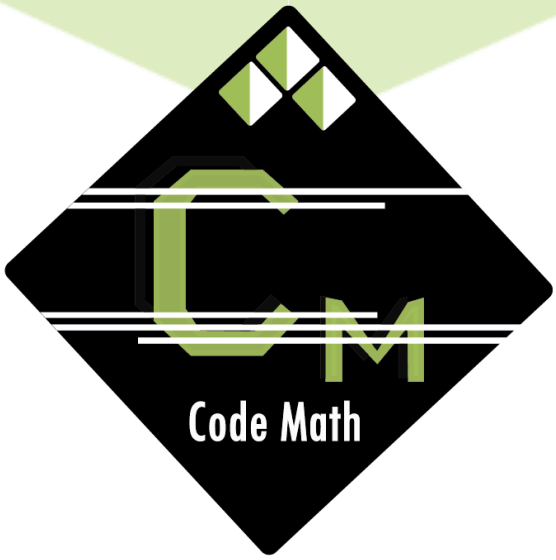


ESTÁNDAR IEEE 754

Conversión de bases IEEE 32 y 64 bits Inversas

Métodos numéricos

Universidad San Buenaventura Cali

CONVERSIÓN DE NUMERO DE MEMORIA DE 32 BITS A BASE DECIMAL

Este formato se almacena en una palabra de 32 Bits organizándose de la siguiente manera:

1 Bit	8 Bits	23 Bits
Signo	Exponente	Mantisa

- El **signo** del número almacenado donde puede ser:
  - 0= positivo
  - 1= negativo.
- El **exponente** se guarda en el espacio de 8 Bits.
- La **mantisa** se guarda en el espacio de 23 Bits.

EJEMPLO:

Escribir el siguiente numero en memoria de 32 bits a decimal.

1 Bit	8 Bits	23 Bits
0	10000110	10100001001110000000000

.

SOLUCIÓN:

Como el primer bit es 0, el número, es positivo.

Convertiremos la mantisa, de binario a decimal, el cual su parte entera será 1, y seguido su mantisa.

1, 10100001001110000000000

Mantisa

Ahora pasaremos el exponente 10000110 a base decimal, y le restaremos 127 como lo indica el estándar.

$$1\,x2^7 + 0\,x2^6 + 0\,x2^5 + 0\,x2^4 + 0\,x2^3 + 1\,x2^2 + 1\,x2^1 + 0\,x2^0$$

$$128 + 0 + 0 + 0 + 0 + 4 + 2 + 0$$

$$128 + 0 + 0 + 0 + 0 + 4 + 2 + 0 = 134$$

$$134 - 127 = 7$$

Ahora correremos la coma 7 veces, y ese número, se pasa de binario a decimal

11010000,10011100000000000

Operaremos la parte entera primero:

$$1\,x2^7 + 1\,x2^6 + 0\,x2^5 + 1\,x2^4 + 0\,x2^3 + 0\,x2^2 + 0\,x2^1 + 0\,x2^0$$

$$128 + 64 + 0 + 16 + 0 + 0 + 0 + 0$$

$$128 + 64 + 0 + 16 + 0 + 0 + 0 + 0 = 208$$

Ahora operaremos la parte fraccionaria:

$$1\,x2^{-1} + 0\,x2^{-2} + 0\,x2^{-3} + 1\,x2^{-4} + 1\,x2^{-5} + 1\,x2^{-6} + 0\,x2^{-7}$$

$$0.5 + 0 + 0 + 0.0625 + 0.03125 + 0.0156 + 0 = 0.609$$

Así que el numero

1 Bit	8 Bits	23 Bits
0	10000110	10100001001110000000000

En decimal, es: 208.609

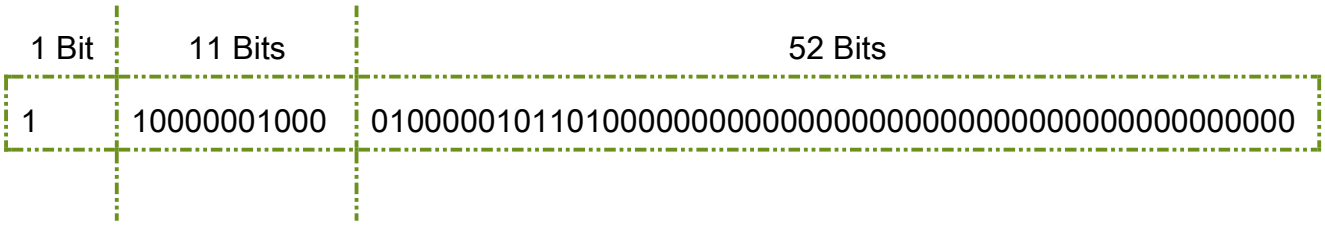
CONVERSIÓN DE NUMERO DE MEMORIA DE 64 BITS A BASE DECIMAL

Este formato se almacena en una palabra de 64 Bits organizándose de la siguiente manera:

- El **signo** del número almacenado donde puede ser:
  - 0= positivo
  - 1= negativo.
- El **exponente** se guarda en el espacio de 11 Bits.
- La **mantisa** se guarda en el espacio de 52 Bits.

EJEMPLO:

Escribir el número en memoria de 64 bits:



A decimal.

Convertiremos la mantisa, de binario a decimal, el cual su parte entera será 1, y seguido su mantisa.

1, 01000001011010000000000000000000000000000000000000

Mantisa

Ahora pasaremos el exponente 10000001000 a base decimal, y le restaremos 127 como lo indica el estándar.

$$1\,x2^{10} + 0\,x2^{9} + 0\,x2^{8} + 0\,x2^{7} + 0\,x2^{6} + 0\,x2^{5} + 0\,x2^{4} + 1\,x2^{3} + 0x2^{2} + 0\,x2^{1} + 0\,x2^{0} -$$

$$1024 + 0 + 0 \quad 0 + 0 + 0 + 0 + 8 + 0 + 0 + 0$$

$$1024 + 8 = 1032$$

Ahora le restamos 1023 como lo indica el estándar:

$$1032 - 1023 = 9$$

Ahora correremos la coma 9 veces, y ese número, se pasa de binario a decimal

1010000010,11010000000000000000000000000000000000000

Operaremos la parte entera primero:

$$1\,x2^{9} + 0\,x2^{8} + 1\,x2^{7} + 0\,x2^{6} + 0\,x2^{5} + 0\,x2^{4} + 0\,x2^{3} + 0x2^{2} + 1\,x2^{1} + 0\,x2^{0}$$

$$512 + 0 + 128 + 0 + 0 + 0 + 0 + 0 + 2 + 0$$

$$512 + 128 + 2 = 642$$

$0\,x2^{-1} + 1\,x2^{-2} + 0\,x2^{-3} + 0\,x2^{-4} + 0\,x2^{-5} + 0\,x2^{-6} + 0\,x2^{-7} + 1\,x2^{-8} + 0\,x2^{-9} + 1x2^{-10} + 1\,x2^{-11} + 0\,x2^{-12} + 1\,x2^{-13}$

$0 + 0.25 + 0 + 0 + 0 + 0 + 0 + 0.0039 + 0 + 0.000976 + 0.000488 + 0 + 0.000122$

$0.25 + 0.0039 + 0.000976 + 0.000488 + 0.000122 = 0.255486$

Así que el numero número en memoria de 64 bits:

1 Bit	11 Bits	52 Bits
1	10000001000	0100000101101000000000000000000000000000000000000000

Es igual al número en base decimal: - 642.255.