Fundamentos de Informática. Representación de la información

L2.0 SISTEMAS DE NUMERACIÓN USUALES EN INFORMÁTICA

Contenidos

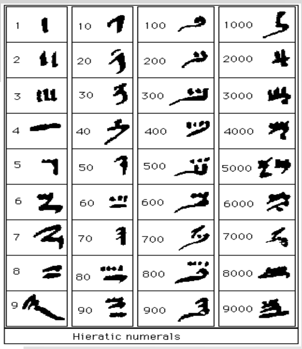
- **Representación posicional de los números.**
- **El sistema de numeración en base 2:**
  - Transformaciones de binario a decimal y viceversa.
  - Operaciones aritméticas básicas.
- **Representación en complementos:**
  - Complemento a uno.
  - Complemento a dos.
- **El sistema de numeración hexadecimal:**
  - Transformaciones entre hexadecimal y binario.
  - Transformaciones entre hexadecimal y decimal.

Sistemas de numeración

	Nº Romano		Nº Romano		Nº Romano		Nº Romano	nº	Nº Romano
1	I	26	XXVI	51	LI	76	LXXVI	200	CC
2	II	27	XXVII	52	LII	77	LXXVII	300	CCC
3	III	28	XXVIII	53	LIII	78	LXXVIII	400	CD
4	IV	29	XXIX	54	LIV	79	LXXIX	500	D
5	V	30	XXX	55	LV	80	LXXX	600	DC
6	VI	31	XXXI	56	LVI	81	LXXXI	700	DCC
7	VII	32	XXXII	57	LVII	82	LXXXII	800	DCCC
8	VIII	33	XXXIII	58	LVIII	83	LXXXIII	900	CM
9	IX	34	XXXIV	59	LIX	84	LXXXIV	1000	M
10	X	35	XXXV	60	LX	85	LXXXV	1100	MC
11	XI	36	XXXVI	61	LXI	86	LXXXVI	1200	MCC
12	XII	37	XXXVII	62	LXII	87	LXXXVII	1300	MCCC
13	XIII	38	XXXVIII	63	LXIII	88	LXXXVIII	1400	MCD
14	XIV	39	XXXIX	64	LXIV	89	LXXXIX	1500	MD
15	XV	40	XL	65	LXV	90	XC	1600	MDC
16	XVI	41	XLI	66	LXVI	91	XCI	1700	MDCC
17	XVII	42	XLII	67	LXVII	92	XCII	1800	MDCCC
18	XVIII	43	XLIII	68	LXVIII	93	XCIII	1900	MCM
19	XIX	44	XLIV	69	LXIX	94	XCIV	2000	MM
20	XX	45	XLV	70	LXX	95	XCV	2100	MMC
21	XXI	46	XLVI	71	LXXI	96	XCVI	2200	MMCC
22	XXII	47	XLVII	72	LXXII	97	XCVII	2300	MMCCC
23	XXIII	48	XLVIII	73	LXXIII	98	XCVIII	2400	MMCD
24	XXIV	49	XLIX	74	LXXIV	99	XCIX	2500	MMD
25	XXV	50	L	75	LXXV	100	C	2600	MMDC

Para números romanos igual o superiores a 4000, se coloca una linea horizontal por encima del número
Para indicar a la base que la multiplicación es por 1000

$\bar{V}$	5.000
$\bar{X}$	10.000
$\bar{L}$	50.000
$\bar{C}$	100.000
$\bar{D}$	500.000
$\bar{M}$	1000.000



Hieratic numerals

REPRESENTACIÓN POSICIONAL DE LOS NÚMEROS

- El mundo le debe a la cultura india el invento trascendental del sistema de numeración **posicional** así como el descubrimiento del 0.
- Un sistema de numeración en base **b** utiliza para representar los números un **conjunto de símbolos** (**S**) compuesto por **b** elementos (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, y
  - de la posición que ocupe dentro del número.

- N = 675,24

Concepto del sistema posicional (*indo-arábigo*)

- **La representación de un número en una base *b*:**

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

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

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

Parámetros: valor relativo de la cifra, posición y peso asociado a la posición

Ejemplo de número decimal:

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

N =	**4**	**5**	**6**	**7**	**,**	**2**	**8**
Posición →	3	2	1	0		-1	-2
Peso →	10^3	10^2	10^1	10^0	,	10^{-1}	10^{-2}

$$4567{,}28 \;=\; 4000 + 500 + 60 + 7 + 0{,}2 + 0{,}08$$

Sistema binario

- Los computadores suelen efectuar 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 octal y hexadecimal 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.

