

CÓDIGO NUMÉRICO GRAY

Este sistema de codificación no corresponde con una representación matemática real de los números. Se trata de una combinación de "1s" y "0s" que son las posibles combinaciones que puede adoptar una variable.

A continuación una tabla con la confección, en código Gray, de los números del 0 al 31 usando hasta 5 bits:

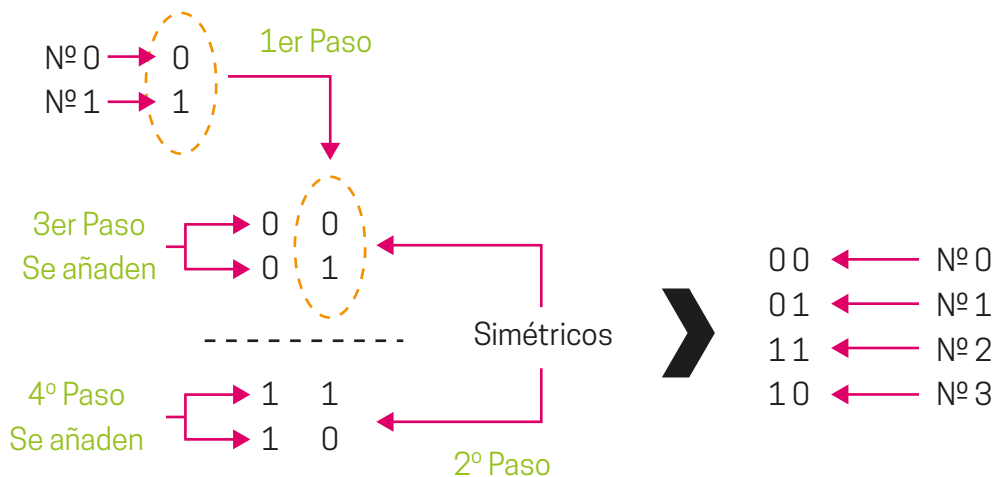
CÓDIGO GRAY											
Nº Decimal	1	2		3			4		5		
	Bit	Bit		Bit			Bit		Bit		
	G1	G2	G1	G3	G1	G4	G1	G5	G1		
0	0	0	0	0	0	0	0	0	0	0	
1	1	0	1	0	0	1	0	0	0	1	
2		1	1	0	1	1	0	0	1	1	
3		3	1	0	1	0	0	0	1	0	
4				1	1	0	0	1	1	0	
5				1	1	1	0	1	1	1	
6				1	0	1	0	1	0	1	
7			7	1	0	0	0	1	0	0	
8							1	1	0	0	
9							1	1	0	1	
10							1	1	1	1	
11							1	1	1	0	
12							1	0	1	0	
13							1	0	1	1	
14							1	0	0	1	
15						15	1	0	0	0	
16								1	1	0	
17								1	1	0	
18								1	1	0	
19								1	1	0	
20								1	1	1	
21								1	1	1	
22								1	1	1	
23								1	1	1	
24								1	0	1	
25								1	0	1	
26								1	0	1	
27								1	0	1	
28								1	0	0	
29								1	0	0	
30								1	0	0	
31								31	1	0	

Partiendo de un número de bits determinado, se realizan las combinaciones de acuerdo a las normas de Gray. La principal característica de este código es que dos números consecutivos, cualquiera que sean, se diferencian solo en un bit (tienen distancia "1"). Además, es un código cíclico. Es decir, el primer número y el último también son adyacentes, es decir, solo se diferencian en un bit.

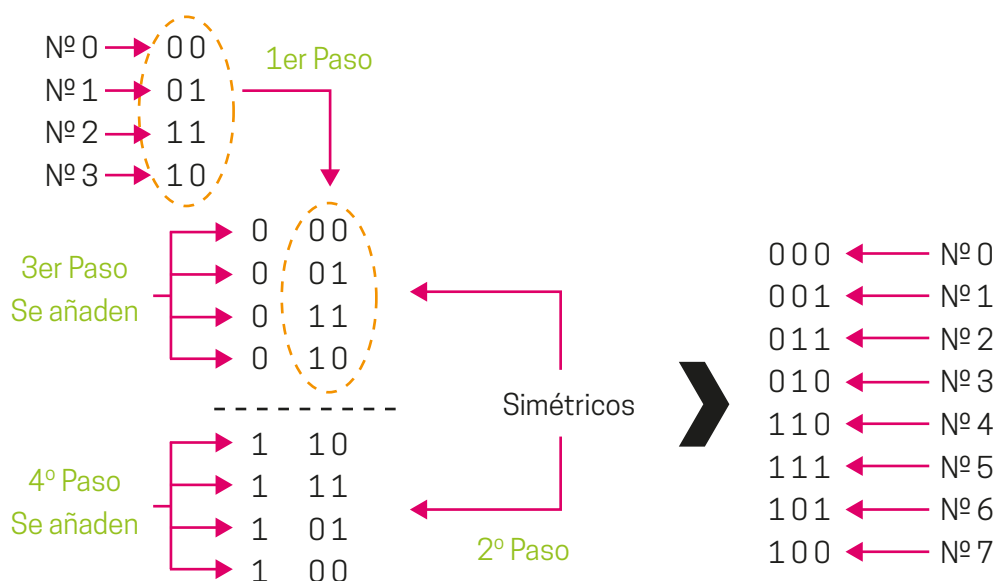
Como se puede apreciar en la tabla anterior, el código Gray es simétrico con respecto a la línea discontinua, en todos sus bits, excepto en el de la izquierda (Gn). En este último bit los de arriba de la línea discontinua son "0" y los de abajo son "1".

