Representación Flotante

Floating Representation

Santiago Runcería Ortiz
Ingeniería de Sistemas y Computación, UTP, Pereira, Colombia
Correo-e: s.runceria@utp.edu.co

***Resumen*— Este documento contiene un resumen sobre el tema: Representación Flotante, tal y como se introdujo en la materia Introducción a la Informática. El objetivo es realizar una revisión de sus propiedades, desarrollo, explicaciones y teoría pertinente.**

***Palabras clave* — posición, base 10, alineación, mantisa, exponente, signo.**

***Abstract*— This document contains a summary on the subject: Floating Representation, as presented in the subject Introduction to Information Technology. The objective is to review its properties, development, explanations and relevant theory.**

***Key Word* — position, base 10, alignment, mantissa, exponent, sign.**

I. INTRODUCCIÓN

Las ciencias de la computación y la informática son disciplinas que se encargan del estudio sistemático de los procesos algorítmicos que describen y transforman información. En una computadora la información está almacenada en forma de bits en una memoria. Para que la máquina pueda acceder a ella y pueda comprender la información, es necesario codificarla en datos numéricos.
Como la memoria de los ordenadores es limitada, no es posible almacenar números con precisión infinita, no importa si se usa fracciones binarias o decimales: en algún momento es necesario que se corte. Pero ¿cuánta precisión se necesita? ¿Y dónde se necesita? ¿Cuántos dígitos enteros y cuántos fraccionarios?

II. CONTENIDO

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.

REPRESENTACIÓN

La notación científica se usa para representar números reales. Siendo r el número real a representar, la representación en notación científica está compuesta de tres partes:

1. (c): El coeficiente, formado por un número real con un solo dígito entero seguido de una coma (o punto) y de varios dígitos fraccionarios.

2. (b): La base, que en nuestro sistema decimal es 10, y en el sistema binario de los computadores es 2.

3. (e): El exponente entero, el cual eleva la base a una potencia.

COEFICIENTE

Un signo en el coeficiente indica si el número real es positivo o negativo.

El coeficiente tiene una cantidad determinada de dígitos significativos, los cuales indican la precisión del número representado, cuantos más dígitos tenga el coeficiente, más precisa es la representación. Por ejemplo, π lo podemos representar en notación científica, con 3 cifras significativas, 3,14 x 100, o con 12 cifras significativas, 3,14159265359 x 100, teniendo en la segunda representación mucha más precisión que la primera.

BASE Y EXPONENTE

El coeficiente es multiplicado por la base elevada a un exponente entero. En nuestro sistema decimal la base es 10. Al multiplicar el coeficiente por la base elevada a una potencia entera, lo que estamos haciendo es desplazando la coma del coeficiente tantas posiciones (tantos dígitos) como indique el exponente. La coma se desplaza hacia la derecha si el exponente es positivo, o hacia la izquierda si es negativo.

Ejemplo de cómo cambia un número al variar el exponente de la base:
2,71828 x 10^-2 representa al número real 0,0271828
2,71828 x 10^-1 representa al número real 0,271828
2,71828 x10^0 representa al número real 2,71828
(el exponente cero indica que la coma no se desplaza)
2,71828 x 10^1 representa al número real 27,1828
2,71828 x 10^2 representa al número real 271,828

EJEMPLO:
Un ejemplo de número en notación científica es el siguiente:

-1,234 567 89 x 10^3
El coeficiente es -1,23456789, tiene 9 dígitos significativos, y está multiplicado por la base diez elevada a la 3. El signo del coeficiente indica si el número real representado por la notación científica es positivo o negativo.

El valor de la potencia nos indica cuántas posiciones (cuántos dígitos) debe ser desplazada la coma del coeficiente para obtener el número real final. El signo de la potencia nos indica si ese desplazamiento de la coma debe hacerse hacia la derecha o hacia la izquierda. Una potencia positiva indica que el desplazamiento de la coma es hacia la derecha, mientras que un signo negativo indica que el desplazamiento debe ser hacia la izquierda. Si el exponente es cero, la coma no se desplaza ninguna posición. La razón de la denominación de "coma flotante", es porque la coma se desplaza o "flota" tantos dígitos como indica el exponente de la base, al cambiar el exponente, la coma "flota" a otra posición.

