Punto Flotante

Muchas aplicaciones requieren trabajar con números que no son enteros. Existen varias formas de representar números no enteros. Una de ellas es usando un punto o coma fijo. Este tipo de representación ubica simpre el punto o coma en alguna posición a la derecha del dígito menos significativo.

Otra alternativa comunmente usada es la que se conoce como representación en punto flotante. Bajo este esquema, un número puede ser expresado mediante un exponente y una mantisa. Por ejemplo el número 10.75 puede ser expresado como

10.75 x	10^{0}
1.075 x	10^{1}
mantisa	exponente

En general, un número en punto flotante puede ser representado como $\pm d_0.d_1d_2d_3...d_k \; x \; b^{exp}$ donde $d_0.d_1d_2d_3...d_k$ se conoce como la mantisa, b es la base y exp es el exponente.

¿Qué se necesita para representar un número en punto flotante?

- el signo del número.
- el signo del exponente.
- Dígitos para el exponente.
- Dígitos para la mantisa.

Dado que un número en punto flotante puede expresarse de distintas formas que son equivalentes, es necesario establecer una única representación. Es por ello que se trabaja con números *normalizados*. Decimos que un número está normalizado si el dígito a la izquierda del punto o coma está entre 0 y la base (0< dígito a la izquierda del punto < b). En particular, decimos que un número binario está normalizado si el dígito a la izquierda del punto es igual a 1.

1.00×10^{-1} normalizado

0.01×10^{2} no normalizado

Estándar IEEE-754 para representación de Punto Flotante

Este estándar se desarrolló para facilitar la portabilidad de los programas de un procesador a otro y para alentar el desarrollo de programas numéricos sofisticados. Este estándar ha sido ampliamente adoptado y se utiliza prácticamente en todos los procesadores y coprocesadores aritméticos actuales. El estándar del IEEE define el formato para precisión simple de 32 bits y para precisión doble de 64 bits.

Precisión Simple

El formato para los números de *precisión simple* es de 32 bits.

signo	exponente con signo	Mantisa
1	8	23

La representación de un número en precisión simple en el formato IEEE-754 consta de las siguientes partes:

- *Signo* se encuentra en el bit más significativo, de esta manera podemos usar la misma circuitería (de enteros) para llevar a cabo comparaciones con respecto al cero.
- *Exponente con signo*. Está conformado por los siguientes 8 bits. Esta ubicación del exponente en la palabra facilita las comparaciones de números. Si los números se encuentran normalizados, comparamos los exponentes. Si son

iguales pasamos a comparar las mantisas. Pero, ¿ qué representación es más conveniente usar para el exponente?. Si utilizamos Complemento a Dos, los exponentes negativos aparecerán como mayores que los exponentes positivos al usar la circuitería de enteros.

C2(-1) = 1111 1111
C2(0) = 0000 0000
C2(1) = 0000 0001

Para evitar este inconveniente, se utiliza 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.

Exponente más negativo representable:

x + 127 = 0000 0001
x = -126

Exponente más grande representable

x + 127 = 1111 1110
x = 127

- *Mantisa.* Está formada por el resto de los bits en la palabra (23). Como los números 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.

Ejercicio

Representar según el estándar IEEE de punto flotante los siguientes valores:

- 7

  - Convertimos el número a binario.

    $7_{10} = 111_2$

  - Normalizamos el número.

    $1.11_2 \times 10_2^2$

  - Calculamos el exponente con exceso127 para precisión simple.

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

  - El número 7_{10} en el estándar IEEE es representado como:

0	10000001	11000000000000000000000
signo	exponente en exceso	mantisa

- 21

  $21_{10} = 10101_2 = 1.0101_2 \times 10_2^4$
  exponente $4 + 127 = 131_{10} = 1000\ 0011_2$

0	10000011	01010000000000000000000

Precisión Doble

La representación de un número en precisión doble en el formato IEEE-754 consta de las siguientes partes:

- *Signo* se encuentra en el bit más significativo

- *Exponente en exceso*. Está conformado por los siguientes 11 bits. Se utiliza una representación en exceso 1023 de forma que el exponente más negativo posible quede en 000 0000 0001 y el más grande de los positivos en 111 1111 1110.
- *Mantisa*. Está formada por 52 bits más el bit implícito (53).

signo	exponente en exceso	Mantisa
1 bit	11 bits	52 bits

Casos Especiales

Para valores de exponente desde 1 hasta 254 en el formato simple y desde 1 a hasta 2046 en el formato doble, se representan números en punto fijo normalizados. El exponente está en exceso, siendo el rango del exponente de -126 a +127 en el formato simple y de -1022 a +1023 en el doble.

Un número normalizado debe contener un bit 1 a la izquierda del punto binario; este bit está implícito, dando una mantisa efectiva de 24 bits para precisión simple o 53 bits para precisión doble.

Un exponente cero junto con una parte fraccionaria cero representa el cero positivo o negativo, dependiendo del bit de signo. Es útil tener una representación del valor 0 exacto.

