Representación de Datos Numéricos

Elementos de Informática

Representación de Datos Numéricos

Representar números no es una operación sencilla y es preciso diferenciar varios casos. Los dispositivos de las computadoras en los que se efectúan las operaciones aritméticas, sólo pueden procesar números que les sean presentados según un formato bien definido.

Representación de Datos Numéricos

Nos centraremos en dos tipos de formatos:

- ✓ los enteros de punto fijo
- ✓ los números de punto flotante.

Formato de Punto Fijo

La primer componente de un formato es su <u>longitud</u>, la segunda es el <u>convenio escogido en la representación o codificación del número</u>.

La representación en formato fijo es la manera más natural de escribir un número en una palabra de memoria.

Para representar enteros grandes, como veremos, necesitamos varios bytes.

Formato de Punto Fijo

En un sistema de representación de números naturales con una base y cantidad de dígitos **(b, d)**,la cantidad de números representables en un sistema es siempre $\mathbf{b^d}$.

Ejemplo: Sistema de representación (3, 2)

Base=3, Cantidad de dígitos=2

00	01	02
10	11	12
20	21	22

Cantidad de números representables= $3^2 = 9$

Formato de Punto Fijo

Estos sistemas de representación de enteros están basados en los sistemas de números naturales
(SN, base, dig, *técnica*).

La técnica me dice a qué valor entero corresponde cada valor natural, con el cual se representará y almacenará en la computadora.

Formato de Punto Fijo

En la electrónica no es posible representar directamente los signos “+” y “-”, sino que se deben utilizar ciertas convenciones.

Existen varias convenciones para simbolizar números con bit de signo. En todas ellas para un formato de trabajo de n bits (en general: 8, 16, 32, 64), se tiene que:

Formato de Punto Fijo

El bit reservado para el signo es 0 si el número es positivo.

El bit reservado para el signo es 1 si el número es negativo.

Forma Binaria Directa

Es la representación decimal del número en el sistema binario.

Con 8 bits pueden representarse directamente los números comprendidos entre 00000000 y 11111111, es decir entre 0 y 255.

Se plantean dos dificultades: primera, que sólo representamos números positivos; segunda, que la magnitud de esos números queda limitada a 255, si trabajamos con sólo ocho bits.

Signo Valor Absoluto o Binario con Signo (SVA)

En un número representado en notación binaria con signo, según dijimos, éste viene indicado por el bit de la izquierda llamado **Bit Más Significativo = BMS**, tradicionalmente "0" si es positivo, y "1", si es negativo;
El rango que se puede representar con 8 bits, va desde:

1111 1111 = -127 0111 1111 = 127

Signo Valor Absoluto o Binario con Signo (SVA)

De esta forma ya se pueden representar números negativos, pero a costa de reducir el rango de enteros.
El Rango que se puede representar es:

$R=[-b^d + 1, b^d-1]$
Ej: si b=2 y uso 4 bits el rango es $R\ [\ -2^3 + 1, 2^3-1]$

Donde **b** es la base y **d** es la cantidad de cifras que soporta. En 16 bits se representa como se indica en la imagen:

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
+/-	Valor Absoluto														

Signo Valor Absoluto o Binario con Signo (SVA)

Considerando el problema de la magnitud, para representar números más grandes, se debe utilizar un número mayor de bits, a la que denominamos "palabra de memoria".

Palabra de memoria se considera a la longitud de la cadena de bits direccionada como un sólo bloque en la memoria de la computadora (8, 16, 32, 64 bits).

Signo Valor Absoluto o Binario con Signo (SVA)

Si utilizamos palabras de 16 bits, b=2 y d=15 el rango disponible es R= [-32767, 32767]

El bit más significativo representa el signo y los 15 restantes (los comprendidos entre el 0 y el 14) expresan hasta la magnitud 2^{15} = 32767

1111 1111 1111 1111 = -32767
0111 1111 1111 1111 = 32767

Ejemplo

Dada la codificación SVA con base 2 y 4 bits codifica los números posibles.

Decimal	SVA				Decimal	SVA			
0	0	0	0	0	0	1	0	0	0
1	0	0	0	1	-1	1	0	0	1
2	0	0	1	0	-2	1	0	1	0
3	0	0	1	1	-3	1	0	1	1
4	0	1	0	0	-4	1	1	0	0
5	0	1	0	1	-5	1	1	0	1
6	0	1	1	0	-6	1	1	1	0
7	0	1	1	1	-7	1	1	1	1

Signo Valor Absoluto o Binario con Signo (SVA)

