Organización de Computadoras 2023

Práctica 2

Sistema de Numeración en Punto Flotante

Objetivos de la práctica: que el alumno domine los tópicos de sistemas de numeración referidos a las representaciones en punto flotante, tales como:

- ***Representación e interpretación.***
- ***Operaciones aritméticas.***
- ***IEEE 754.***

Bibliografía:

- ***"Organización y Arquitectura de Computadores" de W. Stalling, capítulo 8.***
- ***Apunte de la Cátedra, "Sistemas de numeración: Punto flotante".***

1. Considerando el sistema de Punto Flotante cuya mantisa es fraccionaria, con 6 bits , está expresada en BSS (en el inciso a) o BCS (en el inciso b) y su exponente en BCS con 4 bits, escriba el significado de las siguientes cadenas de bits (mantisa a la izquierda):

Cadena	a) Mantisa en BSS	b) Mantisa en BCS
0101110110		
0000010000		
0000111001		
1111111111		
0000000000		
0000001111		
1111110000		
1000000000		
0000011111		

a) Mantisa fraccionaria, 6 bits, BSS, Exponente BCS 4 bits:

0101110110

Los 6 bits de más a la izquierda componen la mantisa, al ser fraccionaria, su forma será 0,010111, usar teorema fundamental de la numeración: 0,010111 = $(2^{-2} + 2^{-4} + 2^{-5} + 2^{-6})$

El exponente es en BCS 0110 => 6 en decimal; entonces:

0,010111 $x\ 2^{0110} = (2^{-2} + 2^{-4} + 2^{-5} + 2^{-6})\ x\ 2^{6}$

Producto de potencias de igual base, se suman los exponentes

$= 2^{4} + 2^{2} + 2^{1} + 2^{0} = 16 + 4 + 2 + 1 = 23$

0000010000

Mantisa:

0,000001 = (2^{-6})

El exponente es en BCS 0000 => 0 en decimal; entonces:

0,000001 $x\ 2^{0000} = (2^{-6})\ x\ 2^{0} = 2^{-6}$

Organización de Computadoras 2023

0000111001
Mantisa:

$0,000011 = (2^{-5} + 2^{-6})$

El exponente es en BCS 1001 => -1 en decimal; entonces:

$0,000011\ x\ 2^{1001} = (2^{-5} + 2^{-6})\ x\ 2^{-1} = 2^{-6} + 2^{-7} = 1/2^{6}+1/2^{7}=1/64+1/128 = (2+1)/128=3/128$

1111111111

Mantisa:
$0,111111 = (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6}) = 1 - 2^{-6} = (64-1) / 64 = 63/64$

El exponente es en BCS 1111 => -7 en decimal; entonces:

$0,111111\ x\ 2^{1111} = (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6})\ x\ 2^{-7} = 2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12} + 2^{-13}$

0000000000

$0,000000\ x\ 2^{0000} = 0\ x\ 2^{0} = 0$

b) Mantisa fraccionaria, 6 bits, BCS, Exponente BCS 4 bits:

0101110110

Los 6 bits de más a la izquierda componen la mantisa. El bit de más a la izquierda es el bit de signo, por tanto, lo separo. Y al ser fraccionaria, la coma está ubicada a la izquierda del primer bit. Usar teorema fundamental de la numeración:

$0\ 0,10111= + (2^{-1} + 2^{-3} + 2^{-4} + 2^{-5})$

El exponente es en BCS 0110 => 6 en decimal; entonces:

$0\ 10111\ 0110 = 0\ 0,10111\ x\ 2^{0110} = + (2^{-1} + 2^{-3} + 2^{-4} + 2^{-5})\ x\ 2^{6} = + (2^{5} + 2^{3} + 2^{2} + 2^{1}) = 32 + 8 + 4 + 2 = 46$

0000010000

$0\ 0,00001= + (2^{-5})$

El exponente es en BCS 0000 => 0 en decimal; entonces:

$0\ 00001\ 0000 => 0\ 0,00001\ x\ 2^{0000} = + (2^{-5})\ x\ 2^{0} = + 2^{-5}$

0000111001

El exponente es en BCS 1001 => -1 en decimal; entonces:

$0\ 00011\ 1001 => 0\ 0,00011\ x\ 2^{1001} = + (2^{-4} + 2^{-5})\ x\ 2^{-1} = + (2^{-5} + 2^{-6})$

1111111111

Mantisa:
$1\ 0,11111 = - (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5})$

El exponente es en BCS 1111 => -7 en decimal; entonces:

$0,11111\ \ x\ 2^{1111} = - (2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5})\ x\ 2^{-7} = - (2^{-8} + 2^{-9} + 2^{-10} + 2^{-11} + 2^{-12})$

0000000000

$0\ 0,00000\ x\ 2^{0000} = 0\ x\ 2^{0} = 0$

Organización de Computadoras 2023

Enlaces asociados :

Introducción a Punto Flotante parte 2 parte 3 parte 4
Ejemplo Ejemplo (parte 2) Ejemplo (parte 3)

2. Dado un sistema de Punto Flotante cuya mantisa es fraccionaria , está expresada en BCS con 5 bits y su exponente en BSS con 3 bits, interprete las siguientes cadenas considerando que la mantisa está sin normalizar, normalizada, o normalizada con bit implícito Identifique aquellas cadenas que no pueden ser interpretadas y mencione por que.

