
	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

SISTEMA NUMERACIÓN

Un ordenador es una máquina que procesa información. La ejecución de una tarea implica la realización de unos tratamientos, según especifica un conjunto ordenado de instrucciones (es decir, un programa) sobre unos datos. Para que el ordenador ejecute un programa es necesario darle información de dos tipos:

- Instrucciones que forman el programa
- Los datos con los que debe operar ese programa

Uno de los aspectos más importantes relacionado con la información, es cómo representarla. Normalmente se le da al ordenador en la forma usual escrita que utilizan los humanos, es decir, con ayuda de un alfabeto o conjunto de símbolos, los caracteres.

Los caracteres que se utilizan para la representación externa son:

- Numéricos: Constituidos por las diez dígitos en el sistema decimal
- Alfabéticos: Letras mayúsculas y minúsculas
- Especiales: Son símbolos no incluidos en los grupos anteriores, como:), (, *, /, +, -, [,]...

Al conjunto de los dos primeros grupos se le denominan caracteres alfanuméricos.



Veremos cómo estos caracteres usados en la representación externa son representables en los ordenadores. Este paso de una representación a otra se denomina codificación y el proceso inverso decodificación.

Por lo tanto hay dos niveles en la representación de la información

- Nivel de representación externa: usada por las personas e inadecuada para el ordenador
- Nivel de representación interna: adecuada al ordenador y no inteligible directamente por el ser humano.

Las informaciones más complejas se reducirán a un conjunto de informaciones elementales por técnicas de codificación.

Los elementos básicos que constituyen un ordenador son de naturaleza binaria, ya que sólo pueden adoptar dos valores, 0 y 1 (corresponden a dos niveles de tensión, dos valores de corriente, dos situaciones de una lámpara...). Al tener que traducir

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

toda la información suministrada a ceros y unos es necesario establecer una correspondencia entre el conjunto de todos los caracteres:

$$\{A, B, C, D, \dots Z, a, b, c, \dots z, 0, 1, \dots 9, /, +, \dots\}$$

y el conjunto binario:

$$\{0, 1\}^n$$

de forma que a cada elemento del primero le corresponda un elemento distinto del segundo.

Estos códigos de transformación se denominan códigos entrada/salida (E/S) o externos y se pueden definir de forma arbitraria. Las operaciones aritméticas con datos numéricos se suelen realizar en una representación más adecuada para este objetivo que la del código de E/S. Por ello en el propio ordenador se efectúa una transformación entre códigos binarios, obteniéndose una representación fundamentada en el sistema de numeración en base dos, que al ser una representación numérica posicional es muy apta para realizar operaciones aritméticas.

5.2 Códigos de entrada/salida.

Los códigos de E/S o externos son códigos que asocian a cada carácter una combinación de bit. En otras palabras, un código de E/S es una correspondencia entre los conjuntos:

$$A = \{0, 1, \dots 9, A, B, \dots Z, a, b, \dots z, *, +, /, \dots\}$$



y

$$B = \{0, 1\}^n$$

Si se usa un número fijo, n , de bit para codificar los símbolos de A , el valor mínimo de n dependerá del número m de elementos de A . Así:

Con 2 bit ($n=2$) podemos hacer 4 combinaciones distintas y se pueden codificar hasta 4 símbolos ($m=4$) distintos

Con 3 bit ($n=3$) podemos hacer 8 combinaciones distintas y se pueden codificar hasta 8 símbolos ($m=8$) distintos

	<p style="text-align: center;">SISTEMAS INFORMÁTICOS</p> <p style="text-align: center;">ANEXO UT1. SISTEMA NUMERACIÓN</p>	 <p style="text-align: center;">Rosa María Zapata Calle</p>
---	--	---

Con 4 bit ($n=4$) podemos realizar 16 combinaciones y se pueden codificar hasta 16 símbolos ($m=16$) distintos

....

Con n bit pueden codificarse $m = 2^n$ símbolos distintos.

