Representación Flotante

floating representation

Michael Stiven Giraldo Henao
Risaralda, UTP, Pereira, Colombia
Correo-e: Michael.giraldo1tp.edu.co

***Resumen*— Este documento cuenta con una explicación de que son las representaciones flotantes y como son usadas en el ámbito de la informática con las cuales se pueden representar números reales extremadamente grandes y pequeños de una manera muy eficiente y compacta, y con la que se pueden realizar operaciones aritméticas**
***Palabras clave*— Abreviación, multiplicación, elevado, base, exponente, espacio**

***Abstract*— This document has an explanation of what are the floating representations and how they are used in the field of computing with which extremely large and small real numbers can be represented in a very efficient and compact way, and with which operations can be performed arithmetic**

***Key Word* — Abbreviation, multiplication, elevated, base, exponent, space**

I. INTRODUCCIÓN

Como la memoria de los ordenadores es limitada, no puedes almacenar números con precisión infinita, no importa si usas fracciones binarias o decimales: en algún momento tienes que cortar. Pero ¿cuánta precisión se necesita? ¿Y dónde se necesita? ¿Cuántos dígitos enteros y cuántos fraccionarios?

Para un ingeniero construyendo una autopista, no importa si tiene 10 metros o 10.0001 metros de ancho ─ posiblemente ni siquiera sus mediciones eran así de precisas.

Para alguien diseñando un microchip, 0.0001 metros (la décima parte de un milímetro) es una diferencia enorme ─ pero nunca tendrá que manejar distancias mayores de 0.1 metros.

Un físico necesita usar la velocidad de la luz (más o menos 300000000) y la constante de gravitación universal (más o menos 0.0000000000667) juntas en el mismo cálculo.

Para satisfacer al ingeniero y al diseñador de circuitos integrados, el formato tiene que ser preciso para números de órdenes de magnitud muy diferentes. Sin embargo, solo se necesita precisión relativa.

Para satisfacer al físico, debe ser posible hacer cálculos que involucren números de órdenes muy dispares.

Básicamente, tener un número fijo de dígitos enteros y fraccionarios no es útil ─ y la solución es un formato con un punto flotante. [1]

II. CONTENIDO

En un ordenador típico los números en punto flotante se representan de la manera descrita en el apartado anterior, pero con ciertas restricciones sobre el número de dígitos de q y m impuestas por la longitud de palabra disponible (es decir, el número de bits que se van a emplear para almacenar un número). Para ilustrar este punto, consideraremos un ordenador hipotético que denominaremos MARC-32 y que dispone de una longitud de palabra de 32 bits (muy similar a la de muchos ordenadores actuales). Para representar un número en punto flotante en el MARC-32, los bits se acomodan del siguiente modo:

Signo del número real x:	1 bit
Signo del exponente m:	1 bit
Exponente (entero $\|m\|$):	7 bits
Mantisa (número real $\|q\|$):	23 bits

Tabla 1: Distribución de bits en la representación flotante [2]

¿Cómo funcionan los números de punto flotante?

La idea es descomponer el número en dos partes:

1) Una mantisa (también llamada coeficiente o significando) que contiene los dígitos del número. Mantisas negativas representan números negativos.

2) Un exponente que indica dónde se coloca el punto decimal (o binario) en relación al inicio de la mantisa. Exponentes negativos representan números menores que uno.

Este formato cumple todos los requisitos:

Puede representar números de órdenes de magnitud enormemente dispares (limitado por la longitud del exponente).
Proporciona la misma precisión relativa para todos los órdenes (limitado por la longitud de la mantisa).

Permite cálculos entre magnitudes: multiplicar un número muy grande y uno muy pequeño conserva la precisión de ambos en el resultado.
Los números de coma flotante decimales normalmente se expresan en notación científica con un punto explícito siempre entre el primer y el segundo dígitos. El exponente o bien se escribe explícitamente incluyendo la base, o se usa una e para separarlo de la mantisa.