En el número representado en la notación científica anterior, -**1,23456789 x 10^3**, el exponente es 3 positivo, lo que indica que la coma del coeficiente -1,23456789 debe ser desplazada 3 posiciones hacia la derecha, si en dado caso el 3 fuera negativo, la coma sería desplazada hacia la izquierda, pero, en este caso el resultado sería:

-1234,567 89 (aplicando la notación científica)

¿CÓMO FUNCIONA LA MISMA TEORÍA EN NÚMERO DE BASE 10 CUANDO SE TRATA DE UNA PC?
La idea es descomponer el número en tres partes:

1. Un ***signo*** que indica si el número es positivo o negativo. Siendo **0** el indicador de un número **positivo**, y **1** el indicador de un número **negativo**.

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.

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

Este formato cumple todos los requisitos:

1. Puede representar números de órdenes de magnitud enormemente dispares (limitado por la longitud del *exponente*).

2. Proporciona la misma precisión relativa para todos los órdenes (limitado por la longitud de la *mantisa*).

3. Permite cálculos entre magnitudes: multiplicar un número muy grande y uno muy pequeño conserva la precisión de ambos en el resultado.

TABLA DE EXPLICACIÓN

SIGNO	MANTISA	EXPONENTE	NOTACIÓN CIENTÍFICA	VALOR EN PUNTO FIJO
+ (0)	1.5	4	1.5 * 10^4	15000
- (1)	2.001	2	-2.001 * 10^2	-200.1
+ (0)	5	-3	5 * 10^-3	0.005
+ (1)	6.667	-6	6.667^-6	0.00000667

Tabla 1. Explicación

EL ESTÁNDAR IEEE 754
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	Número +Pequeño	Número +Grande
Precisión Sencilla	32	23 + 1 signo	8	~1.2 * 10^-38	~3.4 * 10^38
Precisión Doble	64	52 + 1 signo	11	~5.0 * 10^-324	~1.8 * 10^308

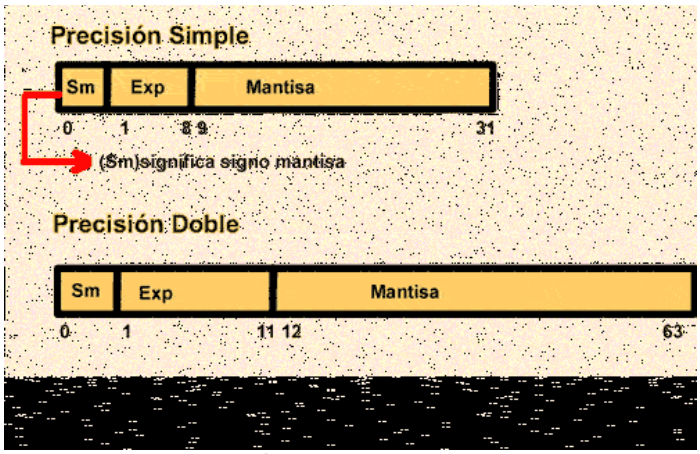

Imagen 1. Ejemplo precisión

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.

Por tanto, los valores de cada formato son los siguientes:
(-1)s*1,M*2^E-127 para 0<E<255
(Los valores para E= 0 y E=255 son especiales como lo veremos enseguida)
(-1)s*1,M*2^E-1023 para 0<E<2047

Obsérvese que se ha colocado la coma a la derecha del dígito
más significativo, lo que significa que la normalización es entre
1,0000... y 1,11111...

Hay algunas peculiaridades:

1. La secuencia de bits es primero el bit del signo, seguido del exponente y finalmente los bits significativos.
2. 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.
3. Se asume que el bit más significativo de la mantisa es 1 y se omite, excepto para casos especiales.
4. 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.
5. 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.

EJEMPLOS CON NÚMEROS BASE 10 (SIN DECIMAL) DISINTAS POSICIONES ORGANIZADOS EN UNA REPRESENTACIÓN DE 8 POSICIONES (8 BITS):

1.
+ . 4 7 6 3 8 2 4 9 6 1 0 2
(12 posiciones y signo positivo)
Al tener 12 posiciones, el exponente será 12, es decir, el número se multiplicará por 10^12 para de esta manera correr el punto decimal hacia la derecha.
Esto mismo ocurrirá con cualquier número; este será multiplicado por la cantidad de dígitos que tenga.
(0.xxxx * 10^#x)

Representación:

Signo	I			Mantisa			I	Exponente
0	4	7	6	3	8	1	2	

Tabla 2. Ejemplo 1 representación

El **2 4 9 6 1 0 2** se pierde al momento de almacenar el número en un conjunto de 8 espacios, puesto que es obligatorio dar los espacios pertinentes al exponente sea cual sea su equivalencia y un espacio al signo sea negativo o positivo.

