

## Sistemas Numéricos

### Repaso

Números naturales: La serie de los números naturales abarca desde 1 a infinito

Números enteros: La serie de los números enteros abarca desde -infinito a +infinito

Estas son definiciones simplificadas para el sistema numérico decimal (base 10)

Este sistema (igual que el resto de los sistemas), está basado en 10 símbolos, del 0 al nueve, que tienen dos valores, el valor absoluto, y el valor posicional y cada cifra está compuesta por números con esos valores. Ejemplo:

4567	es igual a	$7 \times 10^0$	$7 \times 1$	7
		$6 \times 10^1$	$6 \times 10$	60
		$5 \times 10^2$	$5 \times 100$	500
		$4 \times 10^3$	$4 \times 1000$	4000
Total				4567

En el caso de un sistema base 2, es similar, hay dos símbolos (0 y 1), con lo cual vemos un ejemplo

10100111	Es igual a	$1 \times 2^0$	$1 \times 1$	1
		$1 \times 2^1$	$1 \times 2$	2
		$1 \times 2^2$	$1 \times 4$	4
		$0 \times 2^3$	$1 \times 8$	0
		$0 \times 2^4$	$1 \times 16$	0
		$1 \times 2^5$	$1 \times 32$	32
		$0 \times 2^6$	$1 \times 64$	0
		$1 \times 2^7$	$1 \times 128$	128
				167

O sea, el número en base 2 **10100111** es igual a **167** en el sistema numérico base 10

Para pasar de un número base 10 a otro base dos se pueden hacer de múltiples formas, a mano es la siguiente (no es la única ...), se divide por dos los números de la primer columna, colocando el resultado en la fila siguiente (sin decimales), y si es impar en la segunda columna va 1 si es impar y 0 si es par. Ejemplo

167	1
83	1
41	1
20	0
10	0
5	1
2	0
1	1



Acomodando los números de la segunda columna de abajo hacia arriba queda 10100111 que es igual al ejemplo anterior

## Rango de representación

En principio los símbolos numéricos base 2 se denominan **bit** que es la combinación de dos palabras **binary digit**, así que de ahora en adelante se utilizará la palabra bit

¿Cuál es el mayor número decimal que puedo representar con n bits?

Se puede utilizar la siguiente fórmula  $0 \leq Nd \leq (2^n - 1)$

Siendo Nd número decimal y n cantidad de bits, o sea

1	1
2	3
3	7
4	15
5	31
6	63
7	127
8	255
9	511
10	1.023
11	2.047
12	4.095
13	8.191
14	16.383
15	32.767
16	65.535
17	131.071
18	262.143
19	524.287
20	1.048.575
21	2.097.151
22	4.194.303
23	8.388.607
24	16.777.215
25	33.554.431
26	67.108.863
27	134.217.727
28	268.435.455
29	536.870.911
30	1.073.741.823
31	2.147.483.647
32	4.294.967.295

O sea, hay un límite en cualquier computadora para representar número decimales.

Típicamente, cuando programen en C, la representación máxima de números no signados (naturales que incluyen el cero) es 65535 (16 bits)

Cuando queremos representar números signados hay tres formas de representación

1. Signo y Magnitud
2. Complemento a dos
3. Binario Desplazado

## 1. Signo y magnitud

Se debe saber de antemano cuantos bits se van a utilizar. Vamos a suponer, por comodidad 4 bits, el primer bit representa el signo, si es 0 es positivo, si es 1 es negativo

Signo	Magnitud			
0	1	1	1	<b>+7</b>
0	1	1	0	<b>+6</b>
0	1	0	1	<b>+5</b>
0	1	0	0	<b>+4</b>
0	0	1	1	<b>+3</b>
0	0	1	0	<b>+2</b>
0	0	0	1	<b>+1</b>
0	0	0	0	<b>+0</b>
1	0	0	0	<b>-0</b>
1	0	0	1	<b>-1</b>
1	0	1	0	<b>-2</b>
1	0	1	1	<b>-3</b>
1	1	0	0	<b>-4</b>
1	1	0	1	<b>-5</b>
1	1	1	0	<b>-6</b>
1	1	1	1	<b>-7</b>

Este sistema representa el cero tanto como positivo y como negativo, y su rango de representación es  $(-2^{n-1} - 1) \leq N_{dsig} \leq +(2^{n-1} - 1)$

En el caso de 16 bits se puede representar desde -32767 hasta +32767

Para representar números signados en C no se utiliza este sistema, porque no permite realizar operaciones aritméticas.