Para codificar m símbolos distintos necesitamos n bit, siendo,

$$n = \log_2 m = 3.32 * \log m$$

Es decir, n debe ser el menor entero que verifique la relación anterior.

Ejemplo: Para codificar las 10 cifras decimales (0, 1,...,9) se necesitarán:



$$n = 3.32 * \log (m) = 3.32 * \log (10) = 3.32 \text{ bit}$$

es decir 4 bit (ya que con 3 sólo podremos codificar 8 símbolos).

Símbolos	Código 2
0	0000
1	0001
2	1001
3	1000
4	0101
5	0100
6	1100
7	1101
8	0011
9	0010

Pueden hacerse codificaciones con más bit de los necesarios; es decir, podríamos establecer códigos de E/S de forma totalmente aleatoria. Obviamente existen códigos normalizados que suelen ser utilizados por los constructores de ordenadores, son conocidos como:

- BCD de intercambio normalizado.

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

- EBCDIC
- ASCII
- ANSI

5.3 Sistemas de numeración más usuales

Los ordenadores suelen efectuar las operaciones aritméticas utilizando una representación para los datos numéricos basada en el sistema de numeración en base 2 (binario natural). También se utilizan los sistemas de numeración octal y hexadecimal, para obtener códigos intermedios. Un número expresado en uno de estos códigos puede transformarse a binario y viceversa.

Nos basamos en los siguientes:

5.3.2 Sistema de numeración en base dos.

El sistema de numeración en base dos, fue introducido por Gottfried Wilhelm [Leibniz](#) (1646-1716) en el siglo XVII, siendo el más adecuado para usar en las máquinas electrónicas, debido a que utilizan esencialmente sistemas de dos estados, encendido y apagado. En el sistema binario los datos se representan en un sistema que sólo admite dos estados, 0 y 1.

Las operaciones aritméticas se suelen realizar usando una representación de datos y resultados en binario natural.



A) Definición del sistema binario.

En el sistema de numeración binario $b=2$ y el conjunto de símbolos usados es: $\{0, 1\}$

Una muestra de los números enteros binarios que se pueden formar con 3 bit y que corresponden a las cifras decimales $\{0, ..., 7\}$ es:

American National Standards InstituteAmerican National Standards Institute

Binario	Decimal
000	0
001	1
010	2
011	3

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

100	4
101	5
111	6
111	7

B) Transformaciones entre bases binaria y decimal.

Se puede transformar un número binario a decimal sin más que usar la expresión vista anteriormente:

$$.n_4 n_3 n_2 n_1 n_0 n_{-1} n_{-2} \dots)_2 = \dots n_4 2^4 + n_3 2^3 + n_2 2^2 + n_1 2^1 + n_0 2^0 + n_{-1} 2^{-1} + \dots = N_{10}$$

Ejemplo:

Transformar a decimal los siguientes números binarios:

$$\begin{aligned}
 110100_2 &= 1 \cdot 2^5 + 1 \cdot 2^4 + 0 \cdot 2^3 + 1 \cdot 2^2 + 0 \cdot 2^1 + 0 \cdot 2^0 = 2^5 + 2^4 + 2^2 \\
 &= 52_{10} \\
 0.10100_2 &= 0 \cdot 2^0 + 1 \cdot 2^{-1} + 0 \cdot 2^{-2} + 1 \cdot 2^{-3} + 0 \cdot 2^{-4} + 0 \cdot 2^{-5} = 2^{-1} + 2^{-3} \\
 &= 0.625_{10} \\
 10100.001_2 &= 1 \cdot 2^4 + 1 \cdot 2^2 + 1 \cdot 2^{-3} = 20.125_{10}
 \end{aligned}$$



Para transformar un número decimal a binario:

a) La parte entera del nuevo número (binario) se obtiene efectuando divisiones enteras (sin obtener decimales) por dos, de la parte entera del número decimal de partida y de los cocientes que sucesivamente se vayan obteniendo. Los restos de estas divisiones y el último cociente (que serán siempre ceros y unos) son las cifras binarias. El último cociente será el bit más significativo y el primer resto el bit menos significativo (más a la derecha).

