Ejercicio 1. Indique la representación de los siguientes números, razonando su respuesta:

a) -16 en complemento a 2 con 5 bits
b) -16 en complemento a 1 con 5 bits
c) +13 en signo magnitud con 5 bits
d) -14 en complemento a dos con 5 bits

El rango de representación de números en complemento a dos con 5 bits es de -16 a +15. Con n bits, el rango va desde -2^{n-1} hasta $2^{n-1}-1$, por lo que para 5 bits, es desde -2^4 hasta 2^4-1

Solución:

a) El rango de representación de números en complemento a dos con 5 bits es [-25-1..25-1-1] = [-16..15]. 16 en binario es 10000. Tenemos que complementar, 01111 y sumarle 1. Por tanto, -16 en complemento a dos con 5 bits es 10000.
b) El rango de representación de números en complemento a dos con 5 bits es [-25-1..25-1-1] = [-15..15]. Por tanto, el número -16 no se puede representar.
c) 13 en binario puro es 1101. En signo magnitud se introduce al comienzo un bit de signo, en este caso 0 para indicar que es positivo. El número 13 es signo magnitud con 5 bits es 01101.
d) 14 en binario puro con 5 bits es 01110. Se complementa, 10001 y se suma 1, con lo que se obtiene 10010.

Ejercicio 2. Indique la representación de los siguientes números:

a) -64 en complemento a uno con 7 bits
b) -64 en complemento a dos con 7 bits
c) 12 en signo magnitud con 6 bits
d) 18 en complemento a dos con 5 bits

Solución:

a) -64 en complemento a uno con 7 bits
El rango de representación es [-63, 63], por tanto, el número no es representable
b) -64 en complemento a dos con 7 bits
64 en binario con 7 bits es 1000000. Se complementa 0111111 y se suma 1, siendo el resultado 100000
c) 12 en signo magnitud
con 6 bits 001100
d) 18 en complemento a dos con 5 bits
El rango de representación es [-16, 15], por tanto, el número no es representable.

Ejercicio 3. ¿Cómo se detecta un desbordamiento en Complemento a 2 cuando se hace una operación de suma?

Solución:

Se detecta cuando los dos operandos tienen el mismo signo y el resultado tiene signo distinto al de los operandos.

Ejercicio 4. Represente en el estándar IEEE 754 de simple precisión el valor -36.

Solución:

El valor 36 en binario es 100100. $100100 = 1.00100 \times 2^5$. Por tanto:

- El bit de signo es 1, porque el número es negativo.
- El exponente es 5, por tanto el exponente que se almacena es 5 + 127 = 132, que en binario es 10000100
- La mantisa es 001000000 …. 00000

Por tanto, el número -36 se representa como 11000010000100000000000000000000 en binario.

Ejercicio 5. Indique el valor decimal del siguiente número hexadecimal 0x00600000 que representa un número en coma flotante según IEEE 754 (precisión simple)

Solución:

0x00600000 en binario es 00000000011000000000000000000000

Signo = 0, número positivo
Exponente = 00000000
Mantisa = 1100000…0000

Se trata de un número no normalizado cuyo valor es $0,11 \times 2^{-126} = 0,75 \times 2^{-126}$

Ejercicio 6. Represente el número -24,50 utilizando el estándar de coma flotante de simple precisión IEEE 754. Exprese dicha representación en binario y en hexadecimal.

Solución:

$24,5_{(10} = 11000.1_{(2} = 1,10001 \times 2^4$

Signo = 1, número negativo
Exponente = 4 + 127 = 131 = 10000011
Mantisa = 1000100000 .. 00000

En binario es: 11000000111000100000 …….00000 En Hexadecimal es: 0xC1C40000

Ejercicio 7. Se desea representar números enteros dentro del rango -8191…8191. Indicar de forma razonada:

a) ¿Cuál es el número de bits que se necesita si se quiere utilizar una representación en complemento a uno?
b) ¿Cuál es el número de bits que se necesita si se quiere utilizar una representación en signo-magnitud?

Solución:

Se necesitan en los dos casos 14 bits. Con 14 bits el rango de representación en ambos casos es de $-(2^{13}-1)$ …. $2^{13}-1$ = -8191 … 8191

Ejercicio 8. ¿Cuál es el número positivo normalizado más pequeño que se puede representar utilizando el estándar de simple precisión IEEE 754? Justifique su respuesta. Indique también el número positivo no normalizado más pequeño que se puede representar. Justifique de igual forma su respuesta.

Solución:

a) El número positivo normalizado más pequeño representable en el estándar IEEE 754 (32 bits) es:
   0 00000001 00000000000000000000000 positivo normalizado más pequeño
   Cuyo valor es $1.0 \times 2^{1-127} = \mathbf{2^{-126}}$

b) El número positivo no normalizado más pequeño representable en el estándar es:
   0 00000000 00000000000000000000001 positivo no normalizado más pequeño
   Cuyo valor es $2^{-23} \times 2^{-126} = \mathbf{2^{-149}}$

Ejercicio 9. Represente en el estándar de coma flotante IEEE 754 de 32 bits los valores 10,25 y 6,75. Exprese el resultado final en hexadecimal.

Solución:

a) $10{,}25_{(10} = 1010{,}01_{(2} = 1.01001 \times 2^3{}_{(2}$ Signo = 0

   Exponente = $127+3 = 130_{(10} = 10000010_{(2}$

   Mantisa = 010010000…00

   En Binario: 0100 0001 0010 0100 00…00
   En Hexadecimal: 0x41240000

   $6{,}75_{(10} = 110{,}11_{(2} = 1.1011 \times 2^2{}_{(2}$
   Signo = 0
   Exponente = $127+2 = 129_{(10} = 10000001_{(2}$
   Mantisa = 10110000…00

   En Binario: 0100 0000 1101 1000 00…00
   En Hexadecimal: 0x40D80000

Ejercicio 10. Indique el valor decimal del siguiente número representado en el estándar IEEE 754 de simple precisión: 0xBF400000.

Solución:

El valor decimal de 0xBF400000 es el siguiente:

En Binario: 1011 1111 0100 0000…00

Signo= Negativo
Exponente = $01111110_{(2} = 126_{(10}$ = Exponente = -1
Mantisa = 1

Número en binario: $-1{,}1 \times 2^{-1}{}_{(2} = -0{,}11_{(2} = -0{,}75_{(10}$

Ejercicio 11. En relación al estándar IEEE 754 responda a las siguientes preguntas:

a) En una representación IEEE 754 de 32 bits, ¿cuál es el número de valores que hay comprendidos entre el número 8 y el 9?

Solución:

a) Para poder calcular el número de valores comprendidos entre el 8 y el 9 es necesario representar ambos números en coma flotante:

$8_{(10} = 1000_{(2} = 1.000 \times 2^{3}{}_{(2}$

Signo = 1
Exponente = 127+3 = $130_{(10} = 10000010_{(2}$
Mantisa = 000000…. 000000 (23 bits)

$9_{(10} = 1001_{(2} = 1.001 \text{ x } 2^{3}{}_{(2}$

Signo = 1
Exponente = 127+3 = $130_{(10} = 10000010_{(2}$
Mantisa = 001000…. 000000 (23 bits)

La representación del valor 8 es: 0 10000010 00000000000000000000000
La representación del valor 9 es: 0 10000010 00100000000000000000000

Lo único que hay que hacer es calcular el número de valores comprendidos entre ambas representaciones. El primer bit (comenzando desde la derecha) con un valor distinto es el que se encuentra en la posición 21, por tanto el número de valores es 2^{20}.