Mantisa	Exponente	Notación científica	Valor en punto fijo
1.5	4	$1.5 \cdot 10^4$	15000
-2.001	2	$-2.001 \cdot 10^2$	-200.1
5	-3	$5 \cdot 10^{-3}$	0.005
6.667	-11	6.667e-11	0.0000000000667

Tabla 2: Ejemplos

El estándar

Casi todo el hardware y lenguajes de programación utilizan números de punto flotante en los mismos formatos binarios, que están definidos en el estándar IEEE 754. Los formatos más comunes son de 32 o 64 bits de longitud total:

Formato	Bits totales	Bits significativos	Bits del exponente
Precisión sencilla	32	23 + 1 signo	8
Precisión doble	64	52 + 1 signo	11

Hay algunas peculiaridades:

La secuencia de bits es primero el bit del signo, seguido del exponente y finalmente los bits significativos.
El exponente no tiene signo; en su lugar se le resta un desplazamiento (127 para sencilla y 1023 para doble precisión). Esto, junto con la secuencia de bits, permite que los números de punto flotante se puedan comparar y ordenar correctamente incluso cuando se interpretan como enteros.
Se asume que el bit más significativo de la mantisa es 1 y se omite, excepto para casos especiales.
Hay valores diferentes para cero positivo y cero negativo. Estos difieren en el bit del signo, mientras que todos los demás son 0. Deben ser considerados iguales, aunque sus secuencias de bits sean diferentes.
Hay valores especiales no numéricos (NaN, «not a number» en inglés) en los que el exponente es todo unos y la mantisa no es todo ceros. Estos valores representan el resultado de algunas operaciones indefinidas (como multiplicar 0 por infinito, operaciones que involucren NaN, o casos específicos). Incluso valores NaN con idéntica secuencia de bits no deben ser considerados iguales.
[3]

Ejemplo:

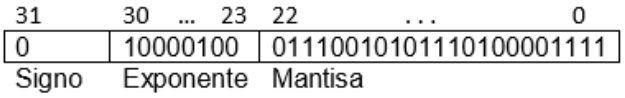
Para escribir el número
101110.01010111010000111110000111111000100112
en el estándar IEEE 754 con precisión simple, con exponente en exceso a 2n-1–1 y mantisa m y signo s, determine su número hexadecimal correspondiente

Primero tendremos que normalizarlo:
101110.01010111010000111110000111111000100112
=1.01110010101110100001111100001111100010011 2 X 25

El exponente E será: exponente representación externa más el exponente en exceso a 2n-1–1 en base 10 en precisión simple es 127. Calculemos ahora a E:

De la mantisa solo se cogerán los 23 bits más significativos:
1.01110010101110100001111

Como el resto de bits no pueden representarse, ya que no caben en la mantisa, entonces se descartarán. Sin embargo, cuando la mantisa se normaliza situando el punto decimal a la derecha del bit más significativo, dicho bit siempre vale 1. Por tanto, se puede prescindir de él, y coger en su lugar un bit más de la mantisa de la parte descartada. De esta forma, la precisión del número representado es mayor. Así, los bits de la mantisa serán: 01110010101110100001111
En consecuencia, el número se puede representar como:



[4]

III. CONCLUSIONES

La representación de punto flotante (en inglés floating point) es una forma de notación científica usada en los computadores con la cual se pueden representar números reales extremadamente grandes y pequeños de una manera muy eficiente y compacta, y con la que se pueden realizar operaciones aritméticas. El estándar actual para la representación en coma flotante es el IEEE 754.

REFERENCIAS

Las fuentes bibliográficas consultadas, pero no citadas en el texto se colocarán al final de las referencias citadas y se numeran de la misma forma.

[1] http://puntoflotante.org/formats/fp/

[2] https://www.uv.es/~diaz/mn/node11.html

[3] http://puntoflotante.org/formats/fp/

[4] https://medium.com/@matematicasdiscretaslibro/