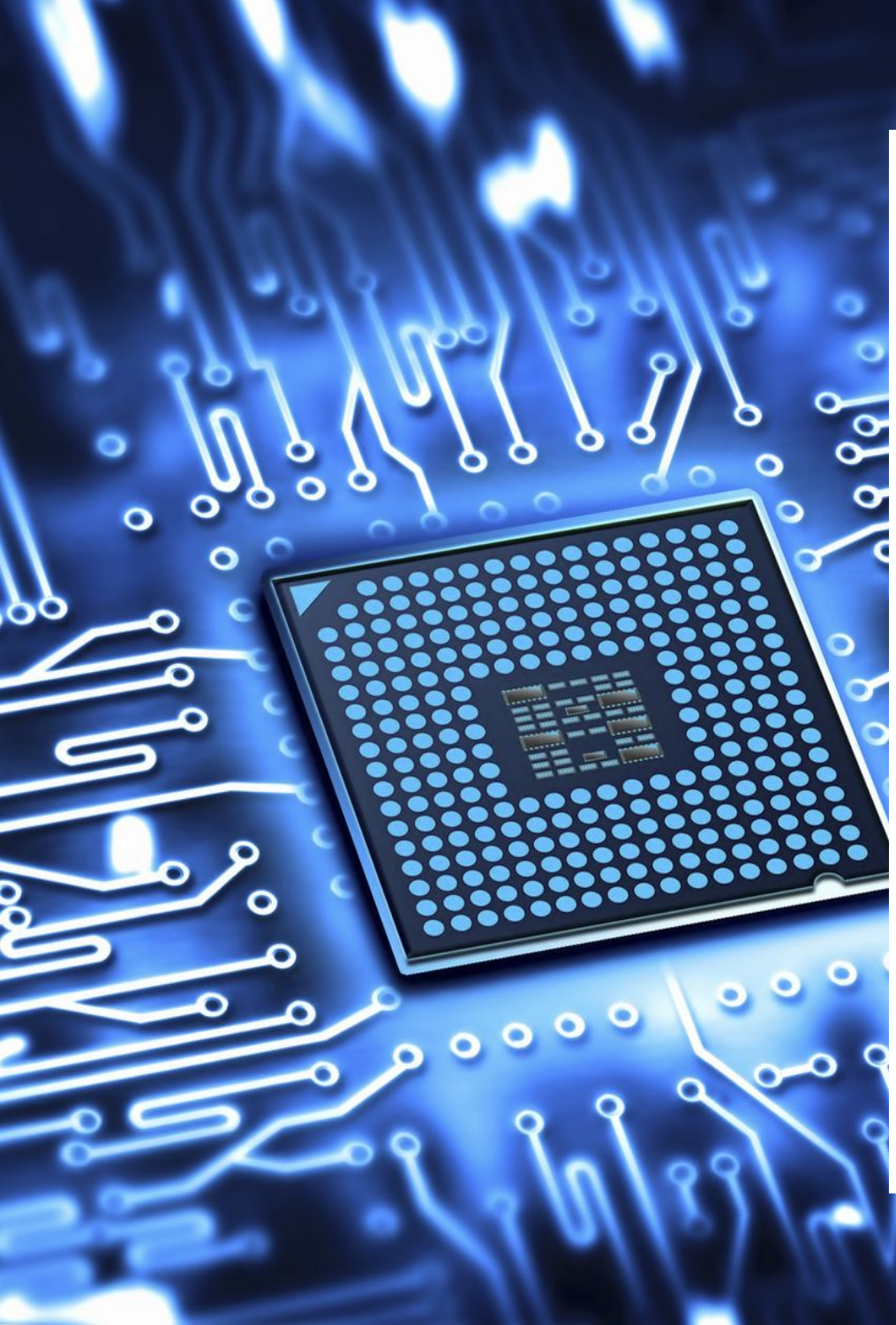
