Sistema binario

En matemática el **sistema binario** es un sistema de numeración en el que los números se representan utilizando las cifras cero y uno ('0' y '1').

Los ordenadores trabajan internamente con dos niveles de voltaje, por lo que su sistema de numeración natural es el sistema binario (encendido '1', apagado '0').

Historia

El antiguo matemático Indio Pingala presentó la primera descripción que se conoce de un sistema de numeración binario en el siglo tercero antes de Cristo, lo cual coincidió con su descubrimiento del concepto del número cero.

El sistema binario moderno fue documentado en su totalidad por Leibniz en el siglo diecisiete en su artículo "*Explication de l'Arithmétique Binaire*". Leibniz usó el 0 y el 1, al igual que el sistema de numeración binario actual.

En 1854, el matemático británico George Boole, publicó un artículo que marcó un antes y un después, detallando un sistema de lógica que terminaría denominándose Álgebra de Boole. Dicho sistema jugaría un papel fundamental en el desarrollo del sistema binario actual, particularmente en el desarrollo de circuitos electrónicos.

En 1937, Claude Shannon realizó su tesis doctoral en el MIT, en la cual implementaba el Álgebra de Boole y aritmética binaria utilizando relés y conmutadores por primera vez en la historia. Titulada *Un Análisis Simbólico de Circuitos Conmutadores y Relés*, la tesis de Shannon básicamente fundó el diseño práctico de circuitos digitales.

En noviembre de 1937, George Stibitz, trabajando por aquel entonces en los Laboratorios Bell, construyó un ordenador basado en relés - al cual apodó "Modelo K" (porque lo construyó en una cocina, en inglés "**k**itchen")- que utilizaba la suma binaria para realizar los cálculos. Los Laboratorios Bell autorizaron un completo programa de investigación a finales de 1938, con Stibitz al mando. El 8 de enero de 1940 terminaron el diseño de una Calculadora de Números Complejos, la cual era capaz de realizar cálculos con números complejos. En una demostración en la conferencia de la Sociedad Americana de Matemáticas, el 11 de septiembre de 1940, Stibitz logró enviar comandos de manera remota a la Calculadora de Números Complejos a través de la línea telefónica mediante un teletipo. Fue la primera máquina computadora utilizada de manera remota a través de la línea de teléfono. Algunos participantes de la conferencia que presenciaron la demostración fueron John Von Neumann, John Mauchly y Norbert Wiener, el cual escribió acerca de dicho suceso en sus diferentes tipos de memorias en la cual alcanzo diferentes logros.

Operaciones con números binarios

Suma de números Binarios

Las posibles combinaciones al sumar dos bits son

- 0+0=0
- 0+1=1
- 1+0=1

- 1+1=0 y *nos llevamos* **1**

```
  100110101
+  11010101
-----------
 1000001010
```

Operamos como en el sistema decimal: comenzamos a sumar desde la derecha, en nuestro ejemplo, 1 + 1 = 10, entonces escribimos 0 y "llevamos" 1 (Esto es lo que se llama el *arrastre*, *acarreo* o *carry* en inglés). Se suma este 1 a la siguiente columna: 1 + 0 + 0 = 1, y seguimos hasta terminar todas la columnas (exactamente como en decimal).

Resta de números binarios

El algoritmo de la resta, en binario, es el mismo que en el sistema decimal. Pero conviene repasar la operación de restar en decimal para comprender la operación binaria, que es más sencilla. Los términos que intervienen en la resta se llaman minuendo, sustraendo y diferencia.

Las restas básicas 0-0, 1-0 y 1-1 son evidentes:

- 0 – 0 = 0
- 1 – 0 = 1
- 1 – 1 = 0

La resta 0 - 1 se resuelve, igual que en el sistema decimal, tomando una unidad prestada de la posición siguiente: 10 - 1 = 1 y me llevo 1, lo que equivale a decir en decimal, 2 – 1 = 1 . Esa unidad prestada debe devolverse, sumándola, a la posición siguiente. Veamos algunos ejemplos:

Restamos 17 - 10 = 7(2=345) Restamos 217 - 171 = 46 (3=690)

```
 10001                          11011001
-01010                         -10101011
------                         ---------
 00111                          00101110
```

A pesar de lo sencillo que es el procedimiento, es fácil confundirse. Tenemos interiorizado el sistema decimal y hemos aprendido a restar mecánicamente, sin detenernos a pensar en el significado del arrastre. Para simplificar las restas y reducir la posibilidad de cometer errores hay varias soluciones:

Dividir los números largos en grupos. En el siguiente ejemplo, vemos cómo se divide una resta larga en tres restas cortas: Restamos

```
 100110011101             1001     1001     1101
-010101110010            -0101    -0111    -0010
-------------      =     -----    -----    -----
 010000101011             0100     0010     1011
```

Utilizando el Complemento a dos. La resta de dos números binarios puede obtenerse sumando al minuendo el complemento a dos del sustraendo. Veamos algunos ejemplos: Hagamos la siguiente resta, 91 – 46 = 45, en binario:

```
 1011011                                    1011011
-0101110                 C246 = 1010010    +1010010
--------                                   --------
 0101101                                   10101101
```

En el resultado nos sobra un bit, que se desborda por la izquierda. Pero, como el número resultante no puede ser más largo que el minuendo, el bit sobrante se desprecia.