Cadena	Sin normalizar	Normalizada	Normalizada con Bit Implícito
01000111			
11000011			
00000000			
11111111			

a) Mantisa fraccionaria, 5 bits, BCS, Exponente BSS 3 bits:

01000111

Mantisa sin Normalizar:

Mantisa: 01000 => 0 0,1000 = + (2^{-1})

Exponente (BSS): 111 = 7 en decimal;

Entonces

01000 111 => 0 0,1000 x 2^{111} = + (2^{-1}) x 2^{7} = + (2^{6}) = 64

Mantisa Normalizada: ‘1’ después de la coma.

Mantisa: 01000 =>0 0,1000 = +(2^{-1})
Exponente (BSS): 111 = 7 en decimal

Entonces
01000 111 => 0 0,1000 x 2^{111} = + (2^{-1}) x 2^{7} = + (2^{6}) = 64

Mantisa Normalizada con bit implícito:

El ‘1’ después de la coma no se “ve” en la representación, pero se debe utilizar para el cálculo, lo agrego:

Mantisa: 01000 => 0 0,**1**1000 = + $(2^{-1} + 2^{-2})$

Exponente (BSS): 111 = 7 en decimal;

Entonces
01000 111 => 0 0,11000 x 2^{111} = + $(2^{-1} + 2^{-2})$ x 2^{7} = + $(2^{6} + 2^{5})$ = 64 + 32 = 96

11000011

Mantisa sin Normalizar:

Mantisa: 11000 => 1 0,1000 = − (2^{-1})

Exponente (BSS): 011 = 3 en decimal;

Entonces
11000 011 => 1 0,1000 $x\ 2^{011} = -(2^{-1})\ x\ 2^{3} = -(2^{2}) = -4$

Mantisa Normalizada:

Mantisa: 11000 => 1 0,1000 = - (2^{-1})
Exponente (BSS): 011 = 3 en decimal;

Entonces
11000 011 => 1 0,1000 $x\ 2^{011} = -(2^{-1})\ x\ 2^{3} = -(2^{2}) = -4$

Mantisa Normalizada con bit implícito:

El ‘1’ después de la coma no se “ve” en la representación, pero se debe utilizar para el cálculo, lo agrego:
Mantisa: 11000 => 1 0,**1**1000 = $-(2^{-1} + 2^{-2})$

Exponente (BSS): 011 = 3 en decimal;

Entonces
11000 011 => 1 0,11000 $x\ 2^{011} = -(2^{-1} + 2^{-2})\ x\ 2^{3} = -(2^{2} + 2^{1}) = -(4 + 2) - 6$

00000000

Mantisa sin Normalizar:

Mantisa: 00000 => 0 0,1000 = +0

Exponente (BSS): 000 = 0 en decimal;

Entonces
00000 000 => 0 0,0000 $x\ 2^{000} = +0$

Mantisa Normalizada:

No se puede calcular, pues la mantisa no empieza con 0,1...

Mantisa Normalizada con bit implícito:

El ‘1’ después de la coma no se “ve” en la representación, pero se debe utilizar para el cálculo, lo agrego:

Mantisa: 00000 => 0 0,**1**0000 = (2^{-1})

Exponente (BSS): 000 = 0 en decimal;

Entonces
00000 000 => 0 0,10000 $x\ 2^{000} = +(2^{-1})\ x\ 2^{0} = +(2^{-1})\ x\ 1 = +0,5$

11111111

Mantisa sin Normalizar:
Mantisa: 11111 => 1 0,1111 = $-(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$
Exponente (BSS): 111 = 7 en decimal;

Entonces

11111 111 => 1 0,1111 $x\ 2^{111} = -(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})\ x\ 2^{7} = -(2^{6} + 2^{5} + 2^{4} + 2^{3}) = -(64 + 32 + 16 + 8) = -120$

Mantisa Normalizada:

Mantisa: 11111 => 1 0,1111 = $-(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$

Exponente (BSS): 111 = 7 en decimal;

Entonces
11111 111 => 1 0,1111 x 2^{111}= − $(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4})$ x 2^7= − $(2^6 + 2^5 + 2^4 + 2^3)$ = −(64 + 32 + 16 + 8) = −120

Mantisa Normalizada con bit implícito:

El ‘1’ después de la coma no se “ve” en la representación, pero se debe utilizar para el cálculo, lo agrego:

Mantisa: 11111 => 1 0,**1**1111 = − $(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4}+ 2^{-5})$

