Representación Numérica en Coma Fija y Coma Flotante

Sistema de Procesamiento de Datos - UTN

Prof. TUP Acierno, German
Prof. DI Guillermo Gimenez
Prof. Lic Verónica Lourdes Tomich

Representación Numérica en Coma Fija y Coma Flotante

Estas representaciones son utilizadas por las computadoras para procesar cálculos numéricos con formatos grandes. Consiste en una cadena de bits que guardan relación con la notación científica, y pueden representar números enteros y números reales tanto negativos como positivos.

Representación Numérica en Coma Fija y Coma Flotante

Los formatos más conocidos son la coma fija y la coma flotante, también denominados punto fijo y punto flotante respectivamente. Se utiliza en ambos caso para representar **Signo-Magnitud**.

Punto Fijo o Coma Fija

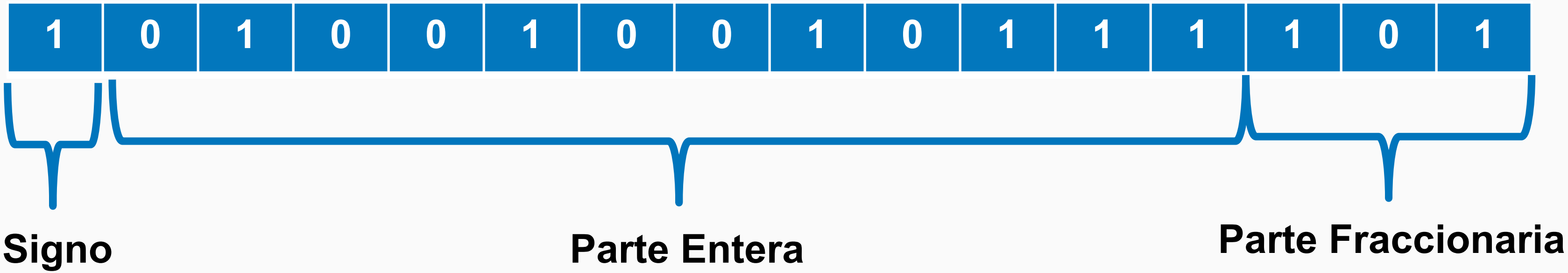
Hasta ahora hemos representado números enteros y fraccionarios separando la parte entera de la fraccionaria mediante una coma. En la representación interna del ordenador, la coma no existe, simplemente sabemos donde está, pero no se usa ningún bit ni ninguna línea eléctrica para representarla.

Punto Fijo o Coma Fija

Por ejemplo, si hemos reservado 2 bits para la parte fraccionaria, sabemos que la secuencia de unos y ceros "101011001" equivale al número "$1010110{,}01_2$".

A esta forma de entender las representaciones la llamamos Numeración en punto fijo, ya que la coma fraccionaria (un punto en la numeración americana), ocupa una posición fija, sea cual sea.

Representación Numérica en Punto Fijo o Coma Fija



Si consideramos **N = 16**, siendo el primer bit el que codifica el signo, los siguientes **12 los de la parte entera**, y los 3 restantes la parte decimal

Representación Numérica en Punto Fijo o Coma Fija

1	0	1	0	0	1	0	0	1	0	1	1	1	1	0	1

Signo **Parte Entera** **Parte Fraccionaria**

Resulta que la representación anterior significaría:

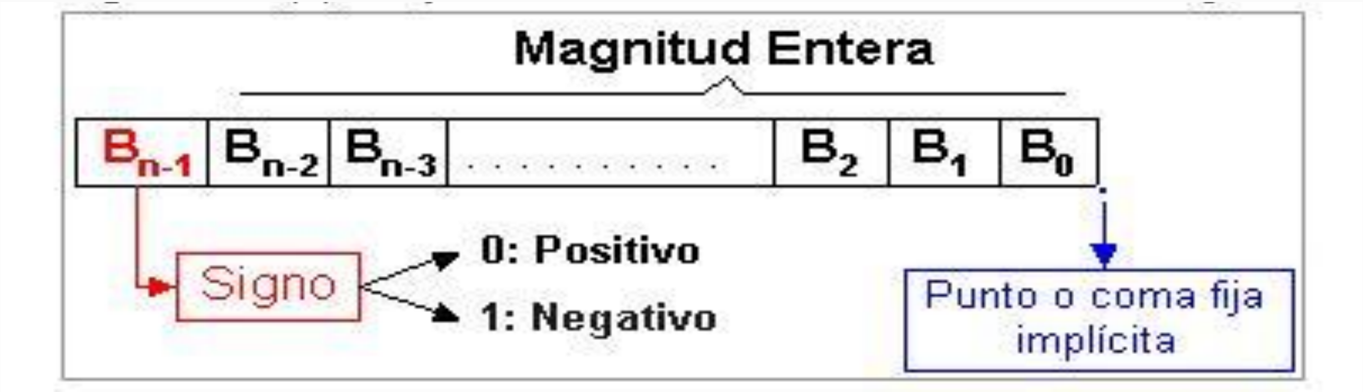
$$-010010010111.101 = -1175.625$$

Representación en Coma Fija

Pero cuando trabajamos con números fraccionarios, muchas veces ocurre que cuanto mayor es la parte entera, menos importancia tiene la fraccionaria, esta puede ser despreciable. Por ejemplo la parte fraccionaria del peso en gramos de un camión, no es relevante el valor exacto a partir de un número de cifras por ejemplo uno que pesa 1,357.475 grs. O 1,357.500 grs.

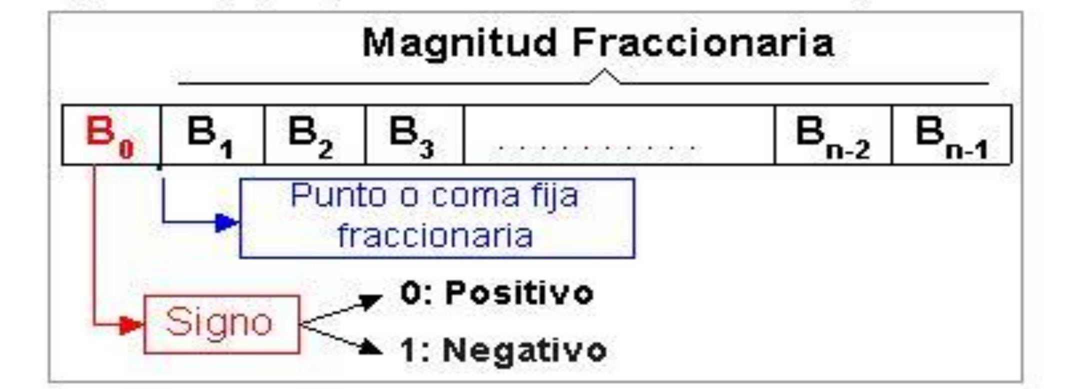
Representación Entera de Coma Fija

Los números fraccionarios y con signo se pueden representar mediante la coma fija; ejemplo de esto se puede apreciar en la figura donde se tiene la representación de números enteros con signo en formato de 16 bits.



Representación Fraccionaria de Coma Fija

Existe otra representación para coma fija, la cual consiste en fijar la posición de la coma después del bit de signo; ver figura respectivamente. Los restantes bits deben indicar la magnitud fraccionaria.



Representación en Coma Fija

Usando representación en punto fijo, para representar número grandes necesitamos muchos bits, y siempre mantenemos la precisión en los dígitos de menor peso, y vemos que esto no siempre es importante, por lo que a veces es conveniente otra representación: el punto flotante.