Vamos a ver como se ha ido formando la tabla:

Como con un bit solo se tiene dos posibilidades (vale "0" ó vale "1") podemos codificar el Nº 0 y el Nº 1. Para poder codificar del Nº 0 al Nº 3 necesitamos dos bits. Según la regla de Gray se procede de la siguiente manera:



Para poder codificar desde el Nº 0 al Nº 7 se necesitan 3 bits y se procede igual que en el paso anterior:



De esta forma se procede sucesivamente añadiendo cada vez un bit más.

Como el Código Gray no es una expresión matemática de los números, siempre hay que convertir el valor Gray en Binario para poder realizar las operaciones matemáticas que sean necesarias para la aplicación donde se esté trabajando.

Para realizar la conversión hay que aplicar la formula siguiente para calcular el valor de cada bit:

$$\text{Bit}(n) = \text{Bit}(n+1) \text{ XOR } G(n)$$

En donde:

Bit(n) es el bit enésimo en Binario

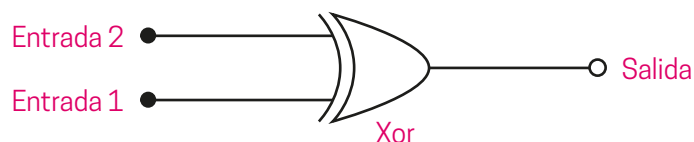
Bit(n+1) es el bit siguiente

G(n) es el bit enésimo en Gray

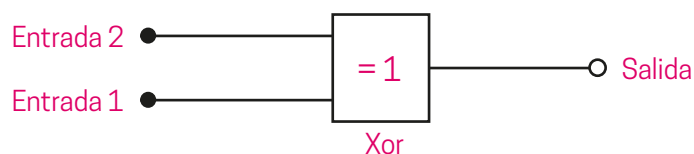
XOR es la operación lógica que se realiza entre los bits

NOTA: La operación XOR es una puerta lógica de dos entradas que suministra un "1" a la salida, solo, cuando las dos entradas son distintas. Si las dos entradas son iguales (tanto "1" como "0") se tiene a la salida un "0".

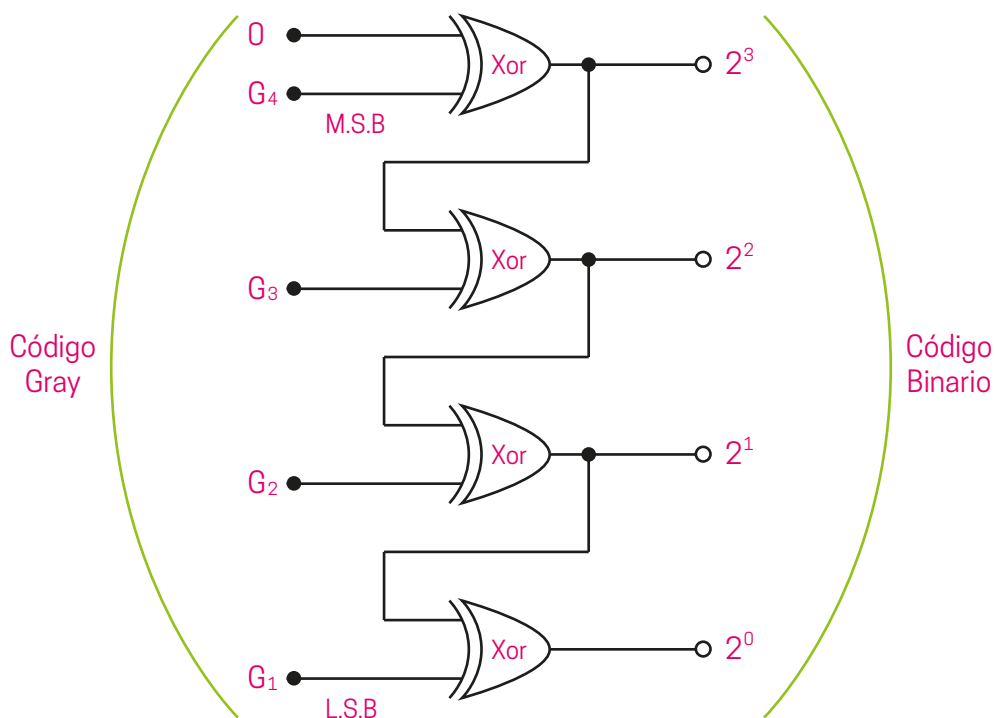
El esquema es el siguiente:



