SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMÁTICA.

TECNOLOGÍA Y ORGANIZACIÓN DE COMPUTADORES

1º Grado en Ingeniería Informática.

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMATICA.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.
2. SISTEMA DE NUMERACIÓN EN BASE DOS.
    1. Definición del sistema binario.
    2. Transformaciones de base binaria a decimal.
    3. Transformaciones de base decimal a binaria.
3. CÓDIGOS INTERMEDIOS.
    1. Base hexadecimal.
    2. Base octal (apéndice).
4. EJERCICIOS.

Bibliografía

[PRI05]: Apéndice 1

[PRI06]: Apéndice A

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMATICA.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.
2. SISTEMA DE NUMERACIÓN EN BASE DOS.
   1. Definición del sistema binario.
   2. Transformaciones de base binaria a decimal.
   3. Transformaciones de base decimal a binaria.
3. CÓDIGOS INTERMEDIOS.
   1. Base hexadecimal.
   2. Base octal (apéndice).
4. EJERCICIOS.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

Los computadores realizan las operaciones aritméticas utilizando una representación para los datos numéricos basada en el sistema de numeración base dos (***binario natural*** o ***binario***).

También se utilizan los sistemas de numeración hexadecimal y octal para obtener códigos intermedios. Un número expresado en uno de estos dos códigos puede transformarse (manual y electrónicamente) directa y fácilmente a binario y viceversa.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

- Un sistema de numeración en base b utiliza para representar los números un alfabeto A compuesto por b símbolos o cifras. Todo número se expresa por un conjunto de cifras, contribuyendo cada una de ellas con un valor que depende:
  - de la cifra en sí
  - de la posición que ocupe dentro del número.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

- Sistema de numeración decimal (base 10):
  b=10 símbolos, A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}
- Sistema de numeración binario (base 2):
  b=2 símbolos, A={0, 1}
- Sistema de numeración hexadecimal (base 16):
  b=16 símbolos, A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}
- Sistema de numeración octal (base 8):
  b=8 símbolos, A={0, 1, 2, 3, 4, 5, 6, 7}

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

- La representación de un número en una base b:

$$N = \ldots\ n_4 n_3 n_2 n_1 n_0\ ,\ n_{-1} n_{-2} n_{-3}\ \ldots;\ n_i \in A$$

es una forma abreviada de expresar su valor, que es:

$$N = \ldots\ n_4 \cdot b^4 + n_3 \cdot b^3 + n_2 \cdot b^2 + n_1 \cdot b^1 + n_0 \cdot b^0 + n_{-1} \cdot b^{-1} \ldots$$

- Para representar un número, resulta más cómodo que los símbolos (cifras) del alfabeto o la base de numeración sean los menos posibles, pero, por otra parte, cuanto menor es la base, mayor es el número de cifras que se necesitan para representar una cantidad dada.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

- Ejemplo: representación de un número en base 10:

$4567{,}28)_{10}$ =	**$4 \cdot 10^3$ +**	**4 000**
	$5 \cdot 10^2$ +	**500**
	$6 \cdot 10^1$ +	**60**
	$7 \cdot 10^0$ +	**7**
	$2 \cdot 10^{-1}$ +	**0,2**
	$8 \cdot 10^{-2}$ =	**0,08**
	4567,28	**4567,28**

- $4567{,}28 = 4 \cdot 10^3 + 5 \cdot 10^2 + 6 \cdot 10^1 + 7 \cdot 10^0 + 2 \cdot 10^{-1} + 8 \cdot 10^{-2}$

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.

- Otros ejemplos:

$3278{,}52)_{10} = 3\cdot10^{3} + 2\cdot10^{2} + 7\cdot10^{1} + 8\cdot10^{0} + 5\cdot10^{-1} + 2\cdot10^{-2}$

$235{,}37)_{8} = 2\cdot8^{2} + 3\cdot8^{1} + 5\cdot8^{0} + 3\cdot8^{-1} + 7\cdot8^{-2}$

$3AB4{,}7)_{16} = 3\cdot16^{3} + A\cdot16^{2} + B\cdot16^{1} + 4\cdot16^{0} + 7\cdot16^{-1}$

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMATICA.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.
2. SISTEMA DE NUMERACIÓN EN BASE DOS.
    1. Definición del sistema binario.
    2. Transformaciones de base binaria a decimal.
    3. Transformaciones de base decimal a binaria.
3. CÓDIGOS INTERMEDIOS.
    1. Base hexadecimal.
    2. Base octal (apéndice).
4. EJERCICIOS.