Ejemplo:

26_{10} es en binario:

$$\begin{array}{r}
 26 \mid 2 \\
 \hline
 0 \quad 13 \mid 2 \\
 \hline
 1 \quad 6 \mid 2 \\
 \hline
 0 \quad 3 \mid 2 \\
 \hline
 \end{array}$$

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

$$26_{10} = 11010_2$$

b) 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 y unos) de los productos obtenidos.

Ejemplo:

Transformar a binario natural el número decimal 0.1875

0.1875	0.3750	0.7500	0.5000
* 2	* 2	* 2	* 2
-----	-----	-----	-----
0.3750	0.7500	1.5000	1.0000



$0.1875_{10} = 0.0011_2$

Ejemplo:

Transformar a binario el número decimal 74.423

a) Parte entera:

74	2	
0	37	2
1	18	2
	0	9
	1	4
	0	2

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

0 1

b) Parte fraccionaria:

0.423	0.846	0.692	0.384	0.768
*2	*2	*2	*2	*2

0.846	1.692	1.384	0.768	1.536
-------	-------	-------	-------	-------

Es decir:

$74.423_{10} = 1001010.01101..._2$

C) Operaciones aritméticas y lógicas con variables binarias

Una variable binaria puede representar, una cifra de un número en el sistema de numeración en base dos. Las operaciones aritméticas básicas con variables binarias naturales son la suma, resta, multiplicación y división. Estas operaciones son análogas a las realizadas en decimal pero usando ceros y unos.



Tabla de operaciones aritméticas:

Suma aritmética		
A	B	A + B
0	0	0
0	1	1
1	0	1
1	1	0 y llevo 1

Producto aritmético		
A	A * B	
0	0	0
0	1	0
1	0	0
1	1	1

Resta aritmética		
A	B	A - B
0	0	0
0	1	1 y debo 1
1	0	1
1	1	0

División aritmética		
A	B	A/B
0	0	-
0	1	0
1	0	-
1	1	1

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

Ejemplo:

1110101	1101010	1101010
1110110	- 1010111	* 11
-----	-----	-----
11101011	0010011	1101010
		+ 1101010

		10011110

Las operaciones lógicas o booleanas con variables binarias son la suma lógica (+), llamada también función OR, el producto lógico (llamado también AND y la complementación (-) o negación o NOT.



Las tablas son las siguientes:

Suma lógica (OR)			Producto lógico (AND)			Complementación (NOT)	
A	B	A + B	A	B	A * B		
0	0	0	0	0	0	A	-A
0	1	1	0	1	0	0	1
1	0	1	1	0	0	1	0
1	1	1	1	1	1		

Es frecuente también la utilización de las operaciones combinadas como NAND (AND y NOT) y NOR (OR y NOT).

Las tablas son las siguientes:

NAND				NOR			
A	B	A * B	-	A	B	A + B	-(A + B)

	<p style="text-align: center;">SISTEMAS INFORMÁTICOS</p>	<div data-bbox="1053 112 1276 313">  </div> <p style="text-align: center;">Rosa María Zapata Calle</p>
	<p style="text-align: center;">ANEXO UT1. SISTEMA NUMERACIÓN</p>	

		(A*B)				
			0	0	0	1
0	0	0	0	1	1	0
0	1	0	1	0	1	0
1	0	0	1	1	1	0
1	1	1				

D) Representación en complementos

Para representar un número negativo se puede utilizar el complemento de ese número a la base. De esta forma las sumas y restas quedan reducidas a sumas. Este sistema de representación es de sumo interés en el caso de los ordenadores ya que al usarlo se reduce la complejidad de los circuitos.

El complemento a la base de un número, es el número que resulta de restar a cada una de las cifras del número N a la base menos uno del sistema que se esté utilizando y posteriormente sumar uno a la diferencia obtenida.

