Unidad Didáctica

Códigos Binarios

FONDO FORMACION

Programa de Formación Abierta y Flexible

Obra colectiva de FONDO FORMACION

Coordinación *Servicio de Producción Didáctica de FONDO FORMACION (Dirección de Recursos)*

Diseño y maquetación *Servicio de Publicaciones de FONDO FORMACION*

Depósito Legal ***AS -1953-2001***

Unidad Didáctica Códigos Binarios

Un sistema digital se relaciona con el mundo exterior mediante numeración decimal y dentro del sistema se debe realizar la conversión al código empleado, después la información ha de convertirse de nuevo a decimal.

En la electrónica digital las magnitudes se representan por medio de símbolos o dígitos; para ello nos basamos en un sistema de numeración. Los sistemas de numeración son códigos que nos sirven para representar magnitudes. La representación binaria es la que emplean los circuitos digitales.

En esta unidad vas a ver diferentes formas de codificar la información en el sistema binario.

En esta unidad veremos los siguientes contenidos:

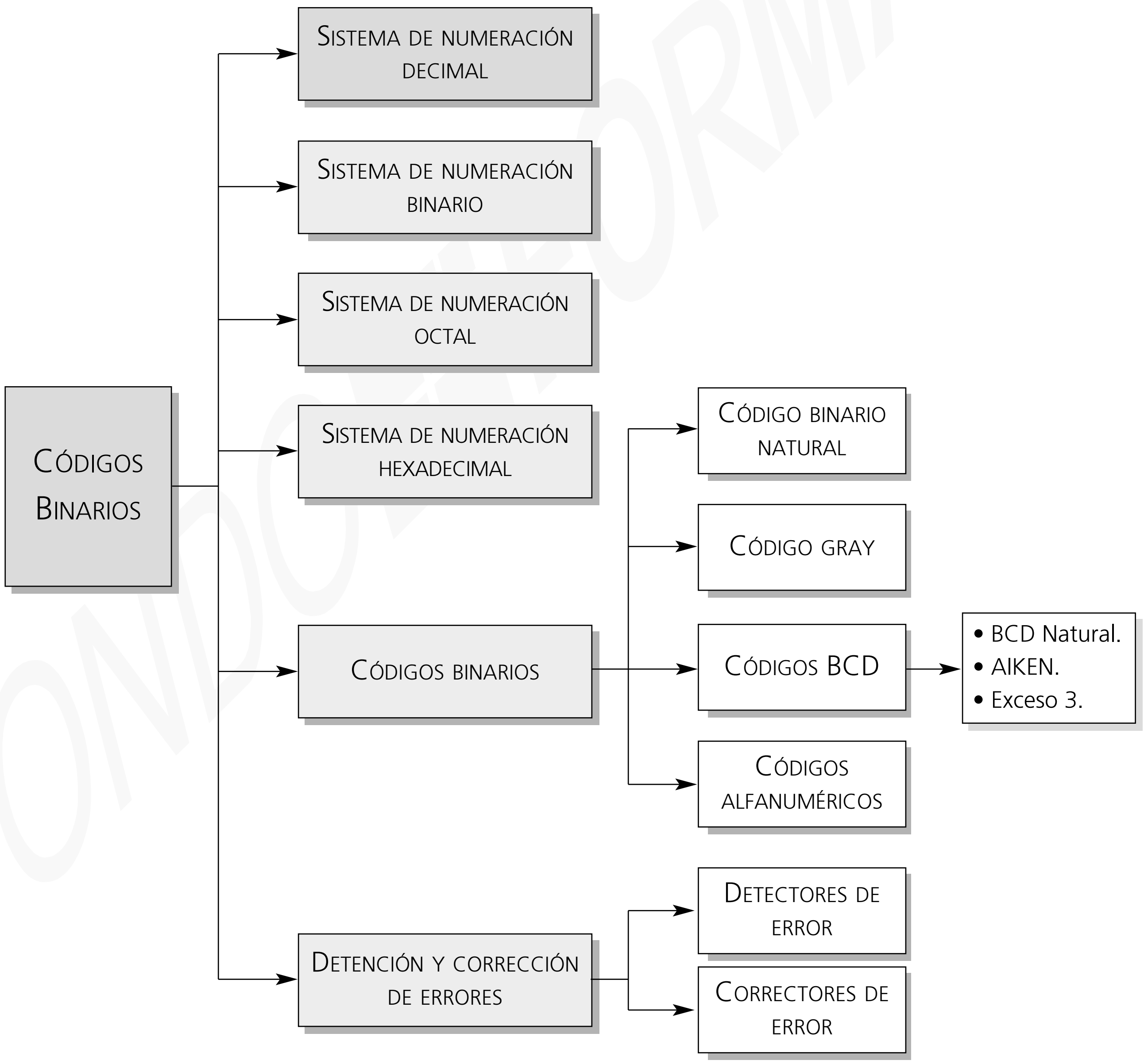
- Sistema de numeración decimal.
- Sistema de numeración binario.
- Sistema de numeración octal.
- Sistema de numeración hexadecimal.
- Códigos binarios.
- Detección y corrección de errores.