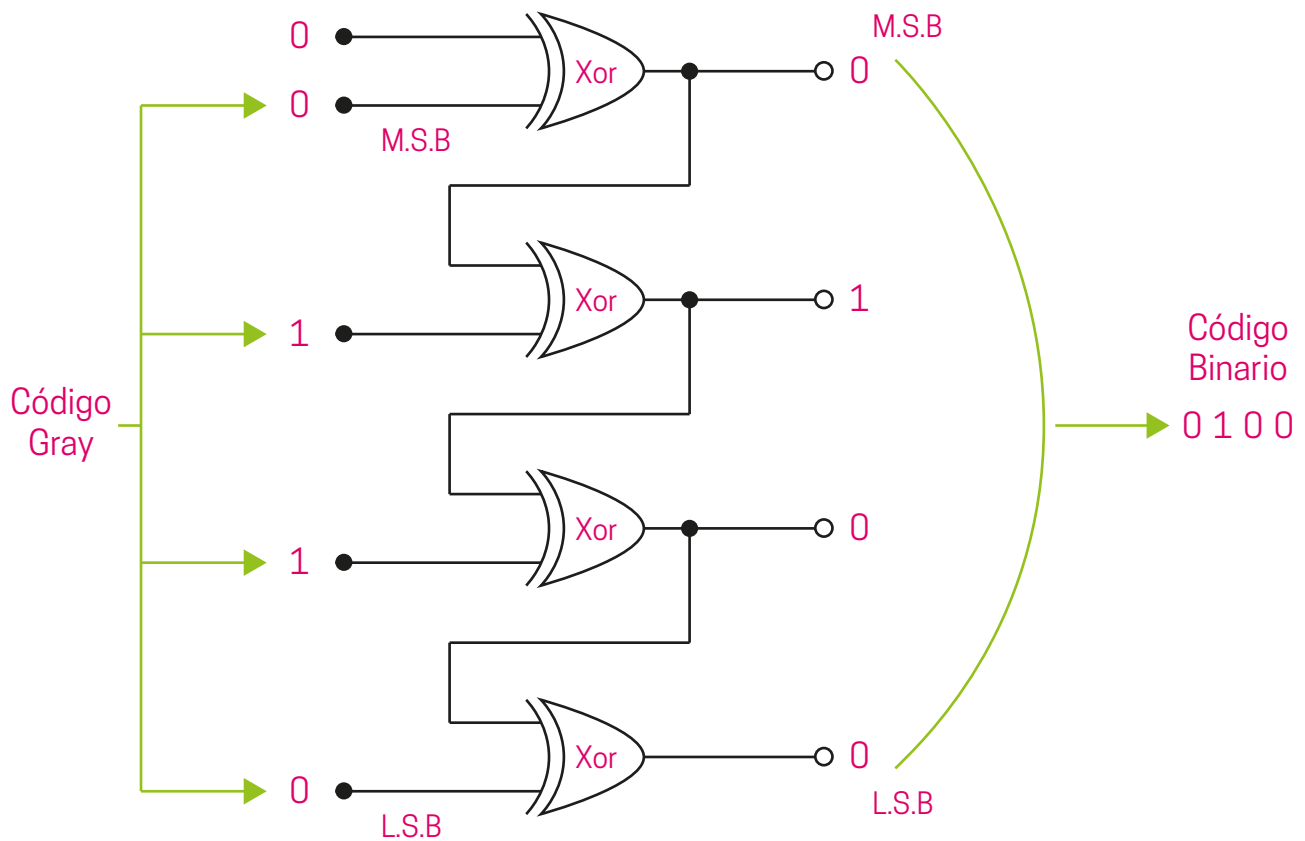
También se puede encontrar el siguiente esquema:



A continuación vamos a aplicar esta función para convertir un número en Código Gray a Binario Puro:



Veamos este mismo esquema con un ejemplo. Vamos a convertir la expresión del N° 4 en código Gray (0110) a código Binario (0100):



NOTA: La entrada N°2 de la puerta Xor del Bit más significativo (M.S.B.) debe estar fija a nivel lógico "0".