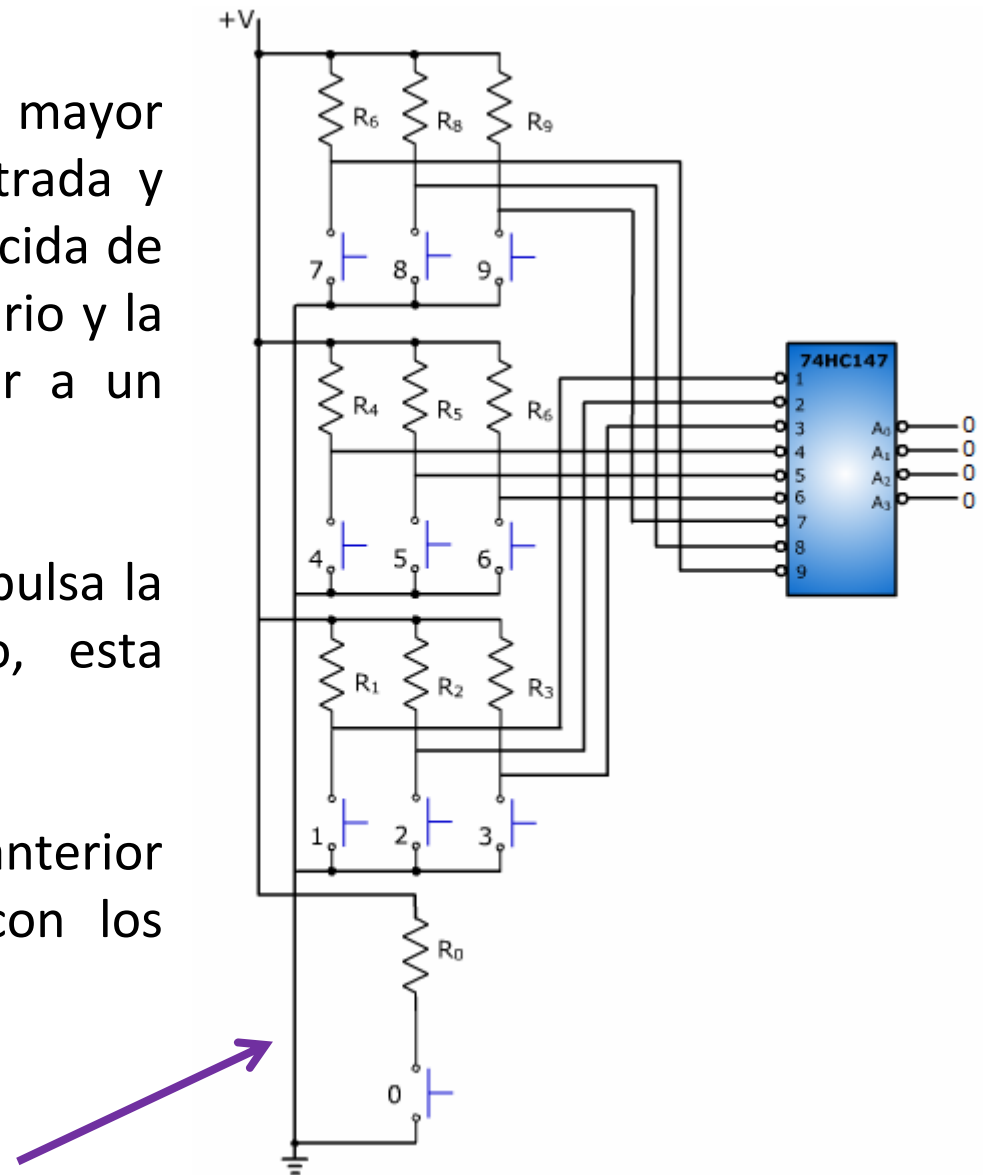


# APLICACIONES POSIBLES DEL CIRCUITO

Los codificadores encuentran mayor aplicación en los dispositivos de entrada y salida. La señal de entrada es introducida de una forma comprensible para el usuario y la "traducción" la realiza el codificador a un código comprensible para el equipo.