2.1. SISTEMA DE NUMERACIÓN BASE 2.

- Sistema de numeración binario: b=2 símbolos , A={0, 1}

Representación binaria de los números decimales del 0 al 9

Decimal	Binario
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

2.2. TRANSFORMACIONES DE BASES BINARIA A DECIMAL.

Se aplica la siguiente expresión:

$$N)_{10} = \ldots\ n_4 \cdot 2^4 + n_3 \cdot 2^3 + n_2 \cdot 2^2 + n_1 \cdot 2^1 + n_0 \cdot 2^0 + n_{-1} \cdot 2^{-1} \ldots$$

Ejemplo:

4 3 **2** 1 0.-1-2-**3**

$10100{,}001)_2 = 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 + 0 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} = 16+4+(1/8) = 20{,}125)_{10}$

Se suman los pesos de las potencias de 2 de las posiciones en las que hay un 1

2.3. TRANSFORMACIONES DE BASES DECIMAL A BINARIA.

La **parte entera** del número binario se obtiene dividiendo entre 2 la parte entera del número decimal de partida, y de los cocientes que sucesivamente se vayan obteniendo (sin obtener decimales en el cociente).

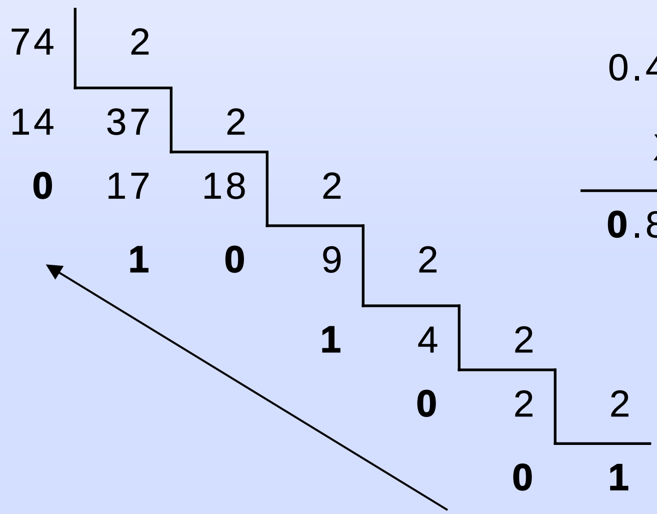
La **parte fraccionaria** del número binario se obtiene multiplicando por 2 sucesivamente la parte fraccionaria del número decimal de partida y las partes fraccionarias que se van obteniendo en los productos sucesivos.

El número binario se forma con **los restos de las divisiones** y el **último cociente** (tomando el último cociente como el bit más significativo y el primer residuo como el bit menos significativo), “,”, las **partes enteras de los productos** obtenidos (siendo el bit más significativo el del primer producto, y el menos significativo el del último producto).

2.3. TRANSFORMACIONES DE BASES DECIMAL A BINARIA.

- **Ejemplo:**

  Transformar de decimal a binario: $74{,}423)_{10}$



0.423	0.846	0.692	0.384	0.768	
X 2	X 2	X 2	X 2	X 2	
0.846	**1**.692	**1**.384	**0**.768	**1**.536	…

Parte fraccionaria: 01101…

Parte entera:1001010

$$\mathbf{74{,}423)_{10} = 1001010{,}\ 01101\ldots)_2}$$

2.3. TRANSFORMACIONES DE BASES DECIMAL A BINARIA.

- Se puede observar que un número decimal con cifras fraccionarias puede dar lugar a un número binario con un número de cifras fraccionarias mucho mayor o incluso infinito. Sin embargo, el número de bits para representar en binario un número decimal es limitado y a veces está prefijado.

- Si para representar en binario tanto la parte entera como la parte fraccionaria de un número decimal se utiliza un número limitado de bits (por ejemplo p bits para la parte entera y q bits para la parte fraccionaria), entonces se puede producir en la representación binaria un error de truncamiento.

2.3. TRANSFORMACIONES DE BASES DECIMAL A BINARIA.

- Ejemplo:

a) Representación binaria del dato decimal

$$74,423)_{10} = 1001010,\ 01101...)_2$$

$$p = 7 \text{ bits} \ ; \ q = 5 \text{ bits}$$

b) Representación decimal del dato binario

$$1001010,01101)_2 = 74,40625)_{10}$$

c) Error cometido

c.1) En la parte fraccionaria: 3,9598 %

c.2) En todo el dato: 0,0225 %

2.3. TRANSFORMACIONES DE BASES DECIMAL A BINARIA.

- La representación anteriormente indicada sólo permite representar números positivos (enteros o reales). En el Tema 1.3 se indicará cómo se realiza la representación de números negativos (enteros o reales)

- La representación anteriormente indicada con un número fijo (p) de bits para representar la parte entera (E) y un número fijo (q) de bits para representar la parte fraccionaria (F) del número decimal recibe el nombre de representación en coma fija o en punto fijo. En el Tema 1.3 se indicará otro tipo de representación (denominada de coma o punto flotante) para representar números reales.