Representación en Coma Flotante

Los números representados en coma flotante tienen la misma forma que la notación científica. La representación tiene la siguiente forma

$$N = Mxb^{E}$$

N= número
M= número normalizado
b= base
E= exponente

Representación en Coma Flotante

Existen varias formas de representar los formatos de coma flotante; sin embargo, los que más se utilizan son los siguientes:

$$N = Mxb^{E}$$

$$N = (M \div b)xb^{E+1}$$

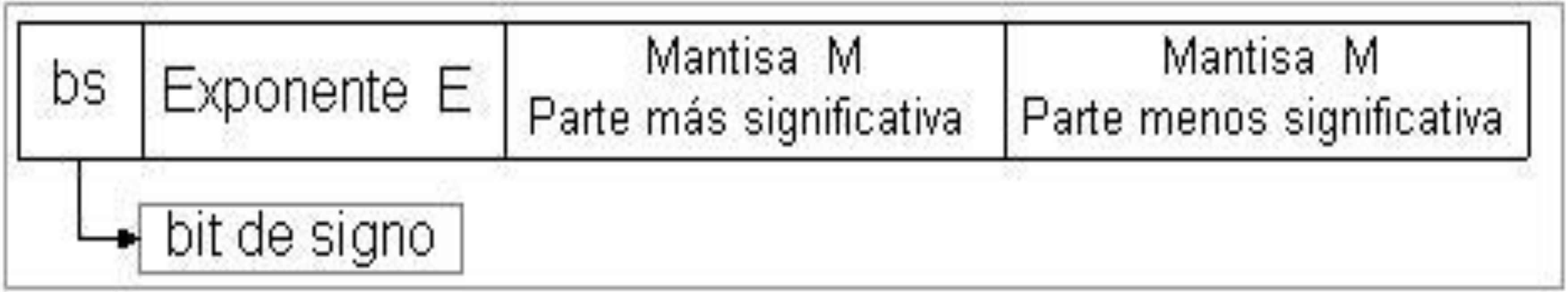
$$N = (Mxb)xb^{E-1}$$

Formatos en coma flotante para datos numéricos reales cortos y largos utilizados en los computadores.

Figura 1.4(a). Declaración de datos cortos en coma flotante.

Formatos en coma flotante para datos numéricos reales cortos y largos utilizados en los computadores.

Figura 1.4(b). Declaración de datos largos en coma flotante.



REPRESENTACIÓN NUMÉRICA EN COMA FLOTANTE

La idea consiste en representar únicamente las cifras más importantes del número e indicar cuántos ceros a la derecha hay que añadir (o a la izquierda si el número queda a la derecha de la coma fraccionaria), o lo que es lo mismo, cuantas posiciones a la izquierda o derecha hay que desplazar la coma decimal respecto a las cifras principales (llamadas mantisa).

REPRESENTACIÓN NUMÉRICA EN COMA FLOTANTE

La representación más habitual es la que muestra la figura, llamada de formato de SIGNO Y MAGNITUD. El tamaño de la palabra, cantidad de bits reservados para la representación del número en punto flotante, se divide en 3 partes.

REPRESENTACIÓN NUMÉRICA EN COMA FLOTANTE

La primera es 1 bit destinado al signo del nro. El resto de bits se reparten entre el exponente y la mantisa. Cuanto más bits se dediquen a exponente, mayor rango de números se podrá representar, ya que la coma decimal se podrá desplazar en un mayor rango de valores

Estándar IEEE de punto flotante

La representación de nros. fraccionarios que necesita de una menor cantidad de dígitos para su representación mediante notación científica, es aquella que usa un punto decimal después de la primera cifra significativa de la mantisa. Esta forma de representación se denomina **normalizada**.

