Bienvenido a mis Tutorías © en la UNED

Ejemplos de utilización IEEE 754

Ejemplo 1:

César Moreno Fernández

El problema es el del examen del 3 de Febrero de 94.
Se pide la representación del Nº 53'2874 en formato normalizado
IEEE 754 para coma flotante de 16 bit.

16 bit quiere decir, a no ser que especifiquen lo contrario, que se utilizan 1 bit para el signo, 8 para el exponente y 7 para la mantisa. El mismo resultado se obtiene con la precisión normal de 32 bits, eliminando los últimos cuatro bytes de la derecha, que corresponden a los 16 bits extras de la mantisa.

Pasos

1º) Calcular el signo:

En nuestro caso es + luego el bit de signo es 0

2º) Normalización de la mantisa:

```
    53'2874 = 110101'010010
```

(se haya dividiendo por dos el entero y quedándote con el resto y multiplicando por 2 la parte fraccionaria y quedándote con los resultados enteros)

Como es el formato IEEE 754 hay un bit implícito. Esto quiere decir que vamos a correr la coma hasta el primer uno pero en lugar de dejarla a su izquierda como se hace en otros formatos lo dejamos a la derecha (específico para IEEE 754). Dicho bit implícito no será representado, con lo que ganamos un bit más en la precisión de la mantisa.

Esto es:

```
    corremos coma   1'10101'010010
                     ^____ /
```

la hemos corrido +5 posiciones. Lo tendremos en cuenta para después.

el bit implícito no lo representamos en la mantisa, luego cogemos los 7 bits siguientes a la coma

```
    Mantisa: 1010101
```

3º) El exponente:

Se calcula en base al exceso 2^(n-1) -1 (específico de IEEE 754)
Entonces tenemos : +5 (de la coma corrida) + 2^(n-1)-1=132= 10000100 (bin)
Luego la representación del número será:

```
    0100001001010101
   _ \_______/\______/
  Sig  Expon   Mantisa
```

Ojo en algunos exámenes los posibles resultados los dan en Hex.
Así que lo pasas de binario a Hex. Luego la solución es 4255 (hex)

Ejemplo 2:

Demetrio Quirós

Este problema es del examen del 26 de Enero de 1999 de ETC-1 (Modelo B - Pregunta 15):

Obtener la representacin del número decimal (-0.00015) en el formato normalizado IEEE754 para coma flotante de 16 bits (igual que el de 32, pero con una mantisa de 7 bits)

A) 377B   B) B77D   C) B91D   D) 377D

Empezaremos por calcular el equivalente de -0.00015 en formato m*2^e. El procedimiento que yo sigo es dividir el número por 2 elevado a 'lo que sea' hasta que el resultado esté en la forma 1.xxxx, en este caso: -0.00015 / 2^-13 = 1.2288, con lo que podemos decir que -0.00015 = 1.2288 * 2^-13, con lo que ya tenemos la mantisa y el exponente.

```
Signo: negativo = 1
Exponente: 127 - 13 = 114 = 01110010
Mantisa: [1.]2288 = [1.]0011101
```

Con lo que el resultado es

```
1011 1001 0001 1101
 B    9    1    D
```

Ejemplo 3:

Ramón Quiñones Lozano

¿Cuál es el error, en valor absoluto, que se comete al representar el número decimal 291.072 con el número en formato IEEE754 (16 bits) 4391?

Primero pasamos el número hexadecimal a binario:

```
   4391 hex  =   0100  0011  1001  0001
```

colocándolos segun el formato IEEE754

```
    signo        exponente      mantisa
      0           10000111      0010001
```

y convirtiéndolo queda:

```
   signo (0) -> positivo
   exp. (10000111) = 135  (quitando el exceso a 127)  135-127 =  8
   mantisa  1.0010001 (ojo al bit implícito)

    1.0010001  x  2^8  =   100100010  =  290 en decimal
```

Así que el error =   291.072 - 290 = 1.072

Ejemplo 4:

Dethais

Convertir el número -2.5675e15 al formato IEEE 754 de 32 bits:

1º Se coge el nº sin el menos, es decir positivo y escribimos lo siguiente:

```
2.5675*10^15 = 2^exponente
```

Nos interesa despejar el "exponente". ¿Cómo se hace esto? Con logaritmos:
La expresión simplificada quedaría:

```
[log(2.5675)+15*log(10)]/log(2) = exponente
```