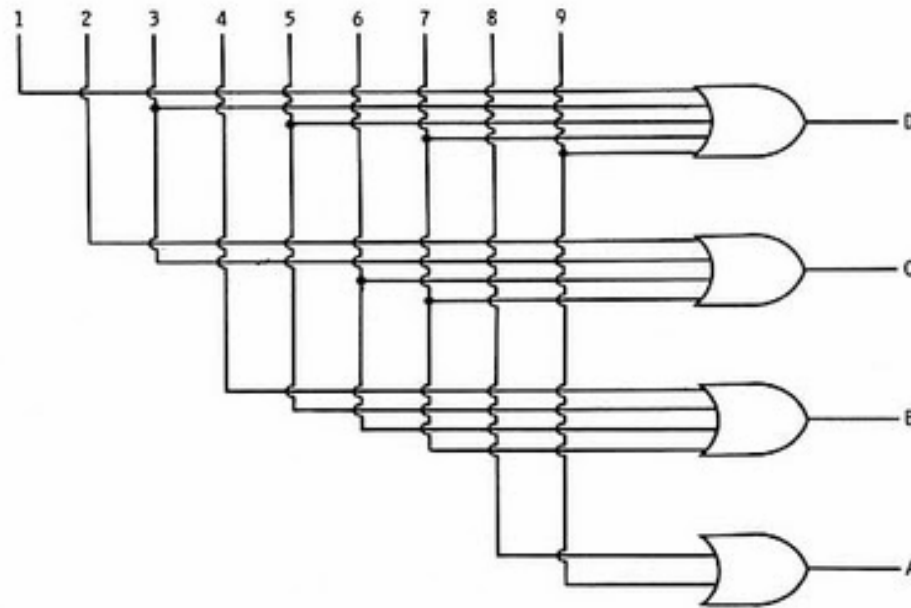
**EJEMPLO:** En un teclado, cuando se pulsa la tecla correspondiente a un dígito, esta entrada se codifica en código *BCD*.

La animación muestra la aplicación anterior por medio de una interactividad con los pulsadores.



# EJERCICIOS

1. Analizar el siguiente circuito construyendo una Tabla de Verdad para el mismo. Solamente una entrada puede estar activada a la vez.



# EJERCICIOS

1	2	3	4	5	6	7	8	9		A	B	C	D
1	0	0	0	0	0	0	0	0		0	0	0	1
0	1	0	0	0	0	0	0	0		0	0	1	0
0	0	1	0	0	0	0	0	0		0	0	1	1
0	0	0	1	0	0	0	0	0		0	1	0	0
0	0	0	0	1	0	0	0	0		0	1	0	1
0	0	0	0	0	1	0	0	0		0	1	1	0
0	0	0	0	0	0	1	0	0		0	1	1	1
0	0	0	0	0	0	0	1	0		1	0	0	0
0	0	0	0	0	0	0	0	1		1	0	0	1

Se puede apreciar en la Tabla de verdad que el circuito "convierte" los números decimales en las terminales de entrada a su equivalente en sistema binario en las terminales de salida. Estos tipos de circuitos reciben el nombre de codificador (*encoder*).

# EJERCICIOS

2. Escriba la tabla de verdad de un codificador de prioridad de 8 entradas con Ein, GS y Eout. Diseñe un codificador de prioridad de 10 entradas utilizando el codificador de prioridad de 8 entradas y el mínimo número de puertas AND, OR y NOT necesarias.

# EJERCICIOS

En la tabla siguiente se puede ver la tabla de verdad del codificador de prioridad de 8 bits mencionado:

Ein	I7	I6	I5	I4	I3	I2	I1	I0	O2	O1	O0	GS	Eout
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

Para diseñar el codificador de prioridad de 10 bits tendremos que conectar un pequeño codificador de 2 bits con el circuito que responda a la tabla de verdad anterior. Para ellos vamos también a obtener en primer lugar su tabla de verdad.