Tus objetivos

Al final de esta unidad serás capaz de:

- Expresar un número en diferentes sistemas de numeración.
- Pasar un número de un sistema de numeración a otro.
- Diferenciar el código binario natural del código BCD natural.
- Codificar un número decimal en diversos códigos BCD.

Esquema de estudio



Sitema de numeración decimal

Para representar las magnitudes nos basamos en un sistema de numeración, cuya base elegimos de tal modo que los dígitos o símbolos empleados nos permiten representar esas magnitudes que queremos medir. La representación de números en sistema-decimal es la habitual en nuestra vida cotidiana.

La base de un sistema de numeración es el número de dígitos distintos que tiene; los dígitos en base diez van del 0 al 9 y con ellos podemos representar cualquier cantidad.

Si tenemos el número en base diez vemos que los dígitos 3, 5, y 7 forman parte del sistema decimal y el peso o valor de cada dígito depende de la posición que ocupa, como puedes comprobar a continuación:

Sea el número 357 en base diez:

$$375_{10} = 3 \cdot 10^2 + 5 \cdot 10^1 + 7 \cdot 10^0 = 300 + 50 + 7 = 357$$

Para **pasar un número de cualquier base, a base 10,** multiplicamos el valor de cada dígito por el peso que tenga, teniendo en cuenta la base en la que estamos.

Así, para pasar el número 357 en base ocho a su equivalente en base diez, hacemos lo siguiente:

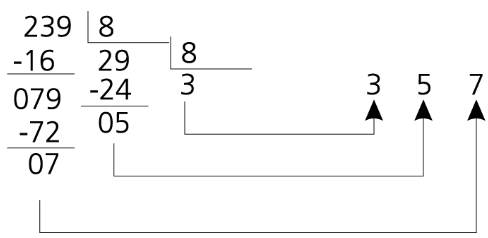
$$375_8 = 3 \cdot 8^2 + 5 \cdot 8^1 + 7 \cdot 8^0 = 192 + 40 + 7 = 239_{10}$$

El número 357 en base cuatro no existe, ya que los dígitos 5 y 7 no forman parte de esta base. El número 123 en base cuatro sí existe y vamos a pasarlo a base diez:

$$123_4 = 1 \cdot 4^2 + 2 \cdot 4^1 + 3 \cdot 4^0 = 16 + 8 + 3 = 27_{10}$$

Para **pasar un número de base 10 a cualquier base** vamos haciendo una serie de divisiones entre el número de la base a la que deseamos pasar.

