

Codificación de la información

Sistemas de numeración

Un sistema de numeración es un conjunto de símbolos y reglas que permiten representar datos numéricos.

Los sistemas de numeración actuales son sistemas posicionales. Cada símbolo tiene distinto valor según la posición que ocupa en la cifra.

Ejemplos:

- Decimal: 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 10, 11, 12, 13, 14, 15, 16, ...
- Binario: 000, 001, 010, 011, 100, 101, 110, 111
- Octal: 00, 01, 02, 03, 04, 05, 06, 07, 10, 11, 12, 13, 14, ...
- Hexadecimal: 00, 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E, 0F

Sistema binario

Este es el sistema utilizado por la electrónica, donde una serie de interruptores y transistores pueden tener dos estados:

- Tienen corriente o no la tienen.
- El primer caso se representa con un 1 y el segundo con un 0.

El sistema binario utiliza 2 dígitos, y cada dígito tiene distinto valor dependiendo de la posición que ocupe.



0



1

Los ordenadores con un sistema binario para:

- Guardar información
- Hacer cálculos
- Enviar y recibir información

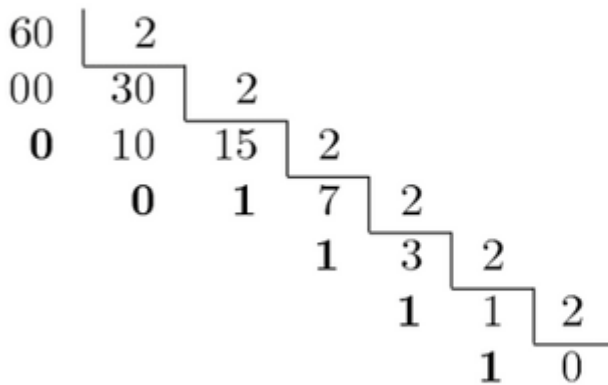


Convertir de decimal a binario

Para pasar un número decimal a binario:

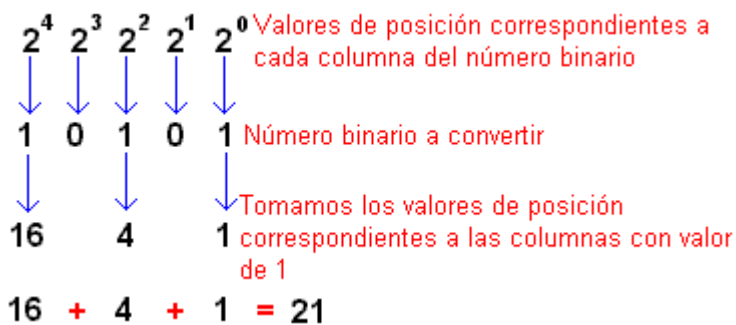
- Realizar divisiones sucesivas por 2
- Al final, escribir los restos obtenidos en cada división en orden inverso

Ejemplo Cálculo del equivalente binario del número decimal 60_{10}



Por tanto, $60_{10} = 111100_2$

Convertir de binario a decimal



Sistema hexadecimal

Este sistema cuenta con 16 dígitos (0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F) y se puede calcular la equivalencia entre el valor decimal de un hexadecimal de forma similar a como se hace con los binarios, pero ahora la base de numeración es 16, valor que habrá que ir elevando a las sucesivas potencias.

Lo que hace interesante el sistema hexadecimal es la inmediatez de transformación entre un número hexadecimal y su equivalente binario natural. Basta con escribir las cuatro cifras binarias de cada dígito para tener la equivalencia

Aplicaciones:

