Documento: Personal

# Organización de Computadoras

CURSO 2024

TURNO RECURSANTES

CLASE 2

Documento: Personal

# Resumen de clase

- Representación de números en punto fijo
  - Rango y resolución
  - Error en punto fijo
  - Representación de números en punto flotante

# Números en punto fijo

- Es posible representar, en un sistema de cómputo, números en punto fijo (números racionales).
- En este caso, la coma fraccionaria está siempre ubicada en el mismo lugar.
- Por lo tanto, todos los números a representar tienen exactamente la misma cantidad de dígitos de parte entera y de parte fraccionaria. Por esa razón no se necesita almacenar la coma.
- La representación en el sistema de cómputo y el papel es similar. La diferencia principal entre ambas representaciones es que no se guarda la coma porque se supone que está en un lugar determinado.

Contenido: Personal

# Rango, resolución y error

Las representaciones numéricas tienen 3 propiedades importantes:

- Rango: diferencia entre el número mayor y el menor. Se obtiene determinando el número más chico y el más grande, admitidos por la representación.
- Resolución: diferencia ("distancia") entre dos números consecutivos. Se obtiene restando 2 valores consecutivos.
- Error: el error de una representación es la diferencia entre el valor real del número que se quiere representar y el valor usado para representarlo.

Documento: Personal

# Rango, resolución y error

Para las propiedades segunda y tercera, surge una pregunta importante:

1) la distancia entre 2 números consecutivos es siempre la misma? En otras palabras, es constante?

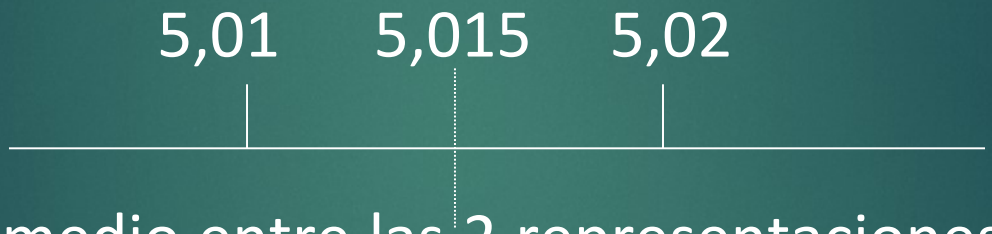
2) El error es el mismo en todo el rango de la representación?

# Error en punto fijo

- Dado que el error de una representación es la diferencia entre el valor real del número que se quiere representar y el valor usado para representarlo, el error es una función que depende del valor de la variable a representar y de la distancia entre los puntos discretos de la representación.
- El error es 0 (E = 0) si la variable tiene exactamente el valor discreto representado, pero crece a medida que nos alejamos de él.
- El error máximo cometido en una representación ocurre en el punto medio de 2 representaciones (puntos) consecutivas.
- El valor máximo del error es igual a la mitad de la diferencia (resolución) entre dos números consecutivos.

# Error en punto fijo

- Supongamos el siguiente ejemplo de números decimales:
  - Se tiene una representación en punto fijo de 1 dígito para la parte entera y 2 para la parte fraccionaria.
  - 2 números consecutivos son, por ejemplo, el 5,01 y el 5,02.