Ein	I1	I0	O0	GS	Eout
0	X	X	0	0	0
1	0	0	0	0	1
1	0	1	0	1	0
1	1	X	1	1	0

# EJERCICIOS

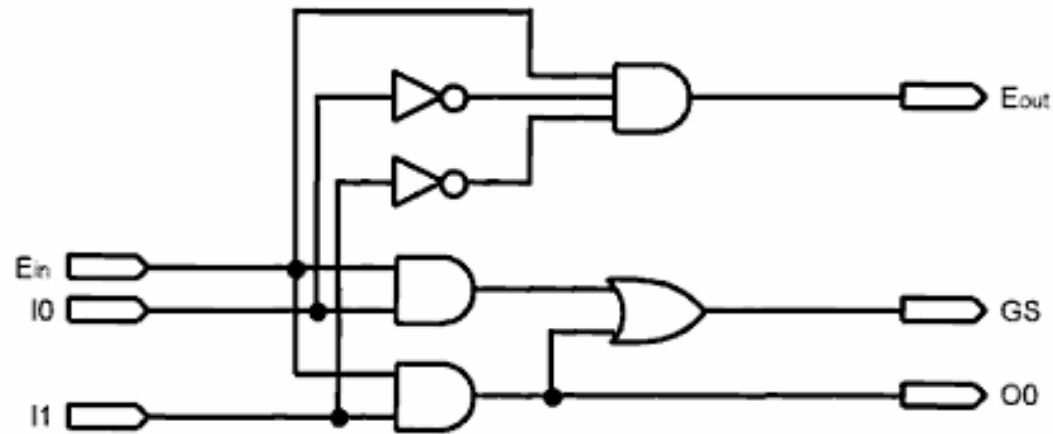
A continuación obtenemos las ecuaciones correspondientes a las tres funciones de salida:

$$O_0 = E_{in} \cdot I_1$$

$$E_{out} = E_{in} \cdot \bar{I}_0 \cdot \bar{I}_1$$

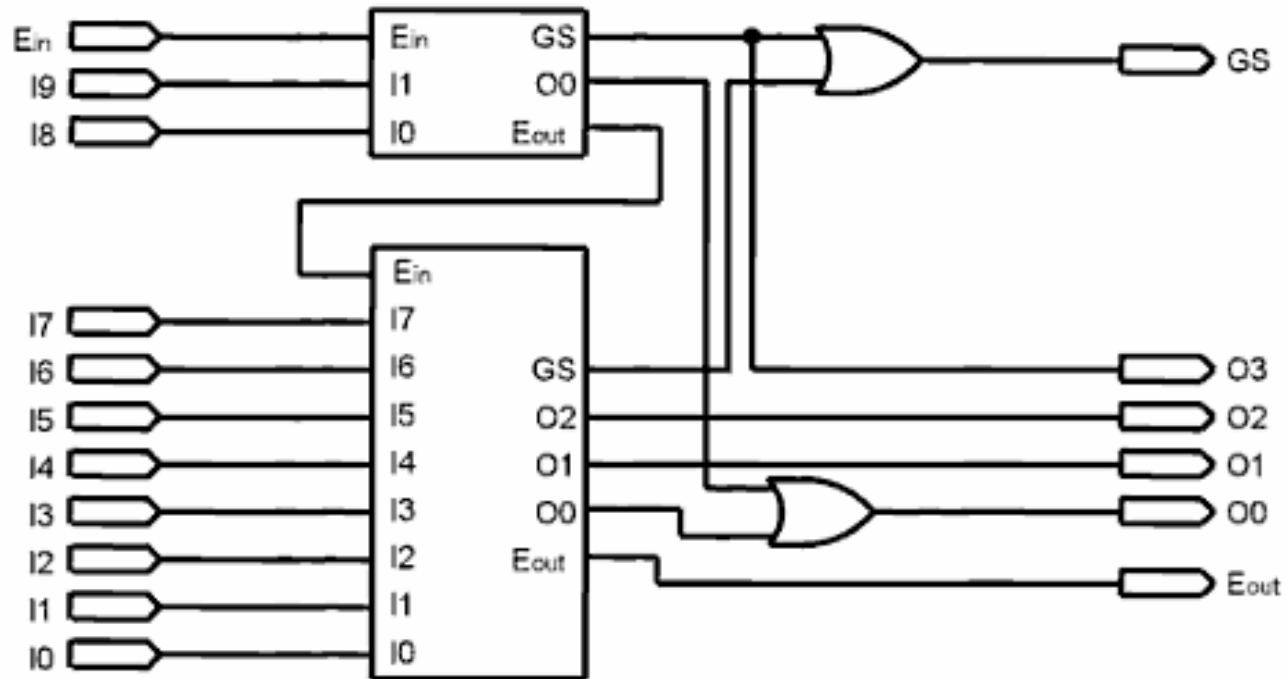
$$G_S = E_{in} \cdot (I_0 + I_1) = E_{in} \cdot I_0 + E_{in} \cdot I_1$$

