SISTEMAS DE NUMERACIÓN

“ Sólo hay 10 tipos de personas en el mundo…
Las que saben sistema binario y las que no. ”

SISTEMAS DE NUMERACIÓN

Los sistemas de numeración binario y hexadecimal son ampliamente utilizados en el campo de la informática y son especialmente relevantes para los analistas de sistemas.

UTILIDADES:

Sistema de numeración binario:

1. Representación de datos digitales: La informática trabaja con señales eléctricas que se pueden encender o apagar, lo que se traduce en los dígitos binarios 0 y 1. El sistema binario es esencial para representar y manipular datos digitales en los sistemas informáticos. Los analistas de sistemas utilizan el sistema binario para comprender cómo se almacenan, se procesan y se transmiten los datos en formato binario.
2. Diseño de circuitos lógicos: Los circuitos lógicos, que son la base de la electrónica digital, utilizan el sistema binario para representar y operar con valores lógicos. Los analistas de sistemas pueden utilizar el sistema binario para diseñar y analizar circuitos lógicos.
3. Programación de sistemas: En la programación de sistemas, el sistema binario es fundamental para representar y manipular datos en la memoria de la computadora. Los analistas de sistemas trabajan con lenguajes de programación que utilizan el sistema binario para realizar operaciones aritméticas, lógicas y manipulación de bits. Comprender y trabajar con el sistema binario es esencial para el desarrollo de software a nivel de sistema.

Sistema de numeración hexadecimal:

1. Representación compacta de números binarios: El sistema hexadecimal proporciona una forma más compacta y legible de representar números binarios largos. Cada dígito hexadecimal corresponde a cuatro dígitos binarios, lo que facilita la visualización y el manejo de valores binarios largos. Podemos utilizar el sistema hexadecimal para representar direcciones de memoria, valores de registros y otros datos binarios en una forma más concisa y fácil de leer.
2. Representación de direcciones de memoria: En sistemas informáticos, las direcciones de memoria se representan a menudo en formato hexadecimal. El sistema hexadecimal permite una representación más clara y concisa, especialmente en sistemas con direcciones de memoria largas.
3. Identificación de patrones y errores: El sistema hexadecimal es útil para identificar patrones y errores en datos binarios. Al representar datos binarios en formato hexadecimal, podemos detectar fácilmente repeticiones, secuencias específicas y errores en los datos.

SISTEMAS DE NUMERACIÓN

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos. Los sistemas de numeración actuales son sistemas posicionales, que se caracterizan porque un símbolo tiene distinto valor según la posición que ocupa en la cifra.

Sistema de numeración decimal:

Nosotros usamos el sistema de numeración decimal, que nació en la India en el siglo VII y llegó a Europa por medio de los árabes. Este sistema se compone de diez símbolos o dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8 y 9) a los que otorga un valor **dependiendo de la posición** que ocupen en la cifra: unidades, decenas, centenas, unidad de mil etc.

Órdenes del Sistema de Numeración Decimal	Potencia de base 10
Unidad	$10^0 = 1$
Decena	$10^1 = 10$
Centena	$10^2 = 100$
Unidad de mil	$10^3 = 1.000$
Decena de mil	$10^4 = 10.000$
Centena de mil	$10^5 = 100.000$
Unidad de millón	$10^6 = 1.000.000$

En este sistema, diez unidades de un orden cualquiera hacen una unidad del orden inmediato superior. Así, el valor de una cifra depende del lugar que ocupa. Por eso decimos que es un sistema posicional.

El valor de cada dígito está asociado al de una potencia de base 10, número que coincide con la cantidad de símbolos o dígitos del sistema decimal, y un exponente igual a la posición que ocupa el dígito menos uno, contando desde la derecha.

En el sistema decimal el número **528**, por ejemplo, significa:

5 centenas + 2 decenas + 8 unidades, es decir:

$528 = 5.10^2 + 2.10^1 + 8.10^0$ o, lo que es lo mismo:

$500 + 20 + 8 = 528$

Sistema de numeración binario.

Nosotros contamos con 10 dígitos:
0, 1, 2, 3, 4, 5, 6, 7, 8 y 9

Supongamos que ahora uno contará solamente con dos dígitos: 0 y 1.

¿Cómo hacer para poder escribir un número?