Exponente (BSS): 111 = 7 en decimal;

Entonces
11111 111 => 1 0,11111 x 2^{111} = − $(2^{-1} + 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5})$ x 2^7 = − $(2^6 + 2^5 + 2^4 + 2^3 + 2^2)$ = −(64 + 32 + 16 + 8 + 4) = −124

Enlaces :
Mantisa normalizada **Normalización** **Normalización (parte 2)**

Normalizada con bit implícito **Normalizada con bit implícito BCS**

3. Calcule rango y resolución en extremos inferior negativo, superior negativo, inferior positivo y superior positivo para los siguientes sistemas de representación en punto flotante:
   a. Mantisa fraccionaria en BSS de 8 bits y exponente en BSS 4 bits
   b. Mantisa fraccionaria normalizada en BSS de 15 bits y exponente en CA1 10 bits
   c. Mantisa fraccionaria normalizada con bit implícito en BCS de 15 bits y exponente en Exceso 5 bits
   d. Mantisa fraccionaria normalizada con bit implícito en BCS de N bits y exponente en CA2 de M bits

Observe que:
- En las mantisas BSS no se puede expresar números negativos, con lo que aun con exponente negativo expresaremos un número positivo por un factor de escala menor a 1, pero también positivo. Ejemplo: 2 x 2^{-4} = 0,125.
- Las mantisas fraccionarias suponen el punto al principio de la mantisa.
- Los exponentes negativos indican factores de escala menores a 1 que mejoran la resolución.
- Mantisa normalizada implica que empieza con 1, o sea mantisa mínima 0,1 para la fraccionaria, igual a 0,5 en decimal. Esto hace que no se pueda representar el 0.
- Mantisa normalizada con bit implícito, significa agregar un 1 al principio de la misma al interpretarla. Ejemplo: 00000 se interpreta 0,100000, o 0,5 en base 10.

a. <u>Mantisa fraccionaria en BSS de 8 bits y exponente en BSS 4 bits</u>

N1 = 0 (el número más chico en este sistema, minimizando la mantisa y exponente)
N2 = 0,11111111 x 2^{1111} =$(1 - 2^{-8})$ x 2^{15}=$(2^{-1}+ 2^{-2} + 2^{-3} + 2^{-4} + 2^{-5} + 2^{-6} + 2^{-7} + 2^{-8})$ x2^{15} (el más grande)

¿Por que 0,11111111 = $(1 - 2^{-8})$?

Pues $(1 - 2^{-8})$ = 1,00000000 - 0,00000001 = 0,11111111

Entonces el rango es: = [N1; N2] = **[0; (1 – 2^{-8}) x 2^{15}]**

La resolución (recordar, diferencia entre 2 números consecutivos), ya no es fija en todo el intervalo, debemos calcularla en cada sector del intervalo de representación:

R1= Resolución extremo inferior: debemos obtener el siguiente número representable en este sistema que sigue al cero, lo llamaremos N3. Para ello, debemos minimizar mantisa al número posterior al cero y minimizar el exponente.

$N3 = 0{,}00000001 \times 2^{0000} = 2^{-8}$ (el más cercano a N1)

Entonces para obtener la resolución en el extremo inferior positivo debo restarle a N3 el valor de N_1

R_1= Resolución extremo inferior = $N3 – N1 = (0{,}00000001 – 0) \times 2^{0000} = \mathbf{2^{-8}}$

R_2= Resolución extremo superior: debemos obtener el número representable en este sistema anterior al más grande, lo llamaremos N_4. Para ello, debemos maximizar mantisa al número anterior a la mantisa máxima y maximizar el exponente.

$N4 = 0{,}1111111\mathbf{0} \times 2^{1111}$ (el más cercano a N2)

Entonces para obtener la resolución en el extremo superior positivo debo restarle a N_2 el valor de N_4

R_2= Resolución extremo superior = $N_2 – N_4 = (0{,}11111111 – 0{,}11111110) \times 2^{15} = 2^{-8} \times 2^{15} = \mathbf{2^7}$

b. <u>Mantisa fraccionaria normalizada en BSS de 15 bits y exponente en CA1 10 bits</u>

Como la mantisa es BSS, los números representables son todos positivos. Para obtener el mínimo, minimizo mantisa (normalizada) y exponente, el cual debe ser el máximo negativo en ca1.

$N_1 = 0{,}100000000000000 \times 2^{1000000000} = 2^{-1} \times 2^{-511} = 0{,}5 \times 2^{-511}$

Se observa que el cero no es representable en este sistema, por estar la mantisa normalizada.

