

7. Convertidores

Los convertidores son circuitos MSI que se utilizan para pasar de un código a otro, es decir, en las entradas se introduce un número descrito mediante una combinación binaria correspondiente a un código y en las salidas se obtiene ese número representado en otro código.

7.1. Diseño

Conversión Binario \Rightarrow Gray

La tabla de verdad, las funciones lógicas y el circuito correspondiente a este tipo de convertidores se representa en la Figura 110.

Binario				Gray			
B ₃	B ₂	B ₁	B ₀	G ₃	G ₂	G ₁	G ₀
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	0
0	1	0	1	0	1	1	1
0	1	1	0	0	1	0	1
0	1	1	1	0	1	0	0
1	0	0	0	1	1	0	0
1	0	0	1	1	1	0	1
1	0	1	0	1	1	1	1
1	0	1	1	1	1	1	0
1	1	0	0	1	0	1	0
1	1	0	1	1	0	1	1
1	1	1	0	1	0	0	1
1	1	1	1	1	0	0	0

$$G_0 = B_1 \oplus B_0$$

$$G_1 = B_1 \oplus B_2$$

$$G_2 = B_2 \oplus B_3$$

$$G_3 = B_3$$

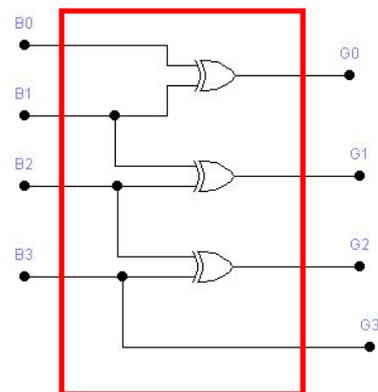
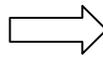


Figura 110

Conversión Gray \Rightarrow Binario

La tabla de verdad, las funciones lógicas y el circuito correspondiente a este tipo de convertidores se representa en la Figura 111.

Gray				Binario			
G_3	G_2	G_1	G_0	B_3	B_2	B_1	B_0
0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	1
0	0	1	0	0	0	1	1
0	0	1	1	0	0	1	0
0	1	0	0	0	1	1	1
0	1	0	1	0	1	1	0
0	1	1	0	0	1	0	0
0	1	1	1	0	1	0	1
1	0	0	0	1	1	1	1
1	0	0	1	1	1	1	0
1	0	1	0	1	1	0	0
1	0	1	1	1	1	0	1
1	1	0	0	1	0	0	0
1	1	0	1	1	0	0	1
1	1	1	0	1	0	1	1
1	1	1	1	1	0	1	0

$$B_0 = B_1 \oplus G_0$$

$$B_1 = G_1 \oplus B_2$$

$$B_2 = G_2 \oplus G_3$$

$$B_3 = G_3$$

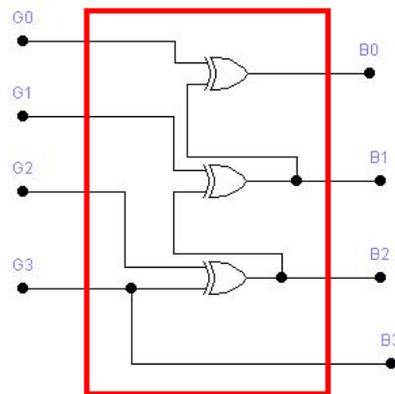


Figura 111

Conversión Binario \Rightarrow BCD

La tabla de verdad correspondiente a este tipo de convertidores se representa en la Tabla 30.

