

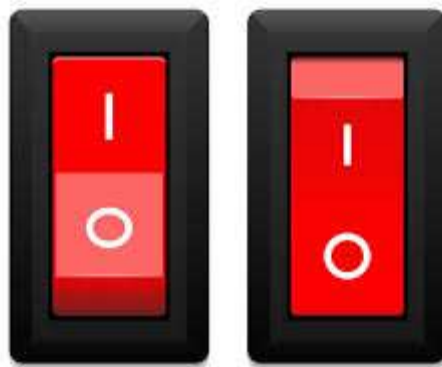
Tema 1

Sistemas de numeración

1.1 - Sistema de numeración BINARIO

En el sistema binario de numeración existen solamente dos símbolos distintos: el “0” y el “1”. A primera vista puede resultar una forma de representación menos complicada que la decimal, con bastantes más símbolos. En la práctica no es así, costando más esfuerzo al ser humano la percepción de cantidades escritas en binario que en decimal.

La razón de su importancia y uso radica en que las máquinas tienen especial facilidad para manejar datos codificados en este sistema de numeración. Así pues, el hombre debe aprender a convertir los estados de activación industriales, magnitudes, mandos y regulaciones para adaptarlos a esta forma de representación.



La ventaja del sistema de base dos o binario consiste precisamente en el hecho de tener solamente dos dígitos, fácilmente aplicables a estados naturales: abierto/cerrado, lleno/vacío, encendido/apagado. En el campo de la electrónica se hacen coincidir los estados lógicos con diodos que conducen o no conducen o con transistores cortados o saturados, estableciendo así la relación entre este sistema de numeración y su aplicación práctica.

La misma teoría explicada para el sistema de numeración decimal sirve para el binario. Así pues, si tenemos un número de cosas codificadas en binario, intentaremos pasarlas a decimal para saber con exactitud a qué nos estamos refiriendo.

Comparación de símbolos entre diferentes sistemas de numeración:

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

Ejemplo práctico:



Una máquina puede indicar refiriéndose al número de coches que han pasado por un lavadero automático:

1001010_2

Si cada coche cuesta 4 euros de limpiar difícilmente se podrá saber el valor total del recaudo si no se sabe pasar ese valor a decimal, pues a primera vista no se posee información directa.

La forma de transformar esos números es parecida a la empleada en el sistema de numeración decimal, es decir, dando un valor determinado a cada posición de un dígito binario. **Al valor que cada dígito posee en su lugar o posición se le denomina “peso”.**

Como el sistema binario contiene dos dígitos, la base será dos, teniendo:

Posición	6	5	4	3	2	1	0
Valor	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Dígito	1	0	0	1	0	1	0

Correspondiendo ese valor en binario al siguiente valor en decimal:

$$\begin{array}{rcl} 1 \times 2^6 & = & 1 \times 64 \\ 0 \times 2^5 & = & 0 \times 32 \\ 0 \times 2^4 & = & 0 \times 16 \\ 1 \times 2^3 & = & 1 \times 8 \\ 0 \times 2^2 & = & 0 \times 4 \\ 1 \times 2^1 & = & 1 \times 2 \\ 0 \times 2^0 & = & 0 \times 1 \end{array}$$

TOTAL 74₁₀

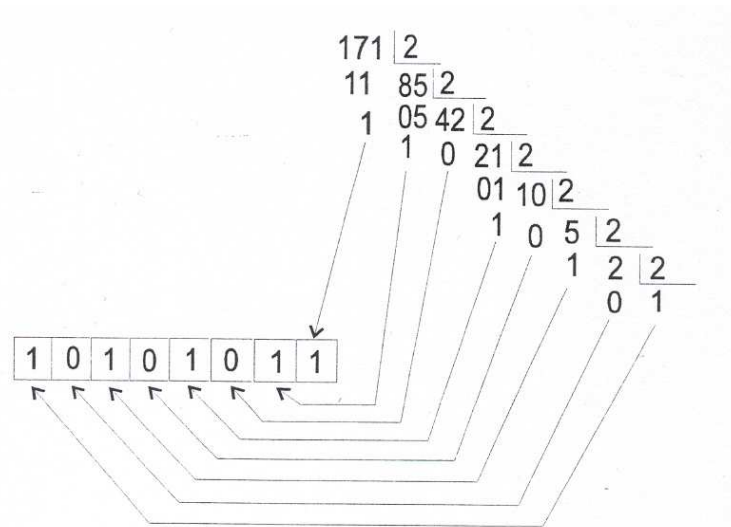
Correspondiendo a la siguiente expresión polinómica:

$$1001010_2 = 1 \times 2^6 + 0 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 0 \times 2^0 = 74_{10}$$

De esta forma se puede decir que los siguientes números representan la misma cantidad de cosas:

$$1001010_2 = 74_{10}$$

Ahora, para convertir un número en base diez a su correspondiente en base dos o binario, el procedimiento consiste en realizar sucesivas divisiones del número en cuestión entre dos, hasta conseguir un valor 0 o 1 de resto. Vamos a verlo con un ejemplo; pasemos el número 171_{10} a base dos:



En la primera división se obtiene un cociente de 85. Hay que seguir dividiendo mientras exista un valor mayor de 1. Los sucesivos restos obtenidos (serán siempre ceros o unos) serán los bits binarios correspondientes al número decimal, siendo el de mayor peso el último y el de menor el primero, tal como se puede observar.

