

75.562 · Fundamentos de Computadores · 2023-24

PEC1 - Primera prueba de evaluación continua

Apellidos: *Gámez García*
Nombre: *Elías*

Formato y fecha de entrega

- Para dudas y aclaraciones sobre el enunciado debéis dirigiros al consultor responsable de vuestra aula.
- Hay que entregar la solución en un fichero PDF utilizando una de las plantillas entregadas conjuntamente con este enunciado.
- Se debe entregar a través de la aplicación de **Entrega de la Actividad** correspondiente del apartado **Contenidos** de vuestra aula.
- La fecha límite de entrega es el **13 de marzo** (a las 24 horas).
- **Razonad la respuesta en todos los ejercicios. Las respuestas sin justificación no recibirán puntuación.**

Respuestas

Ejercicio 1:

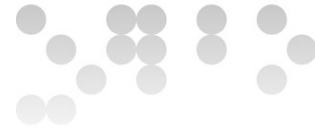
Secuencia de bits: 110101110

- a) Se trata de un número binario natural.

En este caso, sumamos cada dígito de la secuencia por su correspondiente potencia en base 2, empezando desde la derecha por 0. Aplicamos el TFN y nos da como resultado:

$$1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

Resultado: 430



- b) Se trata de un número binario entero codificado en signo y magnitud.

En este caso, el bit signo, que es la cifra del bit más significativo, al corresponderse con un 1 nos indica que el número con el que estamos trabajando es negativo. Procedemos pues a calcular la magnitud del resultado mediante el TFN:

$$1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 174$$

Resultado: -174.

- c) Se trata de un número binario entero codificado en Ca2:

Si analizamos el bit mas significativo, que en este caso es un 1, nos indica que el resultado será negativo. Aplicamos el TFN teniendo en cuenta el signo negativo:

$$-1 \cdot 2^8 + 1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0$$

Resultado: -82

- d) Se trata de un número binario fraccionario en coma fija, con bit de signo y cuatro dígitos fraccionarios

Al tratarse de una parte decimal que consta de 4 dígitos, notamos lo siguiente:

11010,1110

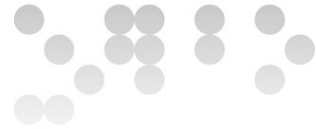
Y si bien se indica que contiene un bit de signo, tenemos en cuenta que será negativo, al ser 1 la cifra más significativa. Por tanto, aplicamos el TFN al aparte entera y sumamos la parte fraccionaria con potencias de 2 negativas:

$$\text{Parte entera: } 1010 = 1 \cdot 2^3 + 0 \cdot 2^2 + 1 \cdot 2^1 + 0 \cdot 2^0 = 10$$

$$\text{Parte fraccionaria: } 1110 = 1 \cdot 2^{-1} + 1 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} = 0,875$$

Por tanto, la representación decimal es de $-10 + 0.875$

Resultado: -10.875



Ejercicio 2:

- a) Número -253 (10 en complemento a 2 y 12 bits).

Como estamos ante un número negativo, para obtener el resultado en ca2 tomamos la secuencia binaria del valor absoluto, en este caso, 253.

253 en binario corresponde a la secuencia de bits: 11111101, mediante el uso del TFN.

Para hacerlo mediante un sistema de 12 bits, añadimos 0s a la izquierda. Por tanto, la secuencia queda tal que:

Resultado: 000011111101.

Seguidamente, realizamos el complemento A1. Esto es, convertir todos los 0s en 1s y viceversa, y finalmente sumamos 1 al bit menos significativo de la secuencia para obtener el complemento a2.

Resultado: 111100000011

- b) Número 253 (10 en complemento a 2 y 12 bits)

Como hemos visto en el apartado a, el número decimal 253 corresponde a la secuencia 000011111101 binaria en 12 bits.

Al ser un número positivo, solo hemos de realizar la conversión usando el TFN, que ya hemos hecho en el apartado anterior.

- c) Número -222 (10 en signo y magnitud y 12 bits).

La representación signo y magnitud consiste en que el bit más significativo de la secuencia corresponde a un 0 si es positivo y 1 si es negativo. El resto de la secuencia corresponde a la conversión binaria de la magnitud.

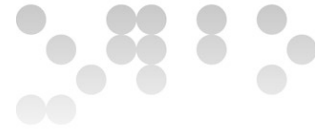
Por tanto, al estar con un número decimal negativo, nuestro bit más significativo corresponde a un 1.

Procedemos a calcular la secuencia 222 en binario usando 12 bits:

222 = 000011011110

Finalmente, añadimos 1 al bit más significativo, por lo que queda en:

Resultado: 100011011110



d) Número $444_{(10)}$ en hexadecimal

Procedemos a realizar la división del número en decimal entre la base, 16, y anotamos los dígitos del residuo para su interpretación en hexadecimal:

División	Cociente	Residuo	Bit
444/16	27	12	0
27/16	1	11	1
1/16	0	1	2

En la representación hexadecimal (base 16), a partir del dígito decimal 10, se utiliza el carácter A para su correspondiente hexadecimal. Entonces, teniendo en cuenta que $12 = C$ en hexadecimal y $11 = B$:

Resultado: $1BC_{16}$

Ejercicio 3:

A = 10101011

B = 11010101

Números codificados en signo y magnitud de 8 bits.

