

CODIFICADORES CON

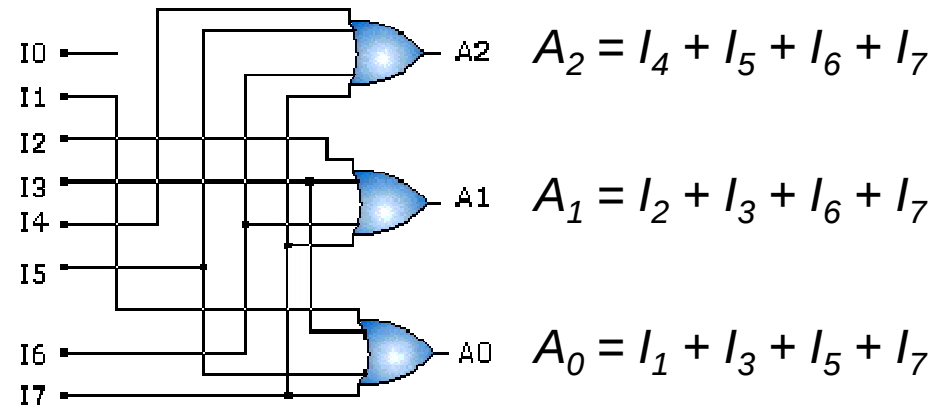
PRIORIDAD

- Codificadores
- Codificadores con prioridad
- Implementación lógica de un codificador con prioridad
- Descripción VHDL
- Familia lógica 74
- Problemas

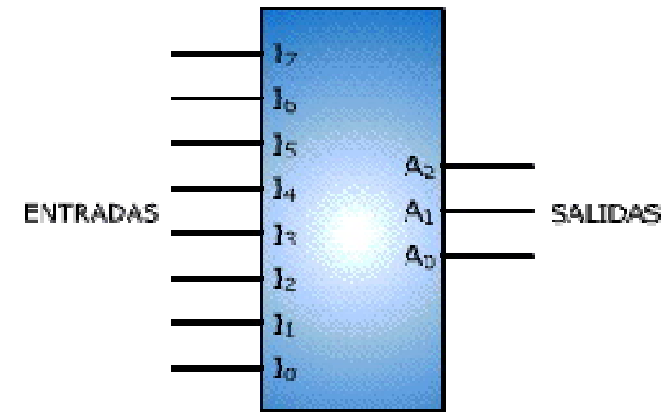
Codificadores

- Es un circuito digital que realiza la operación inversa de la que efectúa un decodificador.
- Convierte la información de un número de entradas M a N bits de salida, donde $M \leq 2^N$. Las salidas generan el código binario correspondiente al valor de entrada.
- Lo normal es que se realicen codificadores de 2^N entradas y N salidas (a binario).
- En general este tipo de codificador puede construirse a partir de n compuertas OR de 2^{n-1} .

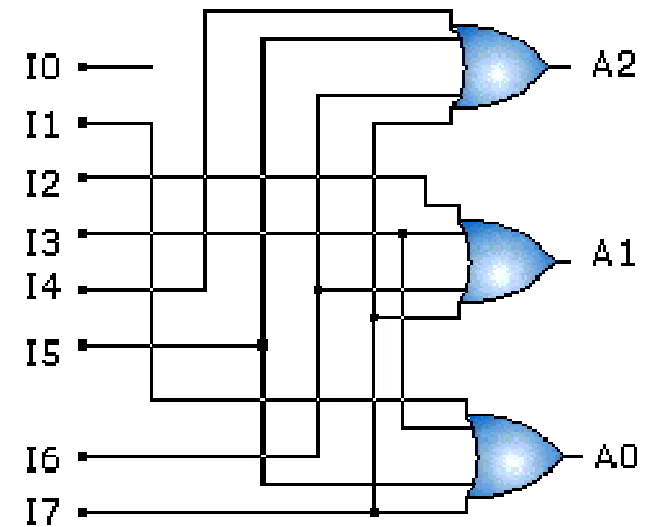
A continuación mostramos la tabla de verdad, las ecuaciones de salida y la implementación con puertas OR de un codificador de 8 entradas y 3 salidas:



Entradas								Salidas		
I0	I1	I2	I3	I4	I5	I6	I7	A ₂	A ₁	A ₀
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



Entradas								Salidas		
I0	I1	I2	I3	I4	I5	I6	I7	X	Y	Z
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	1	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



- Por ejemplo, supongamos que las entradas I2 e I4 del codificador 8 a 3 están asertadas; entonces la salida es 110, el código binario de 6. Las salidas útiles serían 2 o 4 y no 6.