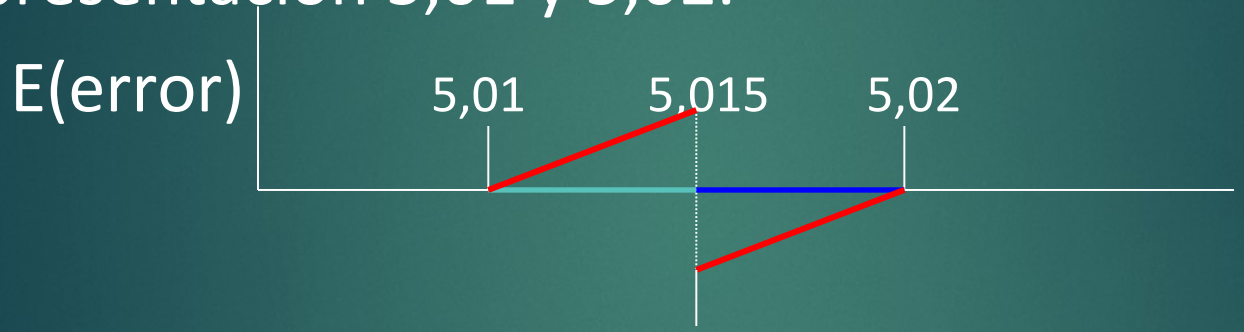



  - El punto medio entre las 2 representaciones consecutivas es 5,015.
  - Si un número es más chico que 5,015 se representará como 5,01. si N < 5,015 entonces N = 5,01
  - Si un número es más grande que 5,015 se representará como 5,02. si N > 5,015 entonces N = 5,02

Documento: Personal

# Error en punto fijo

- Si graficamos en el eje vertical el error E en función del valor de la variable a representar, el error crece linealmente a medida que nos alejamos de los puntos de la representación 5,01 y 5,02.




- El máximo error ocurre en 5,015.
- El valor máximo del error es :

máximo error positivo= 5,015 - 5,01

o bien:

máximo error negativo= 5,02 – 5,015

# Error en punto fijo

- En cualquiera de los dos casos el Error Absoluto máximo resulta ser:

$$EA\ max = 0{,}005$$

- Es decir:

$$EA\ max = 5{,}02 - 5{,}015 = 5{,}015 - 5{,}01 = \frac{5{,}02 - 5{,}01}{2} = 0{,}005$$

- Es decir, el valor máximo del error es igual a la mitad de la resolución. Por eso es tan importante saber cuál es la resolución de una representación numérica.
- Y el máximo error ocurre exactamente en el medio de 2 puntos consecutivos de la representación.

Contenido: Personal

# Conclusiones de la representación en punto fijo

- En punto fijo es posible representar un rango de números fraccionarios centrados en 0.
- La resolución es fija. Siempre hay la misma distancia entre 2 puntos consecutivos de la representación.
- El rango depende de la cantidad de dígitos asignados a la parte entera y a la fraccionaria.
- A continuación vamos a ver como representar números muy grandes o muy chicos.

Uso exclusivo: Personal

# Concepto de representación de números en punto flotante

## Ejemplo 1: representación de números muy grandes

Consideremos el número: 976.000.000.000.000

- Opción 1: en representación en punto fijo se necesitan 15 dígitos para poder escribirlo: 976.000.000.000.000
- Opción 2: también se puede escribir de la siguiente forma:

$$976.000.000.000.000 = 9{,}76 \times 10^{14} = 976 \times 10^{12}$$

Si se escribe usando la notación (compacta) anterior, se requieren:

- 3 dígitos para el número 976 (la "mantisa" M)
- 2 dígitos para el exponente 12 (el "exponente" E)
- 2 dígitos para la base 10 (la "base" B).

Es decir, solo se requieren 7 dígitos para escribir un número de 15 cifras. Esta notación es, entonces, más compacta.

# Números en punto flotante

- Si solo hay que almacenar los dígitos de M, B y E, tenemos la siguiente situación:
  - M: 3 dígitos
  - B: 2 dígitos
  - E: 2 dígitos
  - Total notación compacta: 3+2+2= 7 dígitos
  - El mismo número escrito en punto fijo era:

    976.000.000.000.000 que requiere,
  - Total notación punto fijo = 15 dígitos
  - Conclusión: con esta notación podemos representar números "muy grandes" con una menor cantidad de dígitos.

Documento: Personal

# Concepto de representación de números en punto flotante

## Ejemplo 2: representación de números muy chicos

Consideremos el número: 0,0000000000000976

- Opción 1: en representación en punto fijo se necesitan 17 dígitos para poder escribirlo: 0,0000000000000976
- Opción 2: también se puede escribir de la siguiente forma:

  $0{,}0000000000000976 = 9{,}76 \times 10^{-14} = 976 \times 10^{-16}$

Si se escribe usando la notación compacta última, se requieren:

- 3 dígitos para el número 976 (la "mantisa" M)
- 2 dígitos para el exponente -16 (el "exponente" E)
- 2 dígitos para la base 10 (la "base" B).

