Representación de datos en punto flotante

Arquitectura del Computador LCC-FCEIA

Septiembre de 2020

Introducción

- Para representar números reales es necesario utilizar una coma o punto para separar la parte entera de la parte fraccionaria, independientemente del sistema de representación con que se trabaje (decimal, binario, etc.).
- Existen dos formas de resolver el problema:
  - Se considera la coma o punto en cierta posición fija.
  - Se almacena la posición que ocupa la coma o punto en el número (posición flotante).
- Con punto fijo existe una limitación en cuanto al rango de los números que se pueden representar

Representación de números reales con punto fijo

Ejemplo:
Trabajando con 8 bits de los cuales hemos fijado y reservado 5 para la parte entera y 3 para la fraccionaria:

$$\begin{aligned}(11011.011)_2 &= 1\times 2^4 + 1\times 2^3 + 1\times 2^1 + 1\times 2^0 + 1\times 2^{-2} + 1\times 2^{-3}\\ &= (27.375)_{10}\end{aligned}$$

En este ejemplo se puede observar que existe una limitación en cuanto al rango de los números que se pueden representar.

- Menor número representable: $(00000.001)_2 = 2^{-3} = (0.125)_{10}$
- Mayor número representable: $(11111.111)_2 = 2^5 - 2^{-3} = (31.875)_{10}$

Ventajas: La aritmética de punto fijo es relativamente simple.

Desventajas: El rango de representación, es decir el número de cantidades a representar, es muy limitado.

Representación de números reales con punto flotante

- En números decimales el rango puede ampliarse utilizando la notación científica que permite representar números muy grandes y muy pequeños utilizando relativamente pocos dígitos.

- Todo número real no nulo se puede escribir en forma única en la **notación científica normalizada** como:

$$(-1)^s 0.a_1 a_2 a_3 \ldots a_t \ldots \times 10^e$$

  siendo el dígito $a_1 \neq 0$.

- Ejemplo: 0.976×10^{-15} es un número en notación científica normalizada.

Notación científica normalizada

Definición general

De forma general, todo número real no nulo puede representarse en forma única respecto a la base β de la siguiente forma:

$$N = (-1)^s 0.a_1 a_2 a_3 \ldots a_t \ldots \times \beta^e,$$

donde los "dígitos" a_i son enteros positivos tales que $1 \leq a_1 \leq \beta - 1$, $0 \leq a_i \leq \beta - 1$ para $i = 2, 3, \ldots$ y constituyen la parte fraccional o **mantisa** del número, en tanto que e es el **exponente**, el cual indica la posición del punto correspondiente a la base β.

Standard IEEE 754 para números en punto flotante

El Instituto de Ingenieros Eléctricos y Electrónicos (IEEE) creó un comité para formular una norma en 1985 (Standard IEEE 754) que define 3 formatos estándar para números en punto flotante:

1. Precisión simple con 32 bits (sesgo 127):

1	8	23
S	Exponente	Fracción
b_{31}	b_{30} … b_{23}	b_{22} … b_0

2. Precisión doble con 64 bits (sesgo 1023):

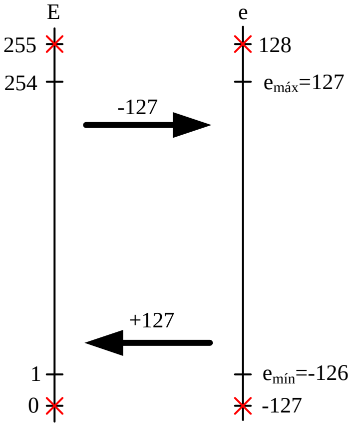
1	11	52
S	Exponente	Fracción
b_{63}	b_{62} … b_{52}	b_{51} … b_0