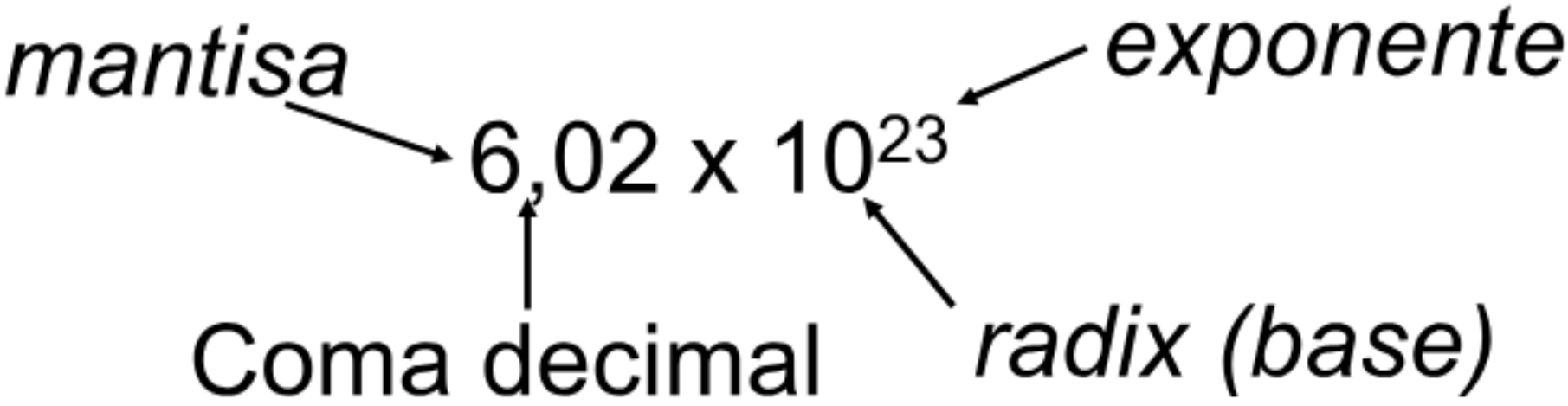


