Tema 1:

Representación de los números

Representación de los números

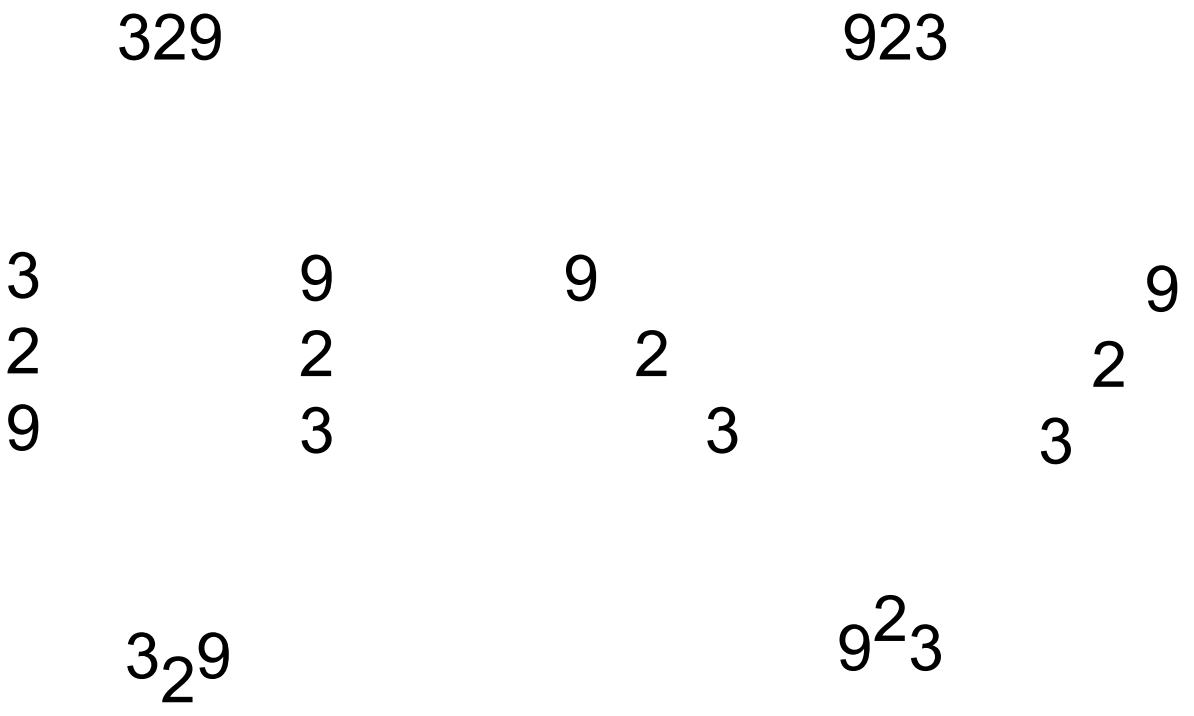
- Objetivos
  - Sistemas de numeración
    - Decimal
    - Binario
    - Octal y hexadecimal
  - Cambios de base
  - Formas de representación de los números
  - Operaciones aritméticas

Representación de la Información

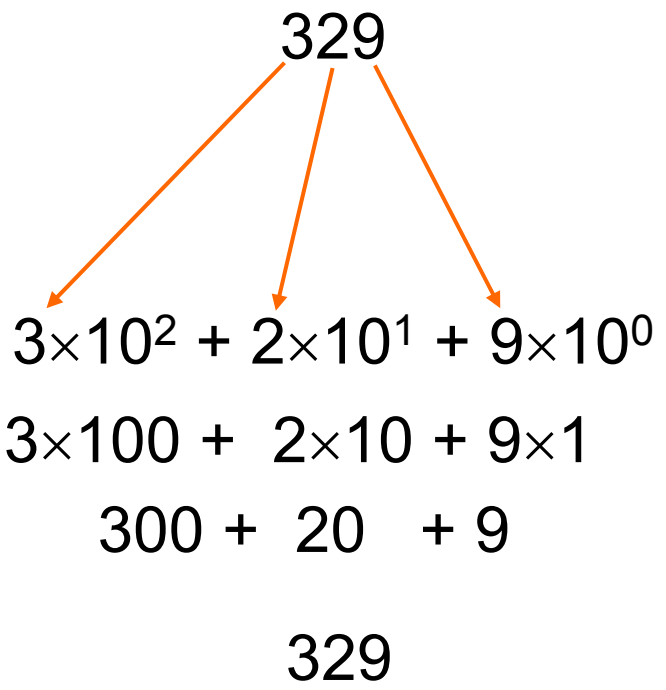
- La información se puede dividir en grupos de símbolos
  - 28 letras abecedario
  - 10 dígitos decimales
  - 7 islas en canarias
- Los sistemas binarios o digitales manipulan la información como 1's y 0's