3. Precisión cuádruple con 128 bits (sesgo 16383):

1	15	112
S	Exponente	Fracción
b_{127}	b_{126} … b_{112}	b_{111} … b_0

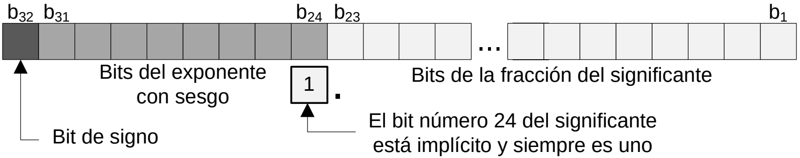
Exponente sesgado

- Los valores mínimos (0) y máximo del exponente (255, 2047 y 32767, según el formato) no se utilizan para número normalizados sino que tienen usos especiales.
- Se utiliza una representación sesgada para el exponente: $E = e + sesgo$



Significante

El significante se define como $1.f$, donde f es la mantisa:



Por lo tanto, el valor de un número en formato IEEE 754 puede ser computado como

$$(-1)^s \times 2^e \times 1.f$$

donde:

$$s = \begin{cases} 0 & \text{si el número es positivo} \\ 1 & \text{si el número es negativo} \end{cases}$$

Notar que el significante es menor que dos y mayor o igual a uno.

Conversión de decimal a IEEE 754 simple precisión

Ver ejemplo en el apunte 2 (página 9).

Conversión de IEEE 754 simple precisión a decimal

Ver ejemplo en el apunte 2 (página 10).

Características principales

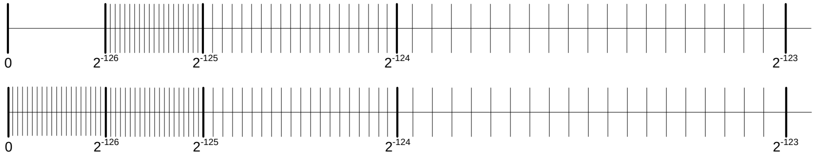
Ver Tabla en apunte 2 (página 12).

Tipos numéricos del standard IEEE 754

Signo	Exponente e	Fracción f	Representa	Denominación
$\pm$	$e = e_{min} - 1$ codificado como $E = (00\ldots0)_2$	$f = 0$	± 0	Ceros
$\pm$	$e = e_{min} - 1$ codificado como $E = (00\ldots0)_2$	$f \neq 0$	$\pm 0.f \times 2^{e_{min}}$	Núm. denormalizados
$\pm$	$e_{min} \leq e \leq e_{max}$	Cualquier patrón	$\pm 1.f \times 2^{e}$	Núm. normalizados
$\pm$	$e = e_{max} + 1$ codificado como $E = (11\ldots1)_2$	$f = 0$	$\pm\infty$	Infinitos
$\pm$	$e = e_{max} + 1$ codificado como $E = (11\ldots1)_2$	$f \neq 0$	NaN	*Not a Number*

Número denormalizados

- El menor número normalizado en simple precisión es 1.0×2^{-126}.
- Los números denormalizados del estándar tienen un exponente con todos los bit en cero (no permitido para los números normalizados) y una fracción donde al menos un bit no nulo.
- El uno implícito del significante es cero para los números denormalizados.
- El exponente es -126 o -1022, según corresponda.



Notar que en la representación de los números denormalizados el exponente es $e = e_{min}$ y no $e = e_{min} - 1$.

Precauciones al operar con números en punto flotante

Ejemplo, si queremos sumar $S = 123_{10} + 2.000001_{10}$ en IEEE 754 simple precisión:

$$
\begin{array}{rcll}
123_{10} & = & 0|10000101|11101100000000000000000 & (e = 6) \\
2.000001_{10} & = & 0|10000000|00000000000000000000100 & (e = 1)
\end{array}
$$

Desplazando el significante del número menor hasta igualar ambos exponentes (cinco veces hacia la derecha), el significante del número 2.000001_{10} resulta $0.000010 \cdots 0$:

$$
\begin{array}{cccccccccccc}
 & 1. & 1 & 1 & 1 & 0 & 1 & 1 & 0 & 0 & \cdots & 0 \\
+ & 0. & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & \cdots & 0 \\
\hline
 & 1. & 1 & 1 & 1 & 1 & 0 & 1 & 0 & 0 & \cdots & 0
\end{array}
$$

Juntando con el exponente resulta:

$$S = 0|10000101|11110100000000000000000 = 125_{10}$$

El resultado obtenido es incorrecto y el error cometido es 1×10^{-6}.