2.3. DATOS DE TIPO ENTERO REPRESENTADOS EN BCD

Representación de dígitos decimales codificados en binario (Binary Coded Decimal - BCD):

Valor decimal	Valor BCD
0	0000
1	0001
2	0010
3	0011
4	0100
5	0101
6	0110
7	0111
8	1000
9	1001

- Se codifica aisladamente cada dígito decimal con cuatro dígitos binarios de acuerdo con la tabla anexa.
- Por ejemplo:

  $0111\ 0010\ 1001)_{BCD} = 729)_{10}$

  $3795)_{10} = 0011\ 0111\ 1001\ 0101)_{BCD}$

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMATICA.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.
2. SISTEMA DE NUMERACIÓN EN BASE DOS.
    1. Definición del sistema binario.
    2. Transformaciones de base binaria a decimal.
    3. Transformaciones de base decimal a binaria.
3. CÓDIGOS INTERMEDIOS.
    1. Base hexadecimal.
    2. Base octal (apéndice).
4. EJERCICIOS.

3. CÓDIGOS INTERMEDIOS: OCTAL Y HEXADECIMAL.

- Los códigos intermedios se fundamentan en la facilidad de transformar un número en base 2 a otra base que sea una potencia de 2 ($2^2=4$; $2^3= 8$; $2^4=16$, etc.), y viceversa.

- Usualmente se utilizan como códigos intermedios los sistemas de numeración en base 16 (o hexadecimal) y en base 8 (u octal).

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

Al ser b = 16 = 2^4, se pueden hacer las conversiones de binario a hexadecimal y viceversa de forma muy sencilla utilizándose grupos de 4 bits.

De la misma forma que manualmente es muy fácil convertir números de binario a hexadecimal y viceversa, también resulta sencillo efectuar esta operación electrónicamente o por programa, por lo que a veces la computadora utiliza este tipo de notaciones intermedias internamente o como entrada/salida.

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

b = 16 símbolos;

A = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Cifras hexadecimales y sus valores decimal y binario

Hexadecimal	Decimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

Ejemplos:

N = AC70,3B =
1010 1100 0111 0000 , 0011 1011

M = 0111 1101 0000 0011 , 0111
0010 = 7D03,72

Hexadecimal	Decimal	Binario
0	0	0000
1	1	0001
2	2	0010
3	3	0011
4	4	0100
5	5	0101
6	6	0110
7	7	0111
8	8	1000
9	9	1001
A	10	1010
B	11	1011
C	12	1100
D	13	1101
E	14	1110
F	15	1111

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

- Ejemplos:

- De hexadecimal a binario:

1ABC701,C4) $_{16}$ =

1 A B C 7 0 1 C 4

= 0001 1010 1011 1100 0111 0000 0001,1100 0100)$_2$

- De binario a hexadecimal:

2 5 D F B A

0010 0101 1101 1111 , 1011 1010)$_2$ = 25DF,BA)$_{16}$

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

- Para transformar un número de hexadecimal a decimal se aplica la expresión general de transformación con $b = 16$.
- Para pasar un número de decimal a hexadecimal se hace de forma análoga al caso binario: la parte entera se divide por 16, así como los cocientes enteros sucesivos, y la parte fraccionaria se multiplica por 16, así como las partes fraccionarias de los productos sucesivos.

El código hexadecimal se suele utilizar cuando el número de bits a representar es múltiplo de 4.

3.1. CÓDIGOS INTERMEDIOS: HEXADECIMAL.

- **Hexadecimal a decimal**

  $A798C,1E)_{16} = 10\cdot16^4+7\cdot16^3+9\cdot16^2+8\cdot16^1+12\cdot16^0+1\cdot16^{-1}+14\cdot16^{-2} =$
  $= 686476,117)_{10}$

- **Decimal a hexadecimal**

  $\mathbf{285,12)_{10}=11D,1E)_{16}}$

$$\begin{array}{r|l}
285 & 16 \\ \hline
13 & 17 \;|\; 16 \\
\mathbf{D} & \mathbf{1} \quad \mathbf{1}
\end{array}$$

$$\begin{array}{r}
0,12 \\
\times\ 16 \\ \hline
\mathbf{1},92
\end{array}
\qquad
\begin{array}{r}
0,92 \\
\times\ 16 \\ \hline
14,72 \\
\mathbf{E}
\end{array}$$

3.2. CÓDIGOS INTERMEDIOS: OCTAL. APÉNDICE.

- Un número octal puede pasarse a binario aplicando los algoritmos vistos; no obstante, al ser $b = 8 = 2^3$, puede hacerse la conversión fácilmente

  - Para transformar un número binario a octal se forman grupos de tres cifras binarias a partir del punto decimal hacia la izquierda y hacia la derecha. Posteriormente se efectúa directamente la conversión a octal de cada grupo individual.