Vamos a pasar el número 239 en base diez a base ocho:



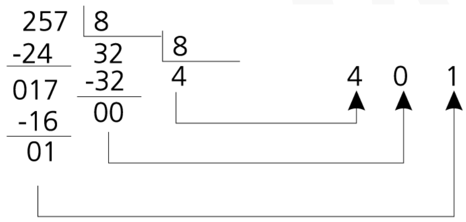
Por tanto, el número 239 en base diez es equivalente al número 357 en base ocho:

$$239_{10} = 357_{g}$$

Un número entero lo es en cualquier base, pero un número con un número finito de decimales en una base, puede tener un número infinito de decimales en otra base.

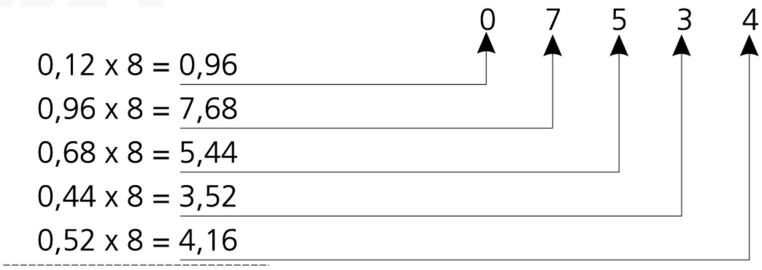
Vamos a pasar el número 257,12 en base diez a base ocho.

Primero, cogemos la parte entera y la pasamos a base ocho:



$$257_{10} \equiv 401_{8}$$

A continuación, pasamos la parte decimal. Para ello vamos haciendo multiplicaciones sucesivas y recogemos los dígitos de la parte entera, como puedes ver:



Por tanto, el número 257,12 en base diez es equivalente al número 401,07534 en base ocho:

$$257{,}12_{10} \equiv 401{,}07534_{8}$$

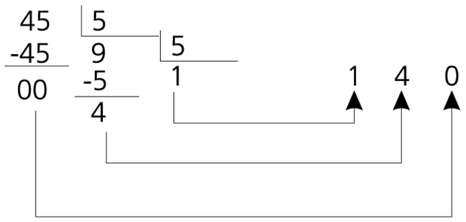
Para **pasar un número de cualquier base a cualquier base** lo que hacemos primero es pasarlo a base diez.

Vamos a pasar el número 231 en base cuatro a base cinco.

Primero, pasamos el número a base diez:

$$231_4 = 2 \cdot 4^2 + 3 \cdot 4^1 + 1 \cdot 4^0 = 32 + 12 + 1 = 45_{10}$$

Y ahora pasamos el resultado a base cinco:



Por tanto, el número 231 en base cuatro es equivalente al número 140 en base cinco:

$$231_4 = 140_5$$

ACTIVIDAD1

Pasa el número 11,3125 en base diez a su equivalente en base dos.

Sistema de numeración binario

Es el sistema más utilizado en la realización de circuitos digitales. En el sistema binario sólo existen dos símbolos: el 0 y el 1, y con ellos, se puede representar cualquier magnitud.

Un número en base dos podría ser el 1011001 en el que –como ves– sólo aparecen los dígitos 0 y 1.

Vamos a pasar el número 1011,1 en base dos a decimal:

$$1011{,}1_2 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-1} = 8 + 0 + 2 + 1 + 0{,}5 = 11{,}5_{10}$$

Vamos a pasar el número 39 en base diez a base dos:

$$\begin{aligned} 39 : 2 &= 1 \\ 19 : 2 &= 1 \\ 9 : 2 &= 1 \\ 4 : 2 &= 0 \\ 2 : 2 &= 0 \\ & 1 \end{aligned}$$

Como hemos visto, se divide sucesivamente el número por dos, y el cociente se vuelve a dividir por dos, y así sucesivamente. El último cociente y los restos obtenidos forman el número en base dos.

Por tanto, tenemos:

$$39_{10} = 100111_2$$

Sistema Octal	Código Binario		
0	0	0	0
1	0	0	1
2	0	1	0
3	0	1	1
4	1	0	0
5	1	0	1
6	1	1	0
7	1	1	1

Tabla 1: Sistema Octal.