B ₃	B ₂	B ₁	B ₀	D ₀	U ₃	U ₂	U ₁	U ₀	
0	0	0	0	0	0	0	0	0	0
0	0	0	1	0	0	0	0	1	1
0	0	1	0	0	0	0	1	0	2
0	0	1	1	0	0	0	1	1	3
0	1	0	0	0	0	1	0	0	4
0	1	0	1	0	0	1	0	1	5
0	1	1	0	0	0	1	1	0	6
0	1	1	1	0	0	1	1	1	7
1	0	0	0	0	1	0	0	0	8
1	0	0	1	0	1	0	0	1	9
1	0	1	0	1	0	0	0	0	16
1	0	1	1	1	0	0	0	1	17
1	1	0	0	1	0	0	1	0	18
1	1	0	1	1	0	0	1	1	19
1	1	1	0	1	0	1	0	0	20
1	1	1	1	1	0	1	0	1	21

Tabla 30

Observando la Tabla 30, se concluye que para las entradas de 0 a 9, la combinación de salida es igual que la de entrada, pero a partir del valor de entrada 10, la combinación de salida difiere de la de entrada. Si se hace una lectura en binario natural de los valores de entrada y salida, existe una diferencia de +6 entre la entrada y salida. Resumiendo:

$$\text{Si } B \in [0-9] \Rightarrow DU = B + 0$$

$$\text{Si } B \in [10-18] \Rightarrow DU = B + 6$$

La operación a realizar es una suma con un operando variable cuyo valor será 0 o 6. Analizando con más detalle esta suma, se observa que en el operando variable, únicamente conmutan dos bits cuyo valor es 1 cuando $B \in [10-18]$.

$$U_3U_2U_1U_0 = B_3B_2B_1B_0 + 0000$$

$$U_3U_2U_1U_0 = B_3B_2B_1B_0 + 0110$$

Así pues, el bloque de diagramas correspondiente a este diseño se muestra en la Figura 112, donde la salida P indicará si $B \in [10-18]$ y el valor del bit variable.

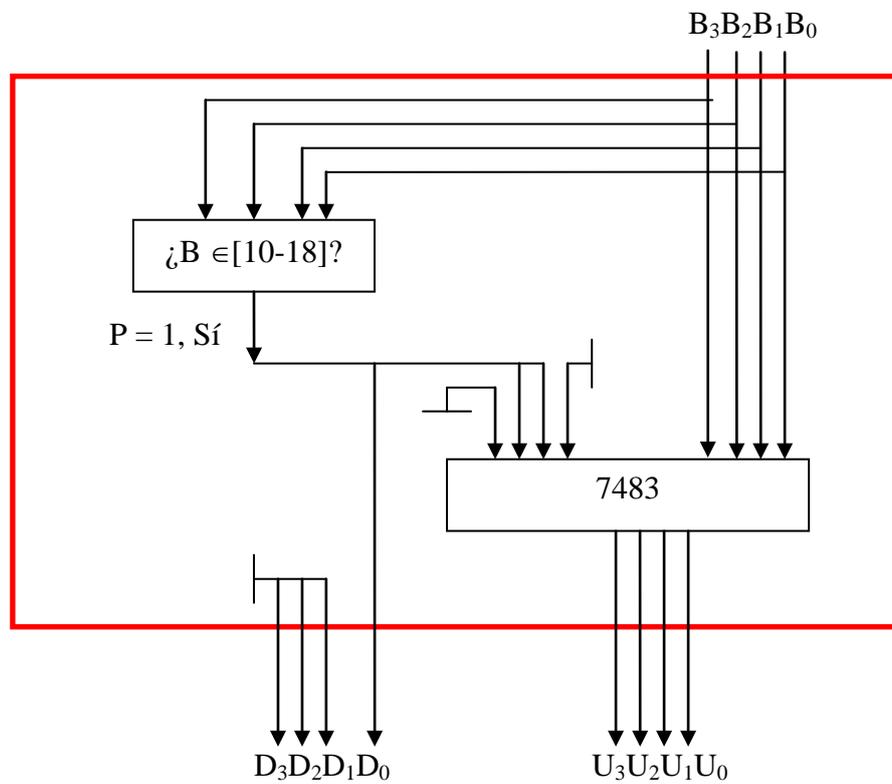


Figura 112

Conversión BCD \Rightarrow Binario

El algoritmo necesario para la implementación de este tipo de circuitos se describe a continuación.