Sistemas de numeración usuales en Informática

Denominación	*Base*	*Símbolos utilizados (alfabeto)*
Decimal	b = 10	$S_{10} = \{0,1,2,3,4,5,6,7,8,9\}$
Binario	b = 2	$S_2 = \{0,1\}$
Hexadecimal	b = 16	$S_{16} = \{0,1,2,3,4,5,6,7,8,9,A,B,C,D,E,F\}$

Ejemplo de número hexadecimal:

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

	3	A	B	4 ,	7
Posiciones →	3	2	1	0	-1
Pesos →	16^3	16^2	16^1	16^0	16^{-1}
	4096	256	16	1	0,0625

$3AB4,7)_{16}$ = 3 x 4096 + A x 256 + B x 16 + 4 + 7 x 0,0625
= 3 x 4096 + (10) x 256 + (11) x 16 + 4 + 7 x 0,0625
= $15.028,44)_{10}$

Ejemplo de número binario:

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

	1	0	1	0	1	1 ,	0	0	1
Posiciones →	5	4	3	2	1	0	-1	-2	-3
Pesos →	2^5	2^4	2^3	2^2	2^1	2^0	2^{-1}	2^{-2}	2^{-3}
	32	16	8	4	2	1	0,5	0,25	0,125

$101011,001)_2$ = $1 \cdot 2^5 + 1 \cdot 2^3 + 1 \cdot 2^1 + 1 \cdot 2^0 + 1 \cdot 2^{-3}$ =
32 + 8 + 2 +1 + 0,125 = $51,125)_{10}$

Para pasar de binario a decimal: se suman los pesos de las posiciones donde hay un 1

SISTEMA DE NUMERACIÓN BASE 2

- b=2, A={0, 1}

Números binarios del 0 al 7

Nº decimal	Nº binario		
0	**0**	**0**	**0**
1	**0**	**0**	**1**
2	**0**	**1**	**0**
3	**0**	**1**	**1**
4	**1**	**0**	**0**
5	**1**	**0**	**1**
6	**1**	**1**	**0**
7	**1**	**1**	**1**
Pesos →	***4***	***2***	***1***

Suma aritmética de nº binarios

Suma aritmética		
a	b	a+b
0	0	0
0	1	1
1	0	1
1	1	10

Acarreos →		*1*	*1*	*1*		
		1	1	0	1	0
+		0	1	1	1	1
Resultado →	1	0	1	0	0	1

Resta aritmética de nº binarios

Resta aritmética		
a	b	A-b
0	0	0
0	1	1 y adeudo 1
1	0	1
1	1	0

	1	0	1	0	1
-	0	1	0	1	0
Adeudos →	1		1		
Resultado →	0	1	0	1	1

Producto aritmético de números binarios

Producto aritmético		
a	b	a·b
0	0	0
0	1	0
1	0	0
1	1	1

			1	0	1	1	0
x					1	0	1
			1	0	1	1	0
		0	0	0	0	0	
	1	0	1	1	0		
Resultado →	1	1	0	1	1	1	0

Representación en complementos

- **Simplifica considerablemente la realización de operaciones de datos con signo**
- **De sumo interés en el caso de los computadores ya que al utilizarla se reduce considerablemente la complejidad de la ALU**
- **Para representar un número negativo se puede utilizar el complemento de ese número a 1, o el complemento a 2.**

En efecto, operar con números con signo es complejo:

- **R = A + B**
  - A= 4 ; B= -3
    - R = |A|- |B| = 4 – 3 = 1
  - A = - 4 ; B = 3
    - R = - (|A|- |B|) = - (4-3) = -1
  - A= 4 ; B = -7
    - R = - (|B|- |A|) = - (7-4) = -3
  - A = -4 ; B = 1
    - R = - (|B|- |A|) = - (4-1) = -3

Algoritmo de suma de dos números con signo R = A+B

- **Si signo de A = signo de B**
  - Sumar los valores absolutos: |R|= |A|+|B|
  - Signo de R = signo de A (o de B)
- **Si signo de A ≠ signo de B**
  - Si A > B; hacer
    - |R| = |A|-|B|
    - signo de R = signo de A
  - Si A<B ; hacer:
    - |R| = |B|-|A|
    - signo de R = signo de B

  - **¡¡ MUCHOS CICLOS DE RELOJ PARA UNA SIMPLE SUMA ¡¡¡**

COMPLEMENTO A UNO

- **Para transformar un número binario, *N*, a complemento a 1 basta con cambiar en *N* los unos por ceros y los ceros por unos.**

0010 1001 → C1(0010 1001) = 1101 0110

1001 0011 → C1(1001 0011) = 0110 1100

RESTA UTILIZANDO EL COMPLEMENTO

- **Podemos restar dos números binarios sumando al minuendo el complemento a uno del substraendo. La cifra que se arrastra del resultado se descarta y se suma al resultado previamente obtenido:**

```
  1011 1101                                    1011 1101
- 1001 0011    Complemento a 1  →       +      0110 1100
  ---------                            -----------------
  0010 1010                             (1)0010  1001
                                                  +  1
                                       -----------------
                                              0010 1010
```

COMPLEMENTO A DOS

- **Para transformar un número binario, N, a complemento a 2 basta con cambiar en N los unos por ceros y los ceros por unos y sumar 1 al resultado anterior.**

1001 0011 → C2(1001 0011) =

= C1(1001 0011) + 1 =

= 0110 1100 + 1 = 0110 1101

Resta utilizando el complemento a dos

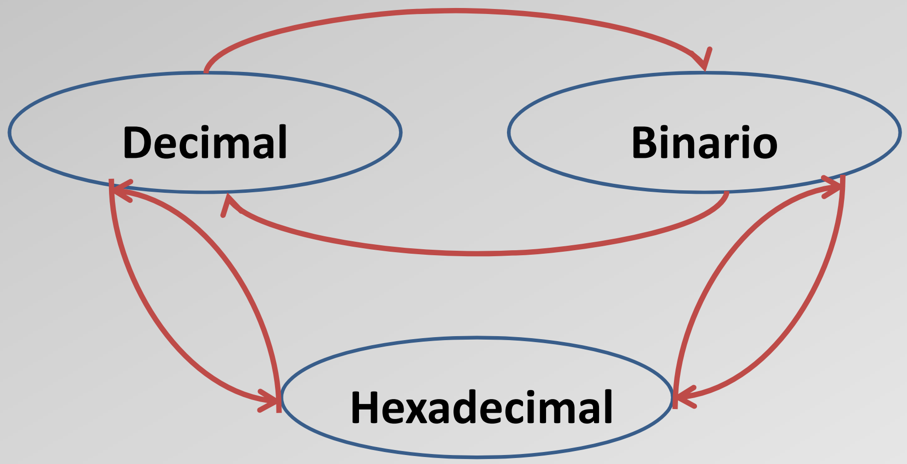
- **Podemos restar dos números sumando al minuendo el complemento a 2 del substraendo. La cifra que se arrastra del resultado se descarta:**

	1011 1101			1011 1101
-	1001 0011	Complemento a 2 →	+	0110 1101
	0010 1010			~~(1)~~ 0010 1010

La notación en complementos simplifica considerablemente las sumas y restas

- **Representando los números negativos en complemento a 2 o a 1, se pueden hacer directamente las operaciones sin tener en cuenta el signo**

Transformaciones de números entre distintas bases



Transformación de binario a decimal

- **Para transformar un número binario a decimal:**
  - Se aplica la expresión:

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

Transformación de decimal a binario (parte entera)

- **La parte entera del número binario se obtiene de la siguiente forma:**
  - Se divide el nº decimal por 2 la parte entera del número decimal de partida, y de los cocientes que sucesivamente se vayan obteniendo (sin obtener decimales en el cociente).
  - Los residuos (restos) de estas divisiones y el último cociente (que serán siempre ceros o unos) son las cifras binarias.
  - El último cociente es el bit más significativo (**MSB**: *Most Significative Bit*) y el primer residuo será el bit menos significativo (**LSB**: *Least Significative Bit*).

Transformación de decimal a binario (parte fraccionaria)

- 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 las partes enteras (que serán ceros o unos) de los productos obtenidos, siendo el bit más significativo el del primer producto, y el menos significativo el del último producto.

Transformar a binario el número decimal 74,423

a) parte entera:

74	2					
14	37	2				
0	17	18	2			
	1	*0*	9	2		
			1	4	2	
				0	2	2
					0	*1*

LSB

MSB

b) parte fraccionaria:

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

MSB

LSB

Es decir:

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

- 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.

- Si el número binario se almacena con un número prefijado de bits se producirá en la representación binaria un error de redondeo.

CÓDIGOS INTERMEDIOS: hexadecimal y octal

- Usualmente se utilizan como códigos intermedios los sistemas de numeración en base 16 (o hexadecimal) o base 8 (octal).
- Los códigos intermedios se fundamentan en la facilidad de transformar un número en base 2 a otra base que sea una potencia entera de 2 ($8=2^3$, $16=2^4$), y viceversa.
- En la actualidad sólo se suele utilizar la base 16 debido a que cada cifra hexadecimal se transforma en 4 bits, y los tamaños de las palabras y datos que se usan en un computador por lo general son múltiplos de 4 (8, 16, 32, 64 y 128 bits)

Código hexadecimal

b=16;
S = {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F}

Cifras hexadecimales y sus valores decimal y binario

Nº hexadecimal	Nº decimal	Nº binario
0	0	0 0 0 0
1	1	0 0 0 1
2	2	0 0 1 0
3	3	0 0 1 1
4	4	0 1 0 0
5	5	0 1 0 1
6	6	0 1 1 0
7	7	0 1 1 1
8	8	1 0 0 0
9	9	1 0 0 1
A	10	1 0 1 0
B	11	1 0 1 1
C	12	1 1 0 0
D	13	1 1 0 1
E	14	1 1 1 0
F	15	1 1 1 1
	Pesos →	*8 4 2 1*

Transformaciones de hexadecimal a binario

- **Un número hexadecimal puede transformarse a binario, y viceversa, aplicando algoritmos similares a los vistos anteriormente para trasformar de decimal a binario y viceversa.**
  - Haciendo divisiones sucesivas de la parte entera del nº hexadecimal por 2, y multiplicando la parte decimal sucesivamente por 2…

Transformación directa de binario a hexadecimal

- **Al ser la base una potencia entera de 2 ($b=18=2^4$) puede hacerse la conversión directamente de la siguiente forma:**
  - Se forman grupos de cuatro cifras binarias a partir del punto decimal hacia la izquierda y hacia la derecha.
  - Posteriormente se efectúa directamente la conversión a hexadecimal de cada grupo individual de cuatro bits.

Transformación directa de binario a hexadecimal

- **Ejemplo:**
  - Transformar a hex el nº binario ***11100010110011,000111***

0011	1000	1011	0011	,	0001	1100
3	8	B	3	,	1	C

  - Luego: ***11100010110011,000111)$_2$ = 38B3,1C)$_{16}$***

Transformación de hexadecimal a binario

– Convertir individualmente a binario (4 bits) cada cifra hexadecimal, manteniendo el orden del número original.

- Ejemplo

2	A	8	7	,	5	C
10	1010	1000	0111	,	0101	11

Nº hexadecimal	Nº decimal	Nº binario
0	0	0 0 0 0
1	1	0 0 0 1
2	2	0 0 1 0
3	3	0 0 1 1
4	4	0 1 0 0
5	5	0 1 0 1
6	6	0 1 1 0
7	7	0 1 1 1
8	8	1 0 0 0
9	9	1 0 0 1
A	10	1 0 1 0
B	11	1 0 1 1
C	12	1 1 0 0
D	13	1 1 0 1
E	14	1 1 1 0
F	15	1 1 1 1
	Pesos →	*8 4 2 1*

Transformaciones entre hexadecimal y decimal

- **Para transformar un número de hexadecimal a decimal se aplica la expresión general con b=16.**

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

- **Para pasar un número de decimal a hexadecimal se hace de forma análoga al paso de decimal a 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.

Ejemplo de conversión de hexadecimal a decimal

Obtener el equivalente decimal del número hexadecimal $A798C,1E)_H$.

