CONVERSIÓN BINARIO – DECIMAL

ÍNDICE

¿QUÉ ES Y PARA QUÉ SIRVE?

El sistema binario es un sistema de numeración que utiliza solo dos dígitos: 0 y 1. Es la base fundamental para la representación de información en sistemas electrónicos y ordenadores. Cada dígito en el sistema binario se llama bit (abreviatura de dígito binario).

En este sistema, los números se expresan mediante combinaciones de los dígitos 0 y 1, de manera similar a cómo en el sistema decimal usamos los dígitos 0 al 9. La posición de cada dígito en un número binario tiene un valor que es una potencia de 2: La posición más a la derecha representa 2^0, la siguiente 2^1, luego 2^2, y así sucesivamente.

Gracias a este recurso aprenderemos a pasar un número en el sistema binario al sistema decimal y viceversa.

MÉTODO: BINARIO A DECIMAL

Ejemplo de número binario: 1011

1. **Inicio:**
    - Se tiene el número binario 1011 que se desea convertir a decimal.
2. **Asignar posiciones:**
    - Comenzar desde el extremo derecho y asignar potencias de 2 a cada posición:
        - $2^0\ para\ las\ unidades, 2^1\ para\ las\ decenas, 2^2\ para\ las\ centenas, 2^3\ para\ las\ u.de\ millar$
3. **Multiplicar y sumar:**
    - Multiplicar cada dígito binario por la potencia de 2 correspondiente y sumar los resultados.
        - $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0$

$$\begin{array}{ccccccc} \big\downarrow & & \big\downarrow & & \big\downarrow & & \big\downarrow \\ 8 & + & 0 & + & 2 & + & 1 \end{array}$$

4. **Calcular resultado:**
    - Realizar las operaciones:
        - $8 + 0 + 2 + 1 = 11$

Resultado final: $\mathbf{1011_2 = 11_{10}}$

Este proceso se repite para cualquier número binario. La clave es asignar potencias de 2 a cada posición en orden creciente comenzando por las unidades, multiplicar cada dígito binario por la potencia correspondiente y sumar los resultados.

NÚMEROS CON DECIMALES

Cuando trabajamos con números binarios que tienen parte decimal, el proceso es similar al de los números enteros, pero se extiende para incluir la parte fraccional.

Ejemplo de número binario: 1011.101

1. **Asignar posiciones:**
    - $2^3, 2^2, 2^1, 2^0, 2^{-1}, 2^{-2}, 2^{-3}$
2. **Multiplicar y sumar:**
    - $1 \times 2^3 + 0 \times 2^2 + 1 \times 2^1 + 1 \times 2^0 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3}$

      ⬇ ⬇ ⬇ ⬇ ⬇ ⬇ ⬇

      $8 \quad + \quad 0 \quad + \quad 2 \quad + \quad 1 \quad + \quad 0.5 \quad + \quad 0 \quad + \quad 0.125$
3. **Calcular resultado:**
    - $8 + 0 + 2 + 1 + 0.5 + 0 + 0.125 = 11.625$

MÉTODO: DECIMAL A BINARIO

Ejemplo de número decimal: 19

1. **Inicio:**
    - Se tiene el número decimal 19 que se desea convertir a binario.
2. **División sucesiva por 2:**
    - Dividir el número por 2 sucesivamente y registrar los restos.
        - 19÷2=9 con un resto de 1.
        - 9÷2=4 con un resto de 1.
        - 4÷2=2 con un resto de 0.
        - 2÷2=1 con un resto de 0.
        - 1÷2=0 con un resto de 1.
3. **Obtener los bits:**
    - Leer los restos en orden inverso para obtener la representación binaria.
        - Restos: 10011 (leídos de abajo hacia arriba).

Resultado final: $19_{10} = 10011_2$

Este método implica dividir sucesivamente el número decimal por 2 y registrar los restos en orden inverso. Los restos representan los bits del número binario.

ALMACENAMIENTO EN BYTES

Un byte es una unidad de medida de información en sistemas de almacenamiento digital y procesamiento de datos. Consiste en 8 bits, y cada bit puede tener un valor de 0 o 1, donde el primer bit de un byte se reserva para indicar el signo.