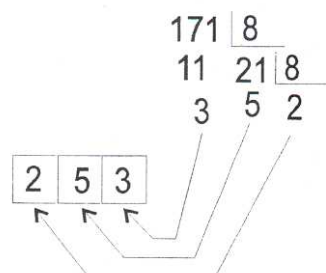
Ahora se puede comprobar el resultado, haciendo el cambio al revés, de binario a decimal:

$$\begin{aligned}
 10101011_2 &= 1 \times 2^7 + 0 \times 2^6 + 1 \times 2^5 + 0 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 = \\
 &= 128 + 32 + 8 + 2 + 1 = \\
 &= 171_{10}
 \end{aligned}$$

1.2 - Sistema de numeración OCTAL

El mismo procedimiento mostrado para la conversión del binario es utilizado para pasar números en base diez a cualquier otra base.

Por ejemplo, si deseamos representar ese mismo número decimal a base 8 (octal), sólo variará el número de base que constituye el dividendo:



Aquí se puede ver una disminución en la cantidad de operaciones a realizar con respecto a la operación de pasar a binario. En este caso, se ha utilizado un sistema con más dígitos.

Y para comprobar el resultado, volvemos a transformarlo a base decimal:

$$253_8 = 2 \times 8^2 + 5 \times 8^1 + 3 \times 8^0 = 2 \times 64 + 5 \times 8 + 3 \times 1 = 128 + 40 + 3 = 171_{10}$$

Ejercicios: Convierte a binario o decimal según corresponda.

$$181_{10} =$$

$$213_{10} =$$

$$77_{10} =$$

$$93_{10} =$$

$$10110101_2 =$$

$$11001011_2 =$$

$$1001101_2 =$$

$$1001101_2 =$$

Para evitar confusiones en el manejo de datos por parte de los ordenadores, se usa otra nomenclatura para determinar el peso de los bits dentro de una **palabra**. Así, se define como **bit más significativo (MSB)** al correspondiente a la izquierda de la palabra, y **bit menos significativo (LSB)** al de la derecha.



El procedimiento para contar en base dos es parecido al decimal, teniendo en cuenta los cambios de posición del bit conforme la cantidad aumenta. Para empezar, nos ayudaremos de una tabla donde aparezcan los valores de los pesos de los bits, además de su correspondiente en decimal:

Binario			Decimal
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
0	0	0	0

El cero es fácil de colocar. Para contar “uno” identificaremos la posición de valor “1” en decimal y escribiremos un “uno” en esa localización.

Binario			Decimal
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
0	0	0	0
0	0	1	1

Seguimos adelante. Ponemos el “uno” correspondiente en la posición de “dos” en decimal, pero debemos poner “cero” en la del valor “uno” en decimal.

Binario			Decimal
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
0	0	0	0
0	0	1	1
0	1	0	2

Para el “tres”, sumaremos “dos” más “uno”, rellenando con “unos” las posiciones de 2^1 y 2^0 :

Binario			Decimal
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3

Continuando con la misma idea se puede rellenar toda la tabla hasta donde podamos, es decir, hasta donde la cantidad de bits de la palabra permita sin repetir combinaciones. La cantidad de estas combinaciones o números que se pueden contar es un concepto muy importante que veremos más adelante. El cálculo se realiza elevando la base, 2, al número de posiciones consideradas en la palabra. Así de esta forma en el ejemplo se dispone de tres dígitos y por tanto se podrá llegar hasta: $2^n = 2^3 = 8$ combinaciones.

Binario			Decimal
$2^2 = 4$	$2^1 = 2$	$2^0 = 1$	
0	0	0	0
0	0	1	1
0	1	0	2
0	1	1	3
1	0	0	4
1	0	1	5
1	1	0	6
1	1	1	7

Si se define un sistema con una longitud de palabra de ocho bits, contaremos con más combinaciones y, por tanto, con más números, y con capacidad de tratar más información.

$$2^n = 2^8 = 256$$



Este es el secreto de la capacidad de almacenamiento de las máquinas procesadoras de información y ordenadores, la llamada “**memoria**”. Para recordar dónde deja un dato, da una determinada dirección numérica. Cuantas más posibilidades de dar números distintos tenga, más capacidad de almacenamiento estará disponible. Así, cuando se dice que un ordenador tiene 64K de memoria, debes saber que son 64.000 posiciones, celdas o lugares distintos determinados por un sistema y que puede contar hasta 64.000, disponiendo de palabras de 16 bits:

$$2^n = 2^{16} = 65.536$$

Por el redondeo se suele decir 64.000.

