Código Decimal Binario (BCD)

El código decimal binario (BCD) es una forma de expresar cada uno de los dígitos decimales con un código binario. Puesto que en el sistema BCD sólo existen diez grupos de código, es muy fácil convertir entre decimal y BCD. Como nosotros leemos y escribimos en decimal, el código BCD proporciona una excelente interfaz para los sistemas binarios.

EJEMPLO

Convertir a BCD los siguientes números decimales: **(a)** 35 **(b)** 98 **(c)** 170 **(d)** 2469

Solución

(a) 3 5 → 0011 0101

(b) 9 8 → 1001 1000

(c) 1 7 0 → 0001 0111 0000

(d) 2 4 6 9 → 0010 0100 0110 1001

Código Gray:

El código Gray es un código sin pesos y no aritmético, es decir, no existen pesos específicos asignados a las posiciones de los bits. La característica más importante del código Gray es que **sólo varía un bit de un código al siguiente**.
Esta propiedad es importante en aplicaciones como codificadores de eje de posición, en donde el error aumenta con el cambio de bit entre números adyacentes dentro de una secuencia.

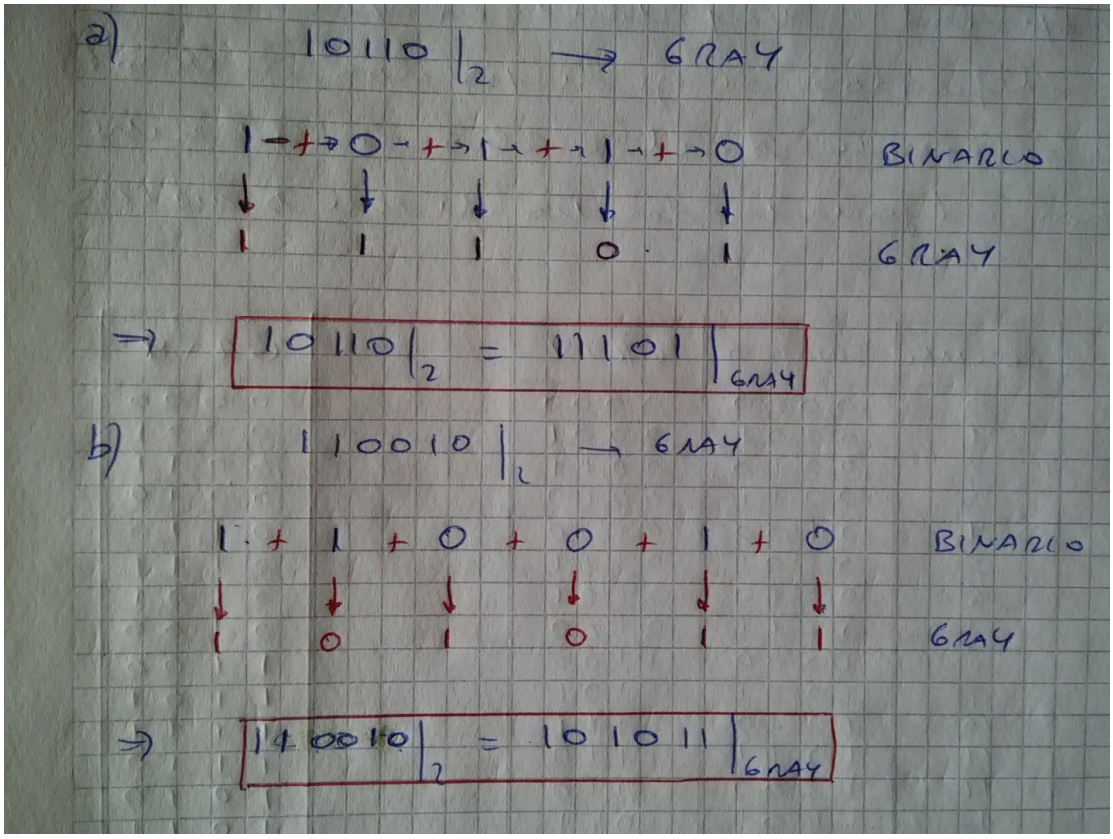
Tabla

Decimal	Binario	Gray
0	0000	0000
1	0001	0001
2	0010	0011
3	0011	0010
4	0100	0110
5	0101	0111
6	0110	0101
7	0111	0100
8	1000	1100
9	1001	1101
10	1010	1111
11	1011	1110
12	1100	1010
13	1101	1011
14	1110	1001
15	1111	1000