Estándar IEEE de punto flotante

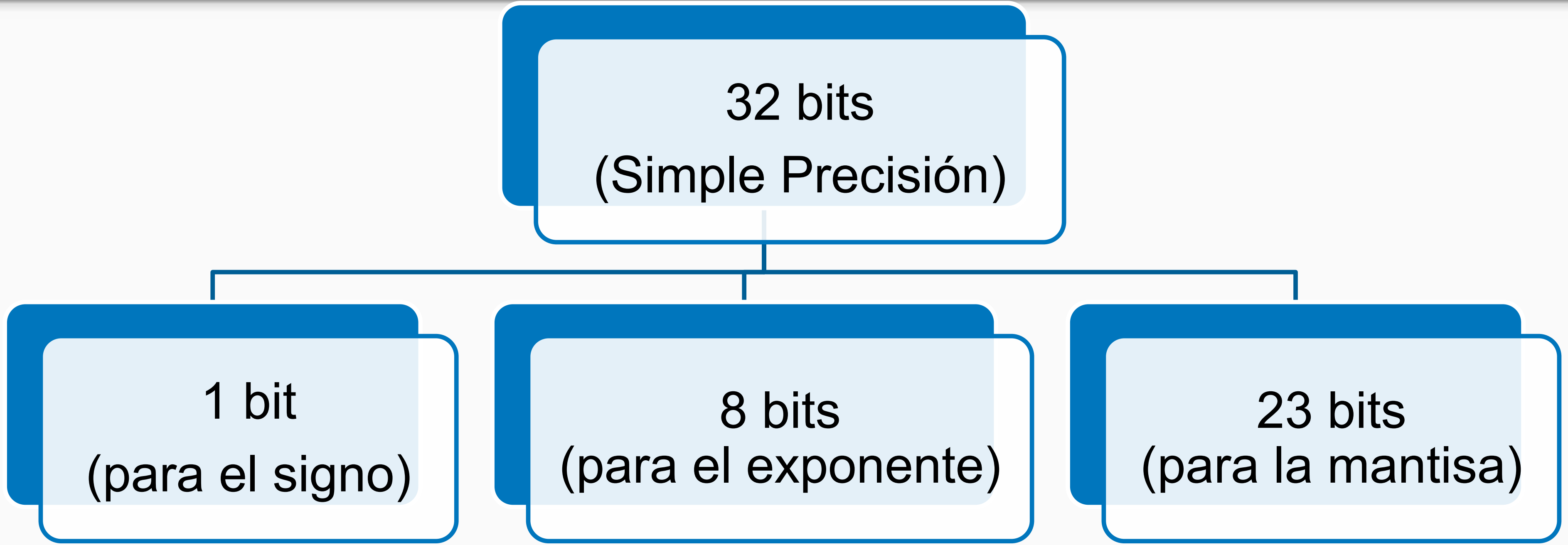
El más utilizado es el estándar IEEE 754, del que hay 2 tipos: IEEE754, simple precisión en el que la palabra es de 32 bits y formato IEEE754 doble precisión en el que la palabra es de 64 bits, Difieren además del tamaño de la palabra, en el nro. de bits que asignan a cada campo

IEEE 754

Este convenio en ambos formatos, usa un truco llamado bit oculto: Como el bit más significativo de la mantisa es un 1, ahorramos un bit simplemente asumiendolo.

El inconveniente de emplear 1 bit oculto es que se requiere una representación especial para el cero

IEEE 754 Precisión Simple



IEEE 754 Precisión Simple

Exponente con signo:

Está conformado por los 8 bits a continuación de signo. Esta ubicación en la palabra facilita las comparaciones de nros. Si los nros. Se encuentran normalizados, comparamos los exponentes. Si son iguales pasamos a comparar las mantisas.

IEEE 754 Precisión Simple

<u>Exponente con signo:</u>

Pero ¿qué representación es más conveniente usar para el exponente?. Si utilizamos Complemento a la base (C2), los exponentes negativos aparecerán como mayores que los positivos al usar la circuitería de enteros.

IEEE 754 Precisión Simple

Exponente con signo:

0 = 0000 0000

1 = 0000 0001

-1 = 1111 1111

IEEE 754 Precisión Simple

Para evitar este inconveniente se usa una representación en exceso N de forma que el exponente más negativo posible quede en 0000 0001 y el más grande de los positivos en 1111 1110. El estándar IEEE 754 usa como exceso 127 para precisión simple.

