Tutorial: Conversion de decimal a flotante IEEE754 de precisión simple

El formato IEEE754 se basa en las ideas de la notación científica. En base dos sería, por ejemplo:

$$- 1101001 \times 2^{-3}$$

A partir de esto, se pueden identificar tres partes: signo, exponente y fracción. Es posible hacerle algunas modificaciones a la notación científica para que quede de la siguiente forma:

$$(-1)^{BS} \times (1 + \text{fracción}) \times 2^{\text{exponente} - \text{bias}}$$

Donde se resaltan las tres partes antes mencionadas. Los 32 bits disponibles se pueden dividir en estas partes con la siguiente distribución:

```
X XXXXXXXX XXXXXXXXXXXXXXXXXXXXXXX
BS Exponente          Fracción
```

- Bit 31: Bit de Signo (BS).
- Bit 30 - 23: Exponente.
- Bit 22 - 0: Fracción.

Bit de Signo

Este bit establece el signo del número. Cuando es cero implica que el número positivo, en el caso en que sea 1, es negativo.

Exponente

El exponente de un número puede ser tanto positivo como negativo. Por esto, en la estandarización se decidió agregar un *bias* que me permite tener valores positivos y negativos sin dificultar aún más el proceso de conversión y la lógica necesaria para comparar flotantes con enteros en la computadora. En el caso de los flotantes de 32 bits, el *bias* es 127. En la siguiente tabla se muestran algunos ejemplos.

Exponente	Exponente + Bias
...	...
-3	124
-2	125
-1	126
0	127
1	128
2	129
3	130
...	....

Fracción
La parte fraccionaria del número debe ser normalizada. Esto implica mover la coma tantos lugares como sea necesario para que el número resultante quede expresado como un uno seguido por la parte fraccionaria:

$$1, xxxxxxxxx \times 2^{yyy}$$

Es importante notar que se debe adaptar el exponente para que el valor no se modifique.

Procedimiento decimal => flotante (IEEE 754 de 32 bits):

1) Encontrar el bit de signo
2) Pasar el número a binario y normalizar
3) Sumar el *bias* al exponente y convertirlo a binario
4) Encontrar la parte fraccionaria
5) Conformar el número

Procedimiento flotante (IEEE 754 de 32 bits) => decimal:

1) Dividir los el conjunto de bits en las tres partes respetando el formato
2) Encontrar el bit de signo
3) Encontrar el exponente
4) Desnormalizar el número y pasar a decimal

Ejemplo 1

Convertir de decimal a flotante IEEE 754 de 32 bits el número 263,3

1) Encontrar el bit de signo
Como 263.3 es positivo, el bit de signo es 0

2) Pasar el número a binario y normalizar
La normalización, como se dijo anteriormente implica mover la coma decimal tantos lugares como sea necesario para que quede un uno seguido por una parte fraccionaria. Para esto se necesita adaptar el exponente para no modificar el valor:

$$263,3_{10} = 100000111,0\overline{1001}\, x\, 2^{0}{}_{2}$$ (Sin normalizar)

$$263,3_{10} = 1,000001110\overline{1001}\, x\, 2^{8}{}_{2}$$ (Normalizado)

(la barra implica que esa parte del número es periódica)

Debido a la normalización, siempre el primer bit será un uno. Entonces, no es necesario almacenarlo. De esta forma ganamos un bit más de precisión.

3) Sumar el *bias* al exponente y convertirlo a binario

$127 + 8 = 135_{10} = 10000111_2$

4) Encontrar la parte fraccionaria
La parte fraccionaria se compone por los 23 bits que siguen a la coma decimal del número normalizado. En caso de que el número necesite más bits de precisión, deberá ser truncado. En este caso, queda:

$00000111010011001100110_2$

5) Conformar el número

Respetando el formato, se une el bit de signo con el exponente y la parte fraccionaria:

$$01000011100000111010011001100110_2$$

Ejemplo 2

Convertir el número obtenido en el punto anterior de flotante IEEE 754 de 32 bits a decimal.

1) Dividir los el conjunto de bits en las tres partes respetando el formato

El bit más significativo es el bit de signo, los bits 30 a 23 son el exponente y del 22 al 0 la parte fraccionaria.

$$0\ \ 10000111\ \ 00000111010011001100110_2$$

2) Encontrar el bit de signo

Como el bit de signo es 0, el número es positivo.

3) Encontrar el exponente

Esto implica pasar a decimal la parte del exponente, y luego restarle el *bias.*

$$10000111_2 = 135_{10}$$

$$135 - 127 = 8$$

4) Desnormalizar el número y pasar a decimal

Esto implica agregar el uno que se elimina al normalizar y luego desplazar la coma tantos lugares (y en el sentido) como indique el exponente

Parte fraccionaria: $00000111010011001100110_2$ => $1{,}00000111010011001100110 \times 2^{8}{}_2$

Desnormalización: $100000111{,}010011001100110_2 = 263{,}299987793$

Aclaración: Como puede verse, el número que se obtuvo al realizar el proceso de conversión y el inverso es distinto al original. Esto se debe a que en los 23 bits de la parte fraccionaria se truncó luego de una cierta cantidad de bits.

Ejemplo 3

Convertir de decimal a flotante IEEE 754 de 32 bits el número $-0{,}000892_{10}$

1) Encontrar el bit de signo

Como -0.000892 es negativo, el bit de signo es 1

2) Pasar el número a binario y normalizar

$$-\ 0,000892_{10} = 0,00000000001110100111010101000110110 \ x\ 2^{0}{}_2 \text{(Sin normalizar)}$$

$$-\ 0,000892_{10} = 1,11010011101010100011011 0\ x\ 2^{-11}{}_2 \text{ (Normalizado)}$$

3) Sumar el *bias* al exponente y convertirlo a binario

$$127 - 11 = 116_{10} = 01110100_2$$

4) Encontrar la parte fraccionaria

$$11010011101010100011011_2$$

5) Conformar el número

$$10111010011010011101010100011011$$