

SISTEMAS DE NUMERACIÓN

INDICE.

1. DEFINICIÓN DE SISTEMAS DE NUMERACIÓN.....	1
2. TEOREMA FUNDAMENTAL DE LA NUMERACIÓN.....	3
3. SISTEMAS DECIMAL, BINARIO, OCTAL Y HEXADECIMAL.....	5
3.1. SISTEMA DECIMAL.....	5
3.2. SISTEMA BINARIO.....	5
3.3. SISTEMA OCTAL.....	5
3.4. SISTEMA HEXADECIMAL.....	5
4. CONVERSIONES.....	7
4.1. CONVERSIÓN DE DECIMAL A OTRO SISTEMA DE NUMERACIÓN.....	7
4.2. CONVERSIÓN DE UN SISTEMA DE NUMERACIÓN A SISTEMA DECIMAL.....	13
4.3. CONVERSIÓN ENTRE LOS SISTEMAS BINARIO Y OCTAL.....	15
4.4. CONVERSIÓN ENTRE LOS SISTEMAS BINARIO Y HEXADECIMAL.....	16
5. REPRESENTACION DE NÚMEROS ENTEROS CON SIGNO.....	17
5.1. MÓDULO Y SIGNO (MS).....	17
5.2. COMPLEMENTO A 1 (C-1).....	18
5.3. COMPLEMENTO A 2 (C-2).....	19
5.4. EXCESO A 2 ELEVADO A N – 1 (EXCESO 2^{N-1}).....	20
6. REPRESENTACION DE NÚMEROS EN COMA FLOTANTE.....	21
6.1. ESTANDAR IEEE P754.....	25

1. DEFINICIÓN DE SISTEMAS DE NUMERACIÓN.

Se define, en general, un sistema de numeración como el conjunto de símbolos y reglas que se utilizan para la representación de cantidades.

En los sistemas de numeración existe un elemento característico que define el sistema y que se denomina **base**, siendo ésta el número de símbolos que se utilizan para dicha representación. La representación de una cantidad se efectúa por medio de cadenas de símbolos, cada uno de ellos con un significado que depende de su **posición**; por ello estos sistemas se denominan posicionales.

La notación matemática de la base, para distinguir a cuál de ellas nos estamos refiriendo, consiste en poner al final del número un subíndice formado por un paréntesis abierto a la derecha y el número de la base correspondiente.

NÚMERO EN BASE B_B

En electrónica digital, los sistemas de numeración utilizados son el decimal, el binario, el octal y el hexadecimal. Para identificar el sistema en que se trabaja se puede utilizar la notación matemática de la base o se puede añadir al final del número una letra para identificar el sistema, según la siguiente tabla:

SISTEMA	IDENTIFICADOR
Decimal	D o ninguna letra
Binario	B
Octal	Q u O
Hexadecimal	H

2. TEOREMA FUNDAMENTAL DE LA NUMERACIÓN.

Todos los sistemas de numeración posicionales toman como referencia el punto decimal y tienen una base de numeración que de forma implícita interviene en la cantidad que con una determinada representación se quiere referenciar. Se trata de un teorema que **relaciona una cantidad expresada en cualquier sistema de numeración con la misma cantidad expresada en el sistema decimal.**

Supongamos una cantidad expresada en un sistema cuya base es B, y representamos por X_i cada uno de los dígitos que contiene dicha cantidad y cuyo subíndice indica la posición de la cifra respecto al punto decimal, posición que hacia la izquierda del punto se numera desde 0 en adelante y de 1 en 1, y hacia la derecha se numeran desde -1 y con incremento -1. En síntesis, el teorema indica el valor decimal de una cantidad expresada en cualquier sistema de representación, y viene dado por la fórmula:

$$N^{\circ} = \sum_{i=-d}^n X_i \times B^i$$

donde:

- B = Base del sistema de numeración.
- i = Posición respecto al punto.
- d = Número de cifras a la derecha del punto.
- n = Número de cifras a la izquierda del número menos 1.
- X = Cada una de las cifras que componen el número.

Así dado el número

$$X_4 X_3 X_2 X_1 X_0 \cdot X_{-1} X_{-2} X_{-3} (B)$$

su valor decimal viene expresado por el sumatorio

$$N^{\circ} = \sum_{i=-3}^4 X_i \times B^i = X_{-3} \times B^{-3} + X_{-2} \times B^{-2} + X_{-1} \times B^{-1} + X_0 \times B^0 + X_1 \times B^1 + X_2 \times B^2 + X_3 \times B^3 + X_4 \times B^4$$

EJEMPLO.

Dada la cantidad 543.3 expresada en el sistema de numeración de base 6, ¿cuál será su valor en el sistema decimal?

$$543.3_{(6)} = 5 \times 6^2 + 4 \times 6^1 + 3 \times 6^0 + 3 \times 6^{-1} = 5 \times 36 + 4 \times 6 + 3 \times 1 + 3 \times \frac{1}{6} = 180 + 24 + 3 + 0.5 = 207.5_{(10)}$$

3. SISTEMAS DECIMAL, BINARIO, OCTAL Y HEXADECIMAL.

3.1. SISTEMA DECIMAL.

El sistema decimal es un sistema posicional que utiliza como base el 10, que corresponde al número de símbolos del sistema y que son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9.