Ejemplos

$$(+17)_{(10)} = \underbrace{0}_{signo\ +} \quad \underbrace{00\ 0000\ 0001\ 0001}_{17\ en\ binario}$$

$$(-17)_{(10)} = \underbrace{1}_{signo\ -} \quad \underbrace{00\ 0000\ 0001\ 0001}_{17\ en\ binario}$$

$$(+127)_{(10)} = \underbrace{0}_{signo\ +} \quad 00\ 0000\ 0111\ 1111$$

$$(-129)_{(10)} = \underbrace{1}_{signo\ -} \quad 00\ 0000\ 1000\ 0001$$

$$(+256)_{(10)} = \underbrace{0}_{signo\ +} \quad 00\ 0001\ 0000\ 0000$$

Ejercicio

Representa los siguientes números utilizando codificación SVA utilizando un byte

9	0	0	0	0	1	0	0	1
-9	1	0	0	0	1	0	0	1
28	0	0	0	1	1	1	0	0
-28	1	0	0	1	1	1	0	0

Signo Valor Absoluto o Binario con Signo (SVA)

Esta representación impone un tratamiento especial del signo y por el mismo hecho, circuitos diferenciales para el procesamiento de sumas y restas, inconveniente que desaparece si se representan los números bajo la forma complementada.

Signo Valor Absoluto o Binario con Signo (SVA)

¿Qué podemos observar en esta representación, en la que una de las representaciones no se condice con la representación numérica utilizada habitualmente?

Complemento a uno

Este sistema de representación es más complejo que SVA.

Divide el espacio entre los números positivos y los negativos.

Para obtener la representación del entero “e” se debe utilizar la siguiente ecuación:

$$r_{c1} = \begin{cases} e & 0 \leq e < \frac{b^d}{2} \\ b^d - 1 + e & -\frac{b^d}{2} < e \leq 0 \end{cases}$$

Complemento a uno

El rango que se puede representar utilizando Complemento a uno es:

$$\left[-\frac{b^d}{2}+1, \qquad \frac{b^d}{2}-1\right]$$

Con un total de b^d representaciones.

Ejemplo

$$r_{c1} = \begin{cases} e & 0 \leq e < \frac{b^d}{2} \\ b^d - 1 + e & -\frac{b^d}{2} < e \leq 0 \end{cases}$$

Dada la codificación Complemento a uno con base 2 y 4 dígitos codifica los números posibles.

Decimal	BINARIO				Decimal	BINARIO			
0	0	0	0	0	0	1	1	1	1
1	0	0	0	1	-1	1	1	1	0
2	0	0	1	0	-2	1	1	0	1
3	0	0	1	1	-3	1	1	0	0
4	0	1	0	0	-4	1	0	1	1
5	0	1	0	1	-5	1	0	1	0
6	0	1	1	0	-6	1	0	0	1
7	0	1	1	1	-7	1	0	0	0

$2^4 = 16$

$16/2 = 8$

$-16/2 = -8$

$e < 8$

$e > -8$

Complemento a uno

Ejemplos

$$
\begin{aligned}
(+3)_{(10)} &= 0000\ 0000\ 0000\ 0011 \\
(-3)_{(10)} &= 1111\ 1111\ 1111\ 1100 \\
(+127)_{(10)} &= 0000\ 0000\ 0111\ 1111 \\
(-127)_{(10)} &= 1111\ 1111\ 1000\ 0000
\end{aligned}
$$

Complemento a la base

Este sistema de representación es similar a Complemento a uno.

La diferencia más importante radica en evitar la doble representación del cero, contando así con una representación adicional.

Para obtener la representación del entero "e" se debe utilizar la siguiente ecuación:

$$r_{cb} = \begin{cases} e & 0 \leq e < \frac{b^d}{2} \\ b^d + e & -\frac{b^d}{2} < e < 0 \end{cases}$$

Complemento a la base

El rango que se puede representar utilizando Complemento a la base es:

$$\left[-\frac{b^d}{2}, \quad \frac{b^d}{2}-1\right]$$

Con un total de b^d representaciones con una única representación del 0 y una representación más para los números negativos.

Complemento a la base

$$r_{cb} = \begin{cases} e & 0 \le e < \frac{b^d}{2} \\ b^d + e & -\frac{b^d}{2} < e < 0 \end{cases}$$

Ejemplos

10

0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0

La Base elevada a la 16 es 2^{16}= 65536 = 1 0000 0000 0000 0000
La resta 10000000000000000
– 0000000000001010, es la que permite obtener el complemento a la base.
Si obtenemos el complemento a la base – 1 y luego sumamos 1 obtenemos el mismo resultado

-10

1	1	1	1	1	1	1	1	1	1	1	1	0	1	1	0

Complemento a la base

Ejemplos

125	0	0	0	0	0	0	0	0	0	1	1	1	1	1	0	1
-125	1	1	1	1	1	1	1	1	1	0	0	0	0	0	1	1

Cero desplazado o en exceso

Este sistema de representación también divide el espacio de representación entre positivos y negativos.

