

Capítulo 4

Como se ha visto en el capítulo anterior, los números fraccionarios en binario son equivalentes al sistema decimal, o sea, después de la coma los símbolos numéricos se multiplican con la base del sistema numérico, elevado a la potencia negativa de acuerdo con la posición dentro de la cifra.

Ese formato se denomina punto fijo, o sea la coma (o el punto, de acuerdo con la denominación inglesa), separa los enteros de los fraccionarios.

Este formato no se utiliza en las computadoras, sino que se utiliza un formato similar al formato de notación científica.

Notación científica.

La notación científica se forma de varios campos. Veamos:

Signo	Primera cifra distinta de cero Primer entero	Mantisa	Base numérica	Exponente incluyendo el signo
-------	---	---------	---------------	-------------------------------

Ejemplo:

1. Velocidad de la luz en el vacío = 299 792 458 m/s o también $2,997\ 924\ 58 \times 10^{+8}$ m/s

+	2	,99 792 458	10	+8
---	---	-------------	----	----

2. Constante de Avogadro = $6,022\ 140\ 857 \times 10^{+23}$ mol

+	6	,022 140 857	10	+23
---	---	--------------	----	-----

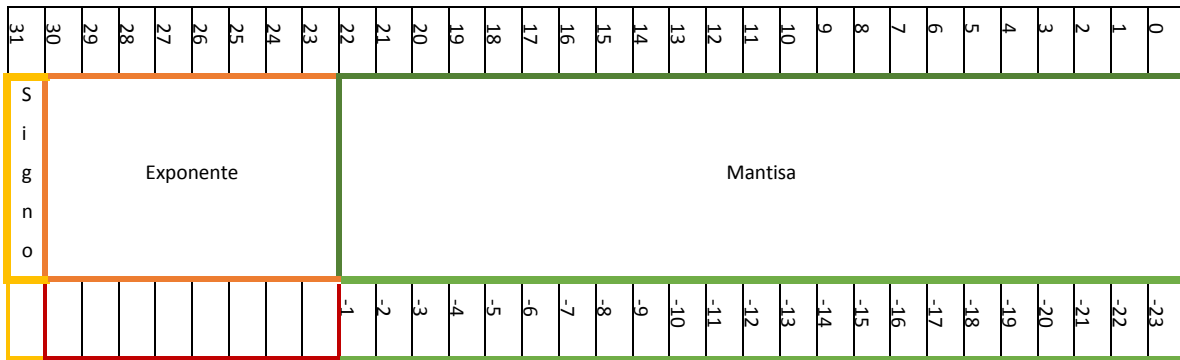
3. Masa atómica relativa del electrón ($A_r(e)$) = $5,485\ 799\ 0943 \times 10^{-4}$

+	5	,485 799 094 3	10	-4
---	---	----------------	----	----

En computación se utiliza un formato similar para representar números fraccionarios y ese formato está normalizado con la norma IEEE 754.

Se la conoce también como punto flotante, y puede tener 32, o 64 bits. Solo vamos a ver la más sencilla (32 bits)

Conversión de float (IEEE 754) a decimal fraccionario



Determinar qué número en decimal representa en el formato IEEE 754 la siguiente combinación.

Resolución:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0				
0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0

1. Separar el número binario en los campos Signo, Exponente y Mantisa (ver recuadros gruesos)
2. El campo **Signo** tiene un 0, por lo que el valor es positivo.
3. El campo **Exponente (1000001)** tiene el valor 129 (en decimal), se le resta 127 (binario desplazado), donde resulta **Exponente = +2**
4. La mantisa queda con 23 bits, que se multiplican por las potencias negativas de 2, desde 2^{-1} hasta 2^{-23} (ver tabla)

0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

5. Se suman todos los valores, (ver tabla que sigue), y se le suma 1 (uno implícito)
6. Ese valor final se multiplica por 2^n , siendo n el valor del exponente con su signo, y queda $1,049999952316280 \times 2^{+2} = 1,049999952316280 \times 4 = +4,199999809265130 \approx +4,2$.