Representación de números con signo en un byte:

- **Bit de signo:**
    - 0: Negativo
    - 1: Positivo
- **7 Bits restantes:**
    - Representan el valor absoluto del número en el sistema binario.

Ejemplo:

Queremos almacenar el número 9527 en un sistema de 16 bits

1. **Reservamos el primer bit para el signo**
    - Como el número es positivo, el primer bit será 1. Son 16 bits en los que el primero determinará el signo y los 15 restantes el valor del número en sistema binario.
2. **División sucesiva por 2**
    - 9527 ÷ 2 = 4763 con un resto de 1
    - 4763 ÷ 2 = 2381 con un resto de 1
    - 2381 ÷ 2 = 1190 con un resto de 1
    - 1190 ÷ 2 = 595 con un resto de 0
    - 595 ÷ 2 = 297 con un resto de 1
    - 297 ÷ 2 = 148 con un resto de 1
    - 148 ÷ 2 = 74 con un resto de 0
    - 74 ÷ 2 = 37 con un resto de 0
    - 37 ÷ 2 = 18 con un resto de 1
    - 18 ÷ 2 = 9 con un resto de 0
    - 9 ÷ 2 = 4 con un resto de 1
    - 4 ÷ 2 = 2 con un resto de 0
    - 2 ÷ 2 = 1 con un resto de 0
    - 1 ÷ 2 = 0 con un resto de 1
3. **Resultado**

    Se han obtenido un total de 14 bits. Como se pide almacenar el número en un sistema de 16 bits y se ha obtenido uno menos (el primer bit ya sabemos que es 1 porque es positivo), el segundo bit va a ser 0.

    De esta manera:

    - $\mathbf{9527_{10} = 1010010100110111\ _2}$

EJERCICIOS PROPUESTOS

EJERCICIO 1

Un sistema informático utiliza 24 bits para codificar números enteros. Teniendo en cuenta que el primer bit se usa para definir el signo, se pide:

1. **Determinar banda de números enteros que se pueden representar**
2. **Representar -2400380**

Solución:

1. Se pueden representar $2^{23} - 1$ números enteros. El exponente es 23 porque son los bits disponibles para el valor del número (el primero se utiliza para determinar el signo) y se resta 1 para considerar el 0.
2. Utilizando el método de la división sucesiva por 2: 000100100011011011111100, primer bit 0 por ser negativo.

EJERCICIO DE EXAMEN PARCIAL 1 23/24

En un sistema de números máquina binario que tenga 5 bits para representar la mantisa (el primero de ellos para el signo) y 4 bits para almacenar el exponente (el primero de ellos para el signo) se pide, en caso de ser posible, el número real que en base 10 se expresa por 250.52

Solución: El número 250.52 puede expresarse en base 2 mediante:

$$250.52_{10} = 1 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + \cdots =$$

$$= (1 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 1 \cdot 2^{-4} + 0 \cdot 2^{-5} + 1 \cdot 2^{-6} + 0 \cdot 2^{-7} + 1 \cdot 2^{-8} + \cdots) \cdot 2^7 =$$

$$=((1.11110101 \ldots) \cdot 2^7)_2$$

El exponente de este número, 7, en binario se expresa como:

$$7_{10} = 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = 111_2$$

Por tanto, el exponente del número puede representarse en el sistema de números máquina considerado (en los cuatro bits) como +111.

La mantisa, al tener sólo 5 bits y ser uno de ellos para el signo, deberá redondearse a sólo cuatro dígitos decimales binarios. Procediendo por redondeo (obsérvese que el quinto decimal es un 1):

1.11110101.... → **+1.1111**

Por tanto, la mantisa del número se codifica como +1111. En resumen, el número máquina que aproxima al número dado (250.52) es el número $+1.1111 \cdot 2^{+111}$

NOTA: Análogamente se podría aplicar la coma flotante de la siguiente manera:

$$=((0.111110101 \ldots) \cdot 2^8)_2$$

Lo que sucede es que, en ese caso, haría falta un bit más para el exponente: 4 para la codificación y uno para el signo.