3.2. SISTEMA BINARIO.

El sistema binario es un sistema posicional que utiliza como base el 2, que corresponde al número de dígitos utilizados por el sistema y que son: 0, 1.

Cada dígito de un número representado en este sistema se denomina **bit**, que viene de la contracción de **binary digit**.

3.3. SISTEMA OCTAL.

El sistema octal es un sistema posicional que utiliza como base el 8, que corresponde al número de símbolos del sistema y que son: 0, 1, 2, 3, 4, 5, 6, 7.

Este sistema se utiliza para trabajar en binario con mayor sencillez ya que cada cifra octal equivale a tres dígitos binarios, tal como se indica en la siguiente tabla:

OCTAL	BINARIO
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

3.4. SISTEMA HEXADECIMAL.

El sistema hexadecimal es un sistema posicional que utiliza como base el 16, que corresponde al número de símbolos del sistema y que son: 0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F.

Las letras A, B, C, D, E y F son símbolos empleados en el sistema hexadecimal para representar las cantidades del sistema decimal 10, 11, 12, 13, 14 y 15 respectivamente.

Este sistema se utiliza para trabajar en binario con mayor sencillez ya que cada cifra hexadecimal equivale a cuatro dígitos binarios, tal como se indica en la siguiente tabla:

HEXADECIMAL	BINARIO
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001
A	1010
B	1011
C	1100
D	1101
E	1110
F	1111

4. CONVERSIONES.

Se denomina conversión a la transformación de una determinada cantidad expresada en un sistema de numeración en su representación equivalente en otro sistema.

Por lo general, para pasar de un sistema de numeración a otro es necesario pasar previamente por la base 10. Por ejemplo, para convertir un número en base 3 a base 7 el proceso a seguir sería:

$$\text{N}^\circ \text{ EN BASE 3} \xrightarrow{\text{CONVERTIR}} \text{N}^\circ \text{ EN BASE 10} \xrightarrow{\text{CONVERTIR}} \text{N}^\circ \text{ EN BASE 7}$$

En el caso de que una de las bases sea una potencia de la otra, la conversión puede hacerse de forma directa, sin pasar por la base 10. Por ejemplo, para convertir un número en base 16 a base 2, o viceversa, el proceso sería directo ya que $16 = 2^4$.

$$\text{N}^\circ \text{ EN BASE 2} \xleftrightarrow{\text{CONVERTIR}} \text{N}^\circ \text{ EN BASE 16}$$

4.1. CONVERSIÓN DE DECIMAL A OTRO SISTEMA DE NUMERACIÓN.

Para convertir números enteros decimales a otra base, la forma más simple es dividir sucesivamente el número decimal y los cocientes que se van obteniendo por la nueva base, hasta que el cociente en una de las divisiones se haga 0. La unión de todos los restos obtenidos en orden inverso nos proporciona el número inicial expresado en el nuevo sistema.

EJEMPLO.

Convertir el número decimal 158 a:

- a) Sistema binario.
- b) Sistema octal.
- c) Sistema hexadecimal.

a) Para convertir el número 158 a binario se ha de dividir el número y sus cocientes por 2.

$$\begin{array}{r} 158 \mid 2 \\ \hline 0 \quad 79 \mid 2 \\ \hline 1 \quad 39 \mid 2 \\ \hline 1 \quad 19 \mid 2 \\ \hline 1 \quad 9 \mid 2 \\ \hline 1 \quad 4 \mid 2 \\ \hline 0 \quad 2 \mid 2 \\ \hline 0 \quad 1 \mid 2 \\ \hline 1 \quad 0 \end{array}$$

El número resultante es:

$$158_{(10)} = 10011110_{(2)}$$

b) Para convertir el número 158 a octal se ha de dividir el número y sus cocientes por 8.

$$\begin{array}{r|l} 158 & 8 \\ \hline 6 & 19 \\ \hline & 3 & 2 \\ \hline & & 2 & 0 \end{array}$$

El número resultante es:

$$158_{(10)} = 236_{(8)}$$

c) Para convertir el número 158 a hexadecimal se ha de dividir el número y sus cocientes por 16.

$$\begin{array}{r|l} 158 & 16 \\ \hline 14 & 9 \\ \hline & 9 & 0 \end{array}$$

El número resultante es:

$$158_{(10)} = 9E_{(16)}$$

Como se puede observar, el número decimal 14 se sustituye por su símbolo equivalente en el sistema hexadecimal, que es el símbolo E.

Para convertir una fracción decimal a otro sistema de numeración, la forma más simple consiste en multiplicar dicha fracción por la nueva base, obteniendo en la parte entera del resultado el primero de los dígitos de la fracción que buscamos. A continuación se repite el proceso con la parte fraccionaria del resultado anterior, obteniendo en la parte entera del nuevo resultado el segundo de los dígitos buscados. Iteramos sucesivamente de esta forma, hasta que desaparezca la parte fraccionaria o hasta que se tengan los suficientes dígitos que nos permitan no sobrepasar un determinado error.