Arquitectura de computadoras

Aritmética de números de coma flotante

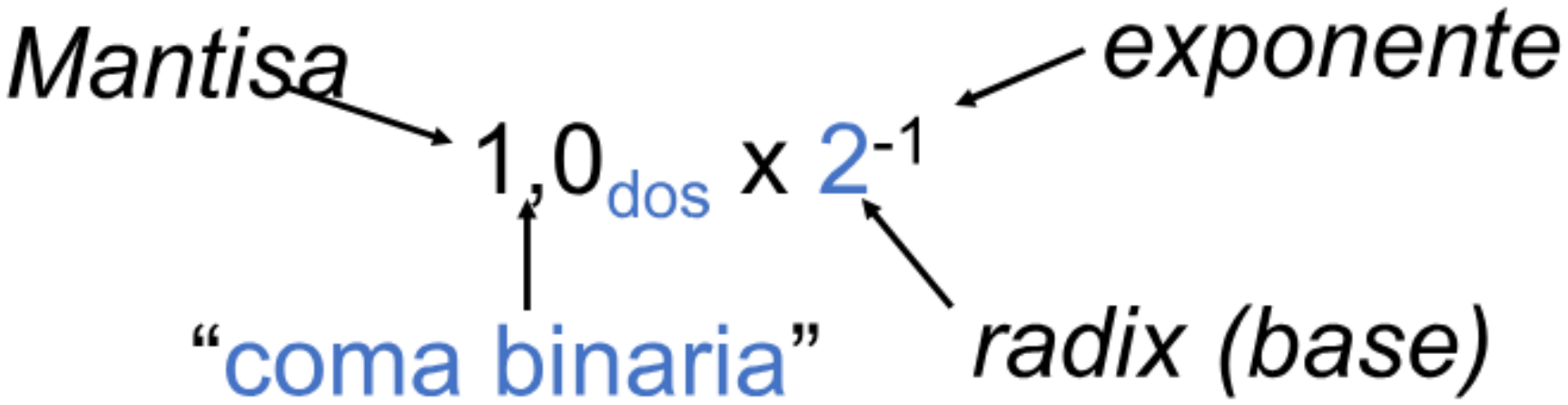
Universidad Nacional de la Patagonia

Revisión de notación científica



- Forma normalizada, sin ceros a la izquierda. Exactamente un dígito a la izquierda de la coma decimal.
- Alternativas de representación de normalizada de $1/1.000.000.000$.
  - Normalizada: $1,0 \times 10^{-9}$
  - No normalizada: $0,1 \times 10^{-8}$; $10,0 \times 10^{-10}$

Notación científica para números binarios



Representación en coma flotante

- Formato normal $+1, xxxxx_2 \times 2^{yyy_2}$
- Múltiplos del tamaño de una palabra (32 bits).

- S representa signo.
- Exponente representa yyyy.
- Mantisa representa xxxxx.
- Representa números tan chicos como $2,0 \times 10^{-38}$ hasta tan grandes como $2,0 \times 10^{38}$

Representación en coma flotante

- Si el numero es demasiado grande ($> 2,0 \times 10^{38}$).
  - Overflow!
  - El exponente a representar es mayor al máximo representable posible.
- Si el numero es demasiado pequeño ($< 2,0 \times 10^{-38}$).
  - Underflow!
  - El exponente a representar es menor al mínimo representable posible.
- Como reducir las chances de que ocurra overflow o underflow?

Representación en doble precisión

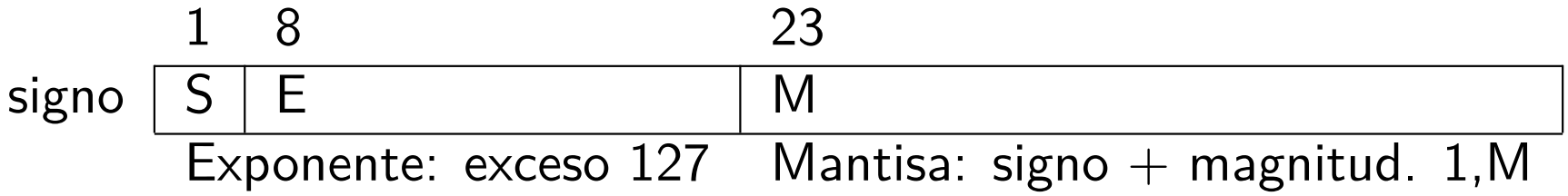
- Formato normal $+1, xxxxx_2 \times 2^{yyy_2}$
- Múltiplos del tamaño de una palabra (64 bits).

- Doble vs simple precisión.
  - Variables en C declaradas como double.
  - La principal ventaja es la mayor precisión debido a una mantisa mas grande.
  - Representa números tan chicos como $2,0 \times 10^{-308}$ hasta tan grandes como $2,0 \times 10^{308}$

IEEE 754 Estándar de coma flotante

- Bit de signo: 1 negativo, 0 positivo.
- Mantisa:
  - Para ganar un bit el primer 1 (izq.) esta implícito.
  - $1 + 23$ bits simple, $1 + 52$ bits doble.
- $(-1)^{signo} \times (1 + significando) \times 2^{exponente}$
- Diseñado para realizar comparaciones rápidas: primero se compara por signo, luego por exponente y luego por mantisa.
- Es deseable que el exponente de menor valor (negativo) se represente como 00000000 y el de mayor valor (positivo) se represente como 11111111
- En IEEE 754 se utiliza exceso 127 para simple precisión y 1023 para doble precisión.