Esto se hace con la calculadora y listo. Se puede hacer con logaritmos o con logaritmos neperianos, lo que se prefiera. El exponente sale:

```
exp = 51.189
```

(aproximamos al inmediatamente inferior, o sea al 51. Esto se hace siempre. Si fuera negativo, por ej. el -81.6, cogeríamos el -82)

Entonces, ¿qué se ha conseguido con esto? Pues una aproximación al nº que nos dan. Para tener el nº exacto entonces tendremos que hacer:

```
2.5675*10^15 = x*2^51
```

Calculamos 2^51 con la calculadora y nos sale:

```
2^51 = 2251799813685248
```

La x es el nº que multiplica a nuestra "aprox. " para que dé el nº exacto (el 2.5675*10^15). Despejamos x y nos queda que:

```
x = 2567500000000000/2^51=1.14019904629003576701
```

2º Ahora tenemos que pasar a binario el nº 1.14. Lo bueno de este método es que sólo tienes que hallar la parte decimal del nº (0.140199...) porque la parte entera es 1 y va a ser el bit implícito o el de ahorro para el IEEE754.

```
0.14019904629003576701(dec) = 0.00100011111001000001011(bin)
```

Con esto se obtiene la mantisa, ahora calculamos el exponente.
En la notación IEEE754 el exponente se pone en exceso, por lo que:

```
2^(n-1)-1 + exponente = 2^7 - 1 + 51 = 127 + 51 = 178
```

3º Ahora se pasa todo a la notación IEEE754:

El primer bit es el de signo, como es negativo se pone un 1. Los siguientes 8 bits son los del exponente, por lo que ponemos el 178 en binario en esos 8 bits y el resto (23 bits) son la mantisa que recuerda que se pone con ahorro de bit, eso quiere decir que el primer bit significativo de la mantisa lo omitimos. Quedaría así:

```
1 10110010 x00100011111001000001011
```

(en la x estaría un 1, pero como es el bit implícito lo quitamos).

Ahora agrupándolos de 4 bits en 4 bits tenemos el nº en hexadecimal

```
1101 1001 0001 0001 1111 0010 0000 1011

 D    9    1    1    F    2    0    B
```

Es decir:

```
-2.5675e15 (dec) = D911F20B (ieee754)
```

Ejemplo 5:

Antonio Bello

Convertir el número -0.01 al formato IEEE 754 de 32 bits:

1º Se coge el nº sin el menos, es decir positivo y escribimos lo siguiente:

    0.01 = 2^exponente

Nos interesa despejar el "exponente". ¿Cómo se hace esto? Con logaritmos:
La expresión simplificada quedaría:

    log(0.01)/log(2) = exponente

Esto se hace con la calculadora y listo. Se puede hacer con logaritmos o con logaritmos neperianos, lo que se prefiera. El exponente sale:

    exp = -6.643856...

(aproximamos al inmediatamente inferior, o sea al -7. Esto se hace siempre. Si fuera positivo, por ej. el 81.6, cogeríamos el 81)

Entonces, ¿qué se ha conseguido con esto? Pues una aproximación al nº que nos dan. Para tener el nº exacto entonces tendremos que hacer:

    0.01 = x*2^-7

Calculamos 2^-7 con la calculadora y nos sale:

    2^-7 = 0.0078125

La x es el nº que multiplica a nuestra "aprox. " para que dé el nº exacto (el 0.01). Despejamos x y nos queda que:

    x = 0.01/0.0078125=1.28

2º Ahora tenemos que pasar a binario el nº 1.28. Lo bueno de este método es que sólo tienes que hallar la parte decimal del nº (0.28) porque la parte entera es 1 y va a ser el bit implícito o el de ahorro para el IEEE754.

    0.28(dec) = 0.0100011110101110000101000(bin)

Con esto se obtiene la mantisa, ahora calculamos el exponente.
En la notación IEEE754 el exponente se pone en exceso, por lo que:

    2^(n-1)-1 + exponente = 2^7 - 1 - 7 = 127 - 7 = 120

3º Ahora se pasa todo a la notación IEEE754:

El primer bit es el de signo, como es negativo se pone un 1. Los siguientes 8 bits son los del exponente, por lo que ponemos el 178 en binario en esos 8 bits y el resto (23 bits) son la mantisa que recuerda que se pone con ahorro de bit, eso quiere decir que el primer bit significativo de la mantisa lo omitimos. Quedaría así:

    1 01111000 x01000111101011100001010

(en la x estaría un 1, pero como es el bit implícito lo quitamos).