  - De octal a binario se pasa sin más que convertir individualmente a binario (tres bits) cada cifra octal, manteniendo el orden del número original.

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

3.2. CÓDIGOS INTERMEDIOS: OCTAL. APÉNDICE.

Ejemplos:

- $75032{,}27\,)_8 = 111\ 101\ 000\ 011\ 010\ ,\ 010\ 111)_2$

- $011\ 000\ 101\ 001\ 111\ 001\ ,\ 101\ 100)_2 = 305171{,}54)_8$

Octal	Binario
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

3.2. CÓDIGOS INTERMEDIOS: OCTAL. APÉNDICE.

- Para transformar un número de octal a decimal se aplica la expresión general de transformación con $b = 8$.
- Para pasar un número de decimal a octal se hace de forma análoga al caso binario: la parte entera se divide por 8, así como los cocientes enteros sucesivos, y la parte fraccionaria se multiplica por 8, así como las partes fraccionarias de los productos sucesivos.

El código octal se suele utilizar cuando el número de bits a representar es múltiplo de 3.

3.2. CÓDIGOS INTERMEDIOS: OCTAL. APÉNDICE.

Ejemplos:

- **Octal a decimal:**

$1367,25)_8 = 1\cdot8^3 + 3\cdot8^2 + 6\cdot8^1 + 7\cdot8^0 + 2\cdot8^{-1} + 5\cdot8^{-2} = 759,33)_{10}$

- **Decimal a octal:**

$\mathbf{760,33)_{10} = 1370,25)_8}$

760	8		
40	95	8	
0	15	11	8
	7	**3**	**1**

$$\begin{array}{r} 0,33 \\ \times\ 8 \\ \hline \mathbf{2},64 \end{array} \qquad \begin{array}{r} 0,64 \\ \times\ 8 \\ \hline \mathbf{5},22 \end{array}$$

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMATICA.

1. REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS.
2. SISTEMA DE NUMERACIÓN EN BASE DOS.
   1. Definición del sistema binario.
   2. Transformaciones de base binaria a decimal.
   3. Transformaciones de base decimal a binaria.
3. CÓDIGOS INTERMEDIOS.
   1. Base hexadecimal.
   2. Base octal (apéndice).
4. EJERCICIOS.

4. EJERCICIOS.

- Pasar los siguientes números de decimal a binario:

  $26{,}1875)_{10}$; $125{,}42)_{10}$
- Pasar de binario a decimal los siguientes números:

  $0{,}10100)_2$; $11001{,}110)_2$

- Transformar de hexadecimal a binario: $A798C{,}1E)_{16}$
- Transformar de binario a hexadecimal:

  $11111111011111000010)_2$

4. EJERCICIOS.

- Transformar de hexadecimal a decimal: $3B5E,34)_{16}$
- Transformar de decimal a hexadecimal: $314,22)_{10}$

- Transformar de BCD a decimal:

  0011 1000 0111 , 1001 0010
- Transformar de decimal a BCD:

  $745,2345)_{10}$

4. EJERCICIOS.

- Sugerencia de debate abierto y trabajo autónomo para el estudiante:

  ¿Cómo cree el estudiante que se podrían representar en binario tanto números positivos como negativos?.

  IDEA: Cuando se representa un número positivo o negativo en decimal aparecen, aparte de los 10 símbolos del alfabeto A={0, 1, 2, 3, 4, 5, 6, 7, 8, 9}, unos nuevos símbolos que representan el signo (+ ó -). Sin embargo, al representar el número en binario sólo se pueden utilizar los dos símbolos del alfabeto binario A={0, 1}. ¿Cuál sería la posible solución para representar el signo?.

- Se hablará de ello en el Tema 1.

SEMINARIO 1. SISTEMAS DE NUMERACIÓN USUALES EN INFORMÁTICA.

TECNOLOGÍA Y ORGANIZACIÓN DE COMPUTADORES

1º Grado en Ingeniería Informática.