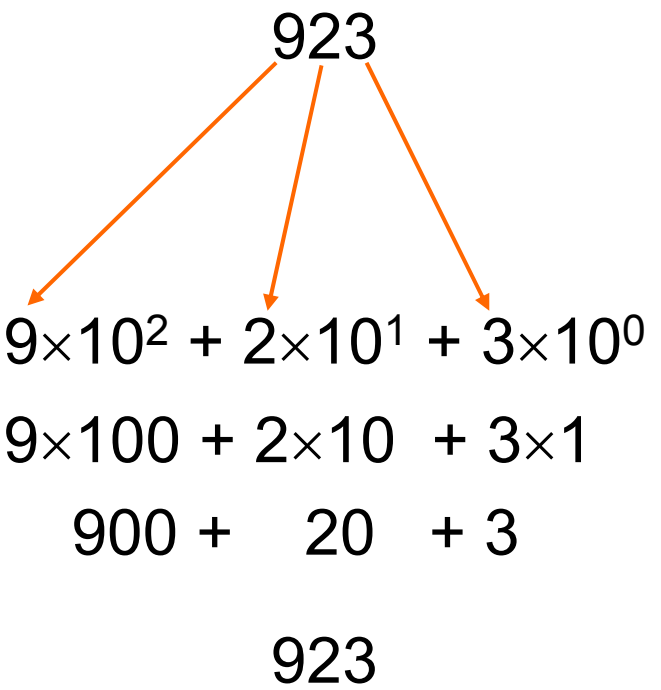
Sistema de numeración posicional

El valor Numérico es representado por una serie de dígitos



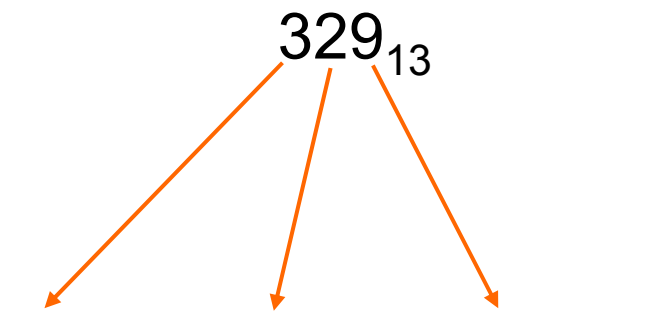
Sistema de numeración posicional

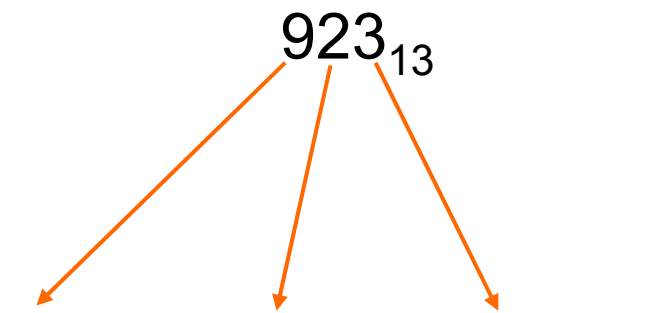




Sistema de numeración posicional

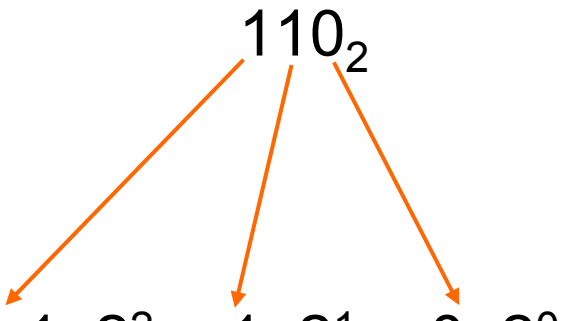
El mismo sistema posicional con diferentes bases:

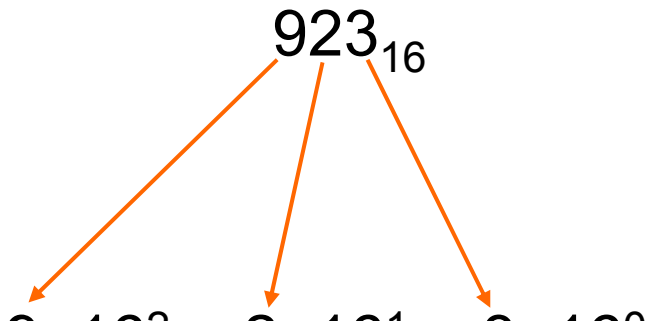




Sistema Binario

En IDL Estamos interesados en las bases 2, 8, y 16.