Ahora agrupándolos de 4 bits en 4 bits tenemos el nº en hexadecimal

    1011 1100 0010 0011 1101 0111 0000 1010

```
    B    C    2    3    D    7    0    A
```

Es decir:

```
   -0.01 (dec) = BC23D70A (ieee754)
```

Ejemplo 6:
[Enrique Buitrón]

Representar el número -480 a coma flotante IEEE754

Las opciones son: A) C3E0 B) C3A0 C) C3F0 D) C3B0

1º/ Pasas 480 a binario: 480 = 111100000.

2º/ Como la repr. IEEE754 tiene bit implícito, colocas la coma después del primer 1, es decir:

```
   1.11100000   Como hemos corrido la coma 8 posiciones hacia la
```

izquierda, tenemos un exponente igual a 8.

3º/ El exponente está en exceso, por lo que sumamos 2^7-1 + 8 = 135

4º/ Lo representamos: El primer bit es el de signo, los 8 siguientes el exp, y los 7 restantes la mantisa.

```
   1 10000111 1.1110000 ---> quitamos el bit implicito y ya nos queda
```

la representación:

```
   1 10000111 1110000 = C 3 F 0
```

Ejemplo 7:
[Francisco Javier Alonso Álvarez]

Convertir el número C19E0000 en formato IEEE754 en su equivalente decinmal:

Primero paso hexadecimal a binario

```
   C19E0000 = 1100 0001 1001 1110 0000 0000 0000 0000
```

s (1) exponente (10000011) mantisa (001 1110 0000 0000 0000 0000)

la formula es

```
   (-1)^s * 1,mantisa * 2^(e-127)
```

s (1) significa, por tanto, signo negativo

exponente 10000011 es 131, como se representa en exceso a 2^(n-1)-1 hay que restar 127 para volver a tener el exponente real, es por tanto 4.

mantisa (hacia la derecha se van multiplicando los bits por 2^-1, 2^-2, 2^-3, 2^-4, ...)

```
   0*0,5
   0*0,25
   1*0,125
```

```
   1*0,0625
   1*0,03125
   1*0,015625
   -----------
   0,234375 en decimal

como está normalizada (se supone un 1 a la izquierda de la coma) es en
realidad 1,234375. Todo junto:

   (-1)^1* 1,234375 * 2^4 = -1,234375*16 = -19,75
```

Ejemplo 8:
Antonio Bello

Se desea normalizar el número fraccionario N=1000011000111010 representado en signo-magnitud sobre una palabra de 16 bits. El byte más significativo contiene la parte entera con signo,y el byte menos significativo la fraccionaria. Normalizarlo según la IEEE754, para 16 bits.

```
Es decir:

   Parte Entera: 10000110 (bin) = -6
   Parte Fracc.: 00111010 (bin) = 0,2265625

Si utilizamos el conversor de la página

   http://www.etsimo.uniovi.es/~antonio/uned/ieee754/IEEE-754.html

sabemos que la solución tendría que ser: C0C7 (IEEE754)

---

1º) Calcular el signo:

Vamos ahora paso a paso. Lo primero es quitar el signo que ya sabemos que
es negativo, entonces el número a normalizar sería:

   110,00111010

2º) Normalización de la mantisa:

Como es el formato IEEE 754 hay un bit implícito. Esto quiere decir que
vamos a correr la coma hasta el primer uno pero en lugar de dejarla a su
izquierda como se hace en otros formatos lo dejamos a la derecha (específico
para IEEE 754). Dicho bit implícito no será representado, con lo que ganamos
un bit más en la precisión de la mantisa.

Esto es:

   corremos coma   1,10,00111010
                    ^__|

la hemos corrido +2 posiciones. Lo tendremos en cuenta para después.

el bit implícito no lo representamos en la mantisa, luego
cogemos los 7 bits siguientes a la coma

   Mantisa: 1000111

3º) El exponente:

se calcula en base al exceso 2^(n-1) -1 (específico de IEEE 754)
```

```
Entonces tenemos : +2 (de la coma corrida) + 2^(n-1)-1=129= 10000001 (bin)
Luego la representación del número será:

   1 10000001 1000111
   _ \______/ \_____/
  Sig  Expon   Mantisa

Ojo en algunos exámenes los posibles resultados los dan en Hex.
Así que lo pasas de binario a Hex. Luego la solución es C0C7 (hex)
```

No dudes en escribir si tienes alguna pregunta sobre estos ejemplos

abello@uniovi.es