A diferencia de los sistemas vistos, los números negativos están primero y los positivos después.

Para calcular la representación en cero desplazado r_{cd} a partir de e, se utiliza la formula:

$$r_{cd} = f + e$$

Donde f representa una constante conocida como frontera siendo el valor que representa al cero

Cero desplazado

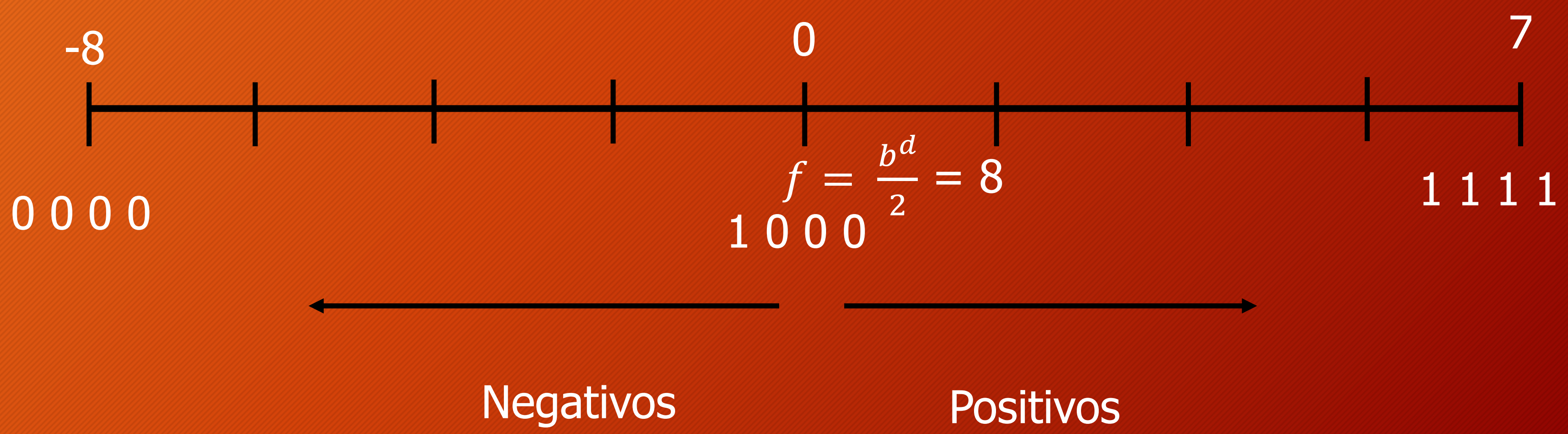
Los valores menores a f representan los números negativos y los mayores a f representan los números positivos.

Si $f = \frac{b^d}{2}$ se denomina frontera equilibrada.

Un sistema de frontera equilibrada representa la mitad de los números negativos y la mitad los números positivos.

Sobre la Recta numérica…

Si pensamos en un recta numérica podemos pensar que lo que ocurre en la codificación Cero Desplazado con 4 bits sería lo siguiente



Cero desplazado

El rango que se puede representar utilizando la representación Cero Desplazado es:

$$[-f\ ,\ b^{d} - f\ - 1]$$

Con un total de b^d representaciones.

Representación de enteros de 4 bits

Decimal	Entero sin signo	Signo y Magnitud	Complemento a uno	Complemento a dos	En exceso a 7
+8	1000	n/d	n/d	n/d	n/d
+7	0111	0111	0111	0111	1111
+6	0110	0110	0110	0110	1110
+5	0101	0101	0101	0101	1101
+4	0100	0100	0100	0100	1100
+3	0011	0011	0011	0011	1011
+2	0010	0010	0010	0010	1010
+1	0001	0001	0001	0001	1001
+0	0000	0000	0000	0000	1000
-0	n/d	1000	1111	n/d	n/d
-1	n/d	1001	1110	1111	0111
-2	n/d	1010	1101	1110	0110
-3	n/d	1011	1100	1101	0101
-4	n/d	1100	1011	1100	0100
-5	n/d	1101	1010	1011	0011
-6	n/d	1110	1001	1010	0010
-7	n/d	1111	1000	1001	0001
-8	n/d	n/d	n/d	1000	0000

Ejercicio

Representa el número -6 en las representaciones vistas en clase

Partimos de 6 positivo 0 0 0 0 0 1 1 0

Codificación	Número	2^7 128	2^6 64	2^5 32	2^4 16	2^3 8	2^2 4	2^1 2	2^0 1
SVA	-6	1	0	0	0	0	1	1	0
CB-1		1	1	1	1	1	0	0	1
CB		1	1	1	1	1	0	1	0
CD		0	1	1	1	1	0	1	0