EJEMPLO.

Convertir los números decimales 0.3125 y 0.537 a:

- Sistema binario.
- Sistema octal.
- Sistema hexadecimal.

a) Para convertir el número 0.3125 a binario se ha de multiplicar el número por 2.

$$0.3125 \times 2 = 0.625$$

$$0.625 \times 2 = 1.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

El número resultante es:

$$0.3125_{(10)} = 0.0101_{(2)}$$

Para convertir el número 0.537 repetimos el proceso.

$$0.537 \times 2 = 1.074$$

$$0.074 \times 2 = 0.148$$

$$0.148 \times 2 = 0.296$$

$$0.296 \times 2 = 0.592$$

$$0.592 \times 2 = 1.184$$

$$0.184 \times 2 = 0.368$$

$$0.368 \times 2 = 0.736$$

$$0.736 \times 2 = 1.472$$

$$0.472 \times 2 = 0.944$$

$$0.944 \times 2 = 1.888$$

$$0.888 \times 2 = 1.776$$

$$0.776 \times 2 = 1.552$$

$$0.552 \times 2 = 1.104$$

$$0.104 \times 2 = 0.208$$

$$0.208 \times 2 = 0.416$$

$$0.416 \times 2 = 0.832$$

$$0.832 \times 2 = 1.664$$

⋮

El número resultante es:

$$0.537_{(10)} = 0.10001001011110001_{(2)}$$

b) Para convertir el número 0.3125 a octal se ha de multiplicar el número por 8.

$$0.3125 \times 8 = 2.5$$

$$0.5 \times 8 = 4.0$$

El número resultante es:

$$0.3125_{(10)} = 0.24_{(8)}$$

Para convertir el número 0.537 repetimos el proceso.

$$0.537 \times 8 = 4.296$$

$$0.296 \times 8 = 2.368$$

$$0.368 \times 8 = 2.944$$

$$0.944 \times 8 = 7.552$$

$$0.552 \times 8 = 4.416$$

$$0.416 \times 8 = 3.328$$

$$0.328 \times 8 = 2.624$$

$$0.624 \times 8 = 4.992$$

$$0.992 \times 8 = 7.936$$

$$0.936 \times 8 = 7.488$$

⋮

El número resultante es:

$$0.537_{(10)} = 0.4227432477_{(8)}$$

c) Para convertir el número 0.3125 a hexadecimal se ha de multiplicar el número por 16.

$$0.3125 \times 16 = 5.0$$

El número resultante es:

$$0.3125_{(10)} = 0.5_{(16)}$$

Para convertir el número 0.537 repetimos el proceso.

$$0.537 \times 16 = 8.592$$

$$0.592 \times 16 = 9.472$$

$$0.472 \times 16 = 7.552$$

$$0.552 \times 16 = 8.832$$

$$0.832 \times 16 = 13.312$$

$$0.312 \times 16 = 4.992$$

$$0.992 \times 16 = 15.872$$

$$0.872 \times 16 = 13.952$$

$$0.952 \times 16 = 15.232$$

$$0.232 \times 16 = 3.712$$

⋮

El número resultante es:

$$0.537_{(10)} = 0.8978D4FDF3_{(16)}$$

Como se puede observar, las partes enteras mayores de 9 se han sustituido por sus símbolos correspondientes del sistema hexadecimal.

Para convertir un número decimal con parte entera y fraccionaria a otro sistema de numeración, se ha de separar la parte entera y la fraccionaria, convertir cada una de ellas por separado y volverlas a juntar para formar el número en la nueva base.

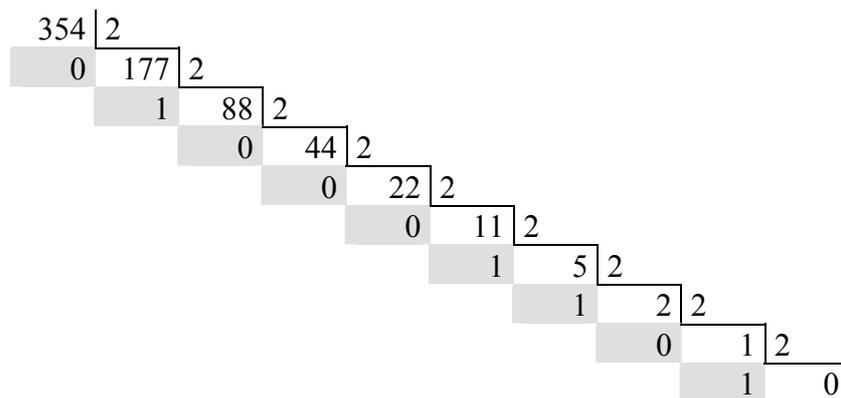
EJEMPLO.

Convertir el número decimal 354.625 a:

- a) Sistema binario.
- b) Sistema octal.
- c) Sistema hexadecimal.

a) Para convertir el número 354.625 a binario se separa la parte entera y la fraccionaria y se convierten por separado.