Codificadores con prioridad

- Este tipo de codificadores permiten que varias entradas estén activadas simultáneamente. La siguiente tabla muestra la lógica de entrada y salida de un codificador con prioridad.

Entradas								Salidas		
I0	I1	I2	I3	I4	I5	I6	I7	A ₂	A ₁	A ₀
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

Codificadores con prioridad de 8 a 3

Entradas								Salidas		
I ₀	I ₁	I ₂	I ₃	I ₄	I ₅	I ₆	I ₇	A ₂	A ₁	A ₀
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

Para calcular las expresiones lógicas de las tres salidas A₂ A₁ A₀, indicamos cada fila de la tabla por una expresión lógica.

- X₇=I₇
- X₆=I₇' I₆
- X₅=I₇' I₆' I₅
- X₄=I₇' I₆' I₅' I₄
-
- X₀=I₇' I₆' I₅' I₄' I₃' I₂' I₁' I₀

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

$$A_2 = X_4 + X_5 + X_6 + X_7 = I_7' I_6' I_5' I_4 + I_7' I_6' I_5 + I_7' I_6 + I_7 = I_6' I_5' I_4 + I_6' I_5 + I_6 + I_7 = I_4 + I_5 + I_6 + I_7$$

$$A_1 = X_2 + X_3 + X_6 + X_7 = I_7' I_6' I_5' I_4' I_3' I_2 + I_7' I_6' I_5' I_4' I_3 + I_7' I_6 + I_7 = I_5' I_4' (I_2 + I_3) + I_6 + I_7 = I_5' I_4' I_2 + I_5' I_4' I_3 + I_6 + I_7$$

$$A_0 = X_1 + X_3 + X_5 + X_7 = I_7' I_6' I_5' I_4' I_3' I_2' I_1 + I_7' I_6' I_5' I_4' I_3 + I_7' I_6' I_5 + I_7 = I_6' I_4' I_2' I_1 + I_6' I_4' I_3 + I_6' I_5 + I_7$$

Codificador con prioridad de 4 entradas y 2 salidas.

1º Hacemos la tabla de verdad, suponiendo que la prioridad la da el de mayor peso.

Entradas				Salidas	
I3	I2	I1	I0	X	Y
0	0	0	1	0	0
0	0	1	X	0	1
0	1	X	X	1	0
1	X	X	X	1	1

2º Sacamos la función lógica a partir de mapas de Karnaugh:

		I1I0			
		00	01	11	10
I3I2	00	X	0	0	0
	01	1	1	1	1
	11	1	1	1	1
	10	1	1	1	1

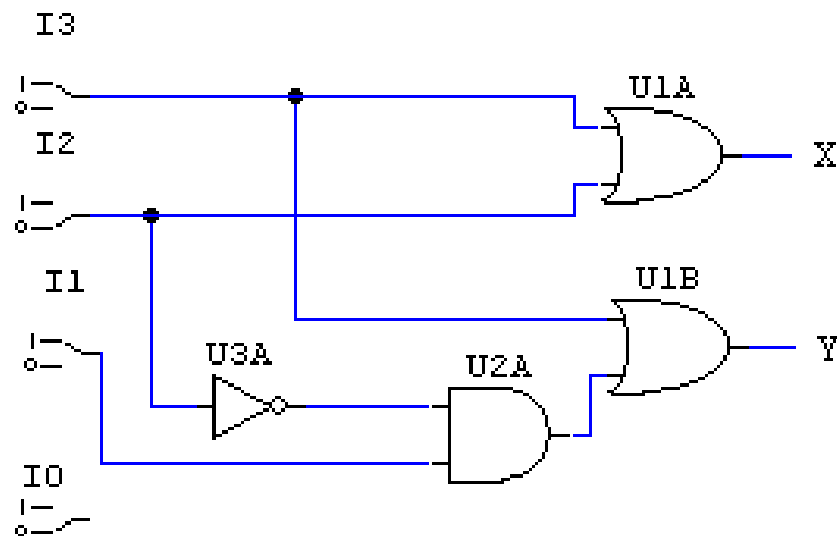
		I1I0			
		00	01	11	10
I3I2	00	X	0	1	1
	01	0	0	0	0
	11	1	1	1	1
	10	1	1	1	1

- La solución de los Karnaugh es la siguiente:

$$X = I_3 + I_2$$

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

- La implementación lógica es:



Descripción VHDL de un codificador con prioridad

```
library ieee;  
use ieee.std_logic_1164.all
```

```
entity cod is  
  port(I:in std_logic_vector(4 downto 1);  
        Z:out std_logic_vector(1 downto 0);  
end cod;
```

