EQUIPOS MICROPROGRAMABLES

TEMA 1 – SISTEMAS NUMERACIÓN

IES FONTEXERIA (MUROS)

Profesor: Javier Fraga Iriso

1. Introducción

Cualquier sistema de numeración se caracteriza por tener una base y un número de símbolos. Siendo el número de símbolos correspondiente a la base del sistema empleado.

$$\begin{cases} \text{Base} = b \\ \text{Nº símbolos} = n \Rightarrow n = b \end{cases}$$

Por ejemplo, el sistema decimal se basa en diez símbolos y el lugar que ocupa cada número indica su valor.

$$\begin{cases} b = 10 \\ n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9. \end{cases}$$

Ejemplo:

$$386 = 3 \cdot 10^2 + 8 \cdot 10^1 + \ 6 \cdot 10^0 = 300 + 80 + 6$$

2. Sistema binario

El sistema binario solo dispone de dos caracteres, conocidos como bit (Binary Digit).

$$\begin{cases} b = 2 \\ n = 0, 1. \end{cases}$$

2.1. Transformación de decimal a binario

Paso 1: Se toma la parte entera del número y se divide entre dos.

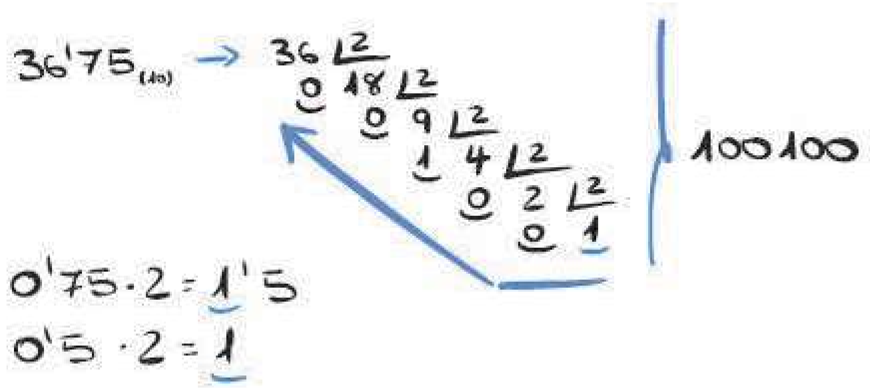
Paso 2: Dividir el resultado de la división anterior entre dos sucesivamente hasta obtener un resultado de 1.

Paso 3: El número binario obtenido será en primer lugar el resultado final de la última división y posteriormente los restos obtenidos de todas las divisiones en orden contrario a su obtención.

***Paso 4:** En el caso de que el número posea una parte decimal, esta será tomada a parte y se multiplicará por dos.

***Paso 5:** Se tomará la parte entera del resultado como un dígito del número en binario y la parte decimal se volverá a multiplicar.

Ejemplo:



Resultado → $100100{,}11_{(2)}$

2.2. Transformación de binario a decimal

Para pasar de binario a decimal, se tomarán los dígitos del número uno a uno, y se multiplicarán por su base elevada a la posición que ocupa dicho digito.

Se empezará a contar por el primer digito de la parte entera que ocupa la posición 0 y se continua hacia la izquierda sumando uno por cada posición que avanzamos. Para la parte decimal se resta de uno en uno.

Ejemplo:

$$\mathbf{100100,11_{(2)} = 1 \cdot 2^5 + 0 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 1 \cdot 2^{-1} + 1 \cdot 2^{-2}}$$

$$\mathbf{= 32 + 4 + 0,5 + 0,25 = 36,75_{(10)}}$$

Este método de multiplicar el dígito por la base del código elevada a su posición se emplea también en el paso de binario a octal y hexadecimal.

$$\boldsymbol{dígito \cdot base^{\,posición}}$$

3. Sistema octal

b = 8

n = 0, 1, 2, 3, 4, 5, 6, 7.

El paso de sistema octal a decimal y de decimal a octal es equivalente a las operaciones realizadas con el sistema binario.

Número decimal	Número octal	Número binario (3 bits)
0	0	000
1	1	001
2	2	010
3	3	011
4	4	100
5	5	101
6	6	110
7	7	111
8	10 (1+0)	001 000
9	11 (1+1)	001 001

Ejemplos:

a) Octal a decimal: $326_{(8)} = 3 \cdot 8^2 + 2 \cdot 8^1 + 6 \cdot 8^0 = 214_{(10)}$

b) Decimal a octal:

$$\begin{array}{l|l} 214_{(10)} & 8 \\ \hline 54 & 26 \;|\; \underline{8} \\ 6 & 2 \quad 3 \end{array}$$

$\longleftarrow$

Resultado → $326_{(8)}$

3.1. Transformación de binario a octal

Para pasar de sistema binario a octal se forman grupos de tres bits y a los que se les asigna el valor equivalente en octal, si es necesario se pueden añadir ceros en los extremos para formar los grupos.