Para obtener el máximo, maximizo la mantisa (en BSS) y el exponente (en Ca1),

$N_2 = 0{,}111111111111111 \times 2^{0111111111} = (1- 2^{-15}) \times 2^{+511}$

Entonces el rango es: = $[N_1; N_2]$ = $\mathbf{[0{,}5 \times 2^{-511}; (1 - 2^{-15}) \times 2^{+511}]}$

Para la resolución en el extremo inferior, debo obtener el número siguiente al mínimo, lo llamamos N3; para ello tomo la siguiente mantisa y utilizo el mismo exponente que para representar ese número

$N_3 = 0{,}100000000000001 \times 2^{1000000000} = (2^{-1} + 2^{-15}) \times 2^{-511}$

R_1= Resolución extremo inferior = $N3 – N1 = (0{,}100000000000001- 0{,}100000000000000) \times 2^{1000000000}$

$= 0{,}000000000000001 \times 2^{1000000000} = \mathbf{2^{-15} \times 2^{-511}}$

Para la resolución extremo superior debemos obtener el número representable en este sistema anterior al más grande, lo llamaremos N_4. Para ello, debemos maximizar mantisa al número anterior a la mantisa máxima y maximizar el exponente:

$N_4 = 0{,}111111111111110 \times 2^{0111111111}$

R_2= Resolución extremo superior = $N_2 – N_4 = (0{,}111111111111111 – 0{,}111111111111110) \times 2^{0111111111}$

$= 0{,}000000000000001 \times 2^{0111111111} \mathbf{= 2^{-15} \times 2^{+511}}$

c. <u>Mantisa fraccionaria normalizada con bit implícito en BCS de 15 bits y exponente en Exceso 5 bits</u>

En este sistema tenemos valores positivos y negativos.
Para obtener el mínimo positivo (N1), colocamos la mantisa mínima positiva (primer bit en cero por ser BCS), pero asumiendo un dígito en 1 de más después de la coma por estar normalizada con bit implícito:

Exponente en exceso = 10000 + 10000 = 00000 (-16 en exceso)

$N_1 = 0\ 0{,}100000000000000 \times 2^{00000} = + (2^{-1}) \times 2^{-16} = \mathbf{+ 0{,}5 \times 2^{-16}}$ (el más chico +)

Para obtener el máximo positivo (N_2), maximizo mantisa (recordando bit implícito) y exponente

N_2 = 0 0,111111111111111 x 2^{11111} = **+ (1 - 2^{-15}) x 2^{+15}** (el más grande +)

Para la resolución en el extremo inferior obtengo el número positivo más cercano a N_1

N_3 = 0 0,100000000000001 x 2^{00000} = + (2^{-1} + 2^{-15}) x 2^{-16} (el más cercano a N_1)

$R_1 = N_3 - N_1 = (2^{-1} + 2^{-15} - 2^{-1}) \times 2^{-16} = \mathbf{(2^{-15}) \times 2^{-16} = 2^{-31}}$

Para la resolución en el extremo superior obtengo el número positivo más cercano a N_2

N_4 = 0 0,111111111111110 x 2^{11111} (el más cercano a N_2)

$R_2 = N_2 - N_4$ = (0,111111111111111-0,111111111111110) x 2^{+15} = 2^{-15} x 2^{+15} = 2^0 = 1

Como tenemos números negativos debemos calcular los valores negativos mínimo y máximo:

Mínimo negativo en módulo:
N_5 = 1 0,100000000000000 x 2^{00000} = - (2^{-1}) x 2^{-16} = - 0,5 x 2^{-16}

Máximo negativo en módulo:
N_6 = 1 0,111111111111111 x 2^{11111} = - (1 - 2^{-15}) x 2^{+15}

En este caso, los valores que se obtienen para las resoluciones en los extremos mínimo y máximo negativo son los mismos que los positivos, en módulo.

Rango : [- (1 - 2^{-15}) x 2^{+15}; + (1 - 2^{-15}) x 2^{+15}]

Enlaces :
Rango y Resolución **Parte 2** **Parte 3** **Parte 4** **Anexo**

Rango y Resolución (normalizada) **Rango y Resolución (con bit implícito)**

4. Dado un sistema de Punto Flotante cuya mantisa es fraccionaria , está expresada en BCS con 10 bits y su exponente en CA2 con 5 bits, obtenga la representación de los siguientes números, considerando que la mantisa está sin normalizar, normalizada, o normalizada con bit implícito

Cadena	Sin normalizar	Normalizada	Normalizada con Bit Implícito
0			
1			
9			
-5,0625			
34000,5			
0,015625			
Nº máximo			

0

Sin normalizar

0 = 0 0,000000000 x 2^E **(el exponente puede ser cualquiera),**

Normalizada

0 = **No se puede representar**. El cero no está en el rango.

Normalizada con bit implícito

0 = **No se puede representar**. El cero no está en el rango.