Sistema de numeración octal

Está compuesto por ocho caracteres que van de 0 a 7, como ves en la tabla 1.

Para convertir un número binario a octal no tenemos más que agrupar los dígitos binarios de tres en tres. Vamos a pasar el número 101110010 en base dos a su equivalente en base ocho.

$$\underbrace{101}_{5}\underbrace{110}_{6}\underbrace{010}_{2}$$

$$101110010_2 = 562_8$$

Vamos a hacer el proceso inverso, convertir el número 562 octal a su equivalente en binario. Para ello, primero lo pasamos a base diez y luego a base dos:

$$562_8 = 5 \cdot 8^2 + 6 \cdot 8^1 + 2 \cdot 8^0 = 320 + 48 + 2 = 370_{10}$$

370 : 2	0
185 : 2	1
92 : 2	0
46 : 2	0
23 : 2	1
11: 2	1
5 : 2	1
2 : 2	0
1	

1 → 1 0 1 1 1 0 0 1 0

Sistema de numeración hexadecimal

Está compuesto por dieciséis caracteres, que son los que ves en la tabla 2. Fíjate que, a partir del dígito 9, pasa a A.

Sistema Hexadecimal	Código Binario	Sistema Hexadecimal	Código Binario
0	0 0 0 0	8	1 0 0 0
1	0 0 O 1	9	1 0 0 1
2	0 0 1 0	A	0 1 1 0
3	0 0 1 1	B	0 1 1 1
4	0 1 0 0	C	1 1 0 0
5	0 1 0 1	D	1 1 0 1
6	0 1 1 0	E	1 1 1 0
7	0 1 1 1	F	1 1 1 1

Tabla 2: Sistema hexadecimal.

A partir del dígito F, viene el número 10, que corresponde al número 10000 en binario. Como ves, para su representación en binario se precisan cinco dígitos.

Para pasar de binario a hexadecimal, agrupamos los dígitos binarios de cuatro en cuatro.

ACTIVIDAD 2

a. Pasar el número 1111100111101100 en base dos, a su equivalente en hexadecimal.

b. Pasar el número A5B en hexadecimal, a su equivalente en binario.

Códigos binarios

La posibilidad de realización de componentes electrónicos con dos estados claramente diferenciados que permiten obtener circuitos electrónicos fiables, ha hecho que el sistema más utilizado en la realización de circuitos digitales sea el binario.

En el sistema binario sólo existen dos símbolos el 0 y el 1, como ya sabes.

Código es una representación de las cantidades, de tal forma que, a cada una de éstas se le asigna una combinación de símbolos determinada. Con n bits se pueden representar 2^n números distintos.

Los códigos pueden ser **ponderados** o **no ponderados.** En los códigos ponderados se asigna un peso a cada posición y el número decimal equivalente se obtiene sumando los pesos de las posiciones que tienen un 1.

1. Código binario natural

El sistema binario descrito hasta ahora se denomina **código binario natural** y es un **código con peso.**

Cada columna tiene un peso de una **potencia de 2.**

Además del código binario natural de la tabla 3, existen otros códigos binarios de gran interés para los sistemas digitales.

Decimal	Código Binario
0	0
1	1
2	1 0
3	1 1
4	1 0 0
5	1 0 1
6	1 1 0
7	1 1 1
8	1 0 0 0
9	1 0 0 1

Decimal	Código Binario
10	1 0 1 0
11	1 0 1 1
12	1 1 0 0
13	1 1 0 1
14	1 1 1 0
15	1 1 1 1
16	1 0 0 0 0
17	1 0 0 0 1
18	1 0 0 1 0
19	1 0 0 1 1

Tabla 3: Código binario natural.

2. Código gray

La formación del código de *n* bits se realiza partiendo del de *n*-1 bits, repitiendo simétricamente las combinaciones de éste, y añadiendo por la izquierda un nuevo bit 0 para las 2^{n-1} combinaciones, y un 1 para las combinaciones siguientes; por esto se dice que es un **código reflejado.**

El código gray es reflejado, continuo*, y cíclico*.

