

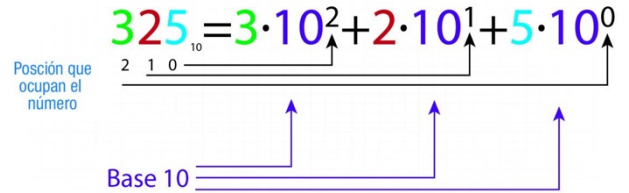
II06. Sistemas de numeración

Existen los siguientes sistemas de numeración:

- Decimal, el que utilizamos nosotros, con 10 dígitos del 0 al 9.
- Binario, el que usan internamente las máquinas, con 2 dígitos del 0 al 1.

Podemos descomponer un número utilizando cada dígito multiplicado por la base (10 en decimal) elevado a su posición (0 para unidades, 1 para decenas, 2 para centenas...):

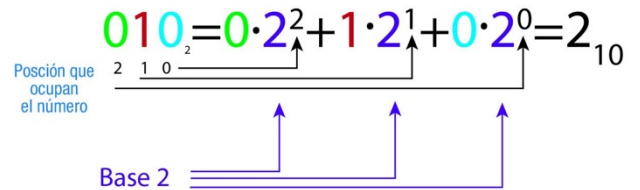
$$\text{Número}_{(\text{decimal} \rightarrow \text{decimal})} = A_n \cdot 2^n + \dots + A_0 \cdot 2^0$$



Conversión binario → decimal

Si aplicamos lo anterior al binario (base 2), podemos pasar cualquier número binario a decimal:

$$\text{Número}_{(\text{binario} \rightarrow \text{decimal})} = A_n \cdot 2^n + \dots + A_0 \cdot 2^0$$

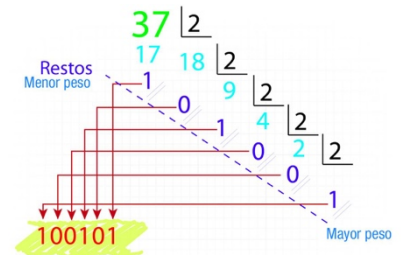


Una clasificación para referirnos a los bits:

- **LSB** (*least significant bit*): El bit de peso "0", el de más a la derecha.
- **MSB** (*most significant bit*): El bit de más a la izquierda.

Conversión decimal → binario

Se divide entre dos hasta que quede cociente "1". Ese es el MSB. Los siguientes bits son los restos de cada



Octal y hexadecimal

Son sistemas basados en binario de tres bits (octal) y de cuatro (hexadecimal). El sistema octal utiliza 8 dígitos (del 0 al 7) y el hexadecimal del 0 a la F. Haciendo grupos de tres y cuatro bits, es muy fácil hacer conversiones usando el binario.

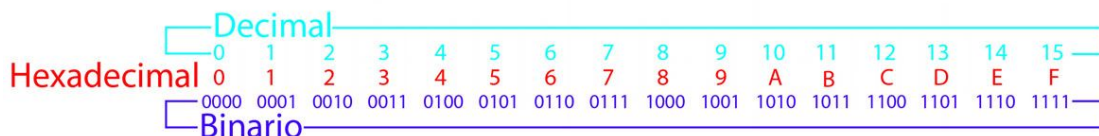
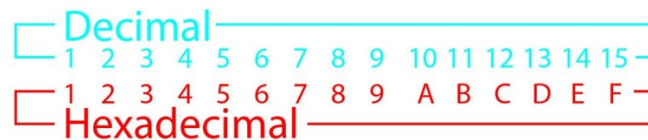


Tabla de números en los cuatro sistemas

Fíjate como en el vídeo “[HCF Reverse Engineering BIOS from IBM PC](#)” de la serie “Halt and Catch Fire” utilizan sistema hexadecimal para **abreviar** dígitos binarios.

Binario				Hexadecimal (= bin 4 bit)	Octal (=bin 3 bit)	Decimal
A ₃	A ₂	A ₁	A ₀			
0	0	0	0	0	0	0
0	0	0	1	1	1	1
0	0	1	0	2	2	2
0	0	1	1	3	3	3
0	1	0	0	4	4	4
0	1	0	1	5	5	5
0	1	1	0	6	6	6
0	1	1	1	7	7	7
1	0	0	0	8	10	8
1	0	0	1	9	11	9
1	0	1	0	A	12	10
1	0	1	1	B	13	11
1	1	0	0	C	14	12
1	1	0	1	D	15	13
1	1	1	0	E	16	14
1	1	1	1	F	17	15

RECUERDA: El número de combinaciones posibles que puedes obtener con un número de bits (binario, base 2) es $2^{n^{\circ} \text{ bits}}$. Así, con los cuatro bits que hay en la tabla: $2^4 = 16$ combinaciones.

Problemas

1. Convierte a binario los siguientes números decimales:

1	3
200	1345
56	3

2. Convierte a decimal los siguientes números binarios:

1	10
101101	10100
111011001	001010110

3. Convierte los siguientes números decimales a hexadecimal.

25	433
344	233
4229	1234
4. Convierte los siguientes números hexadecimales a decimal (IMPORTANTE):

10	AF
231F	3E45A3
34234FAB	025ED
5. Expresa el número 11011011 en decimal, hexadecimal y octal.
6. Completa la tabla:

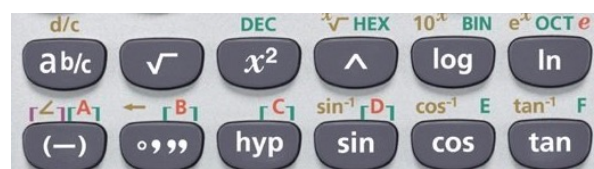
Binario	Octal	Decimal	Hexadecimal
		145	
			FA
	37		
1010110			
	67		
		284	
			C3
10101111			

$n \text{ bits de direcciones} \leftrightarrow 2^n \text{ direcciones de memoria}$

7. Los equipos de 32 y 64 bits:
 - a. ¿Cuántas direcciones de memoria podemos acceder con un equipo de 32 bits?
 - b. ¿Si cada dirección contiene un byte, cuantos bytes de RAM podremos tener como máximo?
 - c. Convierte el resultado anterior a MiB y a GiB.
 - d. Repite el proceso para 64 bits. Resultado en PiB.
8. Expresa el número 11011011 en binario, hexadecimal y octal.
9. Convierte el número 25 (decimal) a hexadecimal.

Calculadora

Ciertas calculadoras lo hacen directamente en el modo "BASE". Fíjate en los números hexadecimales A-F y las teclas para cambiar de base (DEC, HEX...):



Bibliografía

- Materiales formativos de FP Online propiedad del Ministerio de Educación, Cultura y Deporte.
- Límite de memoria: <http://www.zdnet.com/article/clearing-up-the-3264-bit-memory-limit-confusion/>