## REPRESENTACIÓN DE PUNTO FLOTANTE

#### La unidad Fundamental

Normalización de un numero: un número "X" está escrito en su forma normalizada si "X" está escrito en la siguiente forma:  $x = \delta(u.d_1d_2d_3...d_p)_b.b^e$ 

Donde  $\delta$  es el signo (+1) si es positivo; (-1) si es negativo, para  $0 \le d_i \le b-1$  siendo  $d_i \ne 0$  y e es la característica. Por su puesto lo ubicado dentro del paréntesis es la mantisa.

La mantisa denota:  $\frac{d_1}{b_1} \dots + \frac{d_p}{b_p}$  , es decir su parte fraccionaria.

El entero  $\mathcal P$  denota la cantidad de dígitos de precisión y e el entero tal que esta:  $\mathcal L \neq e \neq \mathcal U$  siendo  $\mathcal L$  el exponente más pequeño que pueda tomar y  $\mathcal U$  el más grande.

Signo = 1 bit	Exponencial = 7 bits	Mantisa = 24 bits
0.90		

Los números en punto flotante llamados también números reales, tienen decimales intercalados.

Tales números almacenan y procesan en sus formas exponenciales y binarios.

La localización se memoria se divide comúnmente en 3 campos o bloques de bits.

Un campo, el primer bit, se reserva para el primer número. Un segundo campo para el exponente del número y el ultimo campo para la mantisa.

## Ejemplo:

Representa el número -0,00523 en una palabra de 16 bits siendo b=10.

Solución: primero hay que normalizar el numero

$$-0.00523 = (-1)(0.523)_{10}10^2$$

1	1	2	0	0	0	0	0	0	0	0	0	0	5	2	3

Se pone "1" porque el  $\delta$  es negativo

Si fuese positivo se coloca "0"

#### REPRESENTACION ENTERA

Los enteros o números en punto fijo son números que no tienen punto decimal. Un entero J se representa en la memoria de un computador por medio de su forma binaria si son positivos y por medio de su complemento a 2 de su valor absoluto.

\*Ejemplo:\*

$$4 \ 2 \ 3 = 1 \ 1 \ 0 \ 1 \ 0 \ 1 \ 1 \ 1$$

El computador almacena -423 en una localización de memoria, tomando el complemento a 1 de la anterior representación 423 y sumándole un 1 el computador puede saber si un entero  $\boldsymbol{J}$  en la memoria es positivo o negativo minando el primer digito.

Si 
$$J = 0 \rightarrow +$$

Si 
$$I = 1 \rightarrow -$$

#### **COMPLEMENTO BINARIO**

Sea **A** un número binario el complemento a unos de **A** se obtiene restando **1** a cada digito de **A** y el complemento a dos de **A** es su complemento a **1** más **1**.

$$A = 111100001111_{2}$$

$$A_{1} = 000011110000_{2} + 1$$

$$A_{2} = 000011110001$$

Sea

$$A = -414.8125$$

### Binario

$$A = -110100011.1101_2$$

 $A = -0.1101000111101 * 2^9$  normalización de un número

 $64 + 9 = 73 = 1001001_2$  exponente

# 

... dentro de ese rango se colocan los ceros hasta que se completan los 64 bits