Gray
0000
0001
0011
0010
0110
0111
0101
0100
1100
1101
1111
1110
1010
1011
1001
1000

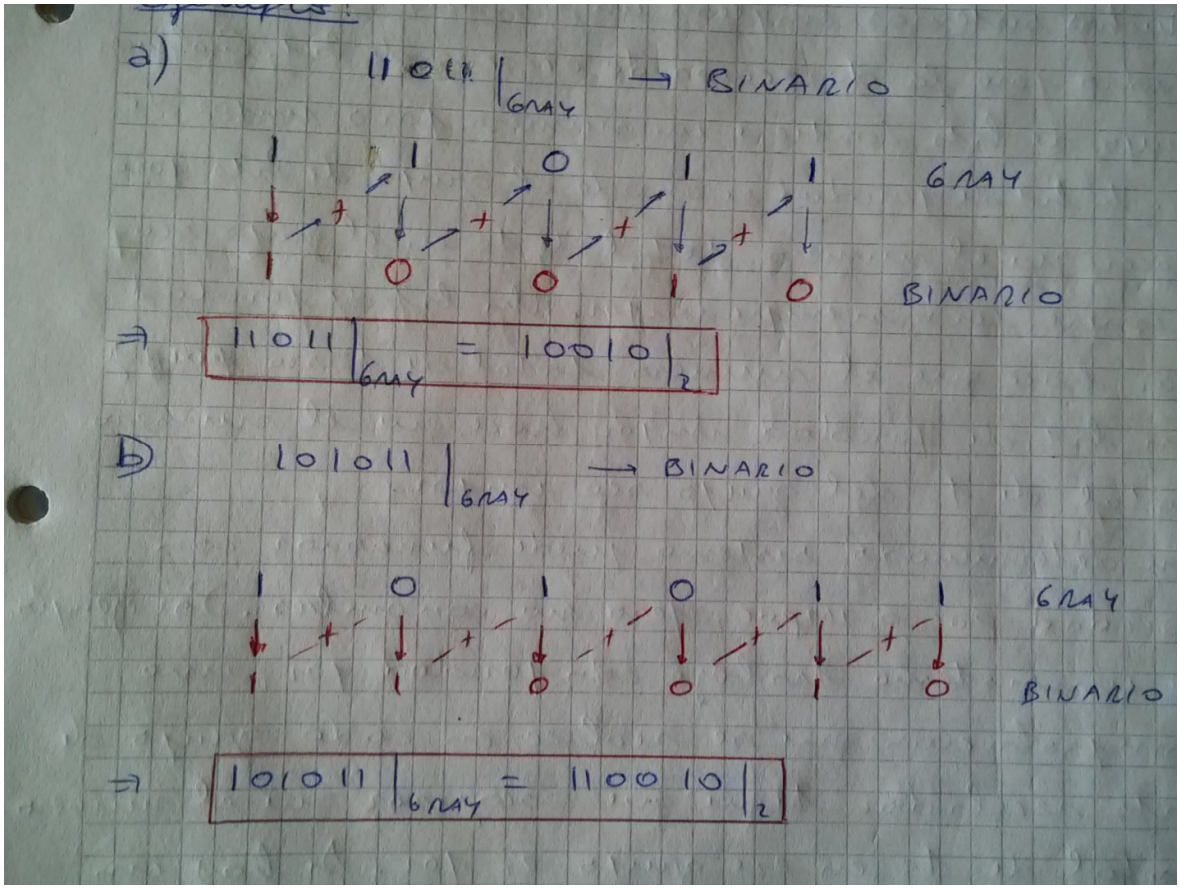
Conversión de código Binario a código Gray

Para realizar esta conversión debemos seguir las siguientes reglas:
1) El bit más significativo (el que está más a la izquierda, MSB) en el código Gray es el mismo que el correspondiente MSB del número binario.
2) Yendo de izquierda a derecha, sumar cada par adyacente de los bits en código binario para obtener el siguiente bit en código Gray. Los acarreos deben descartarse.



Conversión de código Gray a código Binario

En este caso el método es similar, pero con algunas diferencias en la manera de realizar las sumas. Veamos cómo son las reglas:
1) El bit más significativo (bit más a la izquierda) en el código binario es el mismo que el correspondiente bit en código Gray.
2) A cada bit del código binario generado se le suma el bit en código Gray de la siguiente posición adyacente. Los acarreos se descartan.