Sistema de numeración posicional (resumen)

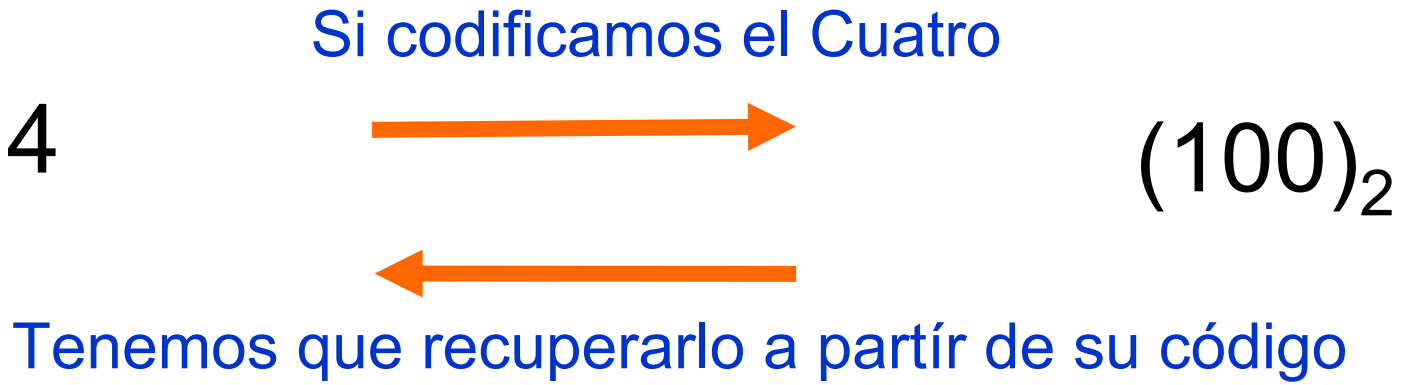
- El valor Numérico es representado por una serie dígitos
  - El número de dígitos usados es fijado por la base
  - Los dígitos son multiplicado por una potencia de la base
  - El orden de los dígitos determina la potencia de la base
- Números muy grandes pueden ser representados al igual que los números fraccionales

El sistema de numeración binaria

- Los números decimales son números de base 10 - los dígitos permitidos son 0,1,2,…,9
- El Sist. Binario es justo como el sistema decimal excepto que:
  - Solamente los dígitos 0 y 1 son validos
  - La base es 2 en lugar de 10

Representación Binaria de la Información

- El mapeo de los símbolos a valores binarios es conocido como "codificacion"
- El Mapeo debe ser único



El sistema de numeración binaria

- En un computador digital convencional–Los enteros son representados como números binarios de longitud fija **n**
- Una secuencia ordenada de dígitos binarios

$$(x_{n-1}, x_{n-2}, \cdots, x_1, x_0)$$

- Cada digito x (bit) es 0 o 1

El sistema de numeración binaria

- La secuencia de arriba representa el valor del entero **X**

$$X = x_{n-1}2^{n-1} + x_{n-2}2^{n-2} + \cdots + x_1 2 + x_0 = \sum_{i=0}^{n-1} x_i 2^i$$

- Letras mayúsculas representan los valores numéricos o secuencia de dígitos. Y las minúsculas, usualmente indexadas, representan dígitos individuales.

Base de sistema de numeración binaria

- El peso del dígito $\mathbf{x_i}$ es la potencia **i** de 2
- 2 es la base de sistema de numeración binaria
- Notación: la base es indicada en el subíndice como un número decimal
  - Ejemplo:
    - $(101)_{10}$ - Valor decimal 101
    - $(101)_2$ - Valor decimal 5

El sist. binario. Rango de representaciones

- En un PC los operandos y resultados son almacenados en registros de longitud fija. – Esto implica que un número finito de distintos valores solo pueden ser representado dentro de una ALU (Unidad Aritmético Lógica)
- Xmin ; Xmax – Representa los valores mas pequeño y el mas grande rtespectivamente
- [Xmin,Xmax] – Representa el rango de números representables

El sistema de numeración binaria

- Un resultado mas grande que el valor Xmax o mas pequeño que Xmin
  - Es incorrectamente representado
- La ALU debería indicar que el resultado generado es erróneo
  - Normalmente indica que se produce desbordamiento *overflow*

Ejemplo.