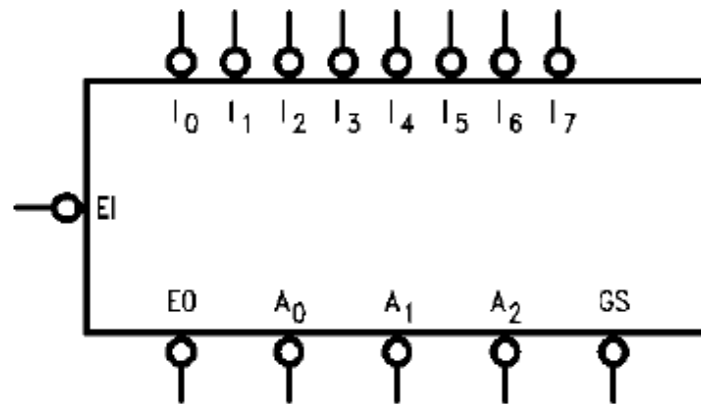
```
architecture vhd1 of cod is  
begin  
  process (I)  
  begin  
    if(I(3)='1') then  
      Z<="11";  
    elsif (I(2)='1') then  
      Z<="10";  
    elsif (I(1)='1') then  
      Z<="01";  
    else  
      Z<="00";  
    end if;  
  end process;  
end vhd1;
```

- La prioridad la da el orden de los if.

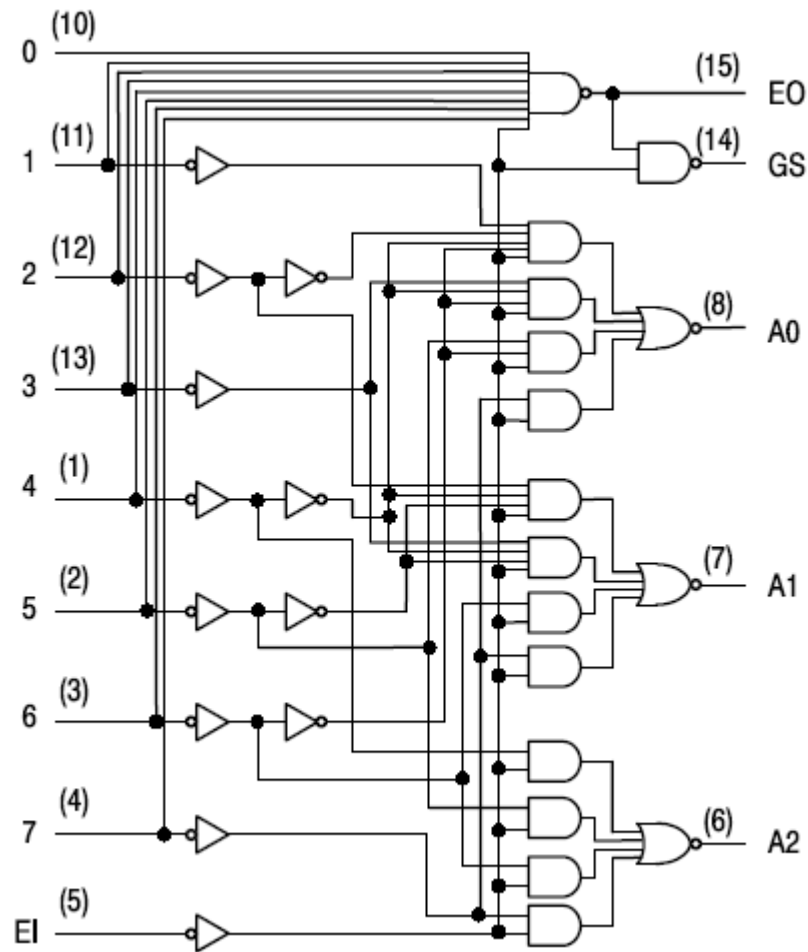
Familia lógica 74

- **El codificador con prioridad 74x148**

Es un codificador de prioridad de 8 entradas MSI. La principal diferencia con el codificador de prioridad “genérico”, es que sus señales activas de entrada y salida están en estado bajo activas y que tiene una entrada de habilitación, EI_L que debe ser asertiva para que cualquiera de sus salidas sea asertiva.



