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 Capitulo 6: *Funciones de la lógica combinacional*, 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.

Introducción teórica

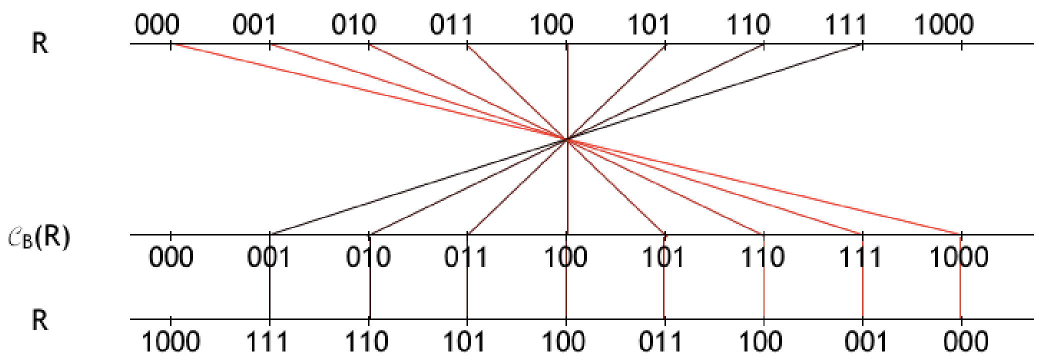
Un sistema de numeración en *base* ***b*** utiliza un alfabeto compuesto por ***b*** símbolos o cifras, para representar valores numéricos. Algunos sistemas de numeración muy empleados son: decimal (b=10), binario (b=2), octal (b=8) y hexadecimal (b=16).

El valor de cada cifra de un número, representado en un sistema de numeración concreto, depende de la cifra en sí y de la posición que ocupa en la secuencia que lo forma (ponderación o peso). El valor N del número representado será:

$$N \equiv \sum_{i} (x_i \cdot b^i) \qquad \text{donde } i \text{ representa la posición que ocupa la cifra } x.$$

Números enteros

La representación gráfica de los valores binarios sobre la recta real ayuda a comprender la causa de las ventajas que tienen determinadas representaciones; p.e., que la notación sesgada es como una traslación del eje X, o que la notación en complemento a 2 es como darle la vuelta al eje X (ver figura)



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. Realizar la siguiente tabla de conversiones:

Número	Binario	Hexadecimal	Octal	Decimal
1022_{10}	1111111110	3FE	1776	-
123727_8	1010011111010111	A7D7	-	42967
10011101_2	-	9D	235	157
$B74FA_H$	10110111010011111010	-	2672372	750842
2011_{10}	11111011011	7DB	3733	-
157_8	1101111	6F	-	111
111110_2	-	3E	76	62
$10AF_H$	1000010101111	-	10257	4271

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

a) $17{,}09375_{10}$ a binario, octal y hexadecimal: $10001{,}00011_2$ $21{,}06_8$ $11{,}18_H$

b) $12{,}15_{10}$ a binario, octal y hexadecimal $1100{,}0010011_2$ $14{,}114_8$ $C{,}26_H$

c) $0{,}17_{10}$ a binario, octal y hexadecimal $0{,}00101011_2$ $0{,}126_8$ $0{,}2B_H$

d) $12F{,}4B_H$ a decimal 303,29296875

e) $1000110001{,}10000011011010101_2$ a hexadecimal, octal y decimal
$231{,}836A8_H$ $1061{,}406652_8$ $561{,}51334381_{10}$

f) $0{,}25_H$ a decimal, octal y binario $0{,}14453125_{10}$ $0{,}112_8$ $0{,}00100101_2$

g) $3{,}05_8$ a decimal, hexadecimal y binario $3{,}14_H$ $3{,}078125_{10}$ $11{,}000101_2$

3. Realizar las siguientes conversiones en 16 bits.

a) $A = -2700_{10}$ a Complemento a 1 1111010101110011_{C1}

b) $B = 75A1_H$ a Complemento a 2 0111010110100001_{C2}

c) $C = -16383_{10}$ a representación sesgada 0100000000000001_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 = $-1001010100010 11_2$

b) A – B = Desbordamiento con 16 bits

c) B + C = $+11010110100010_2$

d) – B – A = -110101100010101_2

Verificar 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:

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

El **campo de exponente** representa el exponente E de forma sesgada, siendo el sesgo igual a $S = 2^{ne-1} – 1$.
Así, con $n_e = 8 \Rightarrow S = 2^{ne-1} – 1 = 127 = 0111\ 1111$

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* $e = 2^{ne} - 1$ *(todo el campo relleno con unos), puede darse alguno de estos casos:*

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

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:

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

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 10000011 01000000000000000000000
$s = 1 \Rightarrow N < 0$
$e = 10000011 \Rightarrow E = 10000011 - 01111111 = 100 = 4_{10}$
$M = 1.01 = 1.25_{10}$
$N_3 = -1.25 \cdot 2^4 = -1.25 \cdot 16 = -20_{10}$

$N_4 =$ 0 01111101 00000000000000000000000
$s = 0 \Rightarrow N \geq 0$
$e = 01111101 \Rightarrow E = 01111101 - 01111111 = -10 = -2_{10}$
$M = 1.0$
$N_4 = 1.0 \cdot 2^{-2} = 2^{-2} = 0.25_{10}$

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 5.

1. Realiza las siguientes conversiones al formato de representación IEEE754-1985 SP:
a) $8539{,}45_{10}$
b) $\sqrt{2}$
c) -10^{-23}
d) $E{,}C50312F_{16} \cdot 16^{-3}$

2. Indicar cuál es el equivalente decimal de los siguientes números expresados en el formato IEEE754-1985 SP:
a) 0 01111100 11000000000000000000000
b) 1 10001001 01100000000000000000000
c) 4B7FFFFF
d) C2C40000

3. Sea el siguiente formato de representación en coma flotante:

Exponente *e* : C2 con 7 bits	Mantisa *m* : sesgada con 9 bits

de forma que cada número N queda representado como $N = m \cdot 2^{e}$

a) Hallar el rango de representación del formato.

$[-256 \cdot 2^{63}, -2^{-64}] \cup \{0\} \cup [2^{-64}, 255 \cdot 2^{63}]$

b) Representar en el formato anterior los siguientes números decimales:
b.1) -12.25 1111110 011001111
b.2) 8192 0000110 110000000

c) Dados los siguientes números, representados en el formato anterior empaquetados en hexadecimal, indicar a qué números en base 10 corresponden
c.1) 0400 -1024_{10}
c.2) FFFF $127{,}5_{10}$