Si uno sigue la misma lógica que cuando tiene los diez dígitos, primero los usa a todos por separado. Es decir, usa: 0, 1,2, 3, 4, 5, 6, 7, 8, 9. Cuando llega hasta aquí, ya no los puede usar a los dígitos solos. Necesita combinarlos. Es decir, necesitamos usar ahora dos de los dígitos. Y empieza con el 10. Y sigue, 11, 12, 13, 14...19... (aquí necesita empezar con el siguiente dígito), y usa el 20,21, 22, 23... 29, 30... etcétera... hasta que llega al 97, 98, 99.
En este punto, ya agotó todas las posibilidades de escribir números que tengan dos dígitos. Y sirvieron para enumerar los primeros cien (porque empezamos con el 0. Hasta el 99, hay justo 100).
¿Y ahora? Necesitamos usar tres dígitos (y que no empiecen con cero, porque si no, es como tener dos dígitos pero en forma encubierta). Entonces, empezamos con 100, 101, 102... etcétera.
Después de llegar a los mil, necesitamos cuatro dígitos. Y así siguiendo. Es decir: cada vez que agotamos todos los posibles números que podemos escribir con un dígito, pasamos a dos.
Cuando agotamos los de dos, pasamos a los de tres. Y luego a los de cuatro. Y así siguiendo.

Cuando uno tiene dos dígitos solamente, digamos el 0 y el 1, ¿cómo hacer? Usamos primero los dos dígitos por separado:
0 = 0
1 = 1
Ahora, necesitamos pasar al siguiente caso, o sea, cuando necesitamos usar dos dígitos (y curiosamente, necesitamos ya usar dos dígitos para escribir el número dos):

10 = 2
11 = 3

Aquí, ya agotamos las posibilidades con dos dígitos. Necesitamos usar más:

100 = 4
101 = 5
110 = 6
111 = 7
Y necesitamos uno más para seguir:
1 000 = 8
1 001 = 9
1 010 = 10
1 011 = 11
1 100 = 12
1 101 = 13

1 110 = 14
1 111 = 15
Escribo sólo un paso más:
10 000 = 16
10 001 = 17
10 010 = 18
10 011 = 19
10 100 = 20
10 101 = 21
10 110 = 22
10 111 = 23
11 000 = 24
11 001 = 25
11 010 = 26
11 011 = 27
11 100 = 28
11 101 = 29
11 110 = 30
11 111 = 31

Y aquí los dejo a ustedes solos. Pero lo que queda claro es que para poder llegar al 32, hace falta agregar un dígito más y usar el 100.000.
Lo notable es que con sólo dos dígitos es posible escribir cualquier número. Los números están ahora escritos en potencias de 2, de la misma forma en que antes estaban escritos en potencias de 10.

En una cifra binaria, cada dígito tiene distinto valor dependiendo de la posición que ocupe. El valor de cada posición es el de una potencia de **base 2**, elevada a un exponente igual a la posición del dígito menos uno. Se puede observar que, tal y como ocurría con el sistema decimal, la base de la potencia coincide con la cantidad de dígitos utilizados (2) para representar los números.

De acuerdo con estas reglas, el número binario 11011 tiene un valor que se calcula así:

$$1.\,2^4 + 1.\,2^3 + 0.\,2^2 + 1.\,2^1 + 1.\,2^0, \text{ es decir:}$$

$$16 + 8 + 0 + 2 + 1 = 27$$

y para expresar que ambas cifras describen la misma cantidad lo escribimos así:

$$1011_2 = 27_{10}$$

Veamos más ejemplos:

a) $111 = 1 \,.\, 2^2 + 1 \,.\, 2^1 + 1 \,.\, 2^0 = 7$
b) $1\,010 = 1 \,.\, 2^3 + 0 \,.\, 2^2 + 1 \,.\, 2^1 + 0 \,.\, 2^0 = 10$
c) $1\,100 = 1 \,.\, 2^3 + 1 \,.\, 2^2 + 0 \,.\, 2^1 + 0 \,.\, 2^0 = 12$
d) $110\,101 = 1 \,.\, 2^5 + 1 \,.\, 2^4 + 0 \,.\, 2^3 + 1 \,.\, 2^2 + 0 \,.\, 2^1 + 1 \,.\, 2^0 = 53$

(Un dato interesante es que todo número par termina en cero,
y todo número impar termina en uno).

Conversión de números decimales a binarios

Convertir un número decimal al sistema binario es muy sencillo: basta con realizar **divisiones sucesivas por 2** y escribir los restos obtenidos en cada división en **orden inverso** al que han sido obtenidos.