Overflow en un Sit. Binario

- Entero sin signo con 5 dígitos binarios (bits)
  - $X_{max} = (31)_{10}$ - representado por $(11111)_2$
  - $X_{min} = (0)_{10}$ - representado por $(00000)_2$
  - Incrementando X_{max} en 1 = $(32)_{10} = (100000)_2$
  - En la representación de 5-bit – Solamente los Los último cinco dígitos son retenidos - produciendo $(00000)_2 = (0)_{10}$

Ejemplo.
Overflow en Un Sit. Binario

- En general -
  - Un numero X fuera del rango [Xmin,Xmax]=[0,31] es representado por :
    - X mod 32
  - Si X+Y excede de X_{max} – el resultado es
    - S = (X+Y) mod 32

5

Ejemplo.
Overflow en Un Sit. Binario

- Ejemplo:

$$\begin{array}{rrrr} X & 10001 & 17 & \\ +\ Y & 10010 & 18 & \\ \hline 1 & 00011 & 3 & = 35 \bmod 32 \end{array}$$

  - El resultado tiene que ser almacenado en un registro de 5-bit – El bit mas significativo (con peso $2^5 = 32$) es descartado

Representación de Números Mixtos

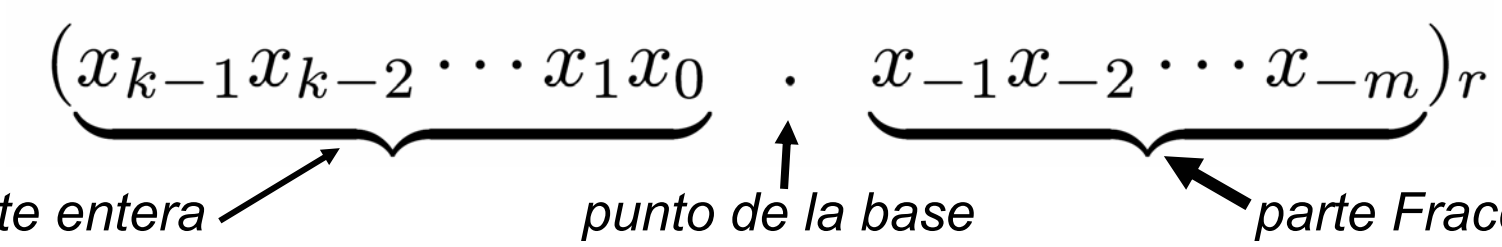
- Una secuencia de n dígitos en un registro – necesariamente no represnta a un entero

$$(x_{n-1}, x_{n-2}, \cdots, x_1, x_0)$$

- Puede estar representando un número mixto con una parte fraccional y una parte entera

Representación de Números Mixtos

- Los n dígitos son divididos en dos - k para la parte entera y m para la parte fraccional (k+m=n)



Representación de Números Mixtos

- El Valor de la secuencia de n dígitos con el punto de la base entre los k mas significativos y los m dígitos menos significativos es:

$$X = x_{k-1}r^{k-1} + x_{k-2}r^{k-2} + \cdots + x_1 r + x_0 + x_{-1}r^{-1} + \cdots + x_{-m}r^{-m}$$

$$= \sum_{i=-m}^{k-1} x_i r^i$$

Conversión de Base

- Es el paso de un número X representado en una base (Origen) a su representación en otra base (destino)
- Razón principal – la mayoría de las ALU operan sobre números binarios, mientras los usuarios estamos acostumbrado a los números decimales (requiere menos dígitos)

Conversión de Base

- Dado un numero X en base r, encontrar su representación en el sistema de numeración con base r_D
- Distinguiremos entre conversión de parte entera X_I y parte fraccional X_F

Conversión parte entera

- Se busca $(x_{k-1}x_{k-2}\cdots x_1x_0)_{r_D}$
-

$$X_I = \{[\cdots(x_{k-1}r_D + x_{k-2})r_D + \cdots + x_2]r_D + x_1\}r_D + x_0;$$