- Conversión de la parte entera : 354.



La parte entera es:

$$354_{(10)} = 101100010_{(2)}$$

- Conversión de la parte fraccionaria : 0.625.

$$0.625 \times 2 = 1.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1.0$$

La parte fraccionaria es:

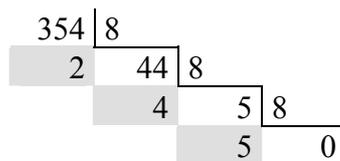
$$0.625_{(10)} = 0.101_{(2)}$$

El número resultante es:

$$354.625_{(10)} = 101100010.101_{(2)}$$

b) Para convertir el número 354.625 a octal se separa la parte entera y la fraccionaria y se convierten por separado.

- Conversión de la parte entera : 354.



La parte entera es:

$$354_{(10)} = 542_{(8)}$$

- Conversión de la parte fraccionaria : 0.625.

$$0.625 \times 8 = 5$$

La parte fraccionaria es:

$$0.625_{(10)} = 0.5_{(8)}$$

El número resultante es:

$$354.625_{(10)} = 542.5_{(8)}$$

c) Para convertir el número 354.625 a hexadecimal se separa la parte entera y la fraccionaria y se convierten por separado.

- Conversión de la parte entera : 354.

$$\begin{array}{r}
 354 \overline{)16} \\
 \underline{2} \\
 22 \\
 \underline{6} \\
 16 \\
 \underline{1} \\
 0
 \end{array}$$

La parte entera es:

$$354_{(10)} = 162_{(16)}$$

- Conversión de la parte fraccionaria : 0.625.

$$0.625 \times 16 = 10$$

La parte fraccionaria es:

$$0.625_{(10)} = 0.A_{(16)}$$

El número resultante es:

$$354.625_{(10)} = 162.A_{(16)}$$

4.2. CONVERSIÓN DE UN SISTEMA DE NUMERACIÓN A SISTEMA DECIMAL.

Para convertir un número expresado en cualquier sistema de numeración al sistema decimal solo es necesario aplicar el **Teorema Fundamental de la Numeración**.

EJEMPLO.

Convertir a sistema decimal los siguientes números:

- 110011.1101₍₂₎.
- 1526.774₍₈₎.
- 4F24.3AB₍₁₆₎.

- Para convertir un número binario aplicamos el Teorema Fundamental de la Numeración utilizando como base el 2.

$$\begin{aligned}
N^\circ &= \sum_{i=-4}^5 X_i \times 2^i = 1 \times 2^{-4} + 0 \times 2^{-3} + 1 \times 2^{-2} + 1 \times 2^{-1} + 1 \times 2^0 + 1 \times 2^1 + 0 \times 2^2 + 0 \times 2^3 + 1 \times 2^4 + 1 \times 2^5 = \\
&= 1 \times \frac{1}{2^4} + 1 \times \frac{1}{2^2} + 1 \times \frac{1}{2^1} + 1 \times 2^0 + 1 \times 2^1 + 1 \times 2^4 + 1 \times 2^5 = \\
&= 1 \times \frac{1}{16} + 1 \times \frac{1}{4} + 1 \times \frac{1}{2} + 1 \times 1 + 1 \times 2 + 1 \times 16 + 1 \times 32 = 51.8125_{(10)}
\end{aligned}$$

El número resultante es:

$$110011.1101_{(2)} = 51.8125_{(10)}$$

b) Para convertir un número octal aplicamos el Teorema Fundamental de la Numeración utilizando como base el 8.

$$\begin{aligned}
N^\circ &= \sum_{i=-3}^3 X_i \times 8^i = 4 \times 8^{-3} + 7 \times 8^{-2} + 7 \times 8^{-1} + 6 \times 8^0 + 2 \times 8^1 + 5 \times 8^2 + 1 \times 8^3 = \\
&= 4 \times \frac{1}{8^3} + 7 \times \frac{1}{8^2} + 7 \times \frac{1}{8^1} + 6 \times 8^0 + 2 \times 8^1 + 5 \times 8^2 + 1 \times 8^3 = \\
&= 4 \times \frac{1}{512} + 7 \times \frac{1}{64} + 7 \times \frac{1}{8} + 6 \times 1 + 2 \times 8 + 5 \times 64 + 1 \times 512 = 854.9921875_{(10)}
\end{aligned}$$

El número resultante es:

$$1526.774_{(8)} = 854.9921875_{(10)}$$

c) Para convertir un número hexadecimal aplicamos el Teorema Fundamental de la Numeración utilizando como base el 16.

$$\begin{aligned}
N^\circ &= \sum_{i=-3}^3 X_i \times 16^i = B \times 16^{-3} + A \times 16^{-2} + 3 \times 16^{-1} + 4 \times 16^0 + 2 \times 16^1 + F \times 16^2 + 4 \times 16^3 = \\
&= 11 \times \frac{1}{16^3} + 10 \times \frac{1}{16^2} + 3 \times \frac{1}{16^1} + 4 \times 16^0 + 2 \times 16^1 + 15 \times 16^2 + 4 \times 16^3 = \\
&= 11 \times \frac{1}{4096} + 10 \times \frac{1}{256} + 3 \times \frac{1}{16} + 4 \times 1 + 2 \times 16 + 15 \times 256 + 4 \times 4096 = 20260.229248046875_{(10)}
\end{aligned}$$

El número resultante es:

$$4F24.3AB_{(16)} = 20260.229248046875_{(10)}$$

4.3. CONVERSIÓN ENTRE LOS SISTEMAS BINARIO Y OCTAL.

La conversión de números octales a binarios consiste en sustituir cada cifra octal por sus tres dígitos binarios correspondientes.