2.
+ . 9 8 2 6 4 1 3 6 7 2 1 4 9 3 2
(15 posiciones y signo positivo)
Como fue explicado previamente, el número sería multiplicado por 10^15 (exponente), debido a la cantidad de dígitos que lo componen.

Representación:

Signo	I			Mantisa			I	Exponente
0	9	8	2	6	1	1	5	

Tabla 3. Ejemplo 2 representación

Nuevamente, como en el caso anterior, los números **3 6 7 2 1 4 9 3 2** se pierden debido a la falta de espacios.

3.
- . 7 2 4 6 2
(5 posiciones y signo negativo)
El número sería multiplicado por 10^5 con base a la cantidad dígitos que lo componen.

Representación:

Signo	I			Mantisa			I	Exponente
1	7	2	4	6	2	0	5	

Tabla 3. Ejemplo 2 representación

En este caso, no se pierde ningún número, únicamente es operado por medio de la notación científica. Esto se da gracias a su poca cantidad de dígitos, permitiendo al exponente y signo del mismo ocupar sus espacios, y así mismo ocupar una casilla por dígito sin problema.

¿CÓMO OPERAR ESTOS NÚMEROS DE FORMA CORRECTA?
Si se quisiera operar estos números es de importancia, primero, tener los mismos organizados según la notación científica, es decir:

1. **0. 4 7 6 3 8 * 10^12** = 476.380.000.000
   (recuperando así las cifras perdidas al momento de almacenarse en el conjunto de 8 posiciones, más no la cantidad)

2. **0. 9 8 2 6 4 * 10^15 =** 982.640.000.000.000
   (recuperando las cifras perdidas)

3. **- 0. 7 2 4 6 2 * 10^5 =** -72.462
   (el número no presenta cambios debido a que cupo en el conjunto de 8 espacios sin perder dígitos)

Una vez se tienen escritos correctamente y con las condiciones pertinentes aplicadas, se procede a organizarlos de la manera correcta para ser operados, teniendo en cuenta sus signos.
Sin embargo, es importante tener en cuenta que la manera de organizarlos es de derecha a izquierda, a partir del número con mayor cantidad de dígitos, es decir:

```
      + 4 7 6 3 8 0 0 0 0 0 0
+ 9 8 2 6 4 0 0 0 0 0 0 0 0 0
                  - 7 2 4 6 2
= 9 8 3 1 1 6 3 7 9 9 2 7 5 3 8
```

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 desnormalizada 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

EJEMPLO:

Emplearemos varios ejemplos para describir la notación de coma flotante. Abajo tenemos 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

Signo:

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.

Exponente:

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 (bits), 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 bit de 31)
```

PARTE SIGNIFICATIVA

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.

III. CONCLUSIONES

Es realmente sencillo almacenar tanto números de base 10 como binarios en conjuntos de determinados espacios, emulando a la computadora, siempre y cuando se le de importancia a las condiciones dadas con base a la teoría, y, de igual forma, se entienda perfectamente el concepto de notación científica, pues de ello, depende el exponente a ingresar dentro del conjunto y la manera en que el número resultante será compuesto tras correr el punto decimal.

Por otro lado, es claro, tras la lectura y comprensión del documento que el proceso de almacenamiento de variables o números en la computadora ha transcurrido por distintas etapas, y que, en la actualidad esto se define por el estándar IEEE 754, siendo el conjunto de bits o espacios más comunes: 32 y 64 bits.

Por último, es claro que para operar números una vez organizados de forma pertinente en notación científica, deben ser alineados según el número más grande de derecha a izquierda, de otro modo, no será posible operarlos.

En conclusión, el documento abarca los temas, conceptos y teoría más importante y relevante acerca de el tema Representación de Punto Fijo (Flotante).

REFERENCIAS:

[1] http://puntoflotante.org/formats/fp/
[2]http://www.portalhuarpe.com.ar/Medhime20/Sitios%20con%20Medhime/Computaci%C3%B3n/COMPUTACION/Menu/modulo%203/paginas/U3-A-SLF-ComaFlotante.htm
[3] https://www.uv.es/~diaz/mn/node11.html
[4] https://www.inf.utfsm.cl/~parce/cc1/clase18-RP.html
[5]https://www.google.com/search?q=conjuntos+de+8+bits+a+16&oq=conjuntos+de+8+bits+a+16+&aqs=chrome..69i57j33.6142j0j4&sourceid=chrome&ie=UTF-8
[6] https://es.wikipedia.org/wiki/Coma_flotante