Decimal	Código Gray de dos bits
0	0 0
1	0 1
2	1 1
3	1 0

Decimal	Código Gray de tres bits
0	0 0 0
1	0 0 1
2	0 1 1
3	0 1 0
4	1 1 0
5	1 1 1
6	1 0 1
7	1 0 0

Tabla 4: Códigos gray de dos y tres bits.

3. Códigos BCD

Son **códigos decimales codificados en binario;** es decir, representan los diez dígitos diferentes del sistema decimal haciendo corresponder a cada dígito decimal cuatro dígitos binarios.

	BCD natural 8 4 2 1	Aiken 2 4 2 1	5 4 2 1	Exceso 3
0	0 0 0 0	0 0 0 0	0 0 0 0	0 0 1 1
1	0 0 0 1	0 0 0 1	0 0 0 1	0 1 0 0
2	0 0 1 0	0 0 1 0	0 0 1 0	0 1 0 1
3	0 0 1 1	0 0 1 1	0 0 1 1	0 1 1 0
4	0 1 0 0	0 1 0 0	0 1 0 0	0 1 1 1
5	0 1 0 1	1 0 1 1	1 0 0 0	1 0 0 0
6	0 1 1 0	1 1 0 0	1 0 0 1	1 0 0 1
7	0 1 1 1	1 1 0 1	1 0 1 0	1 0 1 0
8	1 0 0 0	1 1 1 0	1 0 1 1	1 0 1 0
9	1 0 0 1	1 1 1 1	1 1 0 0	1 1 0 0

Tabla 5: Códigos BCD.

En la tabla 6, puedes ver algunos de los códigos BCD más utilizados. El BCD *Exceso 3* es un código sin peso, como puedes observar.

4. Códigos alfanuméricos

Sirven para representar letras, números y caracteres especiales. Así, tenemos el ASCII (Código americano para el intercambio de información) y el EBCDIC, que es un código de ocho bits muy empleado en sistemas informáticos.

Detección y corrección de errores

En la codificación binaria es importante la detección y corrección de errores. La presencia de ruido y los fallos de componentes originan errores en los sistemas digitales. Hay códigos que permiten la detección de errores, para ello es necesario que no se utilicen todas las combinaciones posibles de los *n* bits del código binario.

1. Códigos detectores de error

Distancia de un código es el número de bits de una configuración binaria del código, que deben ser modificados para obtener otra combinación binaria del código.

Distancia mínima es la menor de las distancias entre dos combinaciones binarias cualesquiera pertenecientes al mismo código.

Para que un código pueda detectar errores, la distancia mínima ha de ser superior a la unidad.

Una forma de detectar errores es introduciendo un bit llamado **de paridad** que tenga en cuenta el número de unos que tiene cada combinación.

Se dice que una palabra tiene paridad par si el número de *unos* que contiene es par.

Se dice que una palabra tiene paridad impar si el número de *unos* que contiene es impar.

Un código es capaz de detectar un error si su distancia mínima es dos, para ello le añadimos un bit a cada palabra de código. Este bit añadido se llama **bit de paridad.**

En la tabla 7, puedes ver el código gray de tres bits al que se le ha añadido un bit de paridad par.

Código Decimal	Código Gray	Bit de Paridad
0	0 0 0	0
1	0 0 1	1
2	0 1 1	0
3	0 1 0	1
4	1 1 0	0
5	1 1 1	1
6	1 0 1	0
7	1 0 0	1

Tabla 6: Código Gray con bit de paridad par.

Este nuevo código formado tendrá ahora cuatro bits. Fíjate que cada combinación lleva un número par de unos.

Si envío el 1111 que se corresponde con el 5 en decimal, y recibo el 0111 vemos que ha cambiado un bit (hay un error) y sabemos que hay error porque ese código no existe.