$$BA_{\text{dec}} = B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0_{\text{BCDnat}}$$

Donde BA_{dec} representa a un número decimal de dos bits, y $B_3 B_2 B_1 B_0 A_3 A_2 A_1 A_0_{\text{BCDnat}}$ la representación de el número anterior codificado en BCD natural. Así pues,

$$\begin{aligned} BA_{\text{dec}} &= B \cdot 10 + A = B \cdot (2^3 + 2^1) + A = B \cdot 2^3 + B \cdot 2^1 + A = \\ &= B_3 B_2 B_1 B_0 0 0 0 + B_3 B_2 B_1 B_0 0 + A_3 A_2 A_1 A_0 \end{aligned}$$

Entonces, el circuito únicamente deberá implementar una suma, que puede representarse igualmente según la siguiente expresión:

$$\begin{array}{r} B_3 \ B_2 \ B_1 \ B_0 \ 0 \ 0 \ 0 \\ \ 0 \ 0 \ 0 \ 0 \ 0 \\ \ A_3 \ A_2 \ A_1 \ A_0 \\ \hline R_7 \ R_6 \ R_5 \ R_4 \ R_3 \ R_2 \ R_1 \end{array}$$

Así pues, para su implementación serán necesarios dos sumadores. Si se utilizan dos 7485 las operaciones a implementar serán las siguientes

$$\begin{aligned} R_1 &= A_0 \\ R_2 &= A_1 + B_0 \Rightarrow C_{11} \\ R_3 &= A_2 + B_1 + C_{11} \Rightarrow C_{21} \\ R_4 &= A_3 + B_0 + B_2 + C_{21} \Rightarrow C_{31}, C_{12} \\ R_5 &= B_1 + C_{31} + B_3 + C_{12} \Rightarrow C_{41}, C_{22} \\ R_6 &= B_2 + C_{41} \Rightarrow C_{32} \\ R_7 &= B_3 + C_{32} \end{aligned}$$

En ellas se han tenido en cuenta los acarreo internos representados por los C_{ij} , donde el subíndice j representa a cada uno de los sumadores ($j = 1, 2$) y la i , la posición del acarreo interno ($i = 1, 2, 3, 4$). El circuito que implementa dichas operaciones se puede estudiar en la Figura 113.

