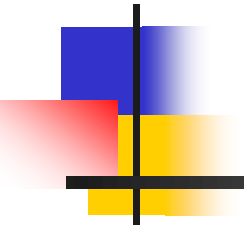
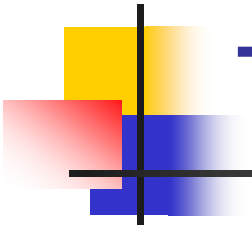


Organización de Computadoras
2021

Turno Recursantes

Clase 2

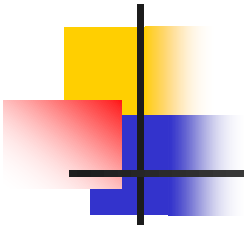


Temas de Clase

- Representación de números en Punto Flotante

Números en punto fijo

- Todos los números a representar tienen exactamente la misma cantidad de dígitos y la coma fraccionaria está siempre ubicada en el mismo lugar. Por esa razón no se necesita almacenarla.
- La representación en el sistema de cómputo y el papel es similar. La diferencia principal entre ambas representaciones es que no se guarda coma alguna, se supone que está en un lugar determinado.



Rango y Resolución

En la representación en punto fijo, 2 propiedades son importantes:

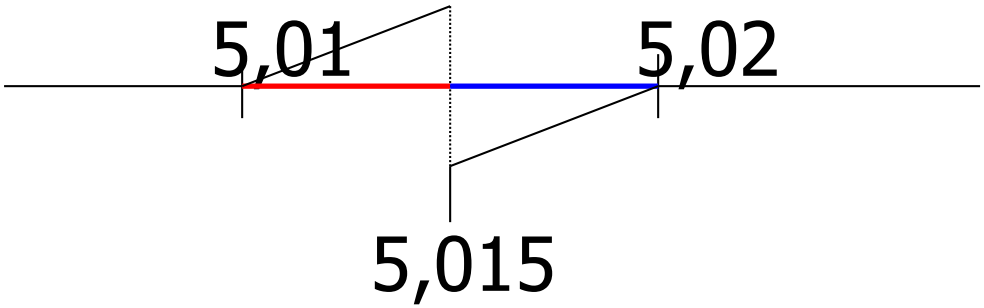
- Rango: diferencia entre el número mayor y el menor. Se obtiene calculando los valores de los números más chico y más grande admitidos por la representación.
- Resolución: diferencia ("distancia") entre dos números consecutivos. Se obtiene determinando valores consecutivos.

Error en punto fijo (1)

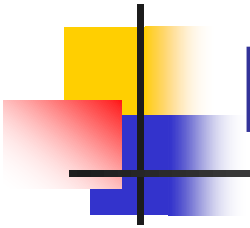
El error es la diferencia entre el valor real del número y el valor usado para representarlo.

El máximo error cometido en una representación puede considerarse como la mitad de la diferencia (resolución) entre dos números consecutivos.

Ej:



- $5{,}01 \leq N^o \leq 5{,}015$ se representa por 5,01
- $5{,}015 < N^o \leq 5{,}02$ se representa por 5,02



Error en punto fijo (2)

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

  EA max = 5,015 - 5,01 = 0,005 ó

  (5,02 - 5,01)/2 = 0,005

  - Es decir, la mitad de la resolución.
  - Exactamente en el medio de 2 puntos consecutivos de la representación.

Posibles representaciones de números fraccionarios

- En punto fijo (ej. Ca2), es posible representar un rango de enteros positivos y negativos centrados en 0.
- Suponiendo un número con componente fraccionaria, en este formato de punto fijo también se pueden representar números.
- Limitaciones: "números muy grandes y números muy pequeños".

Posibles representaciones de números fraccionarios

- Un número decimal "muy grande":

976.000.000.000.000 (15 díg.)

se puede representar como:

$9{,}76 \times 10^{14} = 976 \times 10^{12}$ (5+base)

- Un número decimal "muy pequeño":

0,0000000000000976 (17 díg.)

se puede representar como:

$9{,}76 \times 10^{-14} = 976 \times 10^{-16}$ (5+base)

Números en punto flotante

- Los números anteriores se han escrito con un formato del tipo:

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

donde M representa la parte "entera" (976), 10 es la base numérica, y E el exponente al que se eleva la base (posición de la coma).

- Lo que hemos hecho es desplazar en forma dinámica la coma decimal a una posición conveniente y usar el exponente de base 10 para mantener la "pista" de la coma.

Números en punto flotante

- Con esta notación podemos representar un rango de números desde "muy pequeños" a "muy grandes" con una menor cantidad de dígitos.

Ejemplo:

976.000.000.000.000 → 15 dígitos

976×10^{12} → 3+2 dígitos (sin considerar la base)

- En general, un número se puede representar de la siguiente forma:

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

Números en punto flotante

- Si consideramos que la base B no se requiere almacenar dado que va a ser siempre la misma, un número puede ser almacenado en 2 campos: Mantisa y Exponente.
- El signo del número puede estar