Es decir, solo se requieren 7 dígitos para poder escribirlo. Nuevamente esta notación es más compacta.

# Números en punto flotante

- Si solo hay que almacenar los dígitos de M, B y E, tenemos la siguiente situación:
  - M: 3 dígitos
  - B: 2 dígitos
  - E: 2 dígitos
  - Total notación compacta: 3+2+2= 7 dígitos
  - El mismo número escrito en punto fijo era:

    0,0000000000000976 que requiere,
  - Total notación punto fijo = 17 dígitos
  - Conclusión: con esta notación podemos representar números "muy chicos" con una menor cantidad de dígitos.

# Concepto de representación de números en punto flotante

- En ambos casos (número muy grande y número muy chico) los números anteriores se pudieron escribir de la siguiente forma:

$$M \times 10^{E}$$

Contenido: Personal

# Concepto de representación de números en punto flotante

En general, todos los números pueden escribirse como:

$$M \times B^{E}$$

donde:

- M representa la parte "entera" del número, es decir 976
- B es la base numérica, es decir 10
- E el exponente al que se eleva la base (posición de la coma).

- Lo que implica esta nueva forma de escribir un número es que la coma decimal se puede desplazar en forma dinámica a una posición conveniente, y se usa el exponente de la base para mantener la "pista" de la ubicación de la coma.

# Números en punto flotante

- Si se va a usar siempre la misma base B, entonces no es necesario almacenarla, por lo que el número:

$$M \times B^{E}$$

puede ser almacenado con solo 2 campos: Mantisa y Exponente.

- El signo del número puede estar separado o incluido en el signo de la mantisa.
- En conclusión, un número puede ser representado en punto flotante con solo almacenar M y E.
- La gran ventaja es que se necesitan menos bits para almacenar M y E, que para almacenar el "número completo" en la base correspondiente.

# Números en punto flotante

- M y E estarán representados en alguno de los sistemas de representación en punto fijo ya visto, como por ejemplo BSS, BCS, Ca2, Ca1, Exceso.
- La figura siguiente muestra un formato típico de representación en punto flotante, donde el campo “signo” identifica el signo del número (el exponente tiene implícitamente su propio signo, como vamos se va a ver mas adelante).


- Como elegir las representaciones del campo exponente y del campo mantisa?
- Qué efectos tienen sobre el rango y la resolución?

# Punto flotante – Ejemplo 1

Ejemplo 1:

- Supongamos el siguiente formato en punto flotante de 8 bits:

| | 4 bits | | 4 bits |
|---|---|---|---|
| Mantisa | BSS entera | Exponente | BSS entero |

Determinar el rango y resolución?

El formato del ejemplo es el siguiente:

| Exponente (4) | Mantisa (4) |
|---|---|

# Punto flotante – Ejemplo 1

Ejemplo 1:

- Para hallar el rango debemos calcular el valor más chico y el valor más grande posibles, en la representación numérica asignada.
  - El valor más chico corresponde a la mantisa más chica y el exponente más chico.
  - El valor más grande corresponde a la mantisa más grande y el exponente más grande
- Para encontrar la resolución debemos determinar la distancia entre puntos consecutivos.

# Punto flotante – Ejemplo 1

- Máximo = $1111 \times 2^{1111} = 15 \times 2^{15}$
- Mínimo = $0000 \times 2^{0000} = 0$

Es decir que:

- Rango = $[\ 0,..,15x2^{15}] = [\ 0,..,\ 491520\ ]$
- Resolución: en el extremo superior

  $R = (15 - 14)x2^{15} = 1 \times 2^{15}$
- Resolución: en el extremo inferior

  $R = (1 - 0)x2^{0} = 1$

Las resoluciones no son iguales en ambos intervalos considerados.

# Punto flotante – Ejemplo 1

- Si hubiésemos usado los 8 bits en representación BSS, resultaría:
- Rango = [ 0,..,255 ]
- Resolución: en el extremo superior

  R = 255 – 254 = 1
- Resolución: en el extremo inferior

  R = 1 – 0 = 1

Las resoluciones son iguales en ambos intervalos considerados.