Ejemplo:

- En base 10:
- Base menos uno del sistema: 9
- Representar el número 63 en complemento a la base.

$$\begin{array}{r}
 N = 63 \quad 99 \quad 36 \\
 -63 \quad +1 \\
 \hline
 36 \quad 37
 \end{array}$$



Es decir, el complemento a 10 (base) del número 63 es 37.

En base 2:

Base menos uno: 1

Complemento a 2 del número 10010 es 01110

$$\begin{array}{r}
 1111 \quad 01101 \\
 -10010 \quad +1 \\
 \hline
 \end{array}$$

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

01101	01110
Complemento a 2 del número 101010 es 010110	
11111	010101
101010	+ 1
-----	-----
010101	010110

Observamos que para transformar un número binario N a complemento a 2 basta con cambiar los 0 por 1 y los 1 por 0 de N y sumar 1 al resultado.

Veremos ahora que la utilidad de esta representación es para convertir la realización de las restas a sumas, lo cual simplifica el diseño del procesador.

Ejemplo: Base 10

Supongamos que se ha de realizar la siguiente operación:
77 - 63

Se puede hacer de dos formas diferentes:

- Directamente: $77 - 63 = 14$
- Utilizando el complemento a 10 del substraendo:

- Complemento a 10 del substraendo 63 es 37

```

99 36
-63 +1
-----

```

```

36 37

```

```

77
+37
-----



```

```

114

```

- (No se considera)

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

El resultado es 14

Es decir, para restar basta con sumar el minuendo con el complemento a la base del substraendo y sin considerar el acarreo.

Ejemplo: Base 2

- Supongamos se ha de efectuar la siguiente resta:

11001 - 10010

Se puede hacer de dos formas:

a) Directamente:

```

11001
-10010
-----
00111

```

b) Usando el complemento a 2 del substraendo:

El substraendo es 10010. Su complemento a 2 se obtiene cambiando 0 por 1 y 1 por 0.

01101
y sumándole 1

```

01101
+   1
-----
01110



```

- Ahora sumamos al minuendo el complemento a 2 del substraendo :

```

11001
+01110
-----

```

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

100111

-(No se considera)

E) Códigos intermedios

Los códigos intermedios se basan en la facilidad de transformar un número en base 2 a otra base que sea potencia de 2 y viceversa. Usualmente se usan como códigos intermedios los sistemas de numeración en base 8 y en base 16 (conocidos como octal y hexadecimal).

- a) OCTAL.

En la base octal, $b = 8$ y el conjunto de símbolos utilizados es: $\{0, 1, \dots, 7\}$

Para convertir un número octal a binario sólo debemos sustituir cada dígito octal por su equivalente binario.

Equivalencias



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

Ejemplo:

6 se sustituye por 110

2 se sustituye por 010

$537.24_8 = 101\ 011\ 111 . 010\ 100_2$

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

que equivale según la tabla, a: 5 3 7 . 2 4

La conversión de binario a octal se realiza juntando en grupos de tres dígitos binarios, comenzando por la izquierda desde el punto decimal y sustituyendo cada grupo por el correspondiente dígito octal.

Ejemplo:

El número binario 10001101100.11010_2 es en octal
 $10\ 001\ 101\ 100 . 110\ 10 = 2154.64_8$

Para pasar un número de octal a decimal aplicamos la expresión:

$$N)_8 = \dots n_4 b^4 + n_3 b^3 + \dots + n_1 b^{-1} + n_2 b^{-2} \dots)_{10}$$

con $b = 8$.

Ejemplo:

Para pasar el número octal 1367.25_8 a decimal:
 $1367.25 = 1 \cdot 8^3 + 3 \cdot 8^2 + 6 \cdot 8^1 + 7 \cdot 8^0 + 2 \cdot 8^{-1} + 5 \cdot 8^{-2} =$
 759.328125_{10}

Para pasar un número entero decimal a octal se hacen sucesivas divisiones enteras del número y los subsiguientes cocientes por 8 (al igual que en binario). Para transformar la parte fraccionaria de un número decimal a octal se hacen sucesivas multiplicaciones por 8 (de la misma forma que en binario).



