

Organización de Computadoras 2004

Apunte 2: Sistemas de Numeración: Punto Flotante

La coma o punto flotante surge de la necesidad de representar números reales y enteros con un rango de representación mayor que el que nos ofrece punto fijo. En la representación en coma flotante, se dividen los n bits disponibles para representar un dato, en 2 partes llamadas mantisa M y 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, en tanto que el exponente indica el factor de escala, en forma de una potencia de base r . Por todo ello, el valor del número viene dado por:

$$V(X) = M * r^E$$

El número X viene representado por la cadena:

$$X = (m_{p-1}, \dots, m_1, m_0, e_{q-1}, \dots, e_1, e_0)$$

Cuya representación gráfica es:

Mantisa (p bits)	Exponente (q bits)
n-1 n-1	q-1 0

La mantisa M y el exponente E se representan en alguno de los sistemas de punto fijo.

El rango de representación estará dado por:

Mínimo número negativo:	(mantisa máxima) * base ^{máximo exponente positivo}
Máximo número negativo:	-(mantisa mínima) * base ^{máximo exponente negativo}
Mínimo número positivo:	(mantisa mínima) * base ^{máximo exponente negativo}
Máximo número positivo:	(mantisa máxima) * base ^{máximo exponente positivo}

Ejemplo:

Supongamos que tenemos un sistema de numeración con las siguientes características:

Mantisa de 5 bits, expresada en BSS.

Exponente de 3 bits, expresada en BCS.

Base 2.

Entonces, la cadena

10110 101

representará a $22 * 2^{-1} = 11$

Mantisa fraccionaria

En este caso, la mantisa en lugar de interpretarse como un número entero, se toma como un número real con el punto decimal implícito a la izquierda de sus bits.

Siguiendo con el ejemplo anterior de mantisa de 5 bits en BSS, exponente de 3 bits en BCS y base 2.

Entonces, la cadena

10110 101

representará a $(1/2 + 1/8 + 1/16) * 2^{-1} = 0.6875 * 2^{-1} = 0,34375$

y la cadena

00101 011

representará a $(1/8 + 1/32) * 2^3 = (5/32) * 2^3 = 5/4 = 1.25$

este valor coincide con el de la cadena

01010 010 = $(1/4 + 1/16) * 2^2 = (5/16) * 2^2 = 1.25$

y con el de la cadena

10100 001 = $(1/2 + 1/8) * 2^1 = (5/8) * 2^1 = 1.25$

Para evitar tener múltiples representaciones de un mismo número, se puede representar la mantisa en forma **normalizada**. La normalización consiste en hacer que su primer dígito tenga el valor 1. De esta manera, un número no puede tener mas de una representación.

En el ejemplo anterior, en un sistema con mantisa normalizada, de las 3 representaciones posibles para el número 1,25 solo sería válida 10100 001, ya que en ella la mantisa empieza con un 1.

La desventaja de este sistema de representación es que no se puede representar el número 0.

Dado que las mantisas normalizadas siempre empiezan con un 1, podemos no almacenar este dígito, con lo cual la mantisa tiene un bit de mas para almacenar. Por lo tanto,

$$p + q = n + 1$$

Este bit implícito se agrega en el momento de operar. A este sistema se lo llama ***mantisa fraccionaria normalizada con bit implícito***.

Dado que no siempre podemos representar exactamente el número que queremos, definimos *error absoluto* y *error relativo* de un número en un sistema de la siguiente forma:

$$\begin{aligned} EA(x) &= |x' - x| \\ ER(x) &= EA(x) / x \end{aligned}$$

donde x' es el número representable del sistema más próximo a x .

Ejercicios

1. Mantisa entera en BSS de 8 bits.

Exponente en Ca2 en 4 bits.

Donde los primeros 8 bits representan la mantisa, y los siguientes 4 al exponente.

