Universidad de Huelva

FUNDAMENTOS DE COMPUTADORES

1º Curso del Grado en Ingeniería Informática

TEMA 1

Problemas resueltos

Problemas resueltos del tema 1

1. Convertir a base diez los siguientes números:

a) 1001100010.101_2

b) 610327.5246_8

c) $C9FAB.67E_{16}$

Solución:

a) $1001100010.101_2 = 1 \times 2^9 + 1 \times 2^6 + 1 \times 2^5 + 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-3} =$ **610.625**

b) $610327.5246_8 = 6 \times 8^5 + 1 \times 8^4 + 3 \times 8^2 + 2 \times 8^1 + 7 \times 8^0 + 5 \times 8^{-1} + 2 \times 8^{-2} + 4 \times 8^{-3} + 6 \times 8^{-4} =$ **200919.66552**

c) $C9FAB.67E_{16} = 12 \times 16^4 + 9 \times 16^3 + 15 \times 16^2 + 10 \times 16^1 + 11 \times 16^0 + 6 \times 16^{-1} + 7 \times 16^{-2} + 14 \times 16^{-3} =$ **827307.4058**

2. Volver a convertir los resultados obtenidos en el ejercicio anterior a las respectivas bases de origen y comprobar que los nuevos resultados se corresponden con los datos de partida.

Solución:

a) Para realizar la conversión a binario de la cantidad **610.625** actuamos de la siguiente forma:

Parte entera: Dividimos sucesivamente por 2 y obtenemos: $\mathbf{1001100010_2}$

Parte fraccionaria: Multiplicamos sucesivamente por 2 y obtenemos: $\mathbf{0.101_2}$

El resultado final lo obtenemos colocando a la izquierda y a la derecha de la coma de decimales la parte entera y la parte fraccionaria, respectivamente, como se indica a continuación:

$$610.625 = \mathbf{1001100010.101_2}$$

b) Para realizar la conversión a octal de la cantidad **200919.66552** actuamos de la siguiente forma:

Parte entera: Dividimos sucesivamente por 8 y obtenemos: $\mathbf{610327_8}$

Parte fraccionaria: Multiplicamos sucesivamente por 8 y obtenemos: $\mathbf{0.5245_8}$

El resultado final lo obtenemos colocando a la izquierda y a la derecha de la coma de decimales la parte entera y la parte fraccionaria, respectivamente, como se indica a continuación:

$$200919.66552 = \mathbf{610327.5245_8}$$

La pequeña discrepancia en el resultado es debida a los decimales despreciados en la conversión de octal a decimal.

c) Para realizar la conversión a hexadecimal de la cantidad **827307.4058** actuamos de la siguiente forma:

Parte entera: Dividimos sucesivamente por 16 y obtenemos: $\mathbf{C9FAB_{16}}$

Parte fraccionaria: Multiplicamos sucesivamente por 16 y obtenemos: $\mathbf{0.67E_{16}}$

El resultado final lo obtenemos colocando a la izquierda y a la derecha de la coma de decimales la parte entera y la parte fraccionaria, respectivamente, como se indica a continuación:

$$827307.4058 = \mathbf{C9FAB.67E_{16}}$$

3. Transformar los siguientes números decimales a BCD Natural, BCD Aiken, BCD 5-4-2-1 y BCD exceso 3.

a) 6259.34

b) 3910. 82

c) 7543.21

Solución:

a) 6259.34

BCD Natural	0110	0010	0101	1001	.	0011	0100
BCD Aiken	1100	0010	1011	1111	.	0011	0100
BCD 5-4-2-1	1001	0010	1000	1100	.	0011	0100
BCD exceso 3	1001	0101	1000	1100	.	0110	0111

b) 3910. 82

BCD Natural	0011	1001	0001	0000	.	1000	0010
BCD Aiken	0011	1111	0001	0000	.	1110	0010
BCD 5-4-2-1	0011	1100	0001	0000	.	1011	0010
BCD exceso 3	0110	1100	0100	0011	.	1011	0101

c) 7543.21

BCD Natural	0111	0101	0100	0011	.	0010	0001
BCD Aiken	1101	1011	0100	0011	.	0010	0001
BCD 5-4-2-1	1010	1000	0100	0011	.	0010	0001
BCD exceso 3	1010	1000	0111	0110	.	0101	0100

4. Transformar a decimal los siguientes números:

a) 01010111.01001000 (BCD Natural)

b) 1110.001111001011 (BCD Aiken)

c) 01001000.10111100 (BCD 5-4-2-1)

d) 010001010011.1001 (BCD Exceso 3)

Solución:

a) 01010111.01001000 (BCD Natural) = **57.48**

b) 1110.001111001011 (BCD Aiken) = **8.365**

c) 01001000.10111100 (BCD 5-4-2-1) = **45.89**

d) 010001010011.1001 (BCD Exceso 3) = **120.6**

5. Generar un código Johnson de 6 bits.

Solución:

Dígito decimal	Código Johnson de 6 bits
0	000000
1	000001
2	000011
3	000111
4	001111
5	011111
6	111111
7	111110
8	111100
9	111000
10	110000
11	100000

6. Generar un código de paridad impar a partir de un código BCD 5-4-2-1.

Solución:

Dígito decimal	BCD 5-4-2-1 con bit de paridad impar
0	0000 1
1	0001 0
2	0010 0
3	0011 1
4	0100 0
5	1000 0
6	1001 1
7	1010 1
8	1011 0
9	1100 1

7. Generar un código de Hamming a partir de un código Gray de 3 bits.

Solución:

$$2^c \geq m + c + 1$$

Para c = 3 tenemos → $2^3 \geq 3 + 3 + 1$, por lo que añadiremos 3 bits al código original.

Sobre las combinaciones del código final se realizarán 3 detecciones de paridad (D_2, D_1 y D_0), cuyos resultados indicarán la situación del bit erróneo, en caso de que lo hubiera, según la siguiente tabla:

D_2 D_1 D_0	Bit erróneo
0 0 0	Ninguno
0 0 1	b_1
0 1 0	b_2
0 1 1	b_3
1 0 0	b_4
1 0 1	b_5
1 1 0	b_6

Las detecciones de paridad tendrán las siguientes expresiones:

$$D_0 = b_1 \oplus b_3 \oplus b_5$$

$$D_1 = b_2 \oplus b_3 \oplus b_6$$

$$D_2 = b_4 \oplus b_5 \oplus b_6$$

Los bits a añadir serán b1, b2 y b4, y sus valores se calcularán mediante las siguientes expresiones:

$$b_1 = b_3 \oplus b_5$$

$$b_2 = b_3 \oplus b_6$$

$$b_4 = b_5 \oplus b_6$$

Así, el código resultante quedará como sigue:

Número decimal	**b_6 b_5 b_4 b_3 b_2 b_1**
0	0 0 **0** 0 **0** **0**
1	0 0 **0** 1 **1** **1**
2	0 1 **1** 1 **1** **0**
3	0 1 **1** 0 **0** **1**
4	1 1 **0** 0 **1** **1**
5	1 1 **0** 1 **0** **0**
6	1 0 **1** 1 **0** **1**
7	1 0 **1** 0 **1** **0**