Números de punto flotante

Para representar números reales, muchos formatos de punto flotante emplean notación científica y usan un cierto número de bits para representar una ***mantisa*** y un número pequeño de bits para representar el ***exponente***. Esto resulta en que los números flotantes pueden representar a un número con solo una cantidad específica de dígitos significativos.

Notación científica:

$$n = f \times 10^{e}$$

f es la fracción o mantisa

e es el exponente

Ejemplos:

$$3.14 = 0.314 \times 10^{1} = 3.14 \times 10^{0}$$
$$0.000001 = 0.1 \times 10^{-5} = 1.0 \times 10^{-6}$$
$$1941 = 0.1941 \times 10^{4} = 1.941 \times 10^{3}$$

Estándar IEEE 754

En 1985 se estableció el estándar IEEE 754 para la implementación de números flotantes.

El estándar define tres formatos:

- Precisión sencilla
- Precisión doble
- Precisión extendida

Estándar IEEE 754

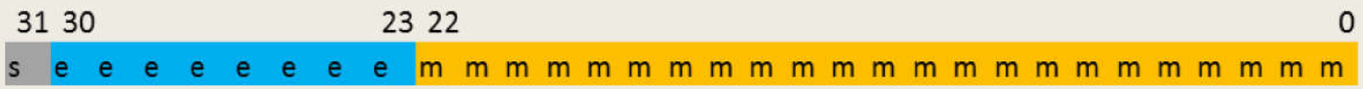
En este estándar la **fracción** consiste en un bit 1 y un punto binario implícito (*1.*), y 23 o 52 bits arbitrarios.

- Si todos los bits de la fracción son ceros, la fracción tiene el valor numérico de 1.0;
- Si todos son uno, la fracción es numéricamente un poco menos que 2.0.

A fin de evitar confusiones con una fracción convencional, se le denomina ***significando*** en vez de fracción o mantisa.

Precisión sencilla

Tiene un tamaño de dato de **32 bits** y su organización es la siguiente:



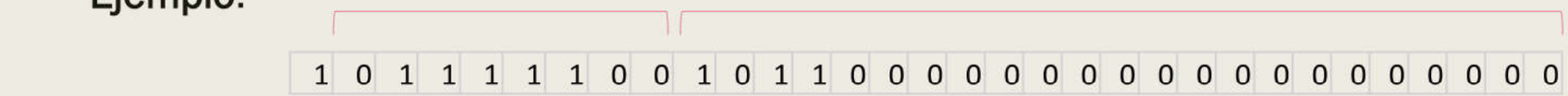
Donde:

- 1 bit corresponde al **signo.** El 0 es positivo y 1 es negativo.
- 8 bits corresponden al **exponente** el cual usa exceso 127.
- 23 bits corresponden al **significando.**

Intervalo decimal que puede representar: aprox. 10^{-38} a 10^{38}

Precisión sencilla

Ejemplo:



Signo = 1 = **número negativo**

Exponente: 01111100_2 = 124 – 127 = **-3**

Significando: $10110000000000000000000_2$

Conversión a decimal:

1.1011_2 (El *1.* es implícito en el estándar)

$1.1011_2 = (1 \times 2^0) + (1 \times 2^{-1}) + (0 \times 2^{-2}) + (1 \times 2^{-3}) + (1 \times 2^{-4}) = 1.6875_{10}$

$1.6875_{10} \times 2^{-3} = 0.2109375$

Valor: - 0.2109375

Precisión doble

Tiene un tamaño de dato de **64 bits** y su organización es la siguiente:

Donde:

- 1 bit corresponde al **signo.** El 0 es positivo y 1 es negativo.
- 11 bits corresponden al **exponente** el cual usa exceso 1023.
- 52 bits corresponden al **significando.**

Intervalo decimal que puede representar: aprox. 10^{-308} a 10^{308}

Precisión extendida

Su implementación es dependiente de la aplicación.

Acotaciones por parte de IEEE 754:

- **Precisión extendida sencilla**:

  Bits de exponente: ≥ 11

  Número de bits significativos (precisión): ≥ 32

- **Precisión extendida doble**:

  Bits de exponente: ≥ 15

  Número de bits significativos (precisión): ≥ 64

Suma y resta en punto flotante

Para sumar y restar dos números en punto flotante:

- Convertir los operandos a notación científica, representando explícitamente el 1 oculto.
- Ajustar los operandos de forma que ambos tengan el mismo exponente.
- Sumar o restar las mantisas de los operandos.
- Ajustar el resultado de forma que esté ***normalizado***, esto es, que se encuentre de la forma $(1.fracción)_2$
- Convertir el valor a formato de punto flotante, remover el 1 implícito del significando.

Suma y resta en punto flotante

Ejemplo:

a = 0.25 b = 100 c = a + b

Representación en punto flotante:

a = 0 01111101 00000000000000000000000 $\rightarrow 1.0 \times 2^{-2}$

b = 0 10000101 10010000000000000000000 $\rightarrow 1.5625 \times 2^{6}$

Ajuste de los exponentes para que sean iguales:

a = 0 10000101 0.00000001000000000000000 $\rightarrow 0.00390625 \times 2^{6}$ No normalizado

b = 0 10000101 10010000000000000000000 $\rightarrow 1.5625 \times 2^{6}$

Suma y resta en punto flotante

Suma de las mantisas:

	0.00000001000000000000000	(a)
+	1.10010000000000000000000	(b)
	1.10010001000000000000000	(c)

En decimal es equivalente a hacer:
$0.00390625 + 1.5625 = 1.56640625$

Suma y resta en punto flotante

Normalización del resultado:

1.10010001000000000000000

El resultado ya se encuentra normalizado, es decir, se encuentra de la forma $(1.fracción)_2$.

Si este no fuera el caso, para normalizarlo hay que realizar corrimientos en la mantisa de forma que haya un único 1 antes del punto, y realizar los correspondientes incrementos o decrementos al exponente.

Suma y resta en punto flotante

Conversión a formato de punto flotante:

c = 0 10000101 10010001000000000000000

Se removió el 1 que se encuentra antes del punto, el cual quedará implícito.

En decimal el valor corresponde a:

$$c = 1.56640625 \times 2^6 = 100.25$$

Algoritmo para multiplicación en punto flotante

Para multiplicar dos números en punto flotante:

- Multiplicar los significandos (incluir el *1.* implícito en ambos).
- Sumar los exponentes (después de restarles a cada uno el exceso-127 o exceso-1023).
- El signo se obtiene de aplicar un xor entre los signos de los operandos.
- Convertir el valor a formato de punto flotante, remover el 1 implícito del significando, sumar el exceso al exponente.

Algoritmo para división en punto flotante

Para dividir dos números en punto flotante:

- Dividir los significandos (incluir el *1.* implícito en ambos).
- Restar los exponentes (después de restarles a cada uno el exceso-127 o exceso-1023). La resta corresponde a

  exponente del dividendo - exponente del divisor

- El signo se obtiene de aplicar un xor entre los signos de los operandos.
- Convertir el valor a formato de punto flotante, remover el 1 implícito del significando, sumar el exceso al exponente.