- Dividiendo X_I by r_D Quedaría un:
  - Resto – X_0
  - Cociente

$$\{[\cdots(x_{k-1}r_D + x_{k-2})r_D + \cdots + x_2]r_D + x_1$$

Conversión parte entera

- Dividiendo de nuevo el cociente entre r_D daría x_1 como nuevo resto
- Dividiendo el cociente resultante de la anterior división repetidamente por r_D hasta que se obtenga un cociente a cero
  – Los sucesivos restos de las diviciones son los dígitos requerídos

Conversión parte fraccionaria

- Buscamos $(x_{-1}x_{-2}\cdots x_{-m})_{r_D}$

$$X_F = r_D^{-1}\left\{x_{-1} + r_D^{-1}\left[x_{-2} + r_D^{-1}(x_{-3} + \cdots)\right]\right\}$$

- Multiplicando X_F por r_D obtenemos un número mixto donde
  - x_{-1} es la parte entera
  - y la parte fraccional es

$$r_D^{-1}\left[x_{-2} + r_D^{-1}(x_{-3} + \cdots)\right]$$

Conversión parte fraccionaria

- La parte fraccional es multiplicada varias veces por r_D Las sucesivas partes enteras representa los dígitos buscados
- El Algoritmo no garantiza un fin.
  - Una fracción puede ser finita en una base pero infinita en otra
  - En la practica el proceso puede terminar despues de m pasos

Conversión de base. Ejemplo

Convierte de decimal a binario el número mixto
$X=(46.375)_{10}$ donde $X_I=46$ y $X_F=0.375$

Parte entera en la nueva base?

- Dividimos repetidamente los cocientes resultantes y hasta que el cero

Cocientes	Restos
23	$0 = x_0$
11	$1 = x_1$
5	$1 = x_2$
2	$1 = x_3$
1	$0 = x_4$
0	$1 = x_5$

Conversión de base. Ejemplo

- Parte fraccional en la nueva base?
  - multiplicamos repetidamente las partes fraccionarias resultantes de cada producto y paramos hasta que tengamos m bit o unos pocos dígitos
  - Los dígitos buscados son las partes enteras resultantes

Parte entera	Par fraccionaria
$0 = x_{-1}$	.75
$1 = x_{-2}$	.5
$1 = x_{-3}$	.0

Conversión de base. Ejemplo

El Resultado Final es:

$$(46.375)_{10} = (101110.011)_2$$

Si la parte decimal fraccionaria fuese $X_F = 0.3$

- El Algoritmo nunca termina

$$\mathbf{(0.0100110011...)_2}$$

Todas las operaciones fueron realizadas sobre la base de origen

Conversión de base. Ejemplo

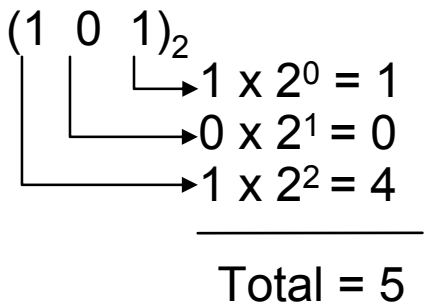
Para la conversión de binario a decimal

- Se puede realizar el algoritmo ahora como base de origen la binaria
- o, mas convenientemente usar la expresión

$$X = \sum_{i=-m}^{k-1} x_i r^i$$

El sistema de numeración binaria

- La conversión de Binario a decimal es justo la expresión explicita de los valores posicionales, tanto para enteros y fracciones
  - Ejemplo.



Representación de los números negativos

Como se representan los números negativos ?