Aritmética de coma flotante



Exponente real es $e = E - 127$

$$N = (-1)^S \times 2^{E-127} \times 1, M$$

Rango 2^{-126} a 2^{127} que es aproximadamente $1,8 \times 10^{-38}$ a $3,4 \times 10^{38}$

Convertir IEEE 754 a decimal

0	01101000	10101010100001101000010

- Signo: 0 (positivo)
- Exponente:
  - $01101000_2 = 104_{10}$
  - Ajuste exceso: $104 - 127 = -23$
- Significando:
  $1 + 1 \times 2^{-1} + 0 \times 2^{-2} + 1 \times 2^{-3} + 0 \times 2^{-4} + 1 \times 2^{-5} + \ldots$
  $= 1 + 2^{-1} + 2^{-3} + 2^{-5} + 2^{-7} + 2^{-9} + 2^{-14} + 2^{-15} + 2^{-17} + 2^{-22}$
  $1 + 0,666115$
- Representa: $1,666115 \times 2^{-23} \approx 1,986 \times 10^{-7} \approx \frac{2}{10.000.000}$

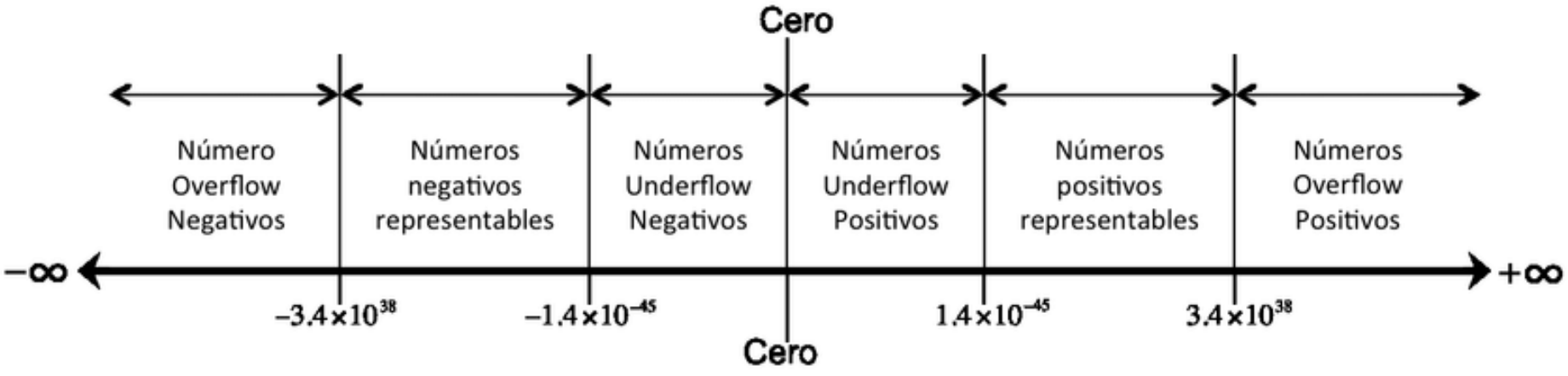
Características de los números IEEE754

Concepto	Simple precision	Doble precision
Bits para el signo	1	1
Bits para el exponente	8	11
Bits para la mantisa	23	52
Total de bits	32	64
Sistema del exponente	Exceso 127	Exceso 1023
Rango del exponente	-126 a 127	-1022 a 1023
Normalizado mas pequeño	2^{-126}	2^{-1022}
Normalizado mas grande	$\approx 2^{127}$	$\approx 2^{1023}$
Rango decimal	$\approx 10^{-38}$ a 10^{38}	$\approx 10^{-308}$ a 10^{308}
Desnormalizado mas pequeño	$\approx 10^{-45}$	$\approx 10^{-324}$

Tipos de numeros IEEE754

Normalizado	$\pm$	$0 < Exp < Max$	M
Desnormalizado	$\pm$	0	$M \neq 0$
Cero	$\pm$	0	0
Infinito	$\pm$	11111111	0
No numero	$\pm$	11111111	$M \neq 0$