Un último ejemplo. Vamos a restar 219 – 23 = 196, directamente y utilizando el complemento a dos:

```
 11011011                                   11011011
-00010111                C223 = 11101001   +11101001
---------                                   --------
 11000100                                  111000100
```

Y, despreciando el bit que se desborda por la izquierda, llegamos al resultado correcto: 11000100 en binario, 196 en decimal

Utilizando el complemento a 1. La resta de dos números binarios puede obtenerse sumando al minuendo el complemento a uno del sustraendo y a su vez sumarle el bit de overflow (bit que se desborda)

Producto de números binarios

El producto de números binarios es especialmente sencillo, ya que **el 0 multiplicado por cualquier número da 0, y el 1 es el elemento neutro del producto**.

Por ejemplo, multipliquemos 10110 por 1001:

```
      10110
       1001
   ---------
      10110
     00000
    00000
   10110
   ---------
    11000110
```

División de números binarios

La división en binario es similar a la decimal, la única diferencia es que a la hora de hacer las restas, dentro de la división, estas deben ser realizadas en binario

Por ejemplo, vamos a dividir 100010010 (274) entre 1101 (13):

```
 100010010 /1101
           ------
-0000      010101
-------
 10001
- 1101
  -------
  01000
 - 0000
  -------
   10000
  - 1101
  -----------
    00111
   - 0000
   -------
     01110
    - 1101
    -------
     00001
      ------------
```

Conversión entre binarios y decimales, binario a octal y de binario a hexadecimal

Binarios a decimal

Dado un número N, binario, para expresarlo en el sistema decimal, se debe escribir cada número que lo compone, multiplicado por la base dos, elevado a la posición que ocupa. Ejemplo.. $11001_2 = 25_{10}$<=>$1 \times 2^4 + 1 \times 2^3 + 0 \times 2^2 + 0 \times 2^1 + 1 \times 2^0$

Decimales a binarios

Se divide el número decimal entre **2** cuyo resultado entero se vuelve a dividir entre 2 y así sucesivamente. Una vez llegados al **1** indivisible se cuentan el último cociente, es decir el uno final (todo número binario excepto el 0 empieza por uno), seguido de los residuos de las divisiones subsiguientes. Del más reciente hasta el primero que resultó. Este número será el binario que buscamos. A continuación se puede ver un ejemplo con el número decimal 100 pasado a binario.

```
100 |_2
 0   50 |_2
       0  25 |_2          --> 100 => 1100100
           1  12 |_2
                0  6 |_2
                   0  3 |_2
                      1  1
```

Otra forma de conversión consiste en un método parecido a la factorización en números primos. Es relativamente fácil dividir cualquier número entre 2. Este método consiste también en divisiones sucesivas. Dependiendo de si el número es par o impar,

```
colocaremos un cero o un uno en la columna de la derecha. si es impar,
le restaremos uno y seguiremos dividiendo por dos,
```

hasta llegar a 1. Después, sólo nos queda tomar el último resultado de la columna izquierda (que siempre será 1) y todos los de la columna de la derecha y ordenar los dígitos de abajo a arriba. Y luego se haría un cuadro con las potencias con el resultado!

Ejemplo:

```
100|0
 50|0
 25|1 --> 1, 25-1=24 y seguimos dividiendo por 2.
 12|0
  6|0
  3|1
  1|   ------->100 => 1100100
```

Y también tenemos otro método el método de distribución en el que distribuimos el número decimal y podemos tener el resultado en binario, trabaja de la siguiente manera tenemos el numero 151 lo que tenemos que hacer es distribuir este número buscando el numero mas próximo en este caso es 128 así que en la casilla donde hay capacidad de contener el numero que tenemos lo vamos marcando. y en las casillas que no empleamos las marcaremos con un 0.

Ejemplo:

```
  1|1
  2|1
  4|1
  8|0
 16|1
 32|0
 64|0
128|1
256|0
```

y sucesivos

Binario a Octal Para convertir un numero binario a octal: Se agrupa el numero binario en grupos de 3 y se convierte a cada grupo en su octal equivalente mediante los método visto para pasar de binario a decimal (recuerde que cada grupo de 3 puede expresar los números del 0 al 7), ejemplo:

```
B(10 101 001) tenemos 3 grupos (010) (101) (001) se completo el primer
grupo agregando un cero, ahora mediante los métodos vistos lo
convertimos y lo volvemos a agrupar 010 = 2, 101 = 5, 001 = 1,
entonces el numero en octal es Oc(251)
```

Octal a Binario Cada digito Octal se lo convierte en su binario equivalente de 3 bits y se lo agrupa, ejemplo:

Oc(247) --> el 2 en binario es 10 pero en binario de 3 bits es Oc(2) = B(010). el Oc(4) = B(100) y el Oc(7) = (111) entonces el numero en binario sera: B(010100111) = O(247)

Binario a hexadecimal Para pasar de binario a Hexadecimal se realiza el mismo proceso de pasar de Binario a Octal pero se agrupa en grupos de 4 bits

Hexadecimal a Binario Idem que para pasar de Octal a Binario, solo que se remplaza por el equivalente de 4 bits.

Tabla de conversión entre decimal, binario, hexadecimal y octal

Decimal	Binario	Hexadecimal	Octal

0	0000	0	0
1	0001	1	1
2	0010	2	2
3	0011	3	3
4	0100	4	4
5	0101	5	5
6	0110	6	6
7	0111	7	7
8	1000	8	10
9	1001	9	11
10	1010	A	12
11	1011	B	13
12	1100	C	14
13	1101	D	15
14	1110	E	16
15	1111	F	17