EJEMPLO.

Convertir el número octal 314.625 al sistema binario.

Para realizar la conversión, sustituimos cada cifra octal por tres dígitos binarios

3	1	4	.	6	2	5
011	001	100	.	110	010	101

El número resultante es:

$$314.625_{(8)} = 11001100.110010101_{(2)}$$

La conversión de números binarios a octales consiste en agrupar los dígitos binarios de tres en tres y sustituirlos por su cifra correspondiente en el sistema octal. La agrupación de los dígitos binarios se ha de hacer teniendo en cuenta las siguientes reglas:

- ♦ La parte entera se agrupa a partir del punto decimal hacia la izquierda. Si en la última agrupación hay menos de 3 bits, se añaden tantos ceros a la izquierda como sean necesarios para tener 3 bits.
 - ♦ La parte fraccionaria se agrupa a partir del punto decimal hacia la derecha. Si en la última agrupación hay menos de 3 bits, se añaden tantos ceros a la derecha como sean necesarios para tener 3 bits.
-

EJEMPLO.

Convertir el número binario 1101111001.1010111 al sistema octal.

Para realizar la conversión, agrupamos los bits de 3 en 3 y sustituimos cada grupo por su cifra octal correspondiente.

001	101	111	001	.	101	011	100
1	5	7	1	.	5	3	4

El número resultante es:

$$1101111001.1010111_{(2)} = 1571.534_{(8)}$$

4.4. CONVERSIÓN ENTRE LOS SISTEMAS BINARIO Y HEXADECIMAL.

La conversión de números hexadecimales a binarios consiste en sustituir cada cifra hexadecimal por sus cuatro dígitos binarios correspondientes.

EJEMPLO.

Convertir el número hexadecimal 5A94.6B6 al sistema binario.

Para realizar la conversión, sustituimos cada cifra hexadecimal por cuatro dígitos binarios

5	A	9	4	.	6	B	6
0101	1010	1001	0100	.	0110	1011	0110

El número resultante es:

$$5A94.6B6_{(16)} = 101101010010100.011010110110_{(2)}$$

La conversión de números binarios a hexadecimales consiste en agrupar los dígitos binarios de cuatro en cuatro y sustituirlos por su cifra correspondiente en el sistema hexadecimal. La agrupación de los dígitos binarios se ha de hacer teniendo en cuenta las siguientes reglas:

- ♦ La parte entera se agrupa a partir del punto decimal hacia la izquierda. Si en la última agrupación hay menos de 4 bits, se añaden tantos ceros a la izquierda como sean necesarios para tener 4 bits.
 - ♦ La parte fraccionaria se agrupa a partir del punto decimal hacia la derecha. Si en la última agrupación hay menos de 4 bits, se añaden tantos ceros a la derecha como sean necesarios para tener 4 bits.
-

EJEMPLO.

Convertir el número binario 1101111001.1010111 al sistema hexadecimal.

Para realizar la conversión, agrupamos los bits de 4 en 4 y sustituimos cada grupo por su cifra hexadecimal correspondiente.

0011	0111	1001	.	1010	1110
3	7	9	.	A	E

El número resultante es:

$$1101111001.1010111_{(2)} = 379.AE_{(16)}$$

5. REPRESENTACION DE NÚMEROS ENTEROS CON SIGNO.

Para poder trabajar con números negativos es necesario utilizar otro sistema de representación en binario. Las cuatro formas habituales que se utilizan para estas representaciones son:

- Módulo y signo.
- Complemento a 1.
- Complemento a 2.
- Exceso a 2 elevado a $N - 1$ (exceso 2^{N-1})

5.1. MÓDULO Y SIGNO (MS).

En este sistema de representación, el bit que está situado más a la izquierda representa el signo, y su valor es 0 si el número es positivo y 1 si el número es negativo. El resto de bits ($N - 1$) representan el módulo del número.

EJEMPLO.

Representar con 8 bits ($N = 8$) los números 25 y -25.

La representación del módulo sería la misma ($25 = 0011001$) y solo varía el bit situado más a la izquierda en función del signo.

$$N^{\circ} +25 \rightarrow 00011001$$

$$N^{\circ} -25 \rightarrow 10011001$$

Se denomina **rango de representación** en un sistema al conjunto de números representables en el mismo. Para el sistema MS, el rango de representación es:

$$-2^{N-1} + 1 \leq X \leq 2^{N-1} - 1$$

siendo N el número de bits.