- Direcciones MAC
- Códigos de colores RGB

TABLA DE COLORES CON SUS VALORES HEXADECIMALES

FFFFFF	FFFCC	FFF99	FFF66	FFF33	FFF00	66FFFF	66FFCC	66FF99	66FF66	66FF33	66FF00
FFCCFF	FFCCCC	FFCC99	FFCC66	FFCC33	FFCC00	66CCFF	66CCCC	66CC99	66CC66	66CC33	66CC00
FF99FF	FF99CC	FF9999	FF9966	FF9933	FF9900	6699FF	6699CC	669999	669966	669933	669900
FF66FF	FF66CC	FF6699	FF6666	FF6633	FF6600	6666FF	6666CC	666699	666666	666633	666600
FF33FF	FF33CC	FF3399	FF3366	FF3333	FF3300	6633FF	6633CC	663399	663366	663333	663300
FF00FF	FF00CC	FF0099	FF0066	FF0033	FF0000	6600FF	6600CC	660099	660066	660033	660000
CCFFFF	CCFFCC	CCFF99	CCFF66	CCFF33	CCFF00	33FFFF	33FFCC	33FF99	33FF66	33FF33	33FF00
CCCCFF	CCCCCC	CCCC99	CCCC66	CCCC33	CCCC00	33CCFF	33CCCC	33CC99	33CC66	33CC33	33CC00
CC99FF	CC99CC	CC9999	CC9966	CC9933	CC9900	3399FF	3399CC	339999	339966	339933	339900
CC66FF	CC66CC	CC6699	CC6666	CC6633	CC6600	3366FF	3366CC	336699	336666	336633	336600
CC33FF	CC33CC	CC3399	CC3366	CC3333	CC3300	3333FF	3333CC	333399	333366	333333	333300
CC00FF	CC00CC	CC0099	CC0066	CC0033	CC0000	3300FF	3300CC	330099	330066	330033	330000
99FFFF	99FFCC	99FF99	99FF66	99FF33	99FF00	00FFFF	00FFCC	00FF99	00FF66	00FF33	00FF00
99CCFF	99CCCC	99CC99	99CC66	99CC33	99CC00	00CCFF	00CCCC	00CC99	00CC66	00CC33	00CC00
9999FF	9999CC	999999	999966	999933	999900	0099FF	0099CC	009999	009966	009933	009900
9966FF	9966CC	996699	996666	996633	996600	0066FF	0066CC	006699	006666	006633	006600
9933FF	9933CC	993399	993366	993333	993300	0033FF	0033CC	003399	003366	003333	003300
9900FF	9900CC	990099	990066	990033	990000	0000FF	0000CC	000099	000066	000033	000000
FFFFFF	EEEEEE	DDDDDD	BBBBBB	AAAAAA	888888	777777	555555	444444	222222	111111	000000

Convertir entre binario y hexadecimal

La conversión de un número hexadecimal a uno binario es muy sencilla: basta con sustituir cada cifra hexadecimal por su equivalente en binario.

Por ejemplo:

3AFF = 0011 1010 1111 1111

(Naturalmente, podemos eliminar los dos ceros de la izquierda).

Asimismo, la conversión de un número binario a hexadecimal es igual de sencilla: se agrupan los bits de cuatro en cuatro, comenzando por la derecha (por el bit de menor peso), y luego se sustituye cada grupo de cuatro bits por su equivalente hexadecimal.

1101101011002 --> 1101 1010 1100 --> DAC16

Códificación de caracteres

Código ASCII

La memoria de un ordenador no guarda caracteres, sino información la guarda en formato binario. Para poder guardar letras, necesito transformarlas a 1s y 0s.

```
01000101 01110011 01110100
00100000 01101001 01101110
01101111 01110010 01101101
01100011 01101001 11110011
00100000 01110011 01100101
01110100 01110010 01100001
01110101 01100011 01100101
01100101 01101110 00100000
01101100 00100000 01101111
01100100 01100101 01101110
01100100 01101111 01110010
```

El código ASCII se crea en 1963. Se inventa una tabla en la que a cada letra y una serie de símbolos (alfanuméricos y otros) se le asigna un código binario.