IEEE 754 Precisión Simple

Exponente más negativo representable:

$$-x + 127 = 0000\ 0001 \quad (x = -126)$$

Exponente mas grande representable:

$$x + 127 = 1111\ 1110 \ (x = 127)$$

IEEE 754 Precisión Simple

Mantisa :

Está formada por el resto de los bits en la palabra (23). Como los nros. Se representan de manera normalizada entonces siempre tendremos un 1 a la izquierda del punto. Por lo tanto este dígito no es necesario almacenarlo en la palabra y se tiene de manera implícita. La mantisa consiste en 24 bits de precisión.

IEEE 754 Precisión Simple

Ejemplos: Representación del 7

- Convertimos el nro. A binario : $7_{10} = 111_2$

- Normalizamos el nro. : $1.11_2 \times 2^2$

IEEE 754 Precisión Simple

-Calculamos el exponente con exceso 127 para precisión simple:

$$2 + 127 = 129_{10} = 1000\ 0001_2$$

El nro. 7 en estándar IEEE es representado como:

$$\underbrace{0}_{\textbf{Signo}}\ \underbrace{1\ 0\ 0\ 0\ 0\ 0\ 0\ 1}_{\textbf{Exponente}}\ \underbrace{1\ 1\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0\ 0}_{\textbf{Parte Decimal}}$$

IEEE 754 Precisión Simple

Ejemplos: Representación del 21

-Convertimos el nro. A binario : $21_{10} = 10101_2$

- Normalizamos el nro. : $1.0101_2 \times 2^4$

IEEE 754 Precisión Simple

-Calculamos el exponente con exceso 127 para precisión simple:

$$4 + 127 = 131_{10} = 1000\ 0011_{2}$$

El nro. 7 en estándar IEE es representado como:

$$\underbrace{0}_{\textbf{BS}}\ \underbrace{1\,0\,0\,0\,0\,0\,1\,1}_{\textbf{Exponente}}\ \underbrace{0\,1\,0\,1\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0\,0}_{\textbf{Mantisa}}$$

IEEE 754 Precisión Simple
(de binario a decimal)

0	0	0	0	0	1	1	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Exponente en exceso $= 00001100 = 12_{10}$

Exponente $= 12_{10} - 127_{10} = -115_{10}$

Los dígitos que están en la mantisa van a formar parte del nro. Decimal, y por tanto el nro. Representado es:

$$1.01_2 \times 2^{-115} = (1 + 0.25) \times 2^{-115} = 1.25_{10} \times 2^{-115}$$

IEEE 754 Precisión Doble

64 bits
(Precisión doble)

1 bit
(para el signo)

11 bits
(para exponente representación en exceso 1023)

52 bits (para la mantisa el bit implícito 53 bits)

IEEE 754 Precisión Doble

Para evitar este inconveniente se usa una representación en exceso N de forma que el exponente más negativo posible quede en 000 0000 0001 y el más grande de los positivos en 111 1111 1110. El estándar IEEE 754 usa como exceso 1023 para precisión doble.

IEEE 754 Precisión Doble

Exponente más negativo representable:

$$-x + 1023 = 000\ 0000\ 0001 \quad (x = -1022)$$

Exponente mas grande representable:

$$x + 1023 = 111\ 1111\ 1110 \quad (x = 1023)$$

IEEE 754 Precisión Doble

Ejemplos: Representación del 1007

-Convertimos el nro. A binario : $1007_{10} = 0000001111101111_2$

- Normalizamos el nro. : $1.111101111_2 \times 2^9$

IEEE 754 Precisión Doble

-Calculamos el exponente con exceso 127 para precisión simple:

$$9 + 1023 = 1023_{10} = 10000001000_2$$

El nro. 1007 en estándar IEE es representado como: