Representación Flotante

Floating Representation

Santiago Valencia Díaz
Ingeniería de Sistemas y Computación, Universidad Tecnológica de Pereira, Pereira, Colombia
Correo-e: santiago.valencia6@utp.edu.co

***Resumen*— Este documento contiene un resumen sobre la representación flotante tal y como se introdujo en la asignatura Introducción a la Informática. El objetivo es realizar una introducción al tema, explicar cómo este método permitió el almacenaje correcto de números reales grandes o pequeños en bloques de 8 dígitos (bits) y algunos ejemplos del método.**

Palabras clave*— *representación, base, exponente, mantisa, signo, alineación, posición

***Abstract*— This document contains a summary about the floating representation, as it's treated in the subject Introduction to the Computer Science. The goal is to make an introduction to the topic, explain how this method allowed big and small real numbers to be correctly stored in blocks of 8 digits (bits) and some examples of the method.**

***Key Words* — representation, base, exponent, mantissa, sign, alignment, position**

I. INTRODUCCIÓN

Las **ciencias de la computación** son las ciencias formales que abarcan las bases teóricas de la información y la computación, así como su aplicación en sistemas computacionales. Una de sus ramas es la rama del almacenamiento de los datos. La información que una computadora almacena en memoria, es codificada en forma de bits. Una computadora no puede leer con exactitud un número real o un número complejo grande o pequeño, por lo que dificulta el almacenaje de los mismos y pueden existir también conflictos en la transmisión de datos entre computadora y computadora.

II. CONTENIDO

La representación de **punto flotante** (o representación flotante) 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. Este sistema de representación utiliza una determinada cantidad de dígitos binarios dependiendo el tipo de precisión (comúnmente 16, 32, 64 y 128 bits).

Un bit es destinado al signo, es decir si ese bit vale 0 se trata de un número positivo, si vale 1 es un número negativo. Los bits restantes se reparten en la representación de los decimales (se suele llamar mantisa) y el exponente.

$$n = (-1)^s . 2^{(e-127)} . (1 + m)$$

Figura 1. En la expresión ***n*** es el número decimal a representar.

En la representación en coma flotante se dividen los n bits, disponibles para representar un dato, en dos partes, una para la mantisa M (o fracción) y otra para el exponente E. Considerando que la mantisa tiene una longitud de p bits y que el exponente la tiene de q bits, se cumple que n = p + q. La mantisa contiene los dígitos significativos del dato, mientras que el exponente indica el factor de escala, en forma de una potencia de base 2.

En la representación flotante, se consideran dos formatos básicos, el de simple y el de doble precisión, que se representan seguidamente.

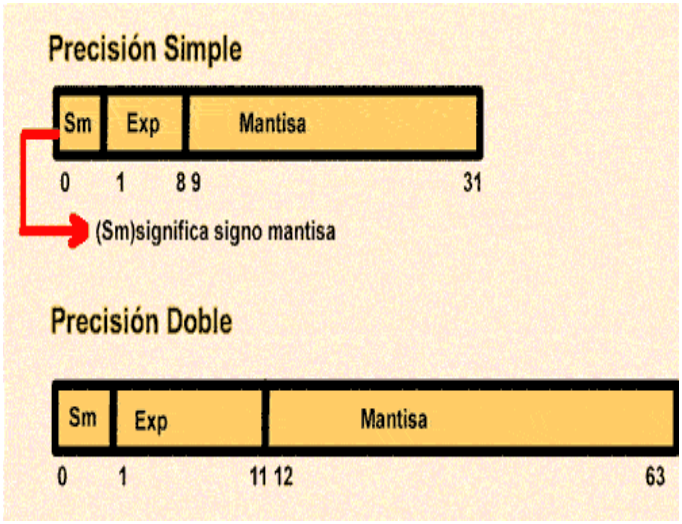

Figura 2. Ejemplo de ambos formatos de representación: Precisión simple y precisión doble. El exponente se representa en exceso a 127 para precisión simple y a 1023 en precisión doble.

La mantisa que se representa es la fracción que queda luego de desplazar la coma detrás del primer 1. Este primer bit significativo de la mantisa que siempre es 1 no se representa, esto permite representar un bit más. La coma fraccionaria de

la mantisa se considera después de dicho 1 de la siguiente manera: 1,M.

En el sistema binario, un valor real se puede extender con una cantidad arbitraria de dígitos. La coma flotante permite representar solo una cantidad limitada de dígitos de un número real, solo se trabajará con los dígitos más significativos, (los de mayor peso) del número real, de tal manera que un número real generalmente no se podrá representar con total precisión sino como una aproximación que dependerá de la cantidad de dígitos significativos que tenga la representación en coma flotante con que se está trabajando. La limitación se halla cuando existen dígitos de peso menor al de los dígitos de la parte significativa. En dicho caso estos suelen ser redondeados, y si son muy pequeños son truncados. Sin embargo, y según el uso, la relevancia de esos datos puede ser despreciable, razón por la cual el método es interesante pese a ser una potencial fuente de error.

En la representación binaria de coma flotante, el bit de mayor peso define el valor del signo, 0 para positivo, 1 para negativo. Le siguen una serie de bits que definen el exponente. El resto de bits son la parte significativa. Debido a que la parte significativa está generalmente normalizada, en estos casos, el bit más significativo de la parte significativa siempre es 1, así que no se representa cuando se almacena, sino que es asumido implícitamente. Para poder realizar los cálculos ese bit implícito se hace explícito antes de operar con el número en coma flotante. Hay otros casos donde el bit más significativo no es un 1, como con la representación del número cero, o cuando el número es muy pequeño en magnitud y rebasa la capacidad del exponente, en cuyo caso los dígitos significativos se representan de una manera denormalizada para así no perder la precisión de un solo golpe sino progresivamente. En estos casos, el bit más significativo es cero y el número va perdiendo precisión poco a poco (mientras que al realizar cálculos este se haga más pequeño en magnitud) hasta que al final se convierte en cero.