a) A + B

Para realizar la operación, primero sumamos los módulos de las magnitudes; que da como resultado:

Bits acarreo		1	1	1	1	1	1	
A		0	1	0	1	0	1	1
B		1	0	1	0	1	0	1
Resultado Operación	1	1	1	1	1	1	1	1
	Desbordamiento							



En este caso, como estamos realizando una operación de suma entre dos números del mismo signo (negativo) sí se produce desbordamiento, al haber un acarreo en la última operación. El resultado no se puede representar al no caber en el formato definido.

b) B - A

En este caso, como estamos realizando una operación de suma entre dos números de signo igual, no se produce desbordamiento

Para realizar la operación, primero restamos los módulos de las magnitudes; que da como resultado:

Bits acarreo	1		1		1		
B	1	0	1	0	1	0	1
A	0	1	0	1	0	1	1
Resultado Operación	0	1	0	1	0	1	0

Para averiguar el signo del resultado, nos fijamos en cual de los dos términos tiene la mayor magnitud; en este caso es B, por lo que el resultado tendrá un signo negativo (un 1 en el bit significativo). Por tanto, que el resultado es de:

Resultado final: 10101010₁₀



Ejercicio 4:

C = 10111001

D = 11110110

Números codificados en complemento a 2 y 8 bits.

a) C + D

Al trabajar en complemento a 2, analizamos primeramente el signo de los operandos, que en este caso es negativo, al ser un 1 el bit mas significativo de C y de D. Analizamos el desbordamiento al comprobar si el resultado de la magnitud da positivo, cosa que en la operación siguiente vemos que no ocurre, al ser 1 el bit mas significativo del resultado.

Bits acarreo	1	1	1	1					
C		1	0	1	1	1	0	0	1
D		1	1	1	1	0	1	1	0
Resultado Operación	X	1	0	1	0	1	1	1	1

La x señala que ignoramos el acarreo final; pero al analizar la magnitud del resultado obtenido, comprobamos aplicando el TFN que en efecto el resultado es negativo:

$$C = 10111001 = -71_{10}$$



$$D = 11110110 = -10_{10}$$

Por tanto, el resultado de $C + D$ debería de ser de -81_{10}

Comprobamos el resultado de la operación y verificamos:

$$\text{Resultado} = 10101111_{\text{Ca2}} = -1 \cdot 2^7 + 0 \cdot 2^6 + 1 \cdot 2^5 + 0 \cdot 2^4 + 1 \cdot 2^3 + 1 \cdot 2^2 + 1 \cdot 2^1 + 1 \cdot 2^0 = -81$$

Resultado final: $C + D = 10101111_{\text{Ca2}}$

Ejercicio 5:

Formato de coma flotante de 14 bits:

S		Exponente			Mantisa	
13	12		8	7		0

- El bit de signo S vale 0 para cantidades positivas y 1 para las negativas.
- El exponente se representa en exceso a 16.
- Hay bit implícito.
- La mantisa está normalizada en la forma 1,X.

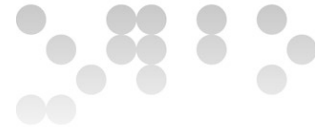
a) Representar el número $50,48_{10}$ en este formato:

Primero de todo procedemos a codificar el número dado en su forma binaria. Esto se hace pasando a binario la parte entera y posteriormente la fraccionaria.

Esto es para la parte entera:

$$50_{10} = 110010_2$$

50	$25 \cdot 2$	0
25	$12 \cdot 2$	1
12	$6 \cdot 2$	0
6	$3 \cdot 2$	0
3	$1 \cdot 2$	1
1	$0 \cdot 2$	1



Esto es para la parte decimal, que como observamos en el enunciado hemos de obtener un total de 8 decimales, al corresponder con una mantisa de 8 bits en el formato especificado:

$$0,48_{10} = 0.01111010_2$$

$0,48 * 2$	0.96	$0 + 0.96$
$0.96 * 2$	1.92	$1 + 0.92$
$0.92 * 2$	1.84	$1 + 0.84$
$0.84 * 2$	1.68	$1 + 0.68$
$0.68 * 2$	1.36	$1 + 0.36$
$0.36 * 2$	0.72	$0 + 0.72$
$0.72 * 2$	1.44	$1 + 0.44$
$0.44 * 2$	0.88	$0 + 0.88$
$0.88 * 2$	1.76	$1 + 0.76$

Por tanto, obtenemos la secuencia binaria del número tal que: $110010,01111010_2$

Seguidamente, normalizamos la expresión obtenida, teniendo en cuenta que la longitud del exponente es de 2 bits según el formato especificado:

$$50,48_{10} = 1,1001001111010_2 * 2^2$$

Por último, obtenemos los valores de la representación:

Como estamos trabajando con un valor positivo, el signo vale 0

El exponente es 2, que hay que codificar a un exceso de 16. Al disponer de 5 bits, hemos de representar $2 + 16 = 18$ con 5 bits.

18	$9 * 2$	0
9	$4 * 2$	1
4	$2 * 2$	0
2	$1 * 2$	0
1	$0 * 2$	1

Es decir, que el exponente = 10010_2 en exceso a 16.

Finalmente, como tenemos bit implícito, solo hay que representar la parte derecha de la coma y disponemos de 8 bits para representar por truncamiento:

Signo	Exponente					Mantisa							
0	1	0	0	1	0	1	0	0	1	0	0	1	1