A partir de estas ecuaciones obtenemos el siguiente circuito:



# EJERCICIOS

Finalmente conectamos el circuito anterior como un bloque con el circuito correspondiente al codificador de prioridad de 8 bits añadiendo las puertas necesarias:



# EJERCICIOS

- 3. Diseñe un codificador de prioridad de cuatro entradas (I0,I1,I2 e I3) que disponga de una entrada de habilitación (E) y una salida de control (G) que indique si la salida de datos es correcto o no. Las señales de control serán activas a nivel bajo. Tanto las entradas como las salidas de datos (O0 y O1) serán activas a nivel alto.***

**La comunicación entre una serie de periféricos y un ordenador se realiza mediante un sistema de prioridades. Para atender el periférico de mayor prioridad se utiliza un sistema de codificadores de prioridad, indicando al ordenador el código binario de dicho periférico. Utilizando el codificador de prioridad anterior, diseñe el sistema de prioridades del ordenador para seis periféricos mediante dos codificadores de prioridad y el mínimo número de puertas lógicas necesarias**



# EJERCICIOS

En primer lugar vamos a realizar la tabla de verdad del circuito solicitado:

E	I3	I2	I1	I0	O1	O0	G
1	X	X	X	X	0	0	1
0	0	0	0	0	0	0	1
0	0	0	0	1	0	0	0
0	0	0	1	X	0	1	0
0	0	1	X	X	1	0	0
0	1	X	X	X	1	1	0

A partir de esta tabla se pueden utilizar los mapas de Karnaugh de 5 variables para obtener las funciones de salida, obteniéndose el siguiente resultado:

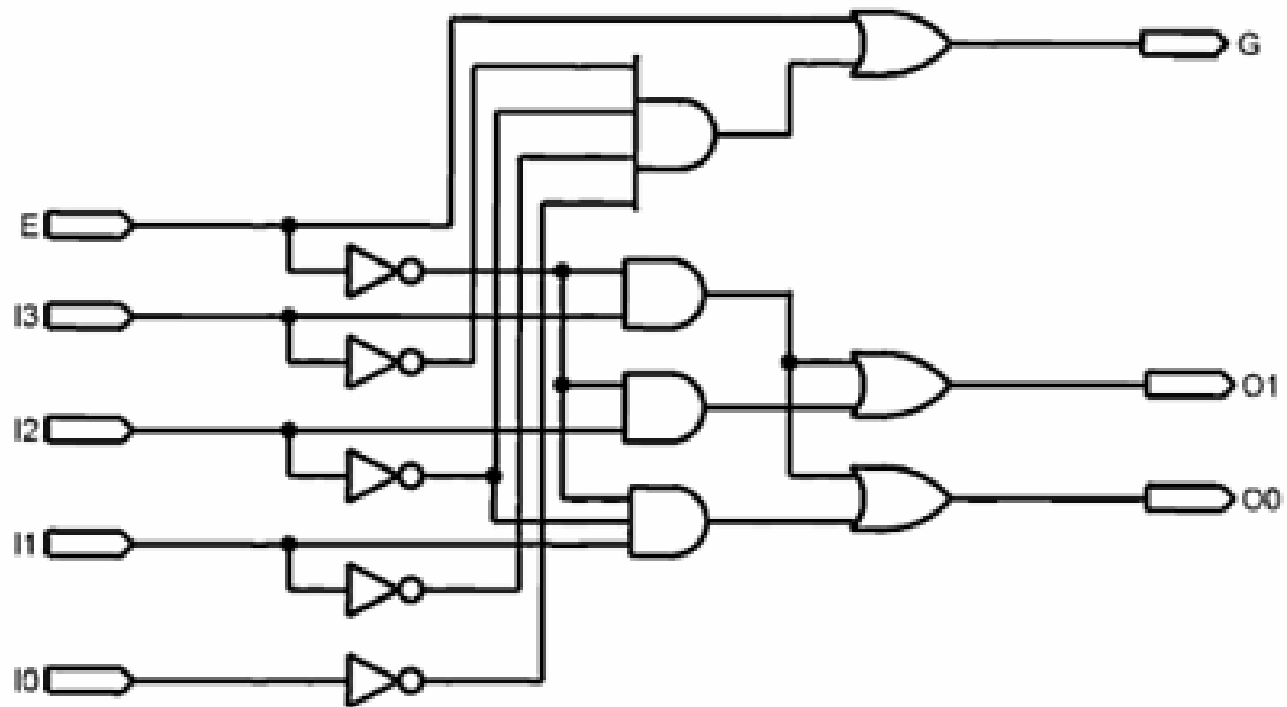
$$O_1 = \bar{E} \cdot I_3 + \bar{E} \cdot I_2$$

