

PRÁCTICA 1

REPRESENTACIÓN DE LA INFORMACIÓN

SOBRE LA PRÁCTICA

En esta práctica se plantean una serie de ejercicios que tienen como objetivo que el estudiante realice operaciones de conversión y representación numérica en los distintos formatos que se abordan en el primer tema de la asignatura.

Para cada apartado, el estudiante dispone de las soluciones **(en rojo en el enunciado)**, de manera que puede comprobar por sí mismo la evolución sobre su aprendizaje. Por tanto, estos ejercicios no son objeto de evaluación sino que sirven de guía orientativa al aprendizaje. No obstante, el profesor observará en clase su realización y orientará al estudiante sobre las posibles dudas que surjan.

A la semana siguiente de la finalización de esta práctica, es decir, la semana en la que empieza la práctica 2, tendrá lugar un examen tipo test que se realizará en clase de prácticas a través de la plataforma Moodle de UA Cloud. En este examen, aparecerán preguntas similares a las realizadas durante la práctica. La nota del examen constituirá la nota de esta práctica.

REFERENCIAS BIBLIOGRÁFICAS

- T. L. Floyd, *Fundamentos de los Sistemas Digitales*, 11ª Edición. Capítulo 2: *Sistemas de numeración, operaciones y códigos*, sección 2-10; y Capítulo 6: *Funciones de la lógica combinatorial*, secciones 6-1 a 6-3.
- Transparencias del Tema 1 de Fundamentos de los Computadores: "Representación de la información".

APARTADO A. CAMBIO DE BASE Y REPRESENTACIÓN DE NÚMEROS ENTEROS

Objetivos

Una vez finalizado este apartado debemos ser capaces de realizar:

- Tareas de representación numérica en los distintos sistemas de codificación.
- Tareas de conversión numérica entre bases.
- Operaciones aritméticas en distintos sistemas de representación.

Elementos necesarios

Los estudiantes deberán realizar las operaciones sobre papel y, posteriormente, comprobar los resultados obtenidos mediante las soluciones aportadas.

Realización práctica

En los siguientes ejercicios, has de anotar **todas** las operaciones efectuadas.

1. Realiza la siguiente tabla de conversiones:

Número	Binario	Hexadecimal	Octal	Decimal
1018 ₁₀	1111111010	3FA	1772	-
121727 ₈	1010001111010111	A3D7	-	41943
10110101 ₂	-	B5	265	181
A750B _H	10100111010100001011	-	2472413	685323
1995 ₁₀	11111001011	7CB	3713	-
147 ₈	1100111	67	-	103
101101 ₂	-	2D	55	45
11AE _H	1000110101110	-	10656	4526

2. Realiza las siguientes conversiones, considerando 8 cifras binarias fraccionarias. Comenta las diferencias que observas al convertir los números a), b) y c) a binario.

a) 18,09162₁₀ a binario, octal y hexadecimal: 10010,00010111₂ 22,056₈ 12,17_H

b) 14,14₁₀ a binario, octal y hexadecimal 1110,00100011₂ 16,106₈ E,23_H

c) 0,21₁₀ a binario, octal y hexadecimal 0,00110101₂ 0,152₈ 0,35_H

d) 13E,5A_H a decimal 318,351562

e) 10000001,100011011010101₂ a hexadecimal, octal y decimal
81.8DAA_H 201.43325₈ 129.553375244₁₀

f) 0,35_H a decimal, octal y binario 0.20703125₁₀ 0,152₈ 0.00110101₂

g) 2,07₈ a decimal, hexadecimal y binario 2,1C_H 2.109375₁₀ 10,000111₂

3. Realiza las siguientes conversiones en 16 bits.

a) $A = -1850_{10}$ a Complemento a 1 **1111100011000101** c_1

b) $B = 64B2_H$ a Complemento a 2 **0110010010110010** c_2

c) $C = -15472_{10}$ a representación sesgada **0100001110010000** s

4. Realiza las siguientes operaciones en complemento a 2 de 16 bits con los datos anteriores. Expresa el resultado final de cada operación en binario natural.

a) $C + A = -100001110101010_2$

b) $A - B = -110101111101100_2$

c) $B + C = +10100001000010_2$

d) $-B - A = -101110101111000_2$

Verifica el resultado de cada una de las operaciones anteriores mediante la realización de dicha operación en decimal.

APARTADO B. REPRESENTACIÓN DE NÚMEROS REALES

Objetivos

Una vez finalizado este apartado debemos ser capaces de:

- Realizar tareas de conversión entre números reales expresados en decimal y representación en coma flotante.
- Comprender las limitaciones a la representación de valores derivadas del uso de un espacio material de representación en un computador.