Para este caso se utiliza el sistema denominado Complemento a dos

## 2. Complemento a dos

Antes de empezar, vamos a dar las reglas de condición de suma en binario; las vamos a utilizar dentro de un rato...

	0	← 0		0	← 0		1	← 0		1	← 0
	+		+		+		+		+		
Transporte	0		0	1	0	1	0	1	1	0	
	0	0	0	1	0	1	1	0	1	0	

	0	← 1		0	← 1		1	← 1		1	← 1
	+		+		+		+		+		
Transporte	0		0	1	0	1	0	1	1	1	
	0	1	1	0	1	0	1	0	1	1	

Ahora arrancamos con la forma de representación convenio a dos. Se debe saber de antemano cuantos bits se van a utilizar. Vamos a suponer, por comodidad 4 bits Los números positivos se representan como en su forma binaria natural y los negativos se hace lo siguiente:

Los números negativos se los representan en su forma binaria, se invierten los 1 por 0 y los ceros por 1 y se les suma 1, de acuerdo con lo visto un poco más arriba.

Vamos a ver un caso en cuatro bits, representar el número -5

Se transforma el módulo del número y se lo representa en binario (en este caso en cuatro bits)

5 = 0101

Se invierten 0101 -> 1010

Se suma 1

1	0	1	0
+			
			1
1	0	1	1

La representación del -5 es entonces 1011

0	1	1	1	<b>+7</b>
0	1	1	0	<b>+6</b>
0	1	0	1	<b>+5</b>
0	1	0	0	<b>+4</b>
0	0	1	1	<b>+3</b>
0	0	1	0	<b>+2</b>
0	0	0	1	<b>+1</b>
0	0	0	0	<b>+0</b>
1	1	1	1	<b>-1</b>
1	1	1	0	<b>-2</b>
1	1	0	1	<b>-3</b>
1	1	0	0	<b>-4</b>
1	0	1	1	<b>-5</b>
1	0	1	0	<b>-6</b>
1	0	0	1	<b>-7</b>
1	0	0	0	<b>-8</b>

Este sistema representa el cero como positivo, y su rango de representación es

$$(-2^{n-1}) \leq N_{sig} \leq +(2^{n-1} - 1)$$

Se puede ver en la tabla que los números positivos siempre empiezan en cero y los negativos con uno, esto no implica que el primer bit sea el signo...

Esto último sirve cuando se quiere saber qué número decimal signado representa una combinación de bits. Ejemplo:

Un número de 4 bits 1100, el primer bit es 1 por lo tanto es negativo, se debe proceder de la siguiente forma: se invierten unos por cero y ceros por uno, se suma uno y con eso da la magnitud del número que es negativo. O sea

1. 1100 -> 0011
2. Se suma 1, 0011 + 1 = 0100
3. 0100 representa como magnitud el 4

4. Como dijimos que era negativo queda que el número 1100 representa a -4 (en cuatro bits).

En caso de los números que empiezan con cero solo hay que ver cual es la magnitud directamente.

Ejemplo:

Un número de 4 bits 0011 empieza con cero por lo tanto es positivo y su magnitud es 3, con lo cual queda que 0011 representa el +3

Veamos algunos ejercicios

- a. Convertir de binario a decimal los siguientes números (no signados)

10110010	
00110011	
11001100	
01010101	
10101010	
11110000	
00001111	
11111111	

- b. Convertir de decimal a binario los siguientes números (no signados)

54	
126	
828	
192	
2000	
20000	
123456	
654321	

- c. Suponiendo 8 bits, convertir los siguientes números decimales signados a la representación Signo y Magnitud y Complemento a dos

Número Decimal Signado	Signo y Magnitud	Complemento a Dos
Ejemplo +4	00000100	00000100
Ejemplo -5	10000101	11111011
-45		
+45		
-100		
+100		
-128		

- d. Suponiendo 16 bits, convertir los siguientes números decimales signados a la representación Signo y Magnitud y Complemento a dos de la tabla anterior

Número Decimal Signado	Signo y Magnitud	Complemento a Dos
Ejemplo +4	0000000000000100	0000000000000100
Ejemplo -5	1000000000000101	1111111111111011
-45		
+45		
-100		
+100		
-128		

e. Indicar que números signados (decimal) representan los siguientes sistemas numéricos

Complemento a Dos	Número Decimal Signado
10110011	
01001100	
01100110	
11100001	
00110011	
10110010	
00011101	
11111100	
Signo y Magnitud	Número Decimal Signado
10110011	
01001100	
01100110	
11100001	
00110011	
10110010	
00011101	
11111100	