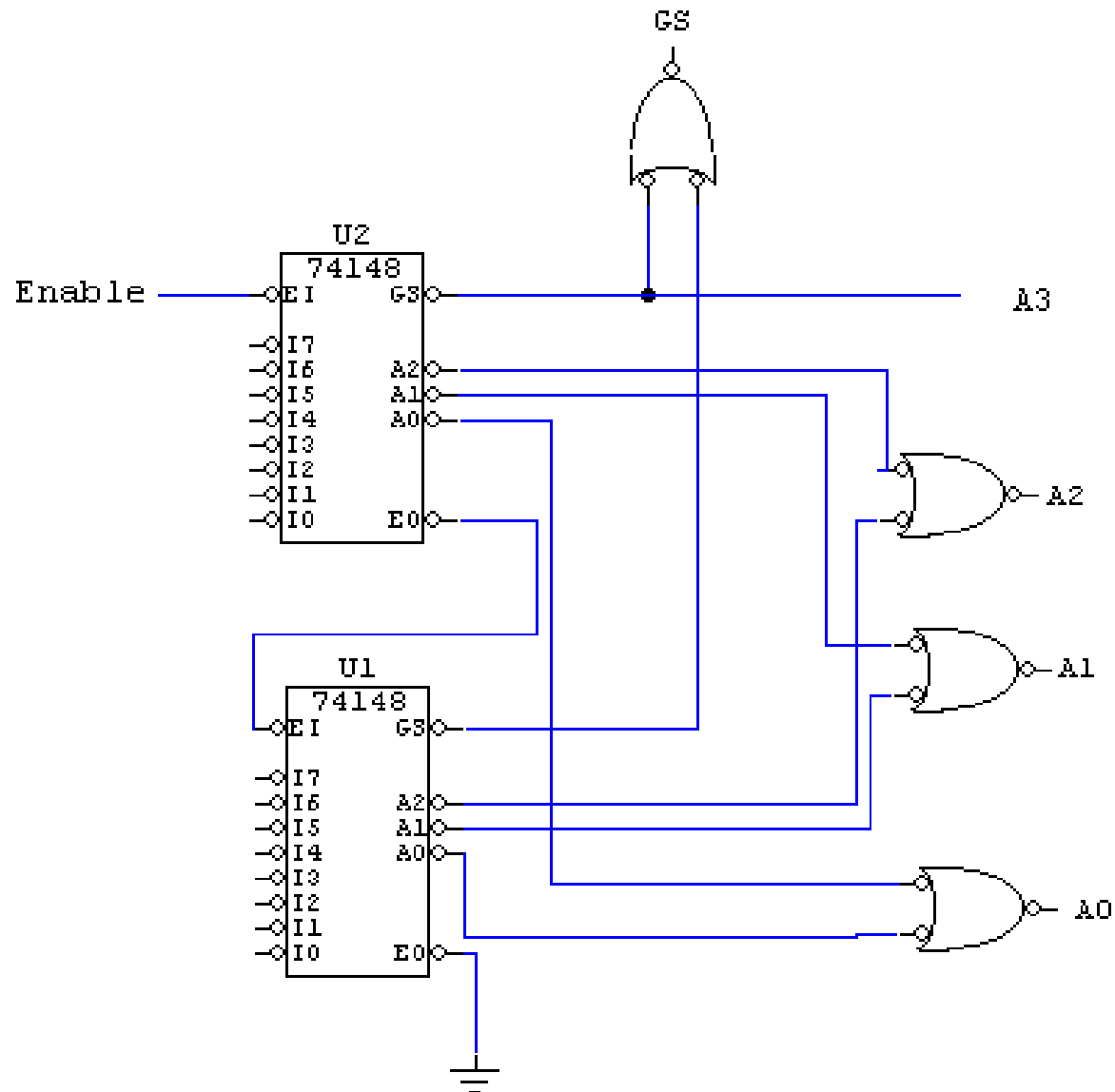
Familia lógica 74



FUNCTION TABLE

[illegible]

Codificador de prioridad 16 a 4 a partir de 2 codificadores de 8 a 3



Problemas Propuestos

Ejercicio 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.

Ejercicio 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.

Ejercicio 3

Se quiere diseñar un circuito que tenga como entrada un número N de 8 bits ($N_7 N_6 \dots N_1 N_0$) y como salida un número C de 8 bits ($C_7 C_6 \dots C_1 C_0$) de forma que C sea el complemento a 2 de N . El circuito debe realizar la transformación siguiendo este algoritmo:

- Sea N_k el bit de índice k menos significativo de N a 1, entonces para cada bit i de C , $C_i = N_i$ para todo i menor o igual que k , y C_i igual a N_i' para todo índice i mayor que k .

El circuito debe diseñarse mediante un circuito formado por dos bloques $B1$ y $B2$:

B1. lee los bits de entrada de N y genera una señal intermedia X de 8 bits ($X_7 X_6 \dots X_1 X_0$). Cada bit X_i indica si C_i debe ser igual a N_i ($X_i=0$), o si debe ser su complemento ($X_i=1$).

B2. En función de cada N_i y X_i generan los bits de salida C_i .

a) Encontrar las ecuaciones que permiten definir las salidas de $B1$ y $B2$, y diseñar el circuito utilizando puertas lógicas de dos entradas.