1.3 - Sistema de Numeración HEXADECIMAL

Tal como se ha visto con el sistema de numeración decimal se emplean diez dígitos. Con ellos es posible representar cantidades del 0 a 9 sin necesidad de combinar cifras. En binario, estos números necesitan el empleo de palabras de cuatro bits, correspondiendo el 0_{10} a 0000_2 y el 9_{10} a 1001_2 . Con cuatro bits en binario se pueden representar más números, hasta el 15_{10} (1111_2).

Para sistemas que necesiten más combinaciones se ha ideado el sistema de numeración hexadecimal, o lo que es lo mismo, en base 16. Para estos se necesita definir 16 símbolos que representen los 16 pesos o valores para combinar.

Se emplean los diez primeros dígitos del sistema decimal, añadiéndole seis letras del abecedario: A, B, C, D, E y F. De esta forma, se puede establecer la siguiente relación entre los tres sistemas de numeración más importantes en electrónica digital.

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

Los números hexadecimales pueden disponer de una longitud de palabra tan larga como se quiera. Así, tenemos números que **mezclan dígitos numéricos con letras**, llamándose **alfanuméricos**.

Por ejemplo:

12AF₁₆ 51D4FB₁₆.

Las operaciones vuelven a encaminarse a conseguir determinar en cantidades decimales las cantidades representadas en base 16, de forma parecida a la vista para los demás.

Ejemplo práctico:

Vamos a intentar pasar el número hexadecimal 8FA3₁₆ a decimal. Para ello aplicamos el concepto del “peso” en relación con la base:

16 ³	16 ²	16 ¹	16 ⁰
8	F	A	3

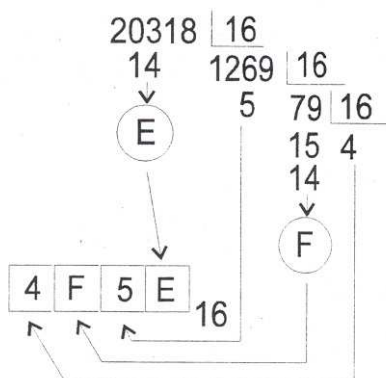
Teniendo en cuenta que F=15₁₀ y A=10₁₀

$$8FA3_{16} = 8 \times 16^3 + F \times 16^2 + A \times 16^1 + 3 \times 16^0 = 8 \times 16^3 + 15_{10} \times 16^2 + 10_{10} \times 16^1 + 3_{10} \times 16^0 = 36.771_{10}$$

Para realizar el paso en sentido contrario se utiliza el procedimiento de la división sucesiva, sustituyendo los valores.

Ejemplo práctico:

Ejemplo de conversión del número 20318_{10} a hexadecimal:



Comprobando el resultado:

$$\begin{aligned}
 4F5E_{16} &= 4 \times 16^3 + F \times 16^2 + 5 \times 16^1 + E \times 16^0 = \\
 &= 4 \times 16^3 + 15_{10} \times 16^2 + 5_{10} \times 16^1 + 14 \times 16^0 = 20.318_{10}
 \end{aligned}$$

Ejemplo práctico:

Para convertir números en formato hexadecimal a binario, se procede adjudicando a cada dígito en base 16 su correspondiente en binario. Por ejemplo, con el $4F5E_{16}$:

4	F	5	E
0100	1111	0101	1110

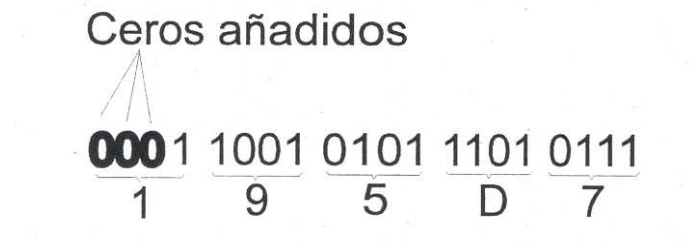
Siendo la representación del número binario la colocación ordenada, de izquierda a derecha, de todos los bits:

$$4F5E_{16} = 0100111101011110_2$$

Para realizar el paso contrario, es decir, de binario a decimal, se forman grupos de cuatro bits de derecha a izquierda, codificando en hexadecimal el número que representa (obviamente empleando el dígito hexadecimal). Si en el último grupo faltan bits para completar un grupo, se añaden ceros a la izquierda del mismo.

Ejemplo práctico:

Deseamos pasar el número 11001010111010111_2 a hexadecimal. Formamos los grupos:



De donde podemos escribir:

$$11001010111010111_2 = 195D7_{16}$$

Ejercicios: Convierte a binario y decimal los siguientes elementos.

$$AF32_{16} =$$

$$AB75_{16} =$$

$$1334_{16} =$$

$$E143_{16} =$$

Ejercicios: Convierte a hexadecimal los siguientes elementos.

$$1010111100110010_2 =$$

$$1010101101110101_2 =$$

$$0001001100110100_2 =$$

$$11100001010000011_2 =$$