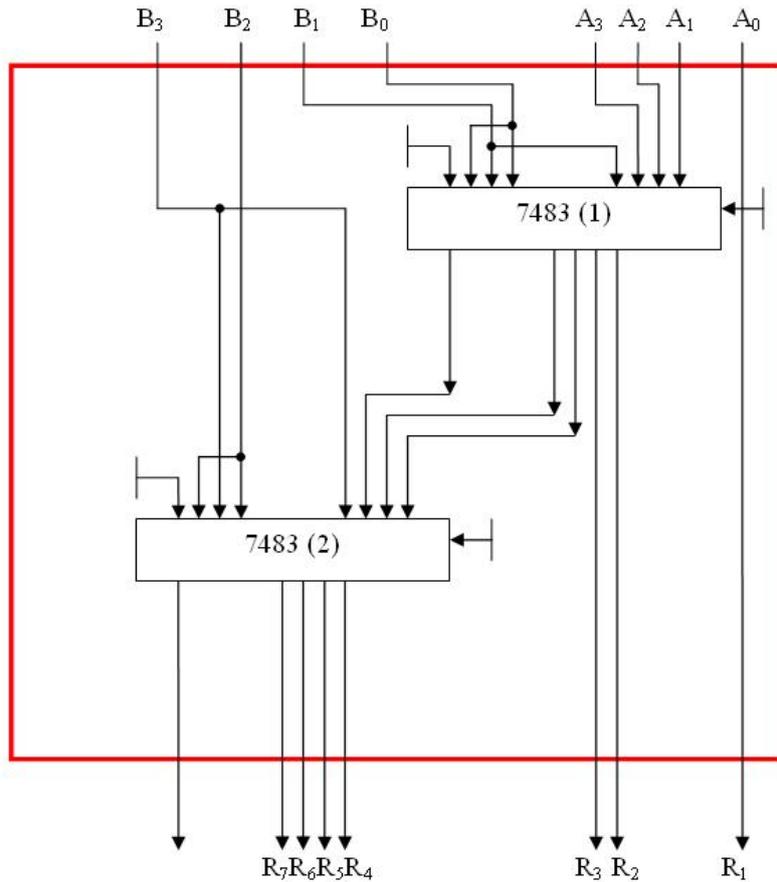


Figura 113

7.2. Circuitos Comerciales: 74185 y 74184

Los convertidores de código se suelen implementar en dispositivos programables. A pesar de ello, también existen circuitos MSI como son el 74185 y el 74184 como el de la Figura 114.

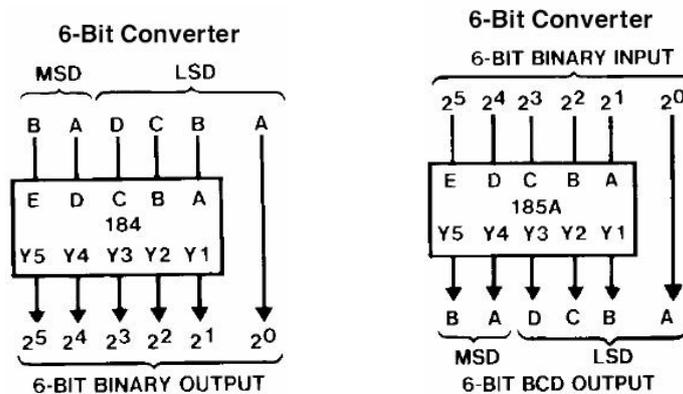


Figura 114

Para convertir números con mayor cantidad de bits, pueden combinarse estos dispositivos tal y como se muestra en la Figura 115 y en la Figura 116. En el primer ejemplo, para convertir un número BCD de dos dígitos se han utilizado dos circuitos integrados 74184; en el segundo caso, para convertir un número binario de 8 bits han sido necesarios tres circuitos integrados 74185.

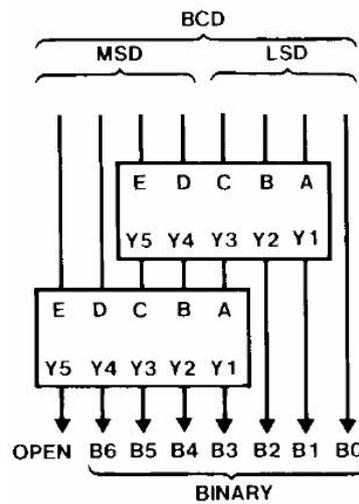


Figura 115.

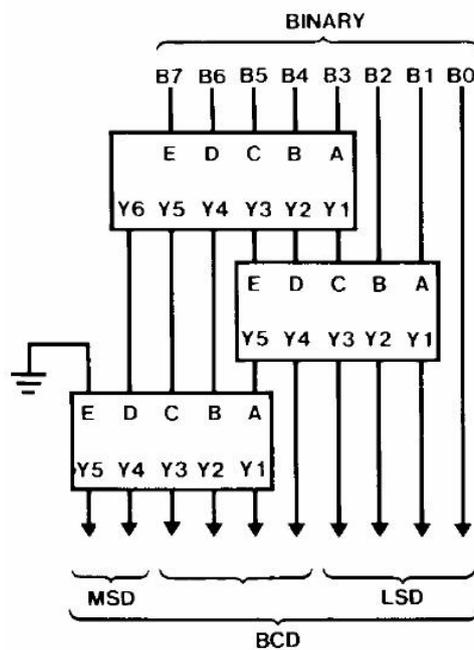


Figura 116.