1. Pasar de decimal a SM

| Decimal | 4 bit | 8 bit |
| --- | --- | --- |
| 0 | 0000 / 1000 | 00000000 / 10000000 |
| 4 | 0100 | 00000100 |
| -4 | 1100 | 10000100 |
| 6 | 0110 | 00000110 |
| -6 | 1110 | 10000110 |
| 9 | No posible | 00001001 |
| -9 | No posible | 10001001 |

1. Pasar el número FFFFFF016 en CA2 a decimal

F →1111

1111 1111 1111 1111 1111 1111 1111 0000

1000 0000 0000 0000 0000 0000 0000 1111

1000 0000 0000 0000 0000 0000 0001 0000

* 16

1. -710 en CA2 16 bits

7: 0000 0000 0000 0111

1111 1111 1111 1001

Comprobación con suma:

1111 1111 1111 1001

0000 0000 0000 0111

—----------------------

0000 0000 0000 0000

1. 8 bits sin signo, 6 para la parte entera y 2 para la fraccionaria.

6: 000110.00

6.5: 000110.10

6.25: 000110.01

6.75: 000110.11

6.8: 000110.11

1. -0.75 en IEEE 754

precision simple:

0.75 = 1.1\*2-1

Signo: 1

Exponente real: -1

Exponente representado: 126 → 0111 1110

Mantisa: 100 0000 0000 0000 0000 0000

Número en precisión simple: 1011 1111 0100 0000 0000 0000 0000 0000

0XBF40000

Número en precisión doble:

Exponente representado: 1022: 011 1111 1110

1011 1111 1110 1000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000

0XBFE8000000000

6. 0XC0A00000

1 100 0000 1 010 0000 0000 0000 0000 0000

Exponente = 129

Expreal = 129-127 = 2

Signo: -

Mantisa: .01

Número: –1.01 \* 22 = -101 = -5

7.

suponiendo que M=1

-2.5675 x 1015 = 2e

log10(2.5675\*1015) = log10(2e)

log10(2.5675) + 15 = e\*log10(2)

e = (log10(2.5675)+15)/log10(2)

e = 51.189, tomamos e=51 y asumimos que M>1

entonces M>1

2.5675 x 1015 = M\*251

M = 2.5675\*1015 / 251 = 1.14019904629003576702

Se pasa a binario

0.14…..\*2 = 0.2803….

0.2803… \* 2 = 0.56….

0.56… \* 2 = 1.12….

0.12 \* 2 = 0.4…..

se van tomando los bits parte entera (0010…)

número en IEEE 754:

expreal = 51, exp representado = 51 + 127 = 178

10110010

signo: -

1101 1001 0001 0001 1111 0010 0000 1010

hex: 0X D 9 1 1 F 2 0 A

10. el símbolo pi tiene U+3C0 (3 bytes)

11001111 10000000

0xCF80