Detección de Errores y Códigos de Corrección

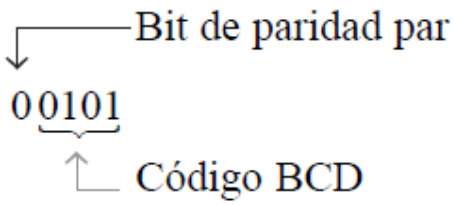
Bits de Paridad:

Paridad par		Paridad impar	
P	BCD	*P*	BCD
0	0000	1	0000
1	0001	0	0001
1	0010	0	0010
0	0011	1	0011
1	0100	0	0100
0	0101	1	0101
0	0110	1	0110
1	0111	0	0111
1	1000	0	1000
0	1001	1	1001

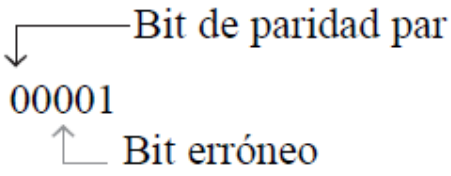
TABLA 2.10 El código BCD con bits de paridad.

Detección de un error:

Transmitimos una palabra (Datos) en este caso de 4 bits y le asignamos un bit de paridad (por ejemplo PAR).



Luego vamos al receptor y vemos que recibimos el siguiente código:



Sabiendo que la paridad de transmisión era PAR, cuando hacemos el chequeo podemos asegurar que ha ocurrido un error en algún bit. Lamentablemente con un solo bit de paridad no podemos asegurar “donde” se produjo el error y de esta manera corregirlo.

Para poder hacer esto necesitamos un código de detección y corrección de errores.

El código Hamming de corrección de errores

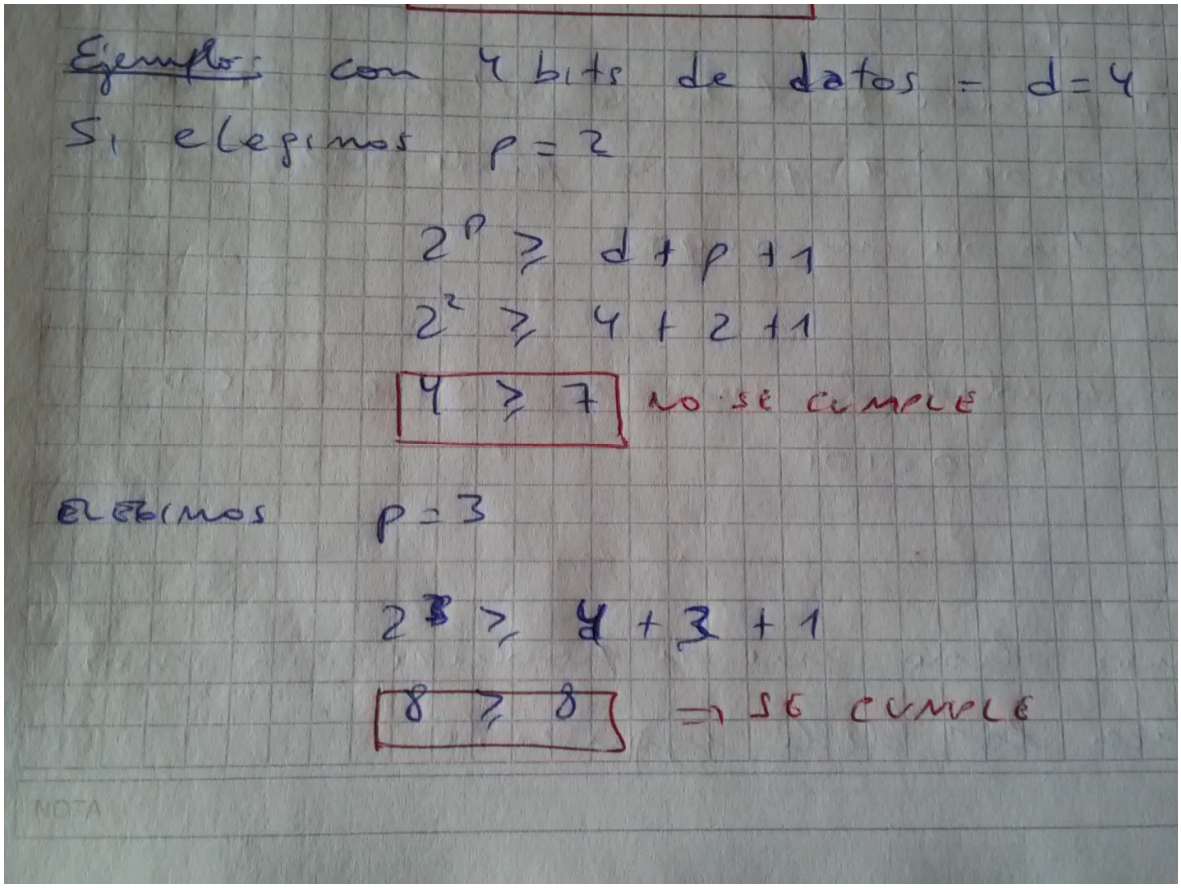
Este método nos permite corregir un único error detectado, identificando la posición del bit erróneo antes de poder corregirlo. Tanto sea que el error haya ocurrido en los bits de datos, como en los bits de paridad.