# Punto flotante – Ejemplo 1

Comparando ambos ejemplos vemos:

- el rango en punto flotante es mayor (491520 contra 255).
- la cantidad de combinaciones binarias distintas es la misma en ambos sistemas $2^8$ =256.
- Pero la cantidad de puntos distintos es menor en punto flotante, porqué hay puntos repetidos. Por ejemplo, cuáles?
- En punto flotante la resolución no es constante a lo largo de la representación, como lo es en el caso de BSS, porqué la distribución de puntos no es constante.

# Punto flotante – Ejemplo 1

## Conclusiones

- En el sistema de punto flotante el rango es mayor.
- Podemos representar números más grandes (y como se verá a continuación también más pequeños) que en un sistema de punto fijo, para igual cantidad de bits.
- Pero pagamos el precio que los números no están igualmente espaciados, como en punto fijo. Es decir perdemos resolución.

# Punto flotante – Ejemplo 2

Ejemplo 2:

- Supongamos el siguiente formato en punto flotante:

Mantisa { 4 bits, Ca2, entera } Exponente { 4 bits, Ca2, entero }

Determinar el rango y resolución?

El formato del ejemplo es el siguiente:

| Exponente Ca2 (4) | Mantisa Ca2(4) |
|---|---|

# Punto flotante – Ejemplo 2

- Máximo=0111 x $2^{0111}$ = +7 x $2^{+7}$=+896
- Mínimo =1000 x $2^{0111}$ = -8 x $2^{+7}$ =-1024
- Rango = [-1024,…,896]
- Resolución en el extremo superior

  R = (7 – 6) x $2^{7}$ = 1 x $2^{7}$ = 128
- Resolución cerca del 0

  R = (1 x $2^{-8}$ – 0 ) = 1 x $2^{-8}$ = 1/256

# Punto flotante – Ejemplo 2

Si hubiésemos usado los 8 bits en representación CA2 resultaría:

- Rango = [ -128,..,127 ]
- Resolución en el extremo superior

  R = 127 – 126 = 1
- Resolución en el 0

  R = 1 – 0 = 1

# Punto flotante – Ejemplo 2

Comparando ambos ejemplos vemos:

- el rango en punto flotante es mayor (pero no tanto, porqué?).
- la cantidad de combinaciones binarias distintas es la misma en ambos sistemas $2^8$ =256. Pero nuevamente la cantidad de puntos distintos es menor en punto flotante (porque hay puntos repetidos).
- en punto flotante la resolución no es constante a lo largo del intervalo (la distribución de puntos no es constante).
- pero la resolución cerca del 0 es mayor, es decir, se pueden representar números más chicos en el entorno del 0.

# Punto flotante – Ejemplo 3

- Ejemplo 3: supongamos un número de 32 bits con el siguiente formato:

| | | | |
|---|---|---|---|
| Mantisa | 23 bits | Exponente | 8 bits |
| | BCS (ó MyS) | | Ca2 |
| | fraccionaria | | entero |
| | 1 bit signo | | |

Determinar el rango y resolución.

El formato del número en punto flotante es el siguiente:

| S | Exp. (8) | Mantisa (23) |
|---|---|---|

# Punto flotante – Ejemplo 3

- ✓ Máximo positivo

0 $0,111..111 \times 2^{01111111} = +(1-2^{-23}).2^{+127}$

- ✓ Mínimo positivo (>0)

0 $0,000..001 \times 2^{10000000} = +(2^{-23}).2^{-128}$

- ✓ Cero

0 $0,000..000 \times 2^{bbbbbbbb}$

- ✓ Máximo negativo (<0)