$A798C,1E)_H = (10) \cdot 16^4 + 7 \cdot 16^3 + 9 \cdot 16^2 + 8 \cdot 16 + 12 + 16^{-1} + (14) \cdot 16^{-2} =$
$= 655360 + 28672 + 2304 + 128 + 12 + 0{,}0625 + 0{,}0546875 =$
$= 686476.1171)_{10}$

Obtener el equivalente hexadecimal del número decimal 4373,79

Parte entera:

```
4573 |16
137   285 |16
 093  125   17 |16
  13   13    1  1
```

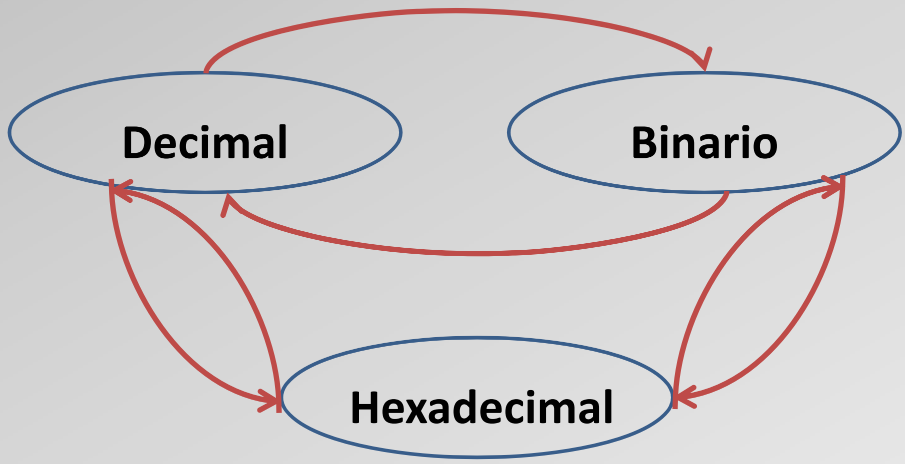
Parte fraccionaria:

```
  0,79     0,64     0,24      0,84  ....
 x  16    x  16    X  16     x  16
 12,64    10,24     3,84     13,44  ....
```

Luego: $4373{,}79)_{10} \approx 11DD{,}CA3D)_H$

Nº hexadecimal	Nº decimal	Nº binario
0	0	0 0 0 0
1	1	0 0 0 1
2	2	0 0 1 0
3	3	0 0 1 1
4	4	0 1 0 0
5	5	0 1 0 1
6	6	0 1 1 0
7	7	0 1 1 1
8	8	1 0 0 0
9	9	1 0 0 1
A	10	1 0 1 0
B	11	1 0 1 1
C	12	1 1 0 0
D	13	1 1 0 1
E	14	1 1 1 0
F	15	1 1 1 1
	Pesos →	*8 4 2 1*

Con muchas frecuencia para pasar de binario a decimal, y al revés, se hace a través de hexadecimal



¿Por qué y para qué se utiliza la notación hexadecimal?

- Facilidad en las conversiones binario ←→hexadecimal
- Facilidad y rapidez de implementación de las conversiones electrónicamente o por programa.
- Por lo general el computador y los programas utilizan esta notación intermedia tanto internamente o como entrada/salida.
- Programa *hexedit*

```
 0  48 6f 79 20 68 61 63 65 20 62 75 65 6e 20 74 69  Hoy hace buen ti
10  65 6d 70 6f 20 79 20 75 6e 61 20 74 65 6d 70 65  empo y una tempe
20  72 61 74 75 72 61 20 64 65 20 32 35 20 67 72 61  ratura de 25 gra
30  64 6f 73 2e 20 41 41 41 41 20 61 61 61 61 20 62  dos. AAAA aaaa b
40  62 62 62 20 30 30 30 30 20 31 31 31 31 20 2e 2e  bbb 0000 1111 ..
50  2e 2e 00                   0                     ..        .
```

Resumen

- **Representación posicional de los números.**
- **El sistema de numeración en base 2:**
  - Transformaciones de binario a decimal y viceversa.
  - Operaciones aritméticas básicas.
- **Representación en complementos:**
  - Complemento a uno.
  - Complemento a dos.
- **El sistema de numeración hexadecimal:**
  - Transformaciones entre hexadecimal y binario.
  - Transformaciones entre hexadecimal y decimal.