En la siguiente tabla se representan algunos rangos en función del número de bits.

N (BITS)	RANGO DE REPRESENTACIÓN
8	$-127 \leq X \leq 127$
16	$-32767 \leq X \leq 32767$
32	$-2147483647 \leq X \leq 2147483647$

La ventaja que representa este sistema frente a otros es la de poseer rango simétrico, mientras que su mayor inconveniente es el de tener dos representaciones para el 0. Para 8 bits el 0 se representa:

00000000	(+0)
10000000	(-0)

5.2. COMPLEMENTO A 1 (C-1).

Este sistema de representación utiliza el bit situado más a la izquierda para el signo, correspondiendo el 0 para el signo + y el 1 para el signo -. Para los números positivos, los N - 1 bits de la derecha representan el módulo, y el negativo de un número se obtiene complementando todos sus dígitos (cambiando ceros por unos y viceversa), incluido el bit de signo.

EJEMPLO.

Representar con 8 bits (N = 8) los números 25 y -25.

La representación del módulo del número positivo será el binario del número 25 = 0011001 y para el número negativo se han de complementar todos los bits.

$$N^{\circ} +25 \rightarrow 00011001$$
$$N^{\circ} -25 \rightarrow 11100110$$

El rango de representación es el mismo que para el MS, o sea:

$$-2^{N-1} + 1 \leq X \leq 2^{N-1} - 1$$

siendo N el número de bits.

La ventaja que representa este sistema es la de poseer rango simétrico, mientras que su mayor inconveniente es el de tener dos representaciones para el 0. Para 8 bits el 0 se representa:

00000000	(+0)
11111111	(-0)

5.3. COMPLEMENTO A 2 (C-2).

Se trata de un método muy utilizado. El complemento a 2 de un número se obtiene restándole de 2^n , siendo n el número de bits del módulo ($n = N - 1$).

Este sistema utiliza el bit situado más a la izquierda para el signo, correspondiendo el 0 para el signo + y el 1 para el signo -. Para los números positivos, los $N - 1$ bits de la derecha representan el módulo y el negativo de un número se obtiene en dos pasos:

1. Se complementa el número positivo en todos sus bits, incluido el bit de signo.
2. Al resultado obtenido se le suma 1 (en binario), despreciando el último acarreo si existe.

EJEMPLO.

Representar con 8 bits ($N = 8$) los números 25 y -25.

La representación del módulo del número positivo será el binario del número $25 = 0011001$ y para el número negativo se han de complementar todos los bits y sumarle 1 en binario.

$$N^{\circ} +25 \rightarrow 00011001$$

$$\begin{array}{r} N^{\circ} -25 \rightarrow 11100110 \\ + \quad \quad 1 \\ \hline 11100111 \end{array}$$

El rango de representación en este caso es asimétrico, lo que representa su mayor inconveniente, y viene dado por:

$$-2^{N-1} \leq X \leq 2^{N-1} - 1$$

siendo N el número de bits.

En la siguiente tabla se representan algunos rangos en función del número de bits.

N (BITS)	RANGO DE REPRESENTACIÓN
8	$-128 \leq X \leq 127$
16	$-32768 \leq X \leq 32767$
32	$-2147483648 \leq X \leq 2147483647$

La principal ventaja es el tener una única representación del 0. Para 8 bits el 0 se representa:

$$\begin{array}{ll} 00000000 & (+0) \\ 00000000 & (-0) \end{array}$$

5.4. EXCESO A 2 ELEVADO A N – 1 (EXCESO 2^{N-1}).

Este sistema no utiliza bit para el signo, con lo cual todos los bits representan un módulo que corresponde al número representado más el exceso, que para N bits viene dado por 2^{N-1} .

EJEMPLO.

Representar con 8 bits ($N = 8$) los números 25 y -25.

Con 8 bits el exceso es $2^{8-1} = 128$, con lo que el número 25 vendrá representado por $25 + 128 = 153$, y para el -25 se tendrá que representar el número $-25 + 128 = 103$. Por lo tanto, la representación de los números es:

$$N^{\circ} + 25 \rightarrow 10011001 \quad (153 \text{ en binario})$$

$$N^{\circ} - 25 \rightarrow 01100111 \quad (103 \text{ en binario})$$

El rango de representación en este caso es asimétrico, lo que representa su mayor inconveniente, y viene dado por:

$$-2^{N-1} \leq X \leq 2^{N-1} - 1$$

siendo N el número de bits.

En la siguiente tabla se representan algunos rangos en función del número de bits.

N (BITS)	RANGO DE REPRESENTACIÓN
8	$-128 \leq X \leq 127$
16	$-32768 \leq X \leq 32767$
32	$-2147483648 \leq X \leq 2147483647$

El 0 tiene una única representación que para 8 bits es:

$$\text{Número } 0 \text{ (} 0 + 128 \text{)} \quad 10000000$$

Resulta interesante observar que todo número representado en exceso es igual a su correspondiente representación en C-2 con el primer dígito a la izquierda cambiado.