En cambio, si envío 1111 y recibo 0011, han cambiado dos bits y no se detecta error porque ese código sí existe.

2. Códigos correctores de error

Se basan en la introducción de varios bits de paridad que permiten deducir la posición del bit erróneo. Se les llama **códigos de Hamming.**

Con una distancia mínima tres, se pueden detectar dos errores o corregir un error.

Si consideras que has concluido el estudio de esta unidad, intenta responder a las siguientes cuestiones de autoevaluación.

Cuestiones de autoevaluación

1

Pasa el número 38
en decimal a:

a. Binario natural.

b. BCD natural.

2

Escribe el código
BCD exceso 3 con bit
de paridad impar.

R ACTIVIDAD 1

Para pasar el número 11,3125 en base diez a su equivalente en base dos, primero, pasamos la parte entera:

11 : 2 1
5 : 2 1
2 : 2 0
1

1 0 1 1

$11_{10} \equiv 1011_2$

y, luego, pasamos la parte decimal:

0 1 0 1

0,3125 x 2 = 0,625
0,625 x 2 = 1,25
0,25 x 2 = 0,50
0,5 x 2 = 1,00

$0{,}3125_{10} \equiv 0101_2$

Por tanto:

$$11{,}3125_{10} = 1011{,}0101_2$$

R ACTIVIDAD 2

a. El número 1111100111101100 en base dos es equivalente a

$$\underbrace{1111}_{F}\ \underbrace{1001}_{9}\ \underbrace{1110}_{E}\ \underbrace{1100}_{C}$$

en hexadecimal.

b. El número A5B en hexadecimal es equivalente a

$$\overbrace{1010}^{A}\ \overbrace{0101}^{5}\ \overbrace{1011}^{B}$$

en binario.

Respuestas a las cuestiones de autoevaluación

1

El número 38 en decimal es:

a. en binario natural: 100110.

b. en BCD natural: 00111000.

2

El código es el que ves en la tabla:

	EXCESO 3 CON BIT PARIDAD				IMPAR
0	0	0	1	1	1
1	0	1	0	0	0
2	0	1	0	1	1
3	0	1	1	0	1
4	0	1	1	1	0
5	1	0	0	0	0
6	1	0	0	1	1
7	1	0	1	0	1
8	1	0	1	1	0
9	1	1	0	0	1

Resumen de Unidad

Sistema de numeración decimal

La representación de números en decimal es la habitual en nuestra vida cotidiana. Los dígitos en base diez van del 0 al 9, y con ellos podemos representar cualquier cantidad.

Sistema de numeración binario

En el sistema binario sólo existen dos símbolos: el 0 y el 1, y con ellos se puede representar cualquier magnitud.

Hay otros sistemas importantes, como son: el **octal** y el **hexadecimal.**

Códigos binarios

El sistema más utilizado en la realización de circuitos digitales es el **binario**.

Hay diferentes tipos de códigos binarios como el **código binario natural,** el **código gray** o los **códigos BCD.**

Los códigos BCD son códigos decimales codificados en binario, y representan los diez dígitos diferentes del sistema decimal, haciendo corresponder a cada dígito decimal cuatro dígitos binarios.

Detección y corrección de errores

Ningún sistema está exento de errores, y, para minimizarlos, existen códigos que permiten la detección de errores.

Para que un código pueda detectar errores la distancia mínima ha de ser superior a la unidad.

Una forma de detectar errores se consigue introduciendo un bit llamado **de paridad,** que tenga en cuenta el número de unos que tiene cada combinación.

Los códigos correctores de error se basan en la introducción de varios bits de paridad.

Con una distancia mínima tres, se pueden detectar dos errores o corregir un error.

Notas

Vocabulario

Cíclico: es aquel código en que la última combinación es adyacente a la primera.

Continuo: un código binario es continuo si las combinaciones correspondientes a dos decimales consecutivos son adyacentes.

ISO 9001
ACREDITADO POR ENAC

FONDO FORMACION