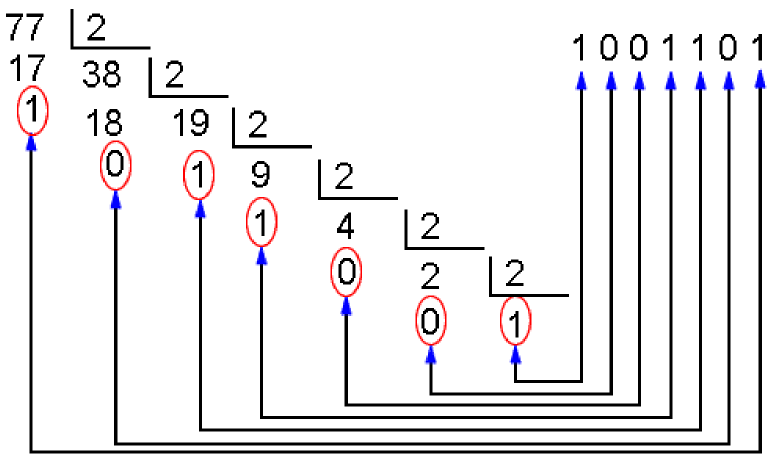
Por ejemplo, para convertir al sistema binario el número 77_{10} haremos una serie de divisiones que arrojarán los restos siguientes:

77 : 2 = 38 Resto: 1
38 : 2 = 19 Resto: 0
19 : 2 = 9 Resto: 1
9 : 2 = 4 Resto: 1
4 : 2 = 2 Resto: 0
2 : 2 = 1 Resto: 0
1 : 2 = 0 Resto: 1

y, tomando los restos en orden inverso obtenemos la cifra binaria:

$$77_{10} = 1001101_2$$

Visto de una forma más sencilla:



Ejercicio 1:

a. Expresa, en sistema binario, los números decimales siguientes: 191, 25, 67, 99, 135, 276
b. Expresar, en sistema decimal, los números binarios siguientes: 1001001, 1111101, 101110

El tamaño de las cifras binarias

La cantidad de dígitos necesarios para representar un número en el sistema binario es mayor que en el sistema decimal. En el ejemplo del párrafo anterior, para representar el número 77, que en el sistema decimal está compuesto tan sólo por dos dígitos, han hecho falta siete dígitos en binario.

Para representar números grandes harán falta muchos más dígitos. Por ejemplo, para representar números mayores de 255 se necesitarán más de ocho dígitos, porque $2^8 = 256$ y podemos afirmar, por tanto, que 255 es el número más grande que puede representarse con ocho dígitos.

Como regla general, con n dígitos binarios pueden representarse un máximo de 2^n, números. El número más grande que puede escribirse con n dígitos es una unidad menos, es decir, $2^n - 1$. Con cuatro bits, por ejemplo, pueden representarse un total de 16 números, porque $2^4 = 16$ y el mayor de dichos números es el 15, porque $2^4 - 1 = 15$.

Ejercicio 2:

Averigua cuántos números pueden representarse con 8, 10, 16 y 32 bits y cuál es el número más grande que puede escribirse en cada caso.

Conversión de binario a decimal

El proceso para convertir un número del sistema binario al decimal es aún más sencillo; basta con desarrollar el número, teniendo en cuenta el valor de cada dígito en su posición, que es el de una potencia de 2, cuyo exponente es 0 en el bit situado más a la derecha, y se incrementa en una unidad según vamos avanzando posiciones hacia la izquierda.

Por ejemplo, para convertir el número binario 1010011 a decimal, lo desarrollamos teniendo en cuenta el valor de cada bit:

$$1.\,2^6 + 0.\,2^5 + 1.\,2^4 + 0.\,2^3 + 0.\,2^2 + 1.\,2^1 + 1.\,2^0 = 83$$

$$1010011_2 = 83_{10}$$

Ejercicio 3:

Expresa, en el sistema decimal, los siguientes números binarios:
110111, 111000, 010101, 101010, 1111110

Sistema de numeración hexadecimal

En el sistema hexadecimal los números se representan con dieciséis símbolos: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F . Se utilizan los caracteres A, B, C, D, E y F representando las cantidades decimales 10, 11, 12, 13, 14 y 15 respectivamente, porque no hay dígitos mayores que 9 en el sistema decimal. El valor de cada uno de estos símbolos depende, como es lógico, de su posición, que se calcula mediante potencias de base 16.