6. REPRESENTACION DE NÚMEROS EN COMA FLOTANTE.

La coma flotante es el sistema utilizado para la representación de números reales con un rango de representación mayor que el que ofrece la representación en coma fija, permitiendo el tratamiento de números muy grandes y muy pequeños.

La representación de números en coma flotante se basa en la representación en notación científica, en la que una cantidad se representa de la forma:

$$N^{\circ} = m \times b^c$$

donde:

- N° = Número representado.
- m = Mantisa (es una fracción con signo).
- b = Base de exponenciación o raíz.
- c = Exponente o característica (número entero con signo).

Un número en esta notación tiene infinitas representaciones, de las que se toma como representación la **normalizada**, que es aquella en la que la mantisa no tiene parte entera y la primera cifra o dígito a la derecha de la coma es distinta de 0, salvo en la representación del número 0.

EJEMPLO.

Representación normalizada del número 123.4 con:

- a) Base 10.
- b) Base 2.

a) Para representar el número 123.4 en notación científica, se va dividiendo por 10 hasta que la parte entera sea igual a 0.

$$123.4 = 123.4 \times 10^0 = 12.34 \times 10^1 = 1.234 \times 10^2 = 0.1234 \times 10^3$$

Por lo tanto, la representación normalizada es:

$$123.4 = 0.1234 \times 10^3$$

b) Se sigue el mismo procedimiento pero dividiendo por 2, hasta que la parte entera sea igual a 0.

$$123.4 = 123.4 \times 2^0 = 61.7 \times 2^1 = 30.85 \times 2^2 = 15.425 \times 2^3 = 7.7125 \times 2^4 = 3.85625 \times 2^5 = 1.928125 \times 2^6 = 0.9640625 \times 2^7$$

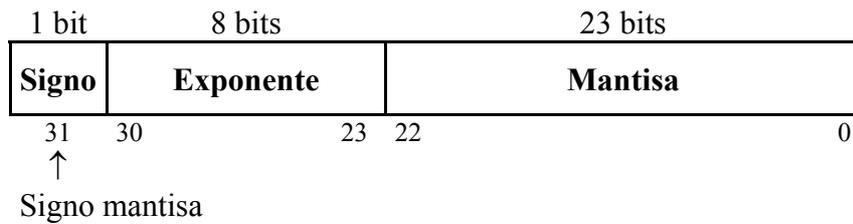
Por lo tanto, la representación normalizada es:

$$123.4 = 0.9640625 \times 2^7$$

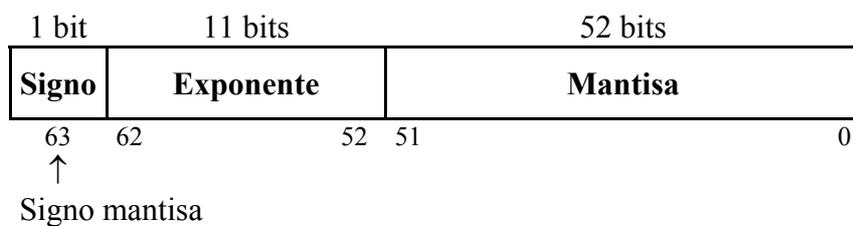
Existen muchos formatos de representación en coma flotante, variando la longitud de la palabra, la base de exponenciación, el número de bits reservados para la mantisa y para el exponente, el sistema de representar la mantisa y el exponente, etc.

Cada fabricante define para la arquitectura de sus máquinas un formato de coma flotante para lo que se denomina **simple precisión**, donde el número de bits corresponde a una palabra, y otro para la **doble precisión**, donde se utiliza una doble palabra.

- Para simple precisión (en máquinas de 32 bits de palabra).



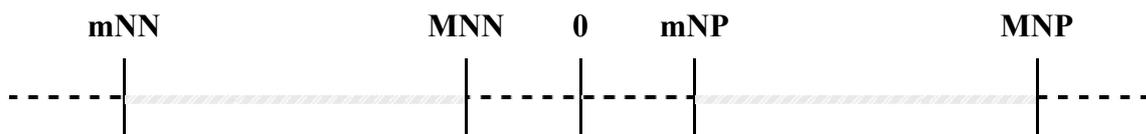
- Para doble precisión (doble palabra de 64 bits).



Para la representación se suelen seguir, en general, las siguientes reglas:

- El exponente se representa en las notaciones MS o exceso 2^{N-1} , siendo un número entero con signo.
- La mantisa es un número real normalizado (sin parte entera) con la coma implícita a la izquierda representada en MS, C-1 o C-2. El signo suele estar situado en el bit de más a la izquierda.
- La base de exponenciación, también denominada raíz, es una potencia de 2 determinada por el fabricante (2, 8 o 16).