$$O_0 = \bar{E} \cdot I_3 + \bar{E} \cdot \bar{I}_2 \cdot I_1$$

$$G = \bar{E} + I_3 \cdot \bar{I}_2 \cdot \bar{I}_1 \cdot \bar{I}_0$$

# EJERCICIOS

Con las ecuaciones anteriores se construye el siguiente circuito:

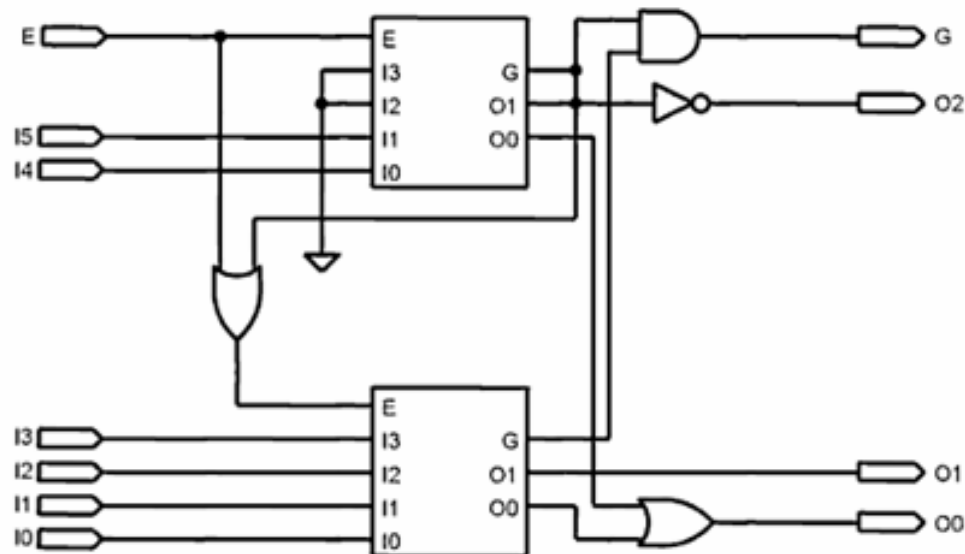


# EJERCICIOS

Para obtener un codificador de prioridad de 6 entradas a partir de dos codificadores de 4 entradas como el diseñado, conectaremos las dos entradas más significativas a tierra. La entrada E del codificador de menor prioridad deberá estar desactivada cuando esté desactivado el circuito completo o haya una entrada activa en el codificador de mayor prioridad (indicado por G). Para obtener las salidas, bastará con tener en cuenta lo siguiente:

- G debe ser activa cuando lo sea cualquiera de las dos salidas homónimas de los dos codificadores. Como es activa a nivel bajo, utilizaremos una puerta AND.
- De la misma forma O0 debe ser activa cuando lo sea cualquiera de las dos salidas homónimas de los dos decodificadores. En este caso son activas a nivel alto por lo que utilizaremos una puerta OR.
- O1 es la correspondiente al codificador de menor prioridad ya que la del de mayor prioridad siempre será 0 puesto que sus entradas más significativas están a tierra.
- O2 es la inversa de G del codificador de mayor prioridad, puesto que cuando éste tenga una entrada, esta salida estará a 0.