Si quisiera hacer una lista de números, luego del 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E y F, vendrían las combinaciones de ellos tomados de a dos cifras… (sin comenzar con el 0)... 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E, 1F, 20, 21, 22, 23....

Calculemos, a modo de ejemplo, el valor del número hexadecimal $(1A3F)_{16}$:

$$1A3F_{16} = 1.16^3 + A.16^2 + 3.16^1 + F.16^0$$
$$1 . 4096 + 10 . 256 + 3. 16 + 15. 1 = 6719$$

$$1A3F_{16} = 6719_{10}$$

Ejercicio 7:

Expresa en el sistema decimal las siguientes cifras hexadecimales: 2BC516, 10016, 1FF16

Ensayemos, utilizando la técnica habitual de divisiones sucesivas, la conversión de un número decimal a hexadecimal. Por ejemplo, para convertir a hexadecimal del número 1735_{10} será necesario hacer las siguientes divisiones:

1735 : 16 = 108 Resto: 7
108 : 16 = 6 Resto: C es decir, 12_{10}
6 : 16 = 0 Resto: 6

De ahí que, tomando los restos en orden inverso, resolvemos el número en hexadecimal:

$$1735_{10} = 6C7_{16}$$

Numeración hexadecimal octal y binario

Podemos usar la tabla de equivalentes, cada cifra hexadecimal corresponde a la combinación de 4 cifras binarias. Y cada cifra octal representa 3 cifras binarias

Por ejemplo: $(145)_8 \Rightarrow 1_8 = 001_2,\ 4_8 = 100_2,\ 5_8 = 101\ _2 \Rightarrow (001100101)_2$
(Los ceros a izquierda no influyen por lo tanto: $(1100101)_2$
$(A23)_{16} \Rightarrow A_{16} = 1010_2,\ 2_{16} = 0010_2, 3_{16} = 0011_2 \Rightarrow (101000100011)_2$

DECIMAL	**BINARIO**	**OCTAL**	**HEXADECIMAL**
0	**0**	**0**	**0**
1	**1**	**1**	**1**
2	**10**	**2**	**2**
3	**11**	**3**	**3**
4	**100**	**4**	**4**
5	**101**	**5**	**5**
6	**110**	**6**	**6**
7	**111**	**7**	**7**
8	**1000**	**10**	**8**
9	**1001**	**11**	**9**
10	**1010**	**12**	**A**
11	**1011**	**13**	**B**
12	**1100**	**14**	**C**
13	**1101**	**15**	**D**
14	**1110**	**16**	**E**
15	**1111**	**17**	**F**

<u>¿Qué pasa después del punto decimal?</u>

Si has entendido cómo hacerlo con números enteros, pasamos ahora a los "decimales" (hmmm... no es exacto llamarlos así porque eso es para base 10, pero nos entendemos).

DECIMAL A HEXADECIMAL

Cuando hay "decimales", hacemos multiplicaciones sucesivas y escribimos los resultados de izquierda a derecha.

Probemos con el número pi (3,1416...), vamos a convertirlo a base 16. La parte entera es fácil, en base 16 es 3, ahora pasamos a la parte "decimal":

.1416 × 16 = 2,2656
.2656 × 16 = 4,2496
.2496 × 16 = 3,9936
etc...

La primera multiplicación nos dice que pongamos 2 en la primera posición "decimal", la segunda dice que pongamos 4 en la segunda, etc...

Entonces, la respuesta es:
3, 243… (base 16)

HEXADECIMAL A DECIMAL

Seguimos la misma lógica que en números enteros pero ahora con potencias negativas de 16

Por ejemplo

$$1A,3F_{16} = 1.16^{1} + A.16^{0} + 3.16^{-1} + F.16^{-2}$$

$$1.16^{1} + 10.16^{0} + 3.16^{-1} + 15.16^{-2} = 26,24609\ldots$$

EJERCICIOS DE SISTEMAS NUMÉRICOS

Ejercicios para practicar
Verificar con calculadora o conversor online

1. Completar la tabla con los números correspondientes en los distintos sistemas numéricos

Decimal	Binario	Octal	Hexadecimal
200			
	11100110		
		352	
			2AF
670			
	11010010101		
		27	
			3C
39			
	101010		
		222	
			40
59			
	111110		