Para poder empezar determinar si es posible aplicar el método debemos cumplir con la siguiente expresión:

Números de bits de Paridad:

Si el número de bits de datos se designa con **"d"**, entonces el número de bits de paridad "p", se calcula así:

$$2^p \geq d + p + 1$$

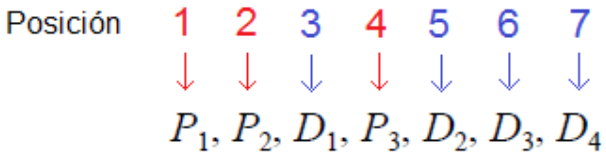


Colocación de bits de paridad en el código:

Una vez sabida la cantidad de bits de paridad, debemos colocarlos en la posición adecuada dentro del código:
Si consideramos que el orden de bits de datos es:

Bit 1, Bit 2, Bit 3, Bit 4, Bit 5, Bit 6, Bit 7.

Los bits de paridad se sitúan haciéndolos corresponder con las potencias en dos en sentido ascendente (1, 2, 4, 8, …).



El símbolo P_n designa un determinado bit de paridad y D_n designa cada uno de los bits de datos.

Asignación de los valores de los bits de paridad:

Designación de bit	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Posición de bit	1	2	3	4	5	6	7
Número de posición en binario	001	010	011	100	101	110	111
Bits de datos (D_n)							
Bits de paridad (P_n)							

Para determinar los bits de paridad se debe aplicar lo siguiente:

a) El bit P1 tiene un 1 a la derecha y comprueba los bits 1, 3, 5 y 7.
b) El bit P2 tiene un 1 en el medio por cual comprueba los bits 2, 3, 6 y 7.
c) El bit P3 tiene un 1 en la izquierda por lo tanto comprueba los bits 4, 5, 6 y 7.

En cada caso se asigna un valor de bit de paridad de modo que la cantidad de 1s en el conjunto que desea comprobar sea par o impar.

Cómo detectar y corregir un error con el código de Hamming

Paso 1. Comience con el grupo comprobado por P_1.

Paso 2. Compruebe si el grupo tiene la paridad correcta. Un 0 representa que la comprobación de paridad es correcta y un 1 que es incorrecta.

Paso3. Repita el paso 2 para cada grupo de paridad.

Paso 4. El número binario formado por los resultados de todas las comprobaciones de paridad indica la posición del bit del código que es erróneo. Es el *código de posición de error*. La primera comprobación de paridad genera el bit menos significativo (LSB). Si todas las comprobaciones son correctas, no habrá error.

Designación de bit	P_1	P_2	D_1	P_3	D_2	D_3	D_4
Posición de bit	1	2	3	4	5	6	7
Número de posición en binario	001	010	011	100	101	110	111
Código recibido							

- ***Primera comprobación de Paridad***

  El bit P1 comprueba 1, 3, 5 y 7
  Verificar si la suma da como resultado la paridad informada (Par o Impar).

- ***Segunda comprobación de Paridad***

  El bit P2 comprueba 2, 3, 6 y 7
  Verificar si la suma da como resultado la paridad informada (Par o Impar).

- ***Tercera comprobación de Paridad***

  El bit P3 comprueba 4, 5, 6 y 7
  Verificar si la suma da como resultado la paridad informada (Par o Impar).

Ejercicios:

1) a) Transmitir datos 1001 con paridad PAR.
   b) Se recibió 0010001, determinar si hubo error y corregirlo.