Con todo ellos el circuito resultante es el siguiente:



# EJERCICIOS PARA TRABAJO PERSONAL

1. Realizar con puertas lógicas, un codificador de 4 a 2 líneas con salida en binario natural con prioridad a la entrada de menor peso.
2.
  - a) Encontrar las ecuaciones lógicas que permiten definir un circuito codificador con prioridad baja de 8 bits de entrada (I7-I0) y salidas en código gray ( de mas a menos significativas: A B C ). Factorizar en lo posible las ecuaciones lógicas suponiendo todas las entradas y salidas asertadas bajas.
  - b) Construir un circuito codificador binario de 8 a 3 con prioridad baja tomando como base el circuito codificador 74LS148, y el menor número posible de puertas lógicas que sean necesarias. Se permite definir como mejor convenga la polaridad de las entradas y de las salidas.
3. Encontrar las ecuaciones lógicas que permiten definir un circuito codificador con prioridad baja de 8 bits de entrada (I7-I0) y salidas en código Gray (de más a menos significativas: A B C).
4. Construir un circuito codificador binario de 8 a 3 con prioridad baja tomando como base el circuito codificador 74LS148, y el menor número posible de puertas lógicas que sean necesarias. Se permite definir como mejor convenga la polaridad de las entradas y de las salidas.

# REFERENCIAS

- <http://es.wikipedia.org/wiki/Codificador>
- [http://www.uhu.es/rafael.lopezahumada/Cursos\\_anteriores/fund97\\_98/combinacional.es.pdf](http://www.uhu.es/rafael.lopezahumada/Cursos_anteriores/fund97_98/combinacional.es.pdf)
- [http://profesormolina2.iespana.es/electronica/componentes/int/sist\\_comb.htm](http://profesormolina2.iespana.es/electronica/componentes/int/sist_comb.htm)
- <http://www.virtual.unal.edu.co/cursos/ingenieria/2000477/lecciones/030501.htm>
- <http://foros.hackerss.com/t9850//getlastpost>
- <http://personales.unican.es/manzanom/EDigitalI/CODG10.pdf>
- <http://personales.unican.es/manzanom/EdigitalI/Familia74.pdf>
- <http://www.datasheetcatalog.org/datasheet2/b/0dzjaudyk7l9gayo3dz9o03g9dyy.pdf>
- [http://www.el.uma.es/MaterialSED/Transp\\_LCModular.pdf](http://www.el.uma.es/MaterialSED/Transp_LCModular.pdf)
- [http://books.google.es/books?id=sHGQT1dxKWgC&pg=PT114&lpg=PT114&dq=ejercicios+resueltos+codificadores&source=bl&ots=WxO3ID5YKo&sig=3LrkBumBZfcQ\\_1IBYERNqqK-Tc&hl=es&ei=hcEXS9zeAc-gjAfwiZD3Aw&sa=X&oi=book\\_result&ct=result&resnum=7&ved=0CB0Q6AEwBg#v=onepage&q=&f=false](http://books.google.es/books?id=sHGQT1dxKWgC&pg=PT114&lpg=PT114&dq=ejercicios+resueltos+codificadores&source=bl&ots=WxO3ID5YKo&sig=3LrkBumBZfcQ_1IBYERNqqK-Tc&hl=es&ei=hcEXS9zeAc-gjAfwiZD3Aw&sa=X&oi=book_result&ct=result&resnum=7&ved=0CB0Q6AEwBg#v=onepage&q=&f=false)
- <http://logica-digital.blogspot.com/2007/11/problemas-resueltos.html>
- LIBRO → Problemas resueltos electronica digital. Jagoba Arias Pérez, Unai Bidarte Peraita, Pedro Ibáñez Ereño....