Los grupos se forman partiendo desde la coma que divide la parte entera de la decimal, es decir en el caso de un número entero de derecha a izquierda, y en sentido contrario para la parte decimal.

Ejemplo: $\mathbf{010\ 001}_{(2)} \rightarrow \mathbf{21}_{(8)}$

3.2. Transformación de octal a binario

Para pasar de octal a binario se toma cada dígito en octal y se le asigna su valor correspondiente en un número binario de tres bits.

Situándolos todos en orden se obtiene el número en binario equivalente.

Ejemplo:

$$\mathbf{21}_{(8)}$$

010 **001** → $\mathbf{010001}_{(2)}$

4. Sistema hexadecimal

$$\begin{cases} b = 16 \\ n = 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. \end{cases}$$

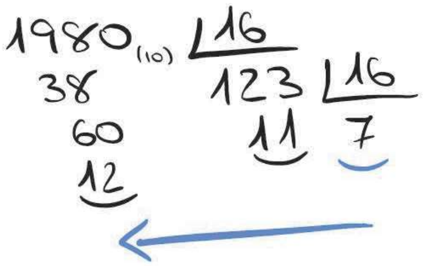
El paso de sistema hexadecimal a decimal es equivalente a las operaciones realizadas con el sistema binario.

Ejemplos:

a) Hexadecimal a decimal:

$$\mathbf{7BC}_{(16)} = \mathbf{7 \cdot 16^2 + 11 \cdot 16^1 + 12 \cdot 16^0 = 1980}_{(10)}$$

b) Decimal a hexadecimal:



Resultado → $7BC_{(16)}$

4.1. Transformación de binario a hexadecimal

Para la realización de esta operación y su inversa haremos uso de la siguiente tabla, la cual relaciona los sistemas decimal hexadecimal y binario.

Decimal	Hexadecimal	Binario (4 bits)
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
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111

Para pasar de sistema binario a hexadecimal se forman grupos de cuatro bits y a estos se les asigna el valor equivalente en hexadecimal.

Los grupos se forman partiendo desde la coma que divide la parte entera de la decimal, es decir en el caso de un número entero de derecha a izquierda, y en sentido contrario para la parte decimal.

Es el mismo proceso realizado para el sistema octal.

Ejemplo:

$$\underbrace{0111}_{7}\ \underbrace{1011}_{B}\ \underbrace{1100}_{C}$$

Resultado → $\boldsymbol{7BC_{(16)}}$

4.2. Transformación de hexadecimal a binario

Se toma cada dígito en hexadecimal y se le asigna su valor en binario correspondiente, situándolos todos en orden se obtiene el número en binario equivalente.

Ejemplo:

$7BC_{(16)}$

0111 1011 1100

Resultado→ $\mathbf{0111100111100_{(2)}}$

5. Códigos binarios BCD

El código BCD (decimal codificado en binario) representa únicamente las cifras numéricas del sistema hexadecimal, por lo que se opera de la misma manera.

A continuación, se muestra una tabla con los diferentes códigos que debes conocer, entre los que se encuentran el BCD natural, el BCD exceso 3 y el BCD Aiken.

El BCD natural emplea el equivalente hexadecimal del sistema decimal, en el BCD exceso 3 debes sumar tres al sistema decimal para hallar su equivalente, y el código Aiken emplea las primeras y las últimas cinco combinaciones del sistema hexadecimal para representar el sistema decimal.

Decimal	Binario	BCD natural	BCD exceso 3	BCD Aiken
0	0000	0000	0011	0000
1	0001	0001	0100	0001
2	0010	0010	0101	0010
3	0011	0011	0110	0011
4	0100	0100	0111	0100
5	0101	0101	1000	1011
6	0110	0110	1001	1100
7	0111	0111	1010	1101
8	1000	1000	1011	1110
9	1001	1001	1100	1111
10	1010	0001 0000	0100 0011	0001 0000
11	1011	0001 0001	0100 0100	0001 0001

6. Codificación binaria

Para codificar el teclado de un ordenador (102 teclas), necesitaríamos como mínimo un número de bits que se puede calcular de varias maneras. La más sencilla es mediante la aplicación de la siguiente formula probar valores de número de bits hasta que se cumpla la ecuación.

$$\boldsymbol{2^{n^{\underline{o}}\ bits} \geq elementos\ a\ codificar}$$

Por lo tanto:

$$2^n \geq 102 \left\{ \begin{array}{l} 2^6 = 64 \rightarrow Menor\ a\ 102 \\ 2^7 = 128 \rightarrow Mayor\ a\ 102 \rightarrow \boldsymbol{Ok} \end{array} \right\} \ \mathbf{n = 7}$$