Representación de los números negativos

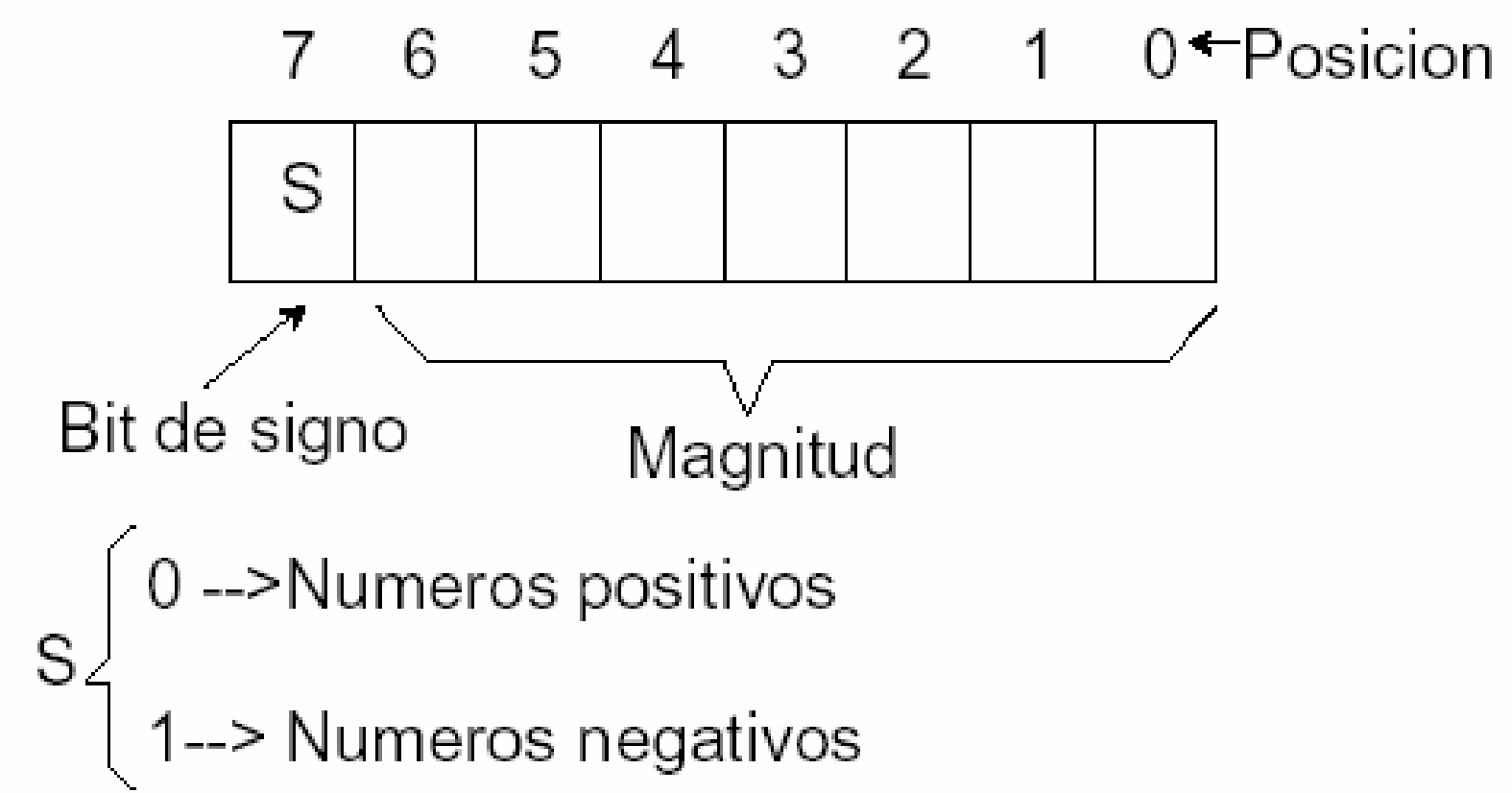
Un numero de sistema de base 2 tiene varias formas de representar un numero negativos:

- Representación magnitud y signo (S-M)
- Representación en complemento a uno (Ca1)
- Representación en complemento a dos (Ca2)

Representación Magnitud y Signo

Es la más "humana" de las representaciones de números con signo, puesto que, al conjunto de los bit que representa la magnitud del número se antepone (en la posición más significativa) un bit, denominado bit de signo, que toma el valor 0 para números positivos y el 1, para los negativos.

Representación Magnitud y Signo



Representación Magnitud y Signo

Ejemplo: usando 4 bits para la magnitud:

- +4 = 00100
- -4 = 10100

En general, se puede afirmar que si se utilizan **n** bits para representar un **número X con signo en notación S-M**, el rango de valores posibles para **X** está comprendido entre:

$$-(2^{n-1}-1) \leqslant x \leqslant 2^{n-1}-1$$

Representación Magnitud y Signo

En el caso **n**= 4 bit Tanto para magnitud y signo tenemos que el rango de los números posibles va desde -7 hasta +7

Código	S-M	Código	S-M
+0	0000	-0	1000
+1	0001	-1	1001
+2	0010	-2	1010
+3	0011	-3	1011
+4	0100	-4	1100
+5	0101	-5	1101
+6	0110	-6	1110
+7	0111	-7	1111

Representación en Complemento a 1

Los números positivos en notación Ca1 se expresan igual que en SM.

En cambio, los números negativos se obtienen a partir de aplicar el operador Ca1 al número expresado como si fuera positivo.

