

**1- Dado el siguiente número decimal 146,750**

- Convertir manualmente a **binario** y explicar el método utilizado.
- Convertir el resultado a **octal**
- Convertir el resultado a **hexadecimal**
- Codificar en **Packed BCD** (4 bits) usando el código "1110" para representar la coma

a)

146	2									0.75	x 2	1.50				
0	73	2								0.50	x 2	1.00				
	1	36	2							0.00	x 2	0.00				
		0	18	2						0.00	x 2	0.00				
			0	9	2					0.00	x 2	0.00				
				1	4	2				0.00	x 2	0.00				
					0	2	2									
						0	1									
1	0	0	1	0	0	1	0			1	1	0	0			

b)

0	1	0	0	1	0	0	1	0		1	1	0
2		2		2						6		

c)

1	0	0	1	0	0	1	0		1	1	0	0
9		2							C			

d)

1	4	6	,	7	5														
0	0	0	1	0	1	0	0	0	1	1	0	1	1	1	0	1	1	0	1

**Tip de resolución:** Los puntos a), b) y c) son ejercicios solamente de conversión de base y responden al Teorema Fundamental de la Numeración.

El punto d) requiere para su representación el reemplazo de cada dígito por cuatro bits que codifique en binario el símbolo decimal correspondiente y la coma se codifica como 1110. Esta representación corresponde a codificación Packed BCD (**B**inary **C**oded **D**ecimal) para una representación de coma fija dentro de un computador. **NO PUEDE USARSE EL TEOREMA FUNDAMENTAL DE LA NUMERACION EN ESTE CASO.** El valor almacenado en el computador es en decimal 146,75.

0x **3F 80 00 00** lsb

- a)

3	F	8	0	0	0	0	0
---	---	---	---	---	---	---	---

[illegible][illegible]

→  $S = (-1)^0 = 1$  (positivo)

[illegible]

$$\text{Exponente} = 0 \cdot 2^7 + 1 \cdot 2^6 + 1 \cdot 2^5 + 1 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 \cdot 127$$

Exponente = 0<sub>10</sub>

Calculamos M la mantisa (llamada también parte significativa)

$$M = 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 0 \cdot 2^{-3} \dots + 0 \cdot 2^{-23} = 0$$

[illegible]

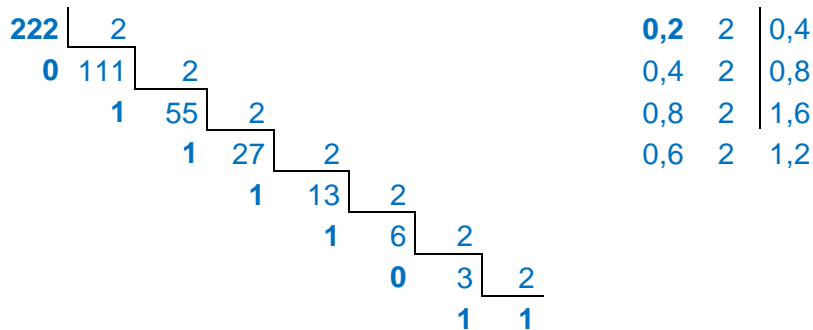
$$N^{\circ} = (-1^S) \cdot [2^{\text{Exp}} \cdot (1 + M)]$$

$$N^{\circ} = (-1^0) \cdot [2^0 \cdot (1+0)]$$

$$N_{10}^{\circ} = 1$$

### 3- Dado el siguiente número decimal 222,2

a) Convertir manualmente a **binario** y explicar el método utilizado.



1	1	0	1	1	1	1	0	0	0	1	1		
128	64	0	16	8	4	2	0	222	0	0	0,125	0,0625	222,1875

b) Convertir el resultado binario a **octal**

Se reemplaza el valor calculado en el punto a) (Valor binario agrupado de a tres bits BCO) por su equivalente octal se completa con un cero a la izquierda. Se agregan períodos de la fracción binaria para determinar el período de la fracción octal

011 011 110,001 100 110 011 001<sub>(2)</sub>... (,0011 periódico)

336,14631<sub>(8)</sub>... (,1463 periódico)

c) Convertir el resultado binario a **hexadecimal**

Se reemplaza el valor calculado en el punto a) (Valor binario agrupado de a cuatro bits BCH) por su equivalente hexadecimal. Como en el binario se convirtió hasta cuatro dígitos en la fracción no hay que completar con ceros

1101 1110,0011<sub>(2)</sub> Agrupamos de a 4 desde la coma y reemplazamos por el dígito Hexa correspondiente

DE,3<sub>(16)</sub> (,3 periódico)

d) Generar en un registro de 32 bits la codificación binaria para representar el número decimal dado en una variable float (Indicar la posición del bit menos significativo lsb)

**Signo** = 0 Positivo

Normalizamos 11011110,0011  $\Rightarrow$  1,10111100011 (Corremos siete lugares la coma dejando un uno significativo en la parte entera)

**Exponente** = 7 (Lugares que se corrió la coma) expresado en exceso a 127)

127 + 7 = 134<sub>(10)</sub> Lo convertimos a binario

134	2							
0	67	2						
	1	33	2					
		1	16	2				
			0	8	2			
				0	4	2		
					0	2	2	
						0	1	
1	0	0	0	0	1	1	0	
128	0	0	0	0	4	2	0	134

**Mantisa** = 110 1000 0010 0000 0000 0000 Parte fraccionaria luego de correr la coma, se completa con el período a la derecha

**Numero float** =

0 1000 0110 101 1110 0011 **0011 0011 0011** lsb  
 s Exponente Mantisa

**4- Expresar en hexadecimal el contenido de una variable short int cuyos valores decimales son los siguientes**

Todos los resultados binarios **short int** se completan con 16 bits ó 4 dígitos Hexa.

a) short int = -222

Para calcular el valor partimos del número binario positivo  
 Binario 0000 0000 1101 1110 lsb  
 Ca1 1111 1111 0010 0001  
 Ca2 1111 1111 0010 0010  
 Hexa FF22

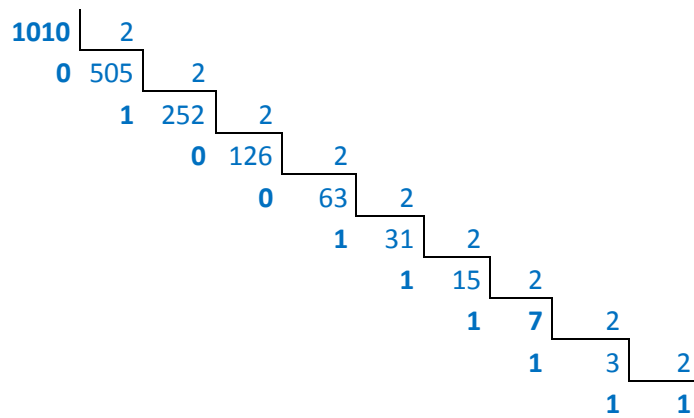
b) short int b = -15

Para calcular el valor partimos del número binario positivo  
 Binario 0000 0000 0000 1111 lsb  
 Ca1 1111 1111 1111 0000  
 Ca2 1111 1111 1111 0001  
 Hexa FFF1

c) short int c = 4096

El número es potencia de dos,  $2^{12}$   
 Binario 0001 0000 0000 0000 lsb  
 Hexa 1000

d) short int d = 1010<sub>(10)</sub>



1	1	1	1	1	1	0	0	1	0
512	256	128	64	32	16	0	0	2	0

Binario 0000 0011 1111 0010 lsb  
 Hexa 03F2