Queremos obtener:

a) La representación del número 0,40625:

$$0,40625 * (2 * 2 * 2 * 2 * 2) = 13$$

$$\begin{aligned}
0,40625 * 2^5 &= 13 \\
0,40625 &= 13/2^5 \\
0,40625 &= 13 * 2^{-5} \\
0,40625 &= 00001101 \ 1011
\end{aligned}$$

b) El número máximo:

$$\text{mayor mantisa por mayor exponente: } 11111111 \ 0111 = 255 * 2^7 = 32640$$

c) El número mínimo:

$$\text{menor mantisa por menor exponente: } 00000000 \ 1000 = 0 * 2^{-8} = 0$$

d) Resolución máxima:

$$\begin{aligned}
\text{mayor número representable: } & 11111111 \ 0111 = 32640 \\
\text{anterior número representable: } & 11111110 \ 0111 = 32512 \\
\text{diferencia: } & = 128
\end{aligned}$$

e) Resolución mínima:

$$\begin{aligned}
\text{menor número representable: } & 00000000 \ 1000 = 0 \\
\text{siguiente número representable: } & 00000001 \ 1000 = 1/256 \\
\text{diferencia: } & = 1/256
\end{aligned}$$

2. Mantisa fraccionaria en BCS de 6 bits.

Exponente en Ex2 en 3 bits.

Donde el primer bit representa el signo, los 5 siguientes representan la mantisa, y los siguientes 3 al exponente.

Queremos obtener:

a) La representación del número 4,75:

$$4,75 * (2 * 2) = 19$$

$$4,75 = 19 / 4$$

tenemos que lograr que la mantisa sea una fracción, por tanto, multiplicamos y dividimos por 8

$$4,75 = (19 * 8) / (4 * 8)$$

reagrupando obtenemos

$$4,75 = (19 / 32) * 8$$

$$4,75 = (19 / 32) * 2^3$$

$$4,75 = 010011 \ 111$$

b) El número máximo positivo:

$$\text{mayor mantisa positiva por mayor exponente positivo: } 011111 \ 111 = (31/32) * 2^3 = 7,75$$

c) El número mínimo positivo:

$$\text{menor mantisa positiva por mayor exponente negativo: } 000000 \ 000 = 0 * 2^{-4} = 0$$

d) El número máximo negativo:

$$\text{mayor mantisa negativa por mayor exponente positivo: } 111111 \ 111 = -(31/32) * 2^3 = -7,75$$

e) El número mínimo negativo:

menor mantisa negativa por menor exponente negativo: $100000\ 000 = 0 * 2^{-4} = 0$

f) Resolución máxima positiva:

mayor número representable:	$011111\ 111 = (31/32) * 2^3 = 7,75$
anterior número representable:	$011110\ 111 = (30/32) * 2^3 = 7,50$
diferencia:	$= 0,25$

g) Resolución mínima positiva:

menor número representable positivo:	$000000\ 000 = 0 * 2^{-4}$
siguiente número representable:	$000001\ 000 = (1/32) * 2^{-4}$
diferencia:	$= (1/32) * 2^{-4} = 2^{-9}$

h) Resolución máxima negativa:

mayor número representable negativo:	$111111\ 111 = (31/32) * 2^3 = -7,75$
anterior número representable:	$111110\ 111 = (30/32) * 2^3 = -7,50$
diferencia:	$= 0,25$

i) Resolución mínima negativa:

menor número representable negativo:	$100000\ 000 = 0 * 2^{-4}$
siguiente número representable:	$100001\ 000 = -(1/32) * 2^{-4}$
diferencia:	$= (1/32) * 2^{-4} = 2^{-9}$

3. Mantisa fraccionaria normalizada en BCS de 6 bits.

Exponente en Ex2 en 3 bits.

Donde el primer bit representa el signo, los 5 siguientes representan la mantisa, y los siguientes 3 al exponente.

Queremos obtener:

a) La representación del número -0,140625:

$-0,140625 * (2 * 2 * 2 * 2 * 2 * 2) = -9$
 $-0,140625 * 2^6 = -9$
 $-0,140625 = -9 / 2^6$
 $-0,140625 = (-9 / 2^5) * 2^{-1}$
 $-0,140625 = 101001\ 011$

como no esta normalizada la mantisa, hacemos un corrimiento

$-0,140625 = 101001\ 011 = 110010\ 010$

b) El número máximo positivo:

mayor mantisa positiva por mayor exponente positivo: $011111\ 111 = (31/32) * 2^3 = 7,75$

c) El número mínimo positivo:

menor mantisa positiva por mayor exponente negativo: $010000\ 000 = (1/2) * 2^{-4} = 1/32$

d) El número máximo negativo:

mayor mantisa negativa por mayor exponente positivo: $111111\ 111 = -(31/32) * 2^3 = -7,75$

e) El número mínimo negativo:

menor mantisa negativa por menor exponente negativo: $110000\ 000 = -(1/2) * 2^{-4} = -1/32$

f) Resolución máxima positiva:

mayor número representable positivo:	011111 111	= 7,75
anterior número representable:	011110 111	= 7,5
diferencia:		= 0,25

g) Resolución mínima positiva:

menor número representable positivo:	010000 000	= $(1/2) * 2^{-4}$
siguiente número representable:	010001 000	= $(17/32) * 2^{-4}$
diferencia:		= $(1/32) * 2^{-4} = 2^{-9}$

h) Resolución máxima negativa:

mayor número representable negativo:	111111 111	= -7,75
anterior número representable:	111110 111	= -7,5
diferencia:		= 0,25

i) Resolución mínima negativa:

menor número representable negativo:	110000 000	= $-(1/2) * 2^{-4}$
siguiente número representable:	110001 000	= $-(17/32) * 2^{-4}$
diferencia:		= $(1/32) * 2^{-4} = 2^{-9}$

4. Mantisa fraccionaria normalizada con bit implícito en BCS de 6 bits.

Exponente en Ex2 en 3 bits.

Donde el primer bit representa el signo, los 5 siguientes representan la mantisa, y los siguientes 3 al exponente.

Queremos obtener:

a) La representación del número 1,96875:

$$\begin{aligned}
 1,96875 * (2 * 2 * 2 * 2 * 2) &= 63 \\
 1,96875 * 2^5 &= 63 \\
 1,96875 &= 63 / 2^5 \\
 1,96875 &= (63 / 2^5) * 2^0 = (63 / 2^6) * 2^1 \\
 1,96875 &= 011111\ 101
 \end{aligned}$$

b) El número máximo positivo:

mayor mantisa positiva por mayor exponente positivo: $011111\ 111 = (63/64) * 2^3 = 7,875$

c) El número mínimo positivo:

menor mantisa positiva por mayor exponente negativo: $000000\ 000 = (1/2) * 2^{-4} = 1/32$

d) El número máximo negativo:

mayor mantisa negativa por mayor exponente positivo: $111111\ 111 = -(63/64) * 2^3 = -7,875$

e) El número mínimo negativo:

menor mantisa negativa por menor exponente negativo: $100000\ 000 = -(1/2) * 2^{-4} = -1/32$

f) Resolución máxima positiva:

mayor número representable positivo:	011111 111	= 7,875
anterior número representable:	011110 111	= 7,75
diferencia:		= 0,125

g) Resolución mínima positiva:

menor número representable positivo:	000000 000	= $(1/2) * 2^{-4}$
siguiente número representable:	000001 000	= $(33/64) * 2^{-4}$
diferencia:		= $(1/64) * 2^{-4} = 2^{-10}$

h) Resolución máxima negativa:

mayor número representable negativo:	111111 111	= -7,875
anterior número representable:	111110 111	= -7,75
diferencia:		= 0,125

i) Resolución mínima negativa:

menor número representable negativo:	100000 000	= -1/32
siguiente número representable:	100001 000	= $-(33/64) * 2^{-4}$
diferencia:		= $(1/64) * 2^{-4} = 2^{-10}$

5. Mantisa fraccionaria entera en Ca1 de 8 bits.

Exponente en BSS en 4 bits.

Donde los primeros 8 bits representan la mantisa, y los siguientes 4 al exponente.

Dadas las siguientes representaciones:

p = 00011010 1101
q = 00001001 1111

Queremos hallar el resultado de la suma de los mismos.

Para esto, tenemos que igualar los exponentes. Dado que el exponente de p es menor que el del q, o bien subimos el exponente de p, o bien bajamos el de q. Subir el exponente de un número implica hacer un corrimiento a derecha de la mantisa; bajar el exponente implica hacer un corrimiento a izquierda de la mantisa.

- Subir el exponente de p:

Subiendo en 1 el exponente de p obtenemos:

00001101 1110

Todavía el exponente sigue siendo menor que el de q, con lo cual tendríamos que hacer otro corrimiento. Pero dado que el último bit de la mantisa es un 1, no lo podemos hacer.

- Bajar el exponente de q:

Bajando en 1 el exponente de q obtenemos:

00010010 1110

Todavía el exponente sigue siendo mayor que el de p , con lo cual tendríamos que hacer otro corrimiento.

Bajando en 1 el exponente obtenemos:

00100100 1101

Con lo cual conseguimos igualar los exponentes.