Organización de Computadoras 2023

+1

Sin normalizar

Tenemos distintas representaciones del valor 1

Por cada decremento la mantisa (es decir por cada corrimiento de la coma a la izquierda de la cadena) debo incrementar en 1 el exponente:

1 = 0 0,100000000 x 2^{00001} = 0 0,010000000 x 2^{00010} =0 0,001000000 x 2^{00011}

Represento el signo (que lo separo de los bits de la mantisa), luego los 5 bits del exponente y finalmente los 9 bits restantes de la mantisa:

0 00011 100000000

Normalizada

1 = 0 0,100000000 x 2^{00001}

0 00001 100000000

Normalizada con bit implícito

1 = 0 0,1000000000 x 2^{00001}

Para el cálculo de la mantisa tengo 10 bits (por el bit implícito, que al representar la cadena no muestro), más el bit de signo.

0 00001 000000000

+9

Sin normalizar

Varias representaciones:

9 = 0 0,100100000 x 2^{00100} = + (2^{-1}+2^{-4}) x 2^4 = + (0,5 + 0,0625) x 16 = 9

9 = 0 0,010010000 x 2^{00101} = + (2^{-2}+2^{-5}) x 2^5 = + (0,25 + 0,03125) x 32 = 9

9 = 0 0,001001000 x 2^{00110} = + (2^{-3}+2^{-6}) x 2^6 = + (0,125 + 0,015625) x 64 = 9, represento este último:

0 00110 001001000

Normalizada

9 = 0 0,100100000 x 2^{00100} = + (2^{-1}+2^{-4}) x 2^4 = + (0,5 + 0,0625) x 16 = 9

0 00100 100100000

Normalizada con bit implícito (recordar que el bit implícito se usa para el cálculo pero no para la representación de la cadena)

9 = 0 0,1001000000 x 2^{00100} = + (2^{-1}+2^{-4}) x 2^4 = + (0,5 + 0,0625) x 16 = 9

0 00100 001000000

-5,0625

Sin normalizar

- 5,0625 =1 101,000100 x 2^0 = 1 0,101000100 x 2^{00011} = - (2^{-1}+2^{-3}+2^{-7}) x 2^3 = -0,6328125 x 8 = - 5,0625

- 5,0625 =1 101,000100 x 2^0 = 1 0,010100010 x 2^{00100} = - (2^{-2}+2^{-4}+2^{-8}) x 2^4 = - 0,31640625 x 16 = - 5,0625

- 5,0625 =1 101,000100 x 2^0 = 1 0,001010001 x 2^{00101} = - (2^{-3}+2^{-5}+2^{-9}) x 2^5 = - 0,158203125 x 32 = - 5,0625 represento este último:

1 00101 001010001

Normalizada

- 5,0625 =1 101,000100 x 2^0 =1 0,101000100 x 2^{00011} = - (2^{-1}+2^{-3}+2^{-7}) x 2^3 = -0,6328125 x 8 = - 5,0625

1 00011 101000100

Normalizada con bit implícito

- 5,0625 =1 0,1010001000x 2^{00011} = - (2^{-1}+2^{-3}+2^{-7}) x 2^3 = -0,6328125 x 8 = - 5,0625

1 00011 010001000

34000,5

Sin normalizar

El más grande = 0 0,111111111 x 2^{01111} = + (1 – 2^{-9}) x 2^{15} = + (0,998046875) x 32768 = + 32704 Por lo tanto 34000,5 está fuera de rango.

Normalizada

El más grande = 0 0,111111111 x 2^{01111} = + (1 – 2^{-9}) x 2^{15} = + (0,998046875) x 32768 = + 32704
Por lo tanto 34000,5 está fuera de rango.

Normalizada con bit implícito

El más grande = 0 0,1111111111 x 2^{01111} = + (1 – 2^{-10}) x 2^{15} = + (0,9990234375) x 32768 = + 32736
Por lo tanto 34000,5 está fuera de rango.

0,015625

Aplicando el cálculo para pasar de decimal a binario:

0,015625 x 2 = 0,03125
0,03125 x 2 = 0,0625
0,0625 x 2 = 0,125 0,125 x
2 = 0,25
0,25 x 2 = 0,5
0,5 x 2 = 1,0
0,0 x 2 = 0,0 Me quedo con los valores enteros de los resultados 0000010

Sin normalizar

+ 0,015625 = 0 0,000001000 x 2^0 = + (2^{-6}) x 2^0 = + 0,015625 x 1 = + 0,015625

+ 0,015625 = 0 0,000010000 x 2^{-1} = + (2^{-5}) x 2^{-1} = + 0,03125 x 0,5 = + 0,015625

+ 0,015625 = 0 0,000100000 x 2^{-2} = + (2^{-4}) x 2^{-2} = + 0,0625 x 0,25 = + 0,015625

0 11110 000100000

Normalizada

+ 0,015625 = 0 0,100000000 x 2^{-5} = 0 0,100000000 x 2^{11011} = + 0,5 x 0,03125 = + 0,015625

0 11011 100000000

Normalizada con bit implícito

+ 0,015625 = 0 0,1000000000 x 2^{-5} = 0 0,100000000 x 2^{11011} =+ 0,5 x 0,03125 = + 0,015625