Ejemplo: usando 4 bits para la magnitud

+4 = 00100 en Ca1

-4 = Ca1(00100) = 11011

Representación en Complemento a 1

En general, se puede afirmar que si se utilizan **n** bits para representar un **número X con signo en notación Ca1,** el rango de valores posibles para **X** está comprendido entre:

$$-(2^{n-1}-1) \leqslant x \leqslant 2^{n-1}-1$$

Representación en Complemento a 1

En el caso **n**= 4 bit tenemos que el rango de los números posibles va desde -7 hasta +7

Código	Ca1	Código	Ca1
+0	0000	-0	1111
+1	0001	-1	1110
+2	0010	-2	1101
+3	0011	-3	1100
+4	0100	-4	1011
+5	0101	-5	1010
+6	0110	-6	1001
+7	0111	-7	1000

Representación en Complemento a 2

Los números positivos en notación Ca2 se expresan igual que en SM y en Ca1 En cambio, los números negativos se obtienen a partir de aplicar el operador Ca2 al número expresado como si fuera positivo.

Ejemplo: usando 4 bits para la magnitud

+4 = 00100 en Ca2

-4 = Ca2(00100) = 11100

Representación en Complemento a 2

En general, se puede afirmar que si se utilizan **n** bits para representar un **número X con signo en notación Ca1,** el rango de valores posibles para **X** está comprendido entre:

$$-2^{n-1} \leqslant x \leqslant 2^{n-1} - 1$$

Representación en Complemento a 2

En el caso **n**= 4 bit tenemos que el rango de los números posibles va desde -8 hasta +7

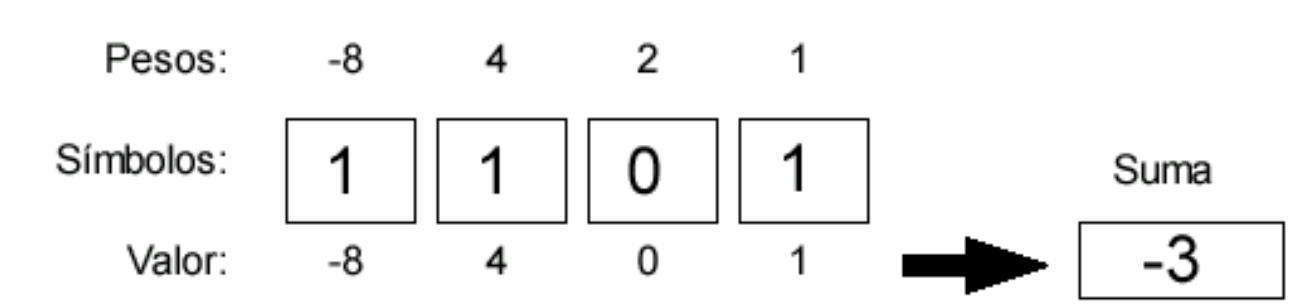
Código	Ca2	Código	Ca2
+0	0000	-1	1111
+1	0001	-2	1110
+2	0010	-3	1101
+3	0011	-4	1100
+4	0100	-5	1011
+5	0101	-6	1010
+6	0110	-7	1001
+7	0111	-8	1000

Representación en Complemento a 2

- Para n bits tenemos el valor de un número representado en Ca2 viene dado por:

$$x = -x_{n-1} \times b^{n-1} + \ldots + x_1 \times b^1 + x_0 \times b^0$$

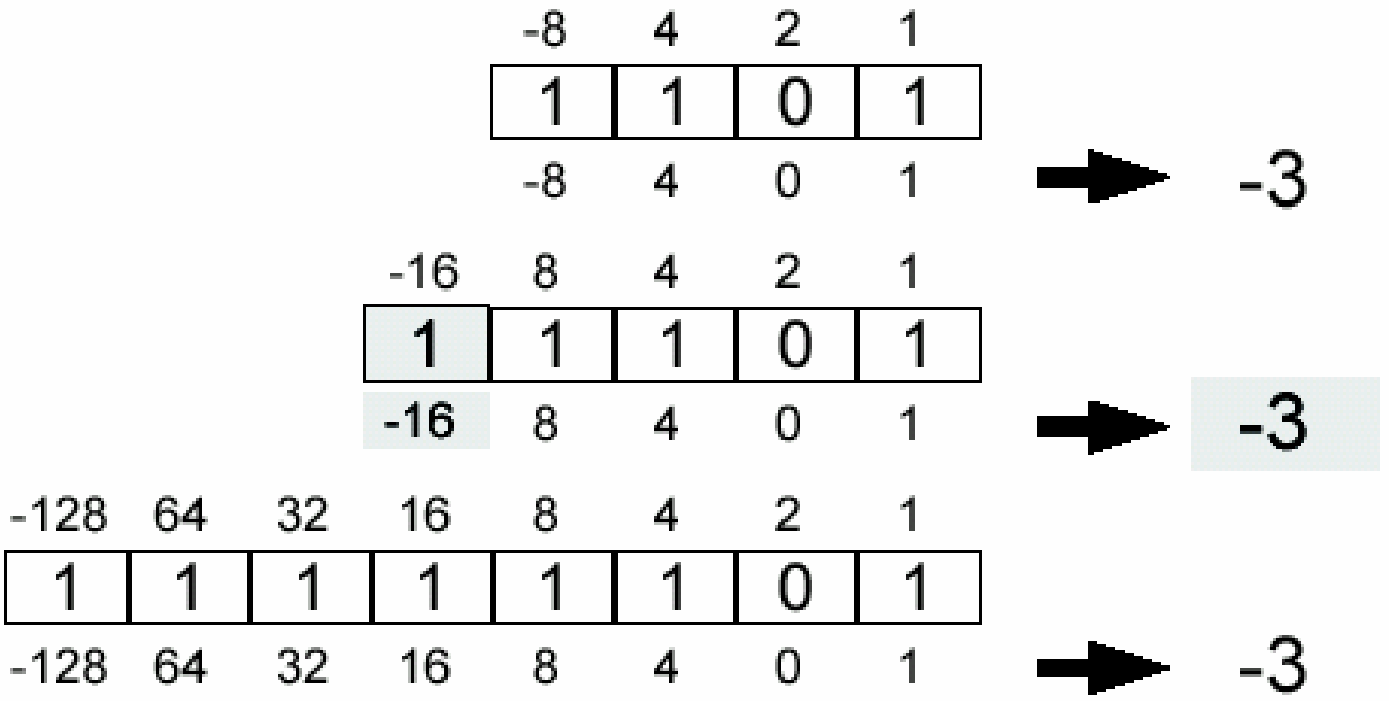
- Para 4 bits



Representación en Complemento a 2

Extensión del signo:

- Al extender el número de bits de un número codificado en Ca2, los bits extendidos toman todos el mismo valor que el antiguo bit de signo.



Repr. en Complemento a 2 propiedades

Si en la representación en Ca2 de una cantidad entera x se complementan todos los bits y, tratando el resultado como un número binario sin signo, se le suma 1, el resultado es la representación en Ca2 de -x

Si las representaciones en Ca2 de dos cantidades enteras x e y se suman, tratándolas como enteros binarios sin signo y despreciando el posible acarreo, el resultado es la representación en Ca2 de la cantidad x+y, salvo que se produzca desbordamiento.

Repr. en Complemento a 2 propiedades

(Regla de desbordamiento): Si dos cantidades binarias representadas en Ca2, ambas con el mismo signo, se suman tratándolas como enteros binarios sin signo, se produce desbordamiento si el signo del resultado, interpretado en Ca2 es distinto al signo de las cantidades sumadas.

Representación en Complemento a 2

- Ejemplos

```
 1001 = -7        1100 = -4        0011 = +3
 0101 = +5        0100 = +4        0100 = +4
-----            -----            -----
 1110 = -2       10000 =  0        0111 = +7


 1100 = -4        0101 = +5        1001 = -7
 1111 = -1        0100 = +4        1010 = -6
-----            -----            -----
11011 = -5        1001 = -7       10011 = +3
```

¡Desbordamiento!

Repre. de los números negativos resumen

Todas ellas requieren de un bit de signo situado en la posición más significativa, y con idéntico significado: un 0 para los números positivos, y un 1 para los negativos.

Los números positivos se representan de forma idéntica en las tres notaciones, sólo cambia para los negativos.

La notación SM y Ca1 tienen dos codificaciones distintas para un mismo número (+0 y -0), situación esta que no ocurre en Ca2.

Representación en Ca 2 resumen

x	s-m	Ca1	Ca2
-8	-	-	1000
-7	1111	1000	1001
-6	1110	1001	1010
-5	1101	1010	1011
-4	1100	1011	1100
-3	1011	1100	1101
-2	1010	1101	1110
-1	1001	1110	1111
0	0000/1000	0000/1111	0000
1	0001	0001	0001
2	0010	0010	0010
3	0011	0011	0011
4	0100	0100	0100
5	0101	0101	0101
6	0110	0110	0110
7	0111	0111	0111