Desnormalización



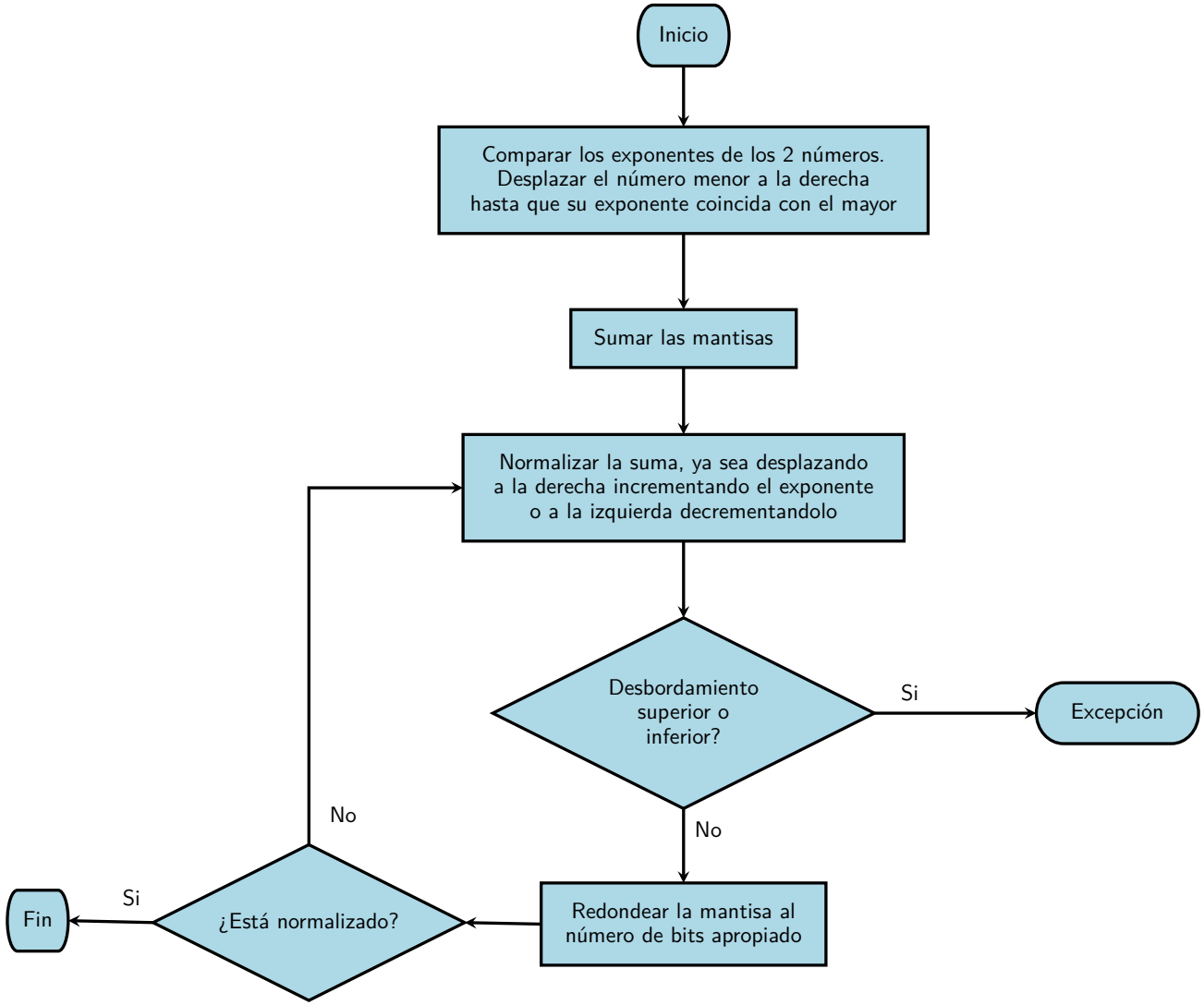
Redondeo

- Al mas cercano.
- Hacia $+\infty$
- Hacia $-\infty$
- Hacia 0

Suma de números decimal en notación científica

Ejemplo: $7\times10^3 + 4\times10^2$	
$7\times\underline{10^3} + 4\times10^2$	Dado que tenemos números con diferentes potencias de base 10, buscamos la potencia con mayor exponente.
$7\times10^3 + 4\times10^2$	Expresaremos ambos valores en función de 10^3, por ser la potencia de base 10 con mayor exponente
$7\times10^3 + 0{,}4\times10^3$ ($\div10^1$ ↓, $\times10^1$ ↓)	La potencia la multiplicamos por 10^1 para convertirla a 10^3, y la mantisa la dividimos entre 10^1.
$7\times\underline{10^3} + 0{,}4\times\underline{10^3}$	Ahora tenemos ambos valores en función de la misma potencia de base 10.
$(7 + 0{,}4) \times 10^3$	Dado que ambos números tienen la misma potencia de base 10, sumamos los números que se encuentran delante de las potencias.
$7{,}4 \times 10^3$	¡Y ya tenemos la respuesta en notación científica!