Elementos Necesarios

Se deben realizar las operaciones sobre papel y, posteriormente, comprobar los resultados obtenidos haciendo uso de asistentes informáticos:

- Calculadora software
- Hojas de cálculo
- Conversor online: <https://babbage.cs.qc.cuny.edu/IEEE-754/>

Introducción Teórica sobre el estándar IEEE 754 de simple precisión

El estándar para aritmética en coma flotante de IEEE indica cómo deben representarse los números reales en un computador y cómo operar con ellos. Esta norma, denominada IEEE-754, data de 1985 y desde entonces ha sido adoptado por la industria informática como norma para codificar dichos números. En agosto de 2008 se aprobó una revisión del estándar que extiende el conjunto de valores representables para cumplir con mayores requerimientos en distintos ámbitos de aplicación.

Características de la norma IEEE 754-1985

Todo número N vendrá dado por un signo s , un exponente E y una mantisa M , de modo que el valor de N se obtendrá mediante la expresión:

$$N = (-1)^s M \cdot 2^E$$

Para representar N , se utilizan tres campos: campo de signo (s), de exponente (e) y de mantisa (m), que quedan agrupados en el formato en el orden **s e m**.

El número de bits utilizados para representar N según el formato anterior es n , que se divide en: 1 bit para el signo, n_e bits para el campo exponente y n_m bits para la mantisa. De modo que $n = 1 + n_e + n_m$.

El estándar IEEE 754-1985 admite distintas longitudes de palabra (precisiones). Por ejemplo, el formato de simple precisión (SP) consta de 32 bits, donde $n_e = 8$ y $n_m = 23$.

El **bit de signo** indica:

- $s = 0 \Rightarrow N \geq 0$
- $s = 1 \Rightarrow N < 0$

El **campo de exponente** representa el exponente E de forma sesgada, siendo el sesgo igual a $S = 2^{n_e-1} - 1$.

Así, con $n_e = 8 \Rightarrow S = 2^{n_e-1} - 1 = 127 = 0111\ 1111$

Ejemplos de exponentes sesgados:

E	E + S	e
0	0 + 127	01111111
2	10 + 01111111 = 129	10000001
127	01111111 + 01111111 = 254	11111110
-1	01111111 - 1 = 126	01111110
-126	01111111 - 01111110 = 1	00000001

Existen codificaciones especiales: cuando $e = 0$ (todo el campo relleno con ceros) o cuando $e = 2^{n_e} - 1$ (todo el campo relleno con unos), puede darse alguno de estos casos:

- $e = 2^{n_e} - 1$ y $m = 0 \Rightarrow N = \infty$
- $e = 2^{n_e} - 1$ y $m = 0 \Rightarrow N = -\infty$
- $e = 0$ y $m = 0 \Rightarrow N = 0$
- $e = 0$ y $m \neq 0 \Rightarrow S = 2^{n_e-1} - 2 \Rightarrow E = -2^{n_e-1} + 2$

*En este último caso, se dice que el número representado está **desnormalizado**.*

El **campo de la mantisa** representa la mantisa del número **normalizado**. Es decir, la mantisa del número debe estar expresada de la forma:

$$M = 1.m, \quad 1 \leq m < 2$$

De forma que se almacenará únicamente la parte fraccionaria de la mantisa normalizada, y se omitirá la parte entera, que siempre es 1 (bit oculto o *implícito*).

Veamos ejemplos de normalización:

$$N_1 = 101.0010011 \cdot 2^{-5} = 0.001010010011 = 1.010010011 \cdot 2^{-3}$$

$$N_2 = 0.0000011101 \cdot 2^{10} = 11101 = 1.1101 \cdot 2^4$$

Veamos algunos ejemplos de representación en este formato para 32 bits:

$$N_3 = 1 \text{ 10000011 010000000000000000000000}$$

$$s = 1 \Rightarrow N < 0$$

$$e = 10000011 \Rightarrow E = 10000011 - 01111111 = 100 = 4_{10}$$

$$M = 1.01 = 1.25_{10}$$

$$N_3 = -1.25 \cdot 2^4 = -1.25 \cdot 16 = -20_{10}$$

$$N_4 = 0 \text{ 01111101 000000000000000000000000}$$

$$s = 0 \Rightarrow N \geq 0$$

$$e = 01111101 \Rightarrow E = 01111101 - 01111111 = -10 = -2_{10}$$

$$M = 1.0$$

$$N_4 = 1.0 \cdot 2^{-2} = 2^{-2} = 0.25_{10}$$

Realización práctica

Nota: no se dan los resultados de los apartados 1 y 2 ya que se pueden obtener directamente de la calculadora online indicada en la página 4.

1. Realiza las siguientes conversiones al formato de representación IEEE754-1985 SP:

- a) $8549,45_{10}$
- b) $\sqrt{3}$
- c) -10^{-23}
- d) $E,C50312F_{16} \cdot 16^{-3}$

2. Indica cuál es el equivalente decimal de los siguientes números expresados en el formato IEEE754-1985 SP:

- a) 0 01111100 110000000000000000000000
- b) 1 10001001 011000000000000000000000
- c) 4B7FFFFF
- d) C2C40000