El rango de representación en la coma flotante tiene la forma:



Donde:

- mNN es el mínimo número negativo que podemos representar y que se corresponde con

$$mNN = - \text{máxima mantisa} \times \text{base}^{\text{máximo exponente}}$$

- MNN es el máximo número negativo, que se corresponde con

$$MNN = - \text{mínima mantisa} \times \text{base}^{-\text{máximo exponente}}$$

- mNP es el mínimo número positivo, que se corresponde con

$$mNP = \text{mínima mantisa} \times \text{base}^{-\text{máximo exponente}}$$

- MNP es el máximo número positivo, que se corresponde con

$$MNP = \text{máxima mantisa} \times \text{base}^{\text{máximo exponente}}$$

En esta representación, además de existir números no representables entre dos cualesquiera, existen cuatro zonas en las que no se pueden representar los números existentes en ellas. Estas zonas son:

- Desbordamiento positivo: números mayores que MNP.
- Subdesbordamiento positivo: números entre 0 y mNP.
- Subdesbordamiento negativo: números entre MNN y 0.
- Desbordamiento negativo: números menores que mNN.

EJEMPLO.

Un ordenador de 32 bits utiliza el siguiente formato para representar números en coma flotante con simple precisión:

- Los bits del 23 al 30 para el exponente, que se representa en exceso a 128.
- Los bits de 0 al 22 para la mantisa normalizada, que se representa en C-1 con el punto a la izquierda.
- El bit 31 para el signo de la mantisa.
- El 0 se representa con todos los bits a 0.

- a) Representar el número 12.5.
- b) Representar el número -12.5
- c) Representar el 0.

- a) En primer lugar, hay que normalizar el número 12.5 con base 2.

$$12.5 \times 2^0 = 6.25 \times 2^1 = 3.125 \times 2^2 = 1.5625 \times 2^3 = 0.78125 \times 2^4$$

Pasamos el exponente 4 a exceso 128.

$$\text{Exponente} = 4 + 128 = 132 = 10000100$$

El signo de la mantisa es positivo, por tanto el bit 31 es igual a 0.

Pasamos la mantisa 0.78125 a C-1:

$$0.78125 \times 2 = 1.5625$$

$$0.5625 \times 2 = 1.125$$

$$0.125 \times 2 = 0.25$$

$$0.25 \times 2 = 0.5$$

$$0.5 \times 2 = 1$$

Por lo tanto, $0.78125 = 0.11001$

La representación del número es:

0	1000 0100	110 0100 0000 0000 0000 0000
---	-----------	------------------------------

b) El exponente es el mismo del caso anterior y la mantisa se pasa a C-1. Por lo tanto:

1	1000 0100	001 1011 1111 1111 1111 1111
---	-----------	------------------------------

c) La representación del 0 será:

1	0000 0000	000 0000 0000 0000 0000 0000
---	-----------	------------------------------

EJEMPLO.

Dado el formato estándar IEEE P754 de simple precisión, se pide:

- a) Representar en coma flotante el número 4568.375
- b) Determinar que número es el representado en coma flotante por:

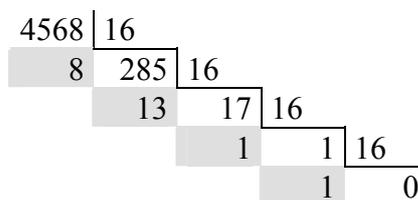
1	0011 0111	011 0111 1001 1110 1001 0000
---	-----------	------------------------------

- c) Determinar que número es el representado en coma flotante por:

0	0000 0000	000 0000 0000 0010 0000 0000
---	-----------	------------------------------

- a) En primer lugar hay que representar en binario el número 4568.375. Una forma de hacerlo es convertirlo en hexadecimal y posteriormente a binario.

- Parte entera



- Parte fraccionaria.

$$0.375 \times 16 = 6$$

El número resultante es:

$$4568.375_{(10)} = 11C8.6_{(16)} = 1000111011000.011_{(2)}$$

A continuación se normaliza el número al formato 1.M:

$$1000111011000.011 \times 2^0 = 1.000111011000011 \times 2^{12}$$

La mantisa, de 23 bits, es:

$$M = 000\ 1110\ 1100\ 0011\ 0000\ 0000$$

La codificación del exponente se realiza en exceso 127, siendo su valor:

$$E' = E + 127 = 12 + 127 = 139 = 10001011$$

El número es positivo, por lo que $S = 0$.

El número codificado es:

0	1000 1011	000 1110 1100 0011 0000 0000
---	-----------	------------------------------

b) El bit S vale 1, luego el número es negativo. Su valor es:

$$\begin{aligned}\text{Valor} &= -1.M \times 2^{E-127} = \\ &= -1.0110111100111101001 \times 2^{00110111-127} = \\ &= -1.0110111100111101001 \times 2^{55-127} = \\ &= -1.0110111100111101001 \times 2^{-72} = \\ &= -1.0110111100111101001 \times 2^{-91} = \\ &= -3.037728 \times 10^{-43}\end{aligned}$$

c) El bit S vale 0, lo cual supone que el número es positivo. Por otra parte el campo $E = 0$, por lo que se trata del caso especial para representar números próximos a 0:

$$\text{Valor} = +0.M \times 2^{-126} = 0.000000000000010000000000 \times 2^{-126} = 1 \times 2^{-140} = 7.174648 \times 10^{-43}$$