Ahora realizamos la suma de las mantisas:

$$\begin{array}{r} 00011010 \\ 00100100 \\ \hline 00111110 \end{array}$$

Por tanto, el resultado de la suma es: 00111110 1101

IEEE 754

El IEEE 754 es un estándar de aritmética en coma flotante. Este estándar especifica como deben representarse los números en coma flotante con simple precisión (32 bits) o doble precisión (64 bits), y también cómo deben realizarse las operaciones aritméticas con ellos.

Emplea mantisa fraccionaria, normalizada y en representación signo magnitud, sin almacenar el primer dígito, que es igual a 1. El exponente se representa en exceso, que en este caso no se toma como 2^{n-1} , sino como $2^{n-1} - 1$

Simple Precisión

El estándar IEEE-754 para la representación en simple precisión de números en coma flotante exige una cadena de 32 bits. El primer bit es el bit de signo (S), los siguientes 8 son los bits del exponente (E) y los restantes 23 son la mantisa (M):

Simple precisión

S	Exponente	mantisa
0 1	8 9	31

En los formatos IEEE no todas las cadenas se interpretan de la misma manera. Algunas combinaciones se utilizan para representar valores especiales.

El valor V representado por esta cadena puede ser determinado como sigue:

Exponente	Mantisa	Signo	Valor
$0 < E < 255$	Cualquiera	1	$-1 * 2^{E-127} * 1.M$ donde "1.M" se emplea para representar el número binario creado por la anteposición a M de un 1 y un punto binario
$0 < E < 255$	Cualquiera	0	$2^{E-127} * 1.M$ donde "1.M" se emplea para representar el número binario creado

			por la anteposición a M de un 1 y un punto binario
255	No nulo	Cualq.	NaN ("Not a number"). Se aplica para señalar varias condiciones de error.
255	0	1	-Infinito
255	0	0	Infinito
0	No nulo	1	$-(2^{-126}) * (0.M)$ (Números sin normalizar)
0	No nulo	0	$2^{-126} * (0.M)$ (Números sin normalizar)
0	0	1	-0
0	0	0	0

En particular,

0 00000000 000000000000000000000000 = 0
 1 00000000 000000000000000000000000 = -0

0 11111111 000000000000000000000000 = Infinito
 1 11111111 000000000000000000000000 = -Infinito

0 11111111 000001000000000000000000 = NaN
 1 11111111 00100010001001010101010 = NaN

0 10000000 000000000000000000000000 = $2^{(128-127)} * 1.0 = 2$
 0 10000001 101000000000000000000000 = $2^{(129-127)} * 1.101 = 6.5$
 1 10000001 101000000000000000000000 = $-1 * 2^{(129-127)} * 1.101 = -6.5$

0 00000001 000000000000000000000000 = $2^{(1-127)} * 1.0 = 2^{(-126)}$
 0 00000000 100000000000000000000000 = $2^{(-126)} * 0.1$

Doble precisión

El estándar IEEE-754 para la representación en doble precisión de números en coma flotante exige una cadena de 64 bits. El primer bit es el bit de signo (S), los siguientes 11 son los bits del exponente (E) y los restantes 52 son la mantisa (M):

Doble Precisión

S	Exponente	mantisa
0 1	11 12	63

El valor V representado por esta cadena puede ser determinado como sigue:

Exponente	Mantisa	Signo	Valor
$0 < E < 2047$	Cualquiera	1	$-1 * 2^{E-1023} * 1.M$ donde "1.M" se emplea para representar el número binario creado por la anteposición a M de un 1 y un punto binario
$0 < E < 2047$	Cualquiera	0	$2^{E-1023} * 1.M$ donde "1.M" se emplea para representar el número binario creado por la anteposición a M de un 1 y un punto binario
2047	No nulo	Cualq.	NaN ("Not a number"). Se aplica para señalar varias condiciones de error.
2047	0	1	-Infinito
2047	0	0	Infinito
0	No nulo	1	$-(2^{-1022}) * (0.M)$ (Números sin normalizar)
0	No nulo	0	$2^{-1022} * (0.M)$ (Números sin normalizar)
0	0	1	-0
0	0	0	0

En particular,

$$0\text{ }0000000000\text{ }000000000000000000000000000000000000000 = 0$$
$$1\text{ }0000000000\text{ }000 = -0$$
[illegible]

`1 1111111111 00 = -Infinito`

```
0 1111111111 0000010000000000000000000000000000000000000000000 = NaN
```

```
1 1111111111 00100010001001010101010000000000000000000000000 = NaN
```

[illegible][illegible][illegible][illegible][illegible]