Caracteres de control				Símbolos gráficos											
Nombre	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex	Símbolo	Dec	Binario	Hex
NUL	0	0000000	00	space	32	0100000	20	@	64	1000000	40	`	96	1100000	60
SOH	1	0000001	01	!	33	0100001	21	A	65	1000001	41	a	97	1100001	61
STX	2	0000010	02	"	34	0100010	22	B	66	1000010	42	b	98	1100010	62
ETX	3	0000011	03	#	35	0100011	23	C	67	1000011	43	c	99	1100011	63
EOT	4	0000100	04	\$	36	0100100	24	D	68	1000100	44	d	100	1100100	64
ENQ	5	0000101	05	%	37	0100101	25	E	69	1000101	45	e	101	1100101	65
ACK	6	0000110	06	&	38	0100110	26	F	70	1000110	46	f	102	1100110	66
BEL	7	0000111	07	'	39	0100111	27	G	71	1000111	47	g	103	1100111	67
BS	8	0001000	08	(40	0101000	28	H	72	1001000	48	h	104	1101000	68
HT	9	0001001	09)	41	0101001	29	I	73	1001001	49	i	105	1101001	69
LF	10	0001010	0A	*	42	0101010	2A	J	74	1001010	4A	j	106	1101010	6A
VT	11	0001011	0B	+	43	0101011	2B	K	75	1001011	4B	k	107	1101011	6B
FF	12	0001100	0C	,	44	0101100	2C	L	76	1001100	4C	l	108	1101100	6C
CR	13	0001101	0D	-	45	0101101	2D	M	77	1001101	4D	m	109	1101101	6D
SO	14	0001110	0E	.	46	0101110	2E	N	78	1001110	4E	n	110	1101110	6E
SI	15	0001111	0F	/	47	0101111	2F	O	79	1001111	4F	o	111	1101111	6F
DLE	16	0010000	10	0	48	0110000	30	P	80	1010000	50	p	112	1110000	70
DC1	17	0010001	11	1	49	0110001	31	Q	81	1010001	51	q	113	1110001	71
DC2	18	0010010	12	2	50	0110010	32	R	82	1010010	52	r	114	1110010	72
DC3	19	0010011	13	3	51	0110011	33	S	83	1010011	53	s	115	1110011	73
DC4	20	0010100	14	4	52	0110100	34	T	84	1010100	54	t	116	1110100	74
NAK	21	0010101	15	5	53	0110101	35	U	85	1010101	55	u	117	1110101	75
SYN	22	0010110	16	6	54	0110110	36	V	86	1010110	56	v	118	1110110	76
ETB	23	0010111	17	7	55	0110111	37	W	87	1010111	57	w	119	1110111	77
CAN	24	0011000	18	8	56	0111000	38	X	88	1011000	58	x	120	1111000	78
EM	25	0011001	19	9	57	0111001	39	Y	89	1011001	59	y	121	1111001	79
SUB	26	0011010	1A	:	58	0111010	3A	Z	90	1011010	5A	z	122	1111010	7A
ESC	27	0011011	1B	;	59	0111011	3B	[91	1011011	5B	{	123	1111011	7B
FS	28	0011100	1C	<	60	0111100	3C	\	92	1011100	5C		124	1111100	7C
GS	29	0011101	1D	=	61	0111101	3D]	93	1011101	5D	}	125	1111101	7D
RS	30	0011110	1E	>	62	0111110	3E	^	94	1011110	5E	~	126	1111110	7E
US	31	0011111	1F	?	63	0111111	3F	_	95	1011111	5F	Del	127	1111111	7F

ASCII extendido

El ASCII se desarrolló para utilizarse con inglés. Por tanto no posee:

- Caracteres acentuados
- Caracteres específicos de otros idiomas

Para codificar estos caracteres, se necesitaba un sistema de códigos distinto. Necesitamos más caracteres, con 7 bits no basta (128 máximo) y 32 son caracteres de control

El Código ASCII se extendió a 8 bits (byte). Se pueden codificar más caracteres (ASCII extendido), hasta 256