A continuación, se presentarán los siguientes ejemplos para describir la notación de coma flotante. Abajo hay 3 números en una representación de coma flotante de 16 bits. El bit de la izquierda es el signo, luego hay 6 bits para el exponente, seguidos de 9 bits para la parte significativa:

Signo	Exponente	Parte Significativa	
1	100011	011101100	= 0xC6EC
0	011011	111001101	= 0x37CD
0	101001	000000001	= 0x5201

Figura 3. Ejemplo de una representación flotante binaria de 16 bits.

El signo es expresado por el bit de la izquierda, con 0 indicando que el número es positivo y 1 indicando que el número es negativo. En los ejemplos de arriba, el primer número es negativo y los dos últimos son positivos.

El exponente indica cuánto se debe desplazar hacia la derecha o hacia la izquierda la coma binaria de la parte significativa. En este caso, el exponente ocupa 6 bits capaces de representar 64 valores diferentes, es decir, es un exponente binario (de base 2) que va desde -31 a +32, representando potencias de 2 entre 2^{-31} y 2^{+32}, indicando que la coma binaria se puede desplazar hasta 31 dígitos binarios hacia la izquierda (un número muy cercano a cero), y hasta 32 dígitos binarios hacia la derecha (un número muy grande).

Pero el exponente no se almacena como un número binario con signo (desde -31 hasta +32) sino como un entero positivo equivalente que va entre 0 y 63. Para ello, al exponente se le debe sumar un desplazamiento (bias), que en este caso de exponente de 6 bits (64 valores), es 31 (31 es la mitad de los 64 valores que se pueden representar, menos 1), y al final, el rango del exponente de -31 a +32 queda representado internamente como un número entre 0 y 63, donde los números entre 31 y 63 representan los exponentes entre 0 y 32, y los números entre 0 y 30 representan los exponentes entre -31 y -1 respectivamente:

```
-31            0             32 <-- Exponente binario real
 +-------+-------+-------+-------+
 0             31            63 <-- Representación en coma flotante
                                    del exponente de 6 bits
                                    (Es el exponente binario más un bits de 31)
```

Figura 4. Representación del exponente como entero positivo equivalente que va desde 0 a 63.

La parte significativa, en este caso, está formada por 10 dígitos binarios significativos, de los cuales tenemos 9 dígitos explícitos más 1 implícito que no se almacena.

Esta parte significativa generalmente está normalizada y tendrá siempre un 1 como el bit más significativo. Debido a que, salvo ciertas excepciones, el bit más significativo del significante siempre es 1, para ahorrar espacio y para aumentar la precisión en un bit, este bit no se almacena, y por ello se denomina bit oculto o implícito, sin embargo, antes de realizar los cálculos este bit implícito debe convertirse en un bit explícito.

La notación genérica para la coma flotante descrita arriba, representa respectivamente los siguientes números reales (expresados en binario). El color rojo indica el bit más significativo, que cuando se almacena es implícito (ver arriba la parte significativa en la representación de coma flotante), pero cuando se hacen los cálculos, o cuando se muestra la información se vuelve explícito:

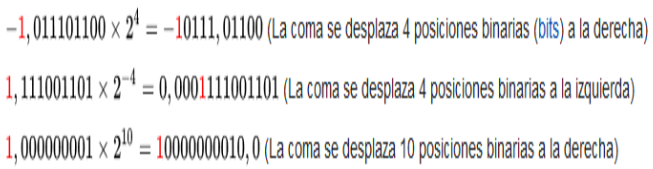

Figura 5. Notación genérica con números reales, representados binariamente.

Para un tamaño determinado de bytes, la notación en coma flotante puede ser más lenta de procesar y es menos precisa que la notación en coma fija, ya que además de almacenar el número (parte significativa), también debe almacenarse el exponente, pero permite un mayor rango en los números que se pueden representar.

Debido a que las operaciones aritméticas que se realizan con números en coma flotante son complejas de realizar, muchos sistemas destinan un procesador especial para la realización específica de este tipo de operaciones, denominado unidad de coma flotante o tienen incorporados componentes especializados. En los casos donde no exista esta facilidad, o que el hardware de coma flotante no pueda realizar determinadas operaciones, se utilizan bibliotecas de software para realizar los cálculos.

III. CONCLUSIONES

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. Debido a la limitación de memoria de los ordenadores, la precisión de los datos debe ser lo más exacta posible, aunque deban sacrificarse parte de esos datos que pueden resultar innecesarios luego. Básicamente, tener un número fijo de dígitos enteros y fraccionarios no es útil ─ y la solución es un formato con un *punto flotante*.

REFERENCIAS

Referencias de páginas web:

[1] http://puntoflotante.org/formats/fp/
[2] https://es.wikipedia.org/wiki/Coma_flotante
[3] http://www.portalhuarpe.com.ar/Medhime20/Sitios%20con%20Medhime/Computaci%C3%B3n/COMPUTACION/Menu/modulo%203/paginas/U3-A-SLF-ComaFlotante.htm
[4] https://www.inf.utfsm.cl/~parce/cc1/clase18-RP.html
[5] http://www2.elo.utfsm.cl/~elo385/docs/Biblio/Lab4/Numeros_Punto_Fijo_y_Flotante.pdf
[6] https://eva.udelar.edu.uy/pluginfile.php/808915/mod_resource/content/3/Clase%2018.pdf
[7] https://medium.com/@matematicasdiscretaslibro/cap%C3%ADtulo-3-punto-flotante-c689043db98bS.