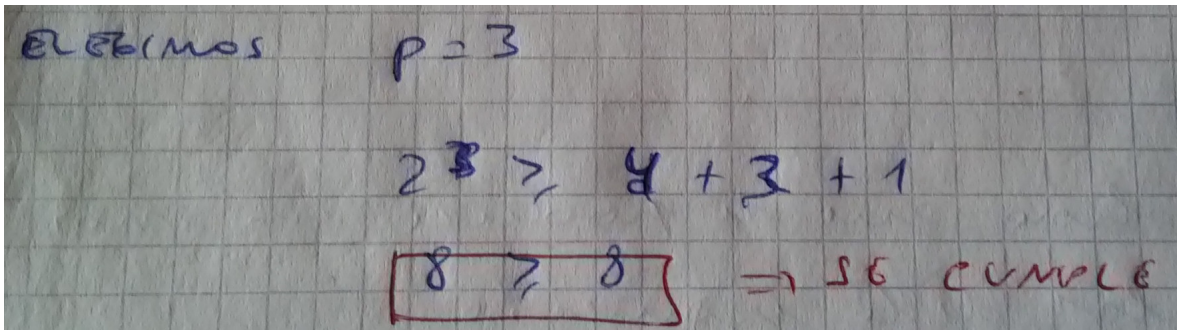
2) Se desea trasmitir una palabra correspondiente al número **1010** **(bits de datos)** utilizando paridad **PAR**. **(3 Pts.)**

   a) Implementar mediante el Código de Hamming cuál será la palabra final transmitida.

   b) Si en el receptor se obtuvo la siguiente palabra **1011110**, indicar si existe un error, en qué bit se produjo y cómo se corrige.

1) **Resolución:**

a) Primero aplicamos la fórmula para determinar la cantidad de bits de paridad que hacen falta



DATOS: **D1 = 1, D2 = 0, D3 = 0, D4 = 1** **es decir 1001**

Designación de Bit	P1	P2	D1	P3	D2	D3	D4
Posición de bit	1	2	3	4	5	6	7
Núm, posición binario	001	010	011	100	101	110	111
Bits de Datos			**1**		**0**	**0**	**1**
Bits de Paridad	**0**	**0**		**1**			

Finalmente la palabra completa a transmitir es: **0011001**

b) Se recibió la palabra 0010001 y sabemos que se utilizó paridad Par.

Designación de Bit	P1	P2	D1	P3	D2	D3	D4
Posición de bit	1	2	3	4	5	6	7
Núm, posición binario	001	010	011	100	101	110	111
Código recibido	**0**	**0**	**1**	**0**	**0**	**0**	**1**

Comprobamos:

Bit P1: verificamos 1, 3, 5 y 7.
En este grupo hay dos 1s. La comprobación de paridad es correcta -> **"0"**

Bit P2: verificamos 2, 3, 6 y 7.
En este grupo hay dos 1s. La comprobación de paridad es correcta -> **"0"**

Bit P3: verificamos 4, 5, 6 y 7.
En este grupo hay un sólo 1. La comprobación de paridad es incorrecta -> **"1"**

Si comparo los tres casos descartamos error en los bits 1, 2, 3, 5, 6 y 7, quedando solamente el **bit 4, que es donde se produjo el error.**
En lugar de un 1 debería haber un 0.
Es decir se había recibido el número **0010001** y la correcta es **0011001**

2) a)

DATOS: D1 = 1, D2 = 0 , D3 = 1 , D4 = 0 es decir 1010 Paridad PAR

Designación de Bit	P1	P2	D1	P3	D2	D3	D4
Posición de bit	1	2	3	4	5	6	7
Núm, posición binario	001	010	011	100	101	110	111
Bits de Datos			**1**		**0**	**1**	**0**
Bits de Paridad	**1**	**0**		**1**			

Paridad:
P1 = 1
P2 = 0
P3 = 1

Respuesta: La palabra a transmitir es: 1011010

b) Palabra recibida: **1011110.**

Designación de Bit	P1	P2	D1	P3	D2	D3	D4
Posición de bit	1	2	3	4	5	6	7
Núm, posición binario	001	010	011	100	101	110	111
Código recibido	**1**	**0**	**1**	**1**	**1**	**1**	**0**

Comprobamos:

Bit P1: verificamos 1, 3, 5 y 7.

En este grupo hay dos 1s. La comprobación de paridad es incorrecta -> **“1”**

Bit P2: **verificamos 2, 3, 6 y 7.**
En este grupo hay dos 1s. La comprobación de paridad es correcta -> **“0”**

Bit P3: **verificamos 4, 5, 6 y 7.**
En este grupo hay un sólo 1. La comprobación de paridad es incorrecta -> **“1”**

Se recibió la palabra **1011110** y como el bit 5 esta mal, se debe cambiar por “0”.
La respuesta es: se debió recibir la palabra 1011010.

Bits de datos recibidos; 1110
Bits de datos correctos: 1010