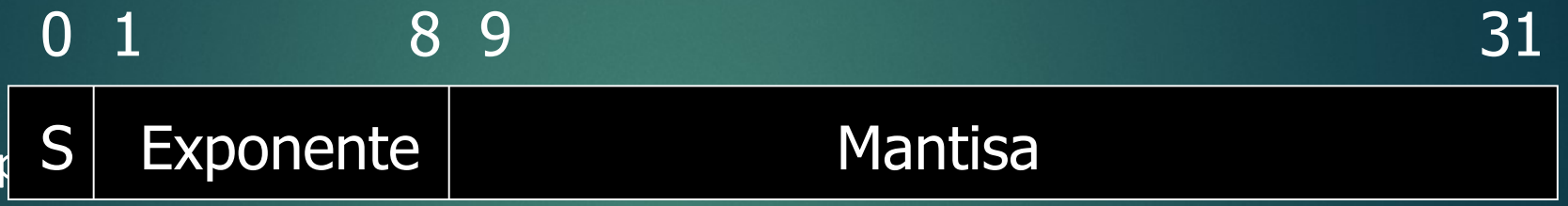
1 $0,000..001 \times 2^{10000000} = -(2^{-23}).2^{-128}$

- ✓ Mínimo negativo

1 $0,111..111 \times 2^{01111111} = -(1-2^{-23}).2^{+127}$

# Punto flotante – Ejemplo 3

- Rango: $[-(1-2^{-23}).2^{+127}, +(1-2^{-23}).2^{+127}]$
- El formato del número queda:




- Ejemp

- Ejemp

| 1 | 01111111 | 1111.....11 |
|---|---|---|

| 0 | 01111111 | 1111.....11 |
|---|---|---|

# Normalización

- Problema: se observa que existen varias combinaciones distintas de mantisa+exponente que representan un mismo número.

Ejemplo (número en decimal):

$$40 = 40x10^{0} = 4x10^{1} = 0{,}4x10^{2} = 400x10^{-1}$$

- Lo mismo sucede en base 2.
- Con el objetivo de tener un único par de valores de mantisa y exponente para un número, se introduce el concepto de normalización.

# Normalización

Una forma de evitar mútiples formas de representación de un mismo número, es definir un formato único de mantisa.

- Por ejemplo, se podría definir que todas las mantisas empiecen con 0,1... con la coma a la izquierda del primer bit en 1. Es decir, todas las mantisas deben ser de la forma:

0,1ddddddd.....ddd

donde d es un dígito binario que vale 0 ó 1.

- También se podría definir que todas las mantisas empiecen con 1,….  con la coma a la derecha del primer bit en 1. Es decir, todas las mantisas deben ser de la forma:

1,ddddddd.....d

donde d es un dígito binario que vale 0 ó 1.

# Normalización – Ejemplo 1

Ejemplo: supongamos un número de 32 bits con el siguiente formato:

23 bits fraccionaria 8 bits

BCS Exponente Exceso 128

Mantisa 1 bit de signo entero

Normalizada 0,1ddddddd.....ddd

Determinar el rango y resolución.

El número en punto flotante tiene el siguiente formato:

| S | Exp. (8) | Mantisa (23) |
|---|---|---|

# Normalización – Ejemplo 1

Si se tiene el siguiente número en punto flotante:

| 0 | 00110011 | 10000000000000000000010 |
|---|---|---|

Representa:

+ 0,10000000000000000000010 x $2^{00110011}$

Los puntos más significativos de la representación se muestran a continuación.

# Normalización – Ejemplo 1

- ✓ Máximo positivo

  0 0,1 11..111 x $2^{11111111}$=+(1-$2^{-23}$).$2^{+127}$ (=+1,7014 x $10^{38}$)

- ✓ Mínimo positivo (≠0)

  0 0,1 00..000 x $2^{00000000}$=+(0,5).$2^{-128}$

- ✓ Máximo negativo (≠0)

  1 0,1 00..000 x $2^{00000000}$= - (0,5).$2^{-128}$

- ✓ Mínimo negativo

  1 0,1 11..111 x $2^{11111111}$= -(1-$2^{-23}$).$2^{+127}$

- ✓ Como se escribe el cero? => no se puede, no existe!