Precisión Simple

Exponente en exceso	Mantisa	Valor
0	0	Cero
0	<>0	Número no normalizado (0. + Mantisa) x 2^{-126}
1 .. 254		(1. + Mantisa) x $2^{exp-127}$
255	0	Infinito
255	<>0	Not a Number

Precisión Doble

Exponente en exceso	Mantisa	Valor
0	0	Cero
0	<>0	Número no normalizado (0. + Mantisa) x 2^{-1022}
1 .. 2046		(1. + Mantisa) x $2^{exp-1023}$
2047	0	Infinito
2047	<>0	Not a Number

Conversión de un número en Punto Flotante Decimal a Binario

Un número $Num_b = d_0.d_1d_2d_3...$ en base b representa

$$Num_{10} = d_0 + d_1 * b^{-1} + d_2 * b^{-2} + d_3 * b^{-3} + \ldots + d_n * b^{-n}$$

podemos reescribirlo de la siguiente forma:

$$Num_{10} = d_0 + b^{-1}\left(d_1 + d_2 * b^{-1} + d_3 * b^{-2} + \ldots + d_n * b^{-n-1}\right)$$

$$Num_{10} = d_0 + b^{-1}\left(d_1 + b^{-1}\left(d_2 + d_3 * b^{-1} + \ldots + d_n * b^{-n-2}\right)\right)$$

$$Num_{10} = d_0 + b^{-1}\left(d_1 + b^{-1}\left(d_2 + d_3 * b^{-1}\left(d_4 + \ldots + b^{-1}\left(d_{n-1} + d_n * b^{-1}\right)\right)\right)\right)$$

De la última expresión podemos deducir el algoritmo de conversión de punto flotante decimal a cualquier base

Dado un número Num_{10} en punto flotante decimal y una base b

d_0= parte entera(Num_{10})
Num_{10} = (Num_{10} - d_0) * b
i=1
Repetir desde i=1 hasta N
  d_i=parte entera (Num_{10})
  Num_{10}=(Num_{10} - d_i) * b

$Num_{10} = d_0.d_1d_2d_3d_4...d_{N\ b}$

Ejemplos

a.) Convertir 0.5_{10} a binario y hallar su representación en IEEE precisión simple

0.50
(0.50-0) * 2 = 1 $d_0=0$
(1.00-1) * 2 = 0 $d_1=1$

$0.50_{10} = 0.1_2 = 1.0 \times 2^{-1}$

exponente en exceso= $-1 + 127 = 126_{10} = 0111\ 1110_2$

0 01111110 00000000000000000000000

0	01111110	00000000000000000000000
signo	exponente en exceso	mantisa

b.) Convertir 3.75_{10} a binario y hallar su representación en IEEE precisión simple

3.75
(3.75-3) * 2 = 1.50 $d_0=3$
(1.50-1) * 2 = 1.00 $d_1=1$
(1.00-1) * 2 = 0.00 $d_2=1$

$3.75_{10} = 11.11_2 = 1.111 \times 2^1$

exponente en exceso= $1 + 127 = 128_{10} = 1000\ 0000_2$

0	1000 0000	11100000000000000000000
signo	exponente en exceso	mantisa

c.) Convertir 0.3_{10} a binario y hallar su representación en IEEE precisión simple

0.3
(0.3-0) * 2 = 0.6 $d_0=0$
(0.6-0) * 2 = 1.2 $d_1=0$
(1.2-1) * 2 = 0.4 $d_2=1$
(0.4-0) * 2 = 0.8 $d_3=0$
(0.8-0) * 2 = 1.6 $d_4=0$
(1.6-1) * 2 = 1.2 $d_5=1$

$0.3_{10} = 0.01001001001..._2 = 1.001001001... \times 2^{-2}$

exponente en exceso= $-2 + 127 = 125_{10} = 0111\ 1101_2$

0	0111 1101	00100100100100100100100
signo	exponente en exceso	Mantisa

Esta representación es una aproximación. No puede ser escrito en forma precisa. Los números punto flotante son normalmente aproximaciones. La razón de esto es que existe un número infinito de números reales entre dos números dados.

d.) ¿Qué número decimal representa el siguiente patrón de bits en IEEE precisión simple?

0 00001100 01000000000000000000000

Calculamos el exponente que va a formar parte del número decimal, restando el valor del exponente menos el exceso de 127.

exponente en exceso= 12_{10} = exponente + 127_{10}
exponente = 12 - 127 = -115_{10}

Los dígitos que están en la mantisa van a formar parte de el número decimal, y por tanto el número representado es

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

e.) ¿Qué número decimal representa el siguiente patrón de bits en IEEE precisión simple?

0 10000011 10100000000000000000000
exponente en exceso = 131
exponente = 131 - 127 = 4
$1.101 \times 2^4 = 11010 = 26_{10}$

Ejercicios Propuestos

- Convertir los siguientes números a punto flotante binario

  -1.756_{10}
  15.75_{10}
  5.625_{10}
  $1.0 \times 10^{-1}{}_{10}$
  $5.7525*10_{10}$

- ¿ Cuál es el menor entero positivo que se puede representar en C2 con 32 bits, pero que no puede ser representado en Punto Flotante IEEE precisión simple?

- ¿Qué número decimal representan los siguientes patrones de bits si se interpretan como punto flotante IEEE precisión simple ?

  c1680000

  7f800000

  fff80000

  42be8000

  ff800000