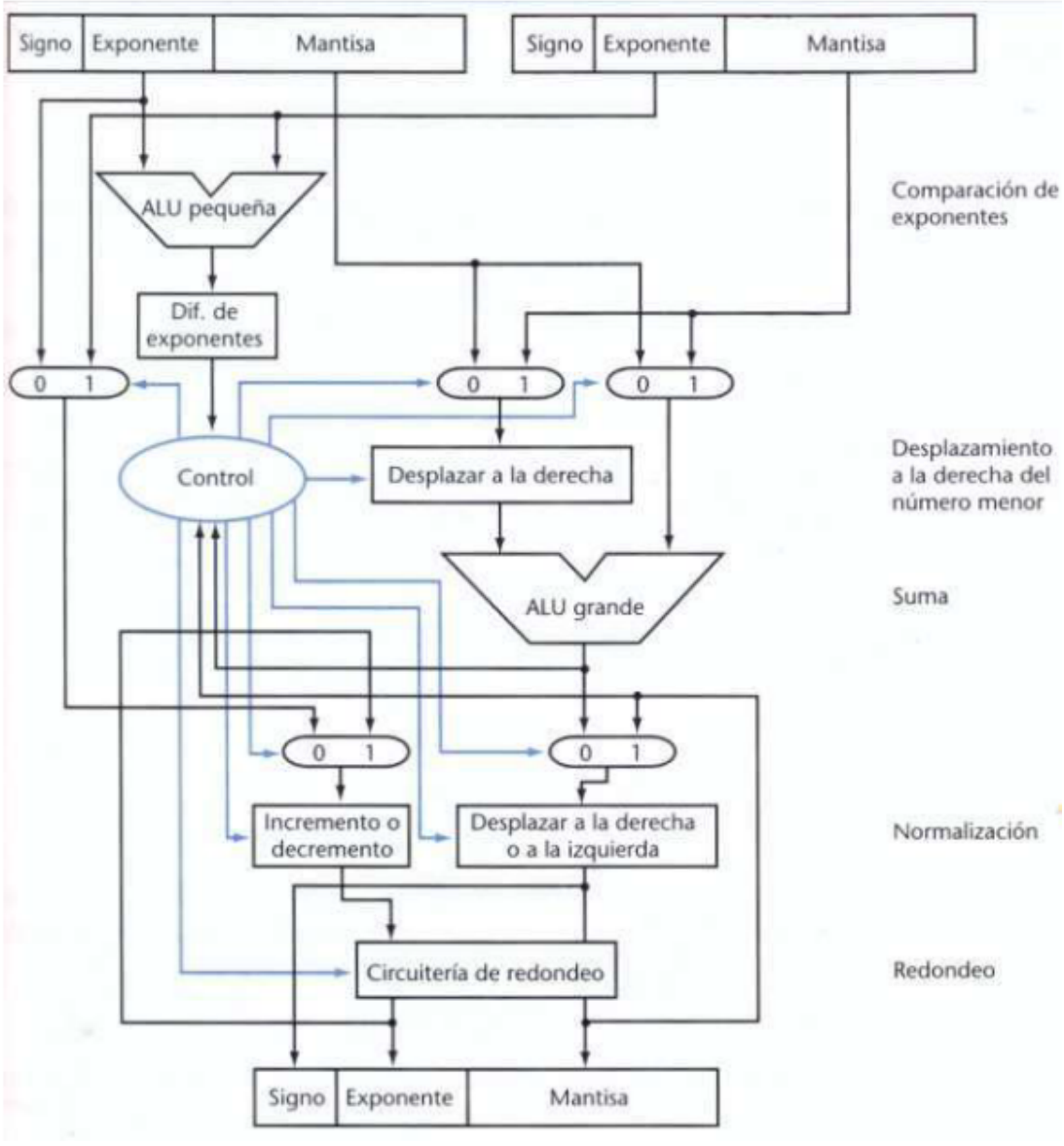
Suma de números flotantes



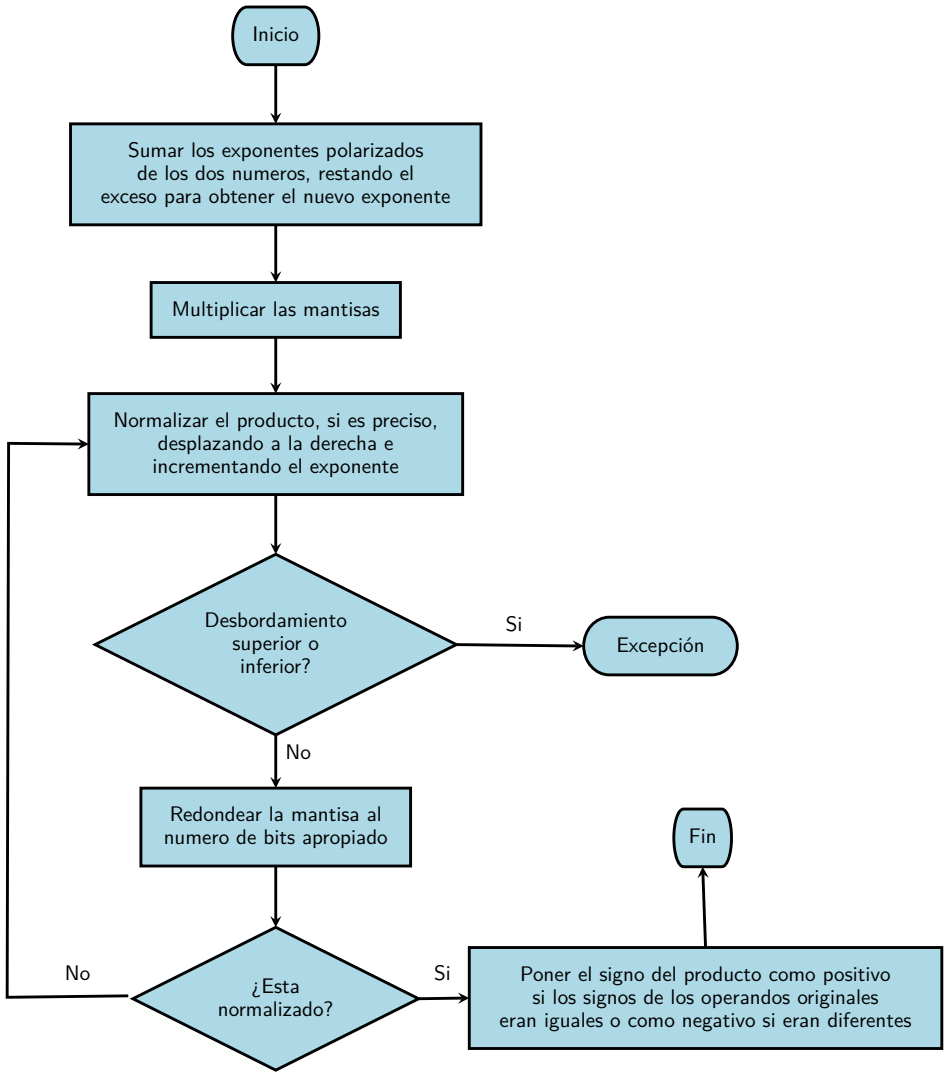
Ejemplo de suma de números IEEE754

- Sumamos 2 números flotantes IEEE754:
  - $1,11100100000000000000010 \times 2^4$
  - $1,10000100000000110000101 \times 2^2$
- Se debe normalizar el numero con el menor exponente para que tenga los dos numeros el mismo exponente.
  - $1,10000100000000110000101 \times 2^2 =$
    $0,01100001000000001100001 \times 2^4$
- Sumamos los valores:
  - $1,11100100000000000000010 \times 2^4$
  - $0,01100001000000001100001 \times 2^4$
  - $10,01000101000000001100011 \times 2^4$
- La suma produce un acarreo, el resultado no esta normalizado.
- Normalizamos
  - $10,01000101000000001100011 \times 2^4$
    $1,00100010100000000110001 \times 2^5$

UNIDAD ARITMÉTICA DE SUMA EN COMA FLOTANTE



Multiplicación de números flotantes



Ejemplo de multiplicación de números IEEE754

- Multiplicamos 2 números flotantes IEEE754:
  - $-1,11010000100000010100001 \times 2^{-4}$
  - $1,10000000001000000000000 \times 2^{-2}$
- A diferencia de la suma, se deben sumar los exponentes
  - $(-4) + (-2) = -6$
- Usando la representación en exceso: $E_z = E_x + E_y - Exceso$
  - $E_x = (-4) + 127 = 123$
  - $E_y = (-2) + 127 = 125$
  - $E_z = 123 + 125 - 127 = -6$
- El signo del producto se calcula aparte mediante un XOR.

Ejemplo de multiplicación de números IEEE754

- Multiplicamos las mantisas

```
          (Multiplicand)     1.11010000100000010100001
          (Multiplier)     × 1.10000000001000000000000
-------------------------------------------------------
                 11101000010000010100001
               11101000010000010100001
             1.11010000100000010100001
-------------------------------------------------------
  10.10111000111110111111001100101000010000000000000
```

- Se duplica la cantidad de bits
- *Multiplicando* $\times 0 = 0$ Se ignoran esas filas.
- *Multiplicando* $\times 1 =$ *Multiplicando* Desplazado a la izquierda

Ejemplo de multiplicación de números IEEE754

- Normalizamos el resultado
  $-10,10111000111110111111001100\cdots \times 2^{-6}$
  Se desplaza a la derecha y se incrementa el exponente
  $-1,01011100011111011111001100\cdots \times 2^{-5}$
- Se trunca y redondea la mantisa
  $-1,01011100011111011111101\cdots \times 2^{-5}$
- Resultado final: 1 01111010 01011100011111011111101

Consideraciones

- Los números representados son aproximaciones
- La cantidad de números reales existentes entre 1.0 y 2.0 es infinita pero en IEEE 754, esta cantidad es limitada
- Se requieren bits extra para la realización de los pasos intermedios
- Este formato minimiza el hardware requerido pero sacrifica exactitud
- Las operaciones son más complicadas pero las comparaciones son sencillas
- Ademas de tener underflow, también se tiene underflow