0 11011 000000000

Nº máximo

Sin normalizar

El más grande = 0 0,111111111 x 2^{01111} = + (1 – 2^{-9}) x 2^{15} = + (0,998046875) x 32768 = + 32704

0 01111 111111111

Normalizada

El más grande = 0 0,111111111 x 2^{01111} = + (1 – 2^{-9}) x 2^{15} = + (0,998046875) x 32768 = + 32704

0 01111 111111111

Normalizada con bit implícito

El más grande = 0 0,1111111111 x 2^{01111} = + (1 – 2^{-10}) x 2^{15} = + (0, 9990234375) x 32768 = + 32736

0 01111 111111111

Nº mínimo

Sin normalizar

El más chico (“negativo más grande”) = 1 0,111111111 x 2^{01111} = - (1 – 2^{-9}) x 2^{15} = - (0,998046875) x 32768 = - 32704

1 01111 111111111

Normalizada

El más chico (“negativo más grande”) = 1 0,111111111 x 2^{01111} = - (1 – 2^{-9}) x 2^{15} = - (0,998046875) x 32768 = - 32704

1 01111 111111111

Normalizada con bit implícito

El más chico (“negativo más grande”) = 1 0,1111111111 x 2^{01111} = + (1 – 2^{-10}) x 2^{15} = + (0, 9990234375) x 32768 = + 32736

Enlaces :

Decimal a IEEE 754 **Parte 2** **Parte 3**

5. Diga cómo influyen las siguientes variantes en el rango y resolución: a.
   Mantisa con signo y sin signo.
   b. Exponente con signo y sin signo.
   c. Tamaño de mantisa. d.
   Tamaño de exponente.
   e. Mantisa fraccionaria, fraccionaria normalizada y fraccionaria normalizada con bit implícito.

Enlaces :
Mantisa normalizada **Rango y Resolución (normalizada)**

Normalizada con bit implícito **Rango y Resolución (con bit implícito)** **Normalizada con bit implícito BCS**

6. Efectúe las siguientes sumas para un sistema de punto flotante con mantisa BSS de 8 bits y exponente en BCS 8 bits.

00001111 00000011 + 00001000 00000010 = ¿?
01111111 00000000 + 11111100 10000001 = ¿?
00000001 00000111 + 00011100 00000000 = ¿?

Observe que los factores de escala deben ser los mismos, sino sumaríamos dos mantisas con pesos distintos (recordar que se puede correr los unos y sumar o restar este corrimiento al exponente para obtener una cadena equivalente).

00001111 00000011 + 00001000 00000010 = 15 x 2^3 + 8 x 2^2 = 120 + 32 = 152
Para hacer la suma, debemos igualar los exponentes y correr la coma de la mantisa hacia la derecha o izquierda según el caso:

Si incremento exponente => decremento mantisa, es decir corro coma a la izquierda
Si decremento exponente => incremento mantisa, es decir corro coma a la derecha
Tener en cuenta que al hacer el corrimiento de mantisa no se pierda la representación exacta del número dado.

En este caso, vamos a incrementar el exponente de la segunda cadena:

a) 00001000 00000010 = 8 x 2^2 = 4 x $\mathbf{2^3}$ 00000100 00000011

00001111 x 2^3
\+ 00000100 x 2^3
00010011 x 2^3 = 19 x 2^3 = 19 x 8 = 152

b) 01111111 00000000 + 11111100 10000001 = 127 x 2^0 + 252 x 2^{-1} = 127 + 126 = 253

Incrementamos exponente 2da cadena, decrementamos mantisa:

111111

01111111 x 2^0
\+ 01111110 x 2^0
11111101 x 2^0 = 253 x 2^0 = 253

c) 00000001 00000111 + 00011100 00000000 = 1 x 2^7 + 28 x 2^0 = 128 + 28 = 156

Decrementamos exponente 1ra cadena en 7, incrementamos mantisa con 7 corrimientos:

10000000 x 2^0
\+ 00011100 x 2^0
10011100 x 2^0 = 156 x 2^0 = 156

7. Suponiendo que los números que no son representables se aproximan al más próximo, obtenga las representaciones o aproximaciones de los números 8,625; 0,4 y 2,5 en los sistemas:
a. Mantisa fraccionaria normalizada de 5 bits BSS exponente 4 bits CA2 b.
Mantisa fraccionaria normalizada de 10 bits BCS exponente 3 bits CA2

Organización de Computadoras 2023

8,625

a. Mantisa fraccionaria normalizada de 5 bits BSS exponente 4 bits CA2

8,625 = 1000,101 x 2^0 = 0,1000101 x 2^4 Pero la mantisa tiene sólo 5 bits, no se puede representar completa; calculamos el número más aproximado:

0,10001 x 2^4 = $(2^{-1} + 2^{-5})$ x 2^4 = **8,5** 0100 10001

El Nº que le sigue = (0,10001 + 0,00001) x 2^4 = 0,10010 x 2^4 = $(2^{-1} + 2^{-4})$ x 2^4 = **9**

Error1 = 8,625 – 8,5 = 0,125

Error2 = 9 – 8,5 = 0,5 => Menor error (valor más cercano) es 8,5

8,625 no tiene una representación exacta en este sistema. 8,5 está más cerca que 9.

b. Mantisa fraccionaria normalizada de 10 bits BCS exponente 3 bits CA2

8,625 = 0 1000,101 x 2^0 = 0 0,100010100 x 2^4

El exponente 4 no se puede expresar con 3 bits en Ca2

El número más grande = 0 0,111111111 x 2^{+3} = + $(1 – 2^{-9})$ x 8 = (0,998046875) x 8 = 7,984375 Éste sería el más cercano a 8,625.
0 011 111111111

0,4

a. Mantisa fraccionaria normalizada de 5 bits BSS exponente 4 bits CA2

0,4 x 2 = 0,8
0,8 x 2 = 1,6
0,6 x 2 = 1,2
0,2 x 2 = 0,4
0,4 x 2 = 0,8
0,8 x 2 = 1,6

0,4 = 0,01100 x 2^0 corriendo la coma a derecha (es decir incrementando mantisa) entra un dígito más (decremento exponente)

0,11001 x 2^{-1} = 0,11001 x 2^{1111} = 0,390625

El que sigue = 0,11010 x 2^{-1} = (0,5+0,25+0,0625) x 0,5 = 0,40625

Error1 = 0,4 – 0,390625 = 0,009375
Error2 = 0,40625 – 0,4 = 0,00625 => Esta representación es más cercana a 0,4.

b. Mantisa fraccionaria normalizada de 10 bits BCS exponente 3 bits CA2

0,4 x 2 = 0,8
0,8 x 2 = 1,6
0,6 x 2 = 1,2
0,2 x 2 = 0,4
0,4 x 2 = 0,8
0,8 x 2 = 1,6
0,6 x 2 = 1,2
0,2 x 2 = 0,4
0,4 x 2 = 0,8

0,4 = 0 0,011001100 x 2^0

Normalizada = 0 0,110011001 x 2^{-1} = $(2^{-1}+2^{-2}+2^{-5}+2^{-6}+2^{-9})$ x 0,5 = 0,399414062

El que sigue = 0 0,110011010 x 2^{-1} = (2^{-1}+2^{-2}+2^{-5}+2^{-6}+2^{-8}) x 0,5 = 0,400390625

Error1 = 0,4 – 0,399414062 = 0,000585938
Error2 = 0,400390625 – 0,4 = 0,000390625 => 0,400390625 está más cerca de 0,4 (menor error)

2,5

a. Mantisa fraccionaria normalizada de 5 bits BSS exponente 4 bits CA2

2,5 = 10,1 x 2^0 = 0,101 x 2^2 = 0,10100 x 2^{0010} = (2^{-1} + 2^{-3}) x 2^2 = (0,5+0,125) x 4 = 2,5

0010 10100

b. Mantisa fraccionaria normalizada de 10 bits BCS exponente 3 bits CA2

2,5 = 0 0,101000000 x 2^{010} = (2^{-1} + 2^{-3}) x 2^2 = (0,5+0,125) x 4 = 2,5

0 010 101000000

8. Definimos Error Absoluto y Error Relativo de un número **x** en un sistema de la siguiente forma:
EA(x) = | x' – x | y ER(x) = EA(x) / x ; donde x' es el número representable del sistema más próximo a x.
Calcule los errores absolutos y relativos para los casos del ejercicio anterior.

8,625

a) Mantisa fraccionaria normalizada de 5 bits BSS exponente 4 bits CA2

Como vimos anteriormente, los valores más aproximados representables en este sistema eran 8,5 y 9.

8,5 < 8,625 < (2^{-1} + 2^{-5}) x 2^4 = 9   8,5 es el más próximo;

EA = 8,625 – 8,5 = 0,125 (Menor error)

ER = EA/Nº a representar = 0,125/8,625 ~ 0,0145

b) Mantisa fraccionaria normalizada de 10 bits BCS exponente 3 bits CA2

En este caso se obtuvo que el número más cercano representable es el 7,984375

EA = 8,625 – 7,984375 = 0,640625

ER = EA/Nº a representar = 0,640625 /8,625 ~ 0,0742753623

0,4

a) 0,390625 < 0,4 < 0,40625
E_A = 0,40625 – 0,4 = 0,00625 (menor error)
E_R = 0,00625/0,4 = 0,015625

b) 0,399414062 < 0,4 < 0,400390625
E_A = 0,400390625 – 0,4 = 0,000390625 (menor error)
E_R = 0,000390625/0,4

2,5

a) E_A = 0   Representación exacta.
b) E_R = 0   Representación exacta.