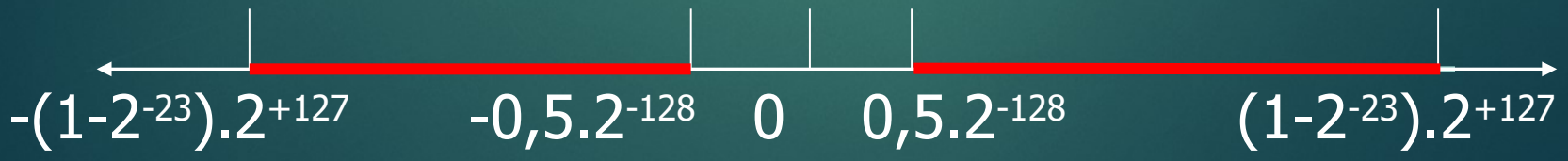
# Normalización – Ejemplo 1

- Formato del número en punto flotante normalizado:


- Ejemplo: el máximo negativo (≠0) es

| 1 | 00000000 | 1000.....00 |
|---|---|---|

- Distribución del rango en la recta



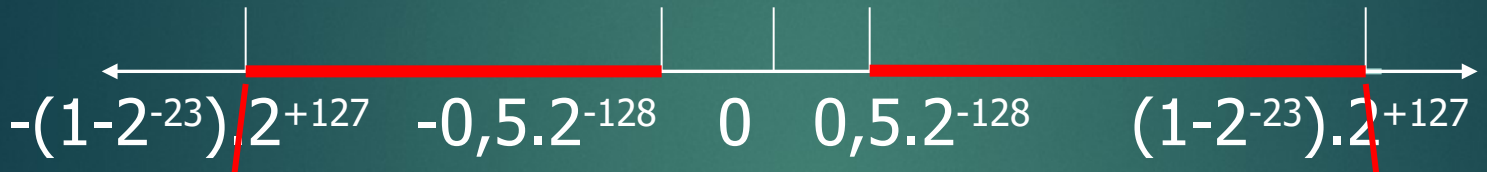
# Normalización – Bit implícito

- Como todos los números comienzan con 0,1  es necesario almacenar el 1?
- Si siempre está presente  no es necesario, y se puede obviar.
- Si no se almacena, se puede “adicionar” un bit más a la mantisa. El bit en 1 no almacenado se conoce como bit implícito.

# Normalización – Bit implícito

Conclusión: hay 2 opciones de normalización:

1) Sin bit implícito




2) Con bit implícito

# Punto flotante normalizado

Problema: como escribir un número en punto flotante normalizado?

1. Se escribe el Nº en el sistema numérico propuesto para la mantisa.
2. Se desplaza la coma hasta obtener la forma normalizada, ajustando el exponente de acuerdo al desplazamiento de la coma.
3. Se convierte el exponente al sistema propuesto para él.

# Punto flotante normalizado

Ejemplo: escribir el número -13,5 en punto flotante normalizado según la siguiente definición:

**Mantisa**

- BCS
- 23 bits fraccionaria
- 1 bit de signo

entero

normalizada

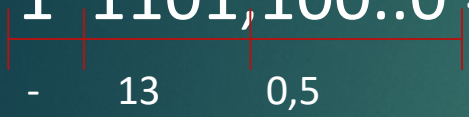
**Exponente**

- Exceso 128
- 8 bits

# Punto flotante normalizado

1) Mantisa en MYS

$1\ 1101,100..0 = 1\ 1101,100..0 \times 2^0$




2) Normalización de mantisa y exponente

$1\ 0,110110..0 \times 2^4$

3) Conversión del exponente

4 en Ca2=00000100

4 en Exceso 128 = 10000100

# Punto flotante normalizado

El número -13,5 se escribe :

1)Sin bit implícito:

| | | |
|---|---|---|
| 1 | 10000100 | 1101100000.....00 |

2)Con bit implícito:

| | | |
|---|---|---|
| 1 | 10000100 | 101100000.....00 |

# Resolución y error en PF

Como son la resolución y el error en punto flotante?

- Resolución: es la diferencia entre dos representaciones consecutivas. En punto flotante varía a lo largo del rango, no es constante como en el caso de punto fijo.
- Error absoluto: es la diferencia entre el valor representado y el valor a representar. Varía a lo largo del rango.
- Error relativo: error absoluto referido al número a representar.