Ejemplo:

Para pasar el número decimal 760.33_{10} a octal:

```

760 | 8__
40
0  95 | 8__
15
7  11 | 8__
3  1
0.33  0.64  0.12  0.96

```

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

*8	*8	*8	*8
2.64	5.12	0.96	7.68
El número en octal es 1370.2507			

b) HEXADECIMAL

Para representar un número en base hexadecimal ($b = 16$) es necesario disponer de un conjunto o alfabeto de 16 símbolos. Se suele usar el conjunto:



$\{0, 1, 2, \dots, 9, A, B, C, D, E, F\}$

Podemos hacer las conversiones de binario a hexadecimal y viceversa en forma análoga al sistema octal. Ahora bien, aquí utilizaremos grupos de 4 bit en lugar de grupos de 3 bit.

Equivalencias

Equivalencias

HEX	BINARIO	DECIMAL
0	0000	0
1	0001	1
2	0010	2
3	0110	3
4	0100	4
5	0101	5
6	0110	6
7	0111	7
8	1000	8
9	1001	9
A	1010	10
B	1011	11

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

C	1100	12
D	1101	13
E	1110	14
F	1111	15

Ejemplo:

Pasar el número binario $010010111011111.1011101_2$ a hexadecimal.

$$010\ 0101\ 1101\ 1111\ .\ 1011\ 101_2 = 25DF.BA_{16}$$

Ejemplo:

Pasar el número $1ABC701.C4_{16}$ a binario:

$$0001\ 1010\ 1011\ 1100\ 0111\ 0000\ 0001\ .\ 1100\ 0100_2$$

1 A B C 7 0 1 C 4

El número binario resultante es:

$$0001101010111100011100000001.11000100_2$$

Para transformar un número hexadecimal a decimal aplicamos la expresión siguiente con $b=16$.

$$N_{16} = \dots n_4 b^4 + n_3 b^3 + \dots + n_{-1} b^{-1} + n_{-2} b^{-2} \dots)_{10}$$



Ejemplo:

Pasar el número hexadecimal $A798C.1E_{16}$ a decimal.

$$10 \cdot 16^4 + 7 \cdot 16^3 + 9 \cdot 16^2 + 8 \cdot 16^1 + 12 \cdot 16^0 + 1 \cdot 16^{-1} + 14 \cdot 16^{-2} = = 686476.1171_{10}$$

Para pasar un número de decimal a hexadecimal se hace de forma análoga a los casos binario y octal: 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:

	<p style="text-align: center;">SISTEMAS INFORMÁTICOS</p>	<div style="text-align: center;">  Rosa María Zapata Calle </div>
	<p style="text-align: center;">ANEXO UT1. SISTEMA NUMERACIÓN</p>	

El número 4573.79_{10} se
corresponde en hexadecimal:

$4573 \mid 16$

137

$093 \ 285 \mid 16$

13 125

13 17 $\mid 16$

1 1

0.79 0.64 0.24

* 16 * 16 * 16

474 384 144

+79 +64 +24

12.64 10.24 3.84

C A 3

El número en hexadecimal es

$11DD.CA3_{16}$



EJERCICIOS:

1. Completa la siguiente tabla:

Observación: Para pasar de Hexadecimal a octal y viceversa se pasará en primer lugar a binario.

DECIMAL	BINARIO	OCTAL	HEXADECIMAL	C1	C2
14					
	111101,021				
		24			
			A3		
				0010110	
23,34					
	1011101				

2. Desarrolla las siguientes operaciones (haz las operaciones y haz la conversión a decimal):

	SISTEMAS INFORMÁTICOS	
	ANEXO UT1. SISTEMA NUMERACIÓN	Rosa María Zapata Calle

$$110110+110110=$$

$$34+15=$$

$$12-5=$$

$$11011*1011=$$