Organización de Computadoras 2023

Enlaces :
Error absoluto y error absoluto máximo **Error relativo**

9. Considerando que en los procesos de truncamiento o redondeo la elección se basa en la representación más cercana, estime el Error Absoluto Máximo cometido en las representaciones del ejercicio 1. Recuerde que la distancia entre 2 representaciones sucesivas se conoce como resolución (R), por lo que EA máx ≤ R / 2.

Enlaces :
Error absoluto y error absoluto máximo

10. Tome un sistema de punto flotante cualquiera y dibuje la forma del gráfico de cada tipo de error en función del número que se quiere representar.

11. Detalle las características del estándar IEEE 754 para simple precisión y doble precisión

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 (M y S), 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):

La mantisa es fraccionaria normalizada, con la coma después del primer bit que es siempre uno (1,), en M y S

El exponente se representa en exceso.

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)

Enlaces :
IEEE 754 **Interpretación** **Interpretación (nº normalizado)** **Interpretación (nº denormalizado)**

12. ¿Qué valores están representados por las siguientes cadenas si responden al estándar IEEE 754?

0 11000100 00000000000000000000000
1 11111110 10100000000000000000000
0 00000000 00000000000000000000001
0 00000000 10011000000000000000000
1 00000000 00000000000000000000000
0 11111111 00000000000000000000000
0 11111111 00000100000000000000000
0 01100010100 0000000000000000000000000000000000000000000000000000
0 11010101110 1010000000000000100000000000000000000000000000000000
0 00000000000 0101000000000000000000000000000000000000000000000000
1 11111111111 1111100000000000000000000000000000000000000000000000

0 11000100 00000000000000000000000

Exponente: en 8 bits, en exceso 128 – 1 = 127
Al valor le restamos el exceso:

11000100 => (128 + 64 + 4) – 127 = 196 - 127 = 69

Mantisa: recordar que se antepone 1, y valor, en este caso 1,0

Entonces la cadena representa el valor **+ 1,0 2^{+69}**

1 11111110 10100000000000000000000

Exponente
11111110 => 254 – 127 = 127

Mantisa: 1 + 0,5 + 0,125 = 1, 625

Signo negativo,

Entonces la cadena representa el valor **-1,625 x 2^{+127}**

0 00000000 00000000000000000000001 = +(2^{-23}) x 2^{-126}

1 00000000 00000000000000000000000 = -0

1 11111111 00000000000000000000000 = + ∞

1 11111111 00000100000000000000000 = NAN

Enlaces :
IEEE 754 **Interpretación** **Interpretación (nº normalizado)** **Interpretación (nº denormalizado)**

13. Hallar la representación en simple precisión del estándar IEEE 754 de los siguientes números **1; 13; 257; -40000; 0,0625**

Escribo la mantisa en binario y luego desplazo la coma hasta el primer 1 de la izquierda (achico mantisa y elevo exponente)

- 40000 = 1 1001110001000000 x 2^{0} = 1 1,00111000100000000000000 x 2^{15} =

Exponente = 15 + 127 = 142 => exponente 10001110

= 1 1,00111000100000000000000 x $2^{10001110}$

Entonces la cadena es **1 10001110 00111000100000000000000**

0,0625 = 0 0,0001000000.......00 x 2^{0} = 0 1,00000..........000 x 2^{4} = 0 1,000000 000 x $2^{10000011}$

Exponente = 4 + 127 = 131 = 10000011 = (+4 en exceso 127)

Entonces la cadena es **0 10000011 00000000000000000000000**

+1 = 0 1,00000………..000 x 2^0 = 0 1,00000….000 x $2^{01111111}$

Exponente = 0 +127 = 127 = 01111111 = (0 en exceso 127)

Entonces la cadena es **0 01111111 00000000000000000000000**

+13 = 0 1101,00000…000 x 2^0 = 0 1,101000….000 x 2^{+3} = 0 1,101000….000 x $2^{10000010}$

Exponente = 3 +127 = 130 = 10000010 = (+3 en exceso 127)

Entonces la cadena es **0 10000010 10100000000000000000000**

Enlaces :
Decimal a IEEE 754 **Parte 2** **Parte 3**

14. Calcule rango y resolución en extremos inferior negativo y superior positivo para los sistemas de simple precisión y doble precisión del estándar IEEE 754. ¿Cuál es el menor número positivo distinto de ‘0’ que se puede representar?

Enlace:
Rango

15. Efectúe las siguientes sumas (las cadenas son representaciones en el estándar IEEE 754)

00001111 010000000000000000000000 + 00010000 010000000000000000000000 = ¿?
11111111 101010101010101010101010 + 11111100 100000011111000001101010 = ¿?

16. En el estándar IEEE 754, ¿para qué sirve, cuando el exponente es 0 y la mantisa no es nula, que la mantisa no esté normalizada?

Enlace:
Interpretación (nº denormalizado)