# Resolución y error

- Error Absoluto = Resolución/2

- Error Relativo = Error Absoluto/Número a representar

# Punto flotante - Estándar IEEE 754

Estandar IEEE 754 de punto flotante normalizado:

- Prevé 2 formatos (con 2 precisiones): en 32 ó 64 bits.
- Formato Simple precisión 32 bits:


- Formato Doble precisión 64 bits

# Punto flotante - Estándar IEEE 754

- Mantisa
  - M y S, con:
    - Formato simple precisión: 23 bits
    - Formato doble precisión: 52 bits
  - 1 bit de signo
  - fraccionaria normalizada, del tipo 1,….
- Exponente
  - Exceso $2^{n-1} – 1$, con:
    - Formato simple precisión: 8 bits
    - Formato doble precisión: 11 bits

# Punto flotante - Estándar IEEE 754

| | Simple precisión | Doble precisión |
|---|---|---|
| Bits en signo | 1 | 1 |
| Bits en exponente | 8 | 11 |
| Bits en fracción | 23 | 52 |
| Total de bits | 32 | 64 |
| Exponente en exceso | 127 | 1023 |
| Rango de exponente | -126 a +127 | –1022 a +1023 |
| Rango de números | $2^{-126} \sim 2^{128}$ | $2^{-1022} \sim 2^{1024}$ |

- Exponentes 00 (-127) y FF (+128) reservados

# Punto flotante - Estándar IEEE 754

Ejemplo:

Determinar el valor que representa el número en punto flotante según el IEEE 754: $3F800000_{16}$?

0011 1111 1000 0000 0000 0000 0000 0000

01111111=127 en exceso 127 representa 0

(1,)000 0000 0000 0000 0000 0000=0

+ $1,0 \times 2^0 = 1$

# Punto flotante - Estándar IEEE 754



¿Qué valor representa el hexadecimal $C0066666_{16}$ según el IEEE 754?

1 100 0000 0 000 0110 0110 0110 0110 0110

10000000=128 en exceso 127 representa 1

(1,)000 0110 0110 0110 0110 0110=0,05

- 1,05 x $2^1$ = -2,1

# Punto flotante - Estándar IEEE 754

El estándar dispone de 4 casos especiales:

- ✓ Si E=FF (255 en SP o 2047 en DP) y M = 0 ⇒Infinito
- ✓ Si E=FF (255 en SP o 2047 en DP) y M ≠ 0⇒NaN (Not a Number)
- ✓ Si E = 00 y M = 0 ⇒ Cero
- ✓ Si E = 00 y M ≠ 0 ⇒ Desnormalizado
  - ± 0,mantisa_s-p $2^{-126}$
  - ± 0,mantisa_d-p $2^{-1022}$

# Operaciones aritméticas en pf

## Sumar y restar

- Alinear mantisas (ajuste de exponentes):

Por ejemplo:

$123x10^{0} + 456 \times 10^{-2} = 123x10^{0} + 4{,}56x10^{0} = 127{,}56 \times 10^{0}$

- Si es resta, se cambia el signo del sustraendo.
- Si algún operando es 0, la operación se iguala al otro.
- Se suman los operandos.
- El resultado puede dar OK, 0, o desborde.
- Normalizar el resultado, desplazando a izquierda hasta que resulte el bit más significativo en 1, y decrementando el exponente.

# Operaciones aritméticas en pf

## Multiplicar y dividir

- Comprobar si algún operando es 0
  - En multiplicación: resultado = 0
  - En división: si dividendo es 0, resultado = 0

    si divisor es 0 , resultado = NaN
- Signo: or-exclusivo de signos de operandos
- Sumar o restar exponentes.
- Multiplicar o dividir mantisas
- Normalizar.
- Redondear.

Todos los resultados intermedios deben doblar su longitud al almacenarse

# mayor información …

- Punto flotante

  Apunte 2 de Cátedra
- Capítulo 8: Aritmética del computador (8.4., 8.5.)

  Stallings, 5ta Ed.
- Aplicación PFI-PFO

  Descargar de página de cátedra

- Link de interés
  - http://babbage.cs.gc.edu/ieee-754/