Representación flotante

Floating Representation

Julian David Siceri Ramirez
Unuversidad Tecnologica de Pereira
Correo-e:j.siceri@utp.edu.co

***Resumen*— El estándar del IEEE para aritmética en coma flotante (IEEE 754) es la norma o estándar técnico para computación en coma flotante, establecida en 1985 por el Instituto de Ingenieros Eléctricos y Electrónicos (IEEE). La norma abordó muchos problemas encontrados en las diversas implementaciones de coma flotante que las hacían difíciles de usar de forma fiable y portátil. Muchas unidades de coma flotante de hardware utilizan ahora el estándar IEEE 754.**

El estándar define:

- **Formatos aritméticos: conjuntos de datos de coma flotante binarios y decimales, que consisten en números finitos, incluidos los ceros con signo y los números desnormalizados o subnormales, infinitos y valores especiales "no numéricos" (NaN).**
- **Formatos de intercambio: codificaciones (cadenas de bits) que se pueden utilizar para intercambiar datos de coma flotante de forma eficiente y compacta.**
- **Reglas de redondeo: propiedades que deben satisfacerse al redondear los números durante las operaciones aritméticas y las conversiones.**
- **Operaciones: operaciones aritméticas y otras (como funciones trigonométricas) en formatos aritméticos.**
- **Manejo de excepciones: indicaciones de condiciones excepcionales, tales como división por cero, desbordamiento, etc.**

Palabras clave— coma flotante, ingenieros, IEEE 754, binarios, decimales, números, codificaciones, operaciones, aritmética, trigonometría, división, desbordamiento.

***Abstract*— The IEEE standard for floating-point arithmetic (IEEE 754) is the technical standard or standard for floating-point computing, established in 1985 by the Institute of Electrical and Electronic Engineers (IEEE). The standard addressed many problems encountered in the various floating point implementations that made them difficult to use reliably and portablely. Many hardware floating point units now use the IEEE 754 standard.**

The standard defines:

• Arithmetic formats: binary and decimal floating point data sets, consisting of finite numbers, including signed zeros and denormalized or subnormal numbers, infinities and special "non-numerical" (NaN) values.

• Exchange formats: encodings (bit strings) that can be used to exchange floating point data efficiently and compactly.

• Rounding rules: properties that must be satisfied when rounding numbers during arithmetic operations and conversions.

• Operations: arithmetic and other operations (such as trig functions) in arithmetic formats.

• Exception handling: indications of exceptional conditions, such as division by zero, overflow, etc.

***Key Word* — floating point, engineers, IEEE 754, binaries, decimals, numbers, encodings, operations, arithmetic, trigonometry, division, overflow.**

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*.

I. CONTENIDO

Cómo funcionan los números de punto flotante

La idea es descomponer el número en dos partes:

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

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: *(Ver tabla 2)*

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.

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 1. Representación punto flotante.

Formato	Bits totales	Bits significativos	Bits del exponente	Número más pequeño	Número más grande
Precisión sencilla	32	23 + 1 signo	8	~1.2 · 10-38	~3.4 · 1038
Precisión doble	64	52 + 1 signo	11	~5.0 · 10-324	~1.8 · 10308

Tabla 2. Representación Punto flotante, estándar.

II. CONCLUSIONES

La Representacion punto flotanto es un método de representación de números de gran cantidad infinitamente grandes en los que podemos acaparar una infinidad de información en un espacio relativamente reduciod utilizando técnicas aritméticas y trigonométricas para llevar algo muy grande a algo reducido.

REFERENCIAS

Referencias de publicaciones periódicas:

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

Reportes Técnicos:

1. ↑ *Mike Cowlishaw (20 de marzo de 2009). «Decimal Arithmetic Encodings» (en inglés). IBM UK Laboratories. Consultado el 18 de marzo de 2017.*

Norrmas:

1. ↑ *«FW: ISO/IEC/IEEE 60559 (IEEE Std 754-2008)»*. *grouper.ieee.org (en inglés). IEEE. 01 de abril de 2011. Archivado desde el original el 19 de marzo de 2012. Consultado el 18 de marzo de 2017.*
2. ↑ *«ISO/IEC/IEEE 60559:2011 - Information technology -- Microprocessor Systems -- Floating-Point arithmetic»*. *www.iso.org (en inglés). ISO. Consultado el 18 de marzo de 2017.*
3. ↑ *«IEEE 754-2008 errata»*. *speleotrove.com (en inglés). Archivado desde el original el 8 de octubre de 2019. Consultado el 18 de marzo de 2017.*
4. ↑ *«Revising ANSI/IEEE Std 754-2008 http://754r.ucbtest.org»*. *754r.ucbtest.org (en inglés). Consultado el 18 de marzo de 2017.*
5. ↑ *Saltar a:* **a** **b** **c** *«IEEE Std 754-2008: IEEE Standard for Floating-Point Arithmetic» (en inglés). IEEE Inc. Archivado desde el original el 6 de noviembre de 2016. Consultado el 27 de febrero de 2017.*
6. *«IEEE Std 1003.1, 2004 Edition» (en inglés). The Open Group. Consultado el 27 de febrero de 2017.*