DEC	BIN	Symbol	DEC	BIN	Symbol	DEC	BIN	Symbol	DEC	BIN	Symbol
128	10000000	€	158	10011110	ž	188	10111100	¼	218	11011010	Ú
129	10000001		159	10011111	ÿ	189	10111101	½	219	11011011	Û
130	10000010	,	160	10100000		190	10111110	¾	220	11011100	Ü
131	10000011	f	161	10100001	i	191	10111111	¿	221	11011101	Ý
132	10000100	n	162	10100010	¢	192	11000000	À	222	11011110	Þ
133	10000101	...	163	10100011	£	193	11000001	Á	223	11011111	ß
134	10000110	†	164	10100100	×	194	11000010	Â	224	11100000	à
135	10000111	#	165	10100101	¥	195	11000011	Ã	225	11100001	á
136	10001000	^	166	10100110		196	11000100	Ä	226	11100010	â
137	10001001	%	167	10100111	§	197	11000101	Å	227	11100011	ã
138	10001010	Š	168	10101000	ˆ	198	11000110	Æ	228	11100100	ä
139	10001011	<	169	10101001	©	199	11000111	Ç	229	11100101	å
140	10001100	Œ	170	10101010	ª	200	11001000	È	230	11100110	æ
141	10001101		171	10101011	«	201	11001001	É	231	11100111	ç
142	10001110	Ž	172	10101100	¬	202	11001010	Ê	232	11101000	è
143	10001111		173	10101101		203	11001011	Ë	233	11101001	é
144	10010000		174	10101110	®	204	11001100	Ì	234	11101010	ê
145	10010001	`	175	10101111	¯	205	11001101	Í	235	11101011	ë
146	10010010	'	176	10110000	°	206	11001110	Î	236	11101100	ì
147	10010011	"	177	10110001	±	207	11001111	Ï	237	11101101	í
148	10010100	"	178	10110010	²	208	11010000	Ð	238	11101110	î
149	10010101	•	179	10110011	³	209	11010001	Ñ	239	11101111	ï
150	10010110	–	180	10110100	´	210	11010010	Ò	240	11110000	ð
151	10010111	—	181	10110101	µ	211	11010011	Ó	241	11110001	ñ
152	10011000	˜	182	10110110	¶	212	11010100	Ô	242	11110010	ò
153	10011001	™	183	10110111	·	213	11010101	Õ	243	11110011	ó
154	10011010	š	184	10111000	,	214	11010110	Ö	244	11110100	ô
155	10011011	>	185	10111001	¹	215	11010111	×	245	11110101	õ
156	10011100	œ	186	10111010	º	216	11011000	Ø	246	11110110	ö
157	10011101		187	10111011	»	217	11011001	Ù	247	11110111	÷

Para no romper la compatibilidad con ASCII, se hace que el primer bit signifique

- 0: Los 7 bits inferiores siguen la tabla ASCII
- 1: Los 7 bits inferiores siguen otra tabla

Ejercicios

1. Convierte los siguientes números decimales al sistema binario:

31
65
100
144
256

2. Convierte los siguientes números binarios a decimal:

11011101
1000001
11101110
1110001101

3. Convierte los siguientes números binarios al sistema hexadecimal:

```
110010001011101
1000111110001011101
11011000100101
```

4. Convierte los siguientes números hexadecimales al sistema binario:

```
AB34
F22
344
```

5. Averigua cuántos números pueden representarse con 8, 10, 16 y 32 bits y cuál es el número más grande que puede escribirse en cada caso.

6. Dados dos números binarios: 01001000 y 01000100 ¿Cuál de ellos es el mayor? ¿Podrías compararlos sin necesidad de convertirlos al sistema decimal?

7. Escribe los nombres de tus compañeros de la derecha e izquierda tuyos en ASCII.

8. Convierte la fecha de tu cumpleaños en binario. ¿Cuántos bits necesitarías?

9. ¿Cuántos KB se necesitan aproximadamente para guardar todos los nombres y fechas de nacimiento de todos los alumnos del colegio?

10. ¿Cómo sabrías si un número binario es par o impar sin convertirlo?

Practicar online

0	0	1	1	0	0	1	0	=	158
0	1	1	1	1	1	0	0	=	172
0	1	1	0	0	0	1	0	=	157
0	0	0	0	1	0	1	0	=	69
1	0	0	0	1	0	1	1	=	96
1	0	1	1	0	1	0	0	=	90