O sea, el +4,2 en decimal se representa en **float** como:

0	1	0	0	0	0	0	0	1	0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0
---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---	---

Tabla de Potencias Negativas																						
0	0	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0	0	1	1	0



-23	0,00000011920928955078125	0	0
-22	0,0000002384185791015625	1	2,38419E-07
-21	0,000000476837158203125	1	4,76837E-07
-20	0,00000095367431640625	0	0
-19	0,0000019073486328125	0	0
-18	0,000003814697265625	1	3,8147E-06
-17	0,00000762939453125	1	7,62939E-06
-16	0,0000152587890625	0	0
-15	0,000030517578125	0	0
-14	0,00006103515625	1	6,10352E-05
-13	0,0001220703125	1	0,00012207
-12	0,000244140625	0	0
-11	0,00048828125	0	0
-10	0,0009765625	1	0,000976563
-9	0,001953125	1	0,001953125
-8	0,00390625	0	0
-7	0,0078125	0	0
-6	0,015625	1	0,015625
-5	0,03125	1	0,03125
-4	0,0625	0	0
-3	0,125	0	0
-2	0,25	0	0
-1	0,50	0	0

0,049999952316280

0	1,0	1	1,00
---	-----	---	------

1,049999952316280

Conversión de decimal fraccionario a formato float (IEEE 754)

Supongamos el caso anterior ... +4,2

1. Se separa enteros y decimales y se procede como se hizo en el capítulo anterior para pasar a binario fraccionario (punto fijo..).
 - a. $4 = 100$
 - b. $0,2 = 0,001100110011001100.....$

2. Se unen los dos términos 100,00110011001100110011... hasta completar 24 bits desde el primer 1 (viendo desde la izquierda...)
3. Se corre la coma hacia el primer 1, en este caso hacia la izquierda; la cantidad de desplazamientos determina la potencia del exponente:
 - a. $1,00001100110011001100110 \times 2^{+2}$
4. Se descarta la parte entera y queda 0,00001100110011001100110, y los bits después de la coma corresponden a la mantisa
 - a. Mantisa = 00001100110011001100110
5. Al exponente +2 se le suma 127, y queda 129
6. Ese valor pasa a binario en 8 bits $129 = 10000001$, y ya tenemos el exponente.
7. Finalmente, como definimos que ese número es positivo (+ 4,2), el bit de signo es 0
8. Uniendo los campos queda:

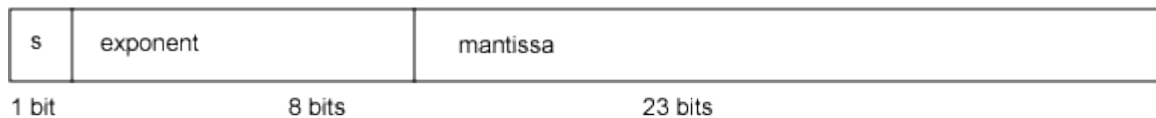
0 10000001 00001100110011001100110 (32 bits...)

Representación

Simplificando a dos modalidades la norma define dos resoluciones posibles para los números.

Simple precisión (32 bits) y doble precisión (64 bits).

IEEE Floating Point Representation



IEEE Double Precision Floating Point Representation



Formato de representación para el estándar IEEE 754 (en simple y doble precisión).

Rango de representación

El rango de representación del formato float es, a nivel de módulo, desde el módulo más pequeño $1,17549 \times 10^{-38}$ hasta el módulo más grande $3,40282 \times 10^{+38}$

El rango de representación del formato double float es, a nivel de módulo, desde el módulo más pequeño $4,4501477170144 \times 10^{-308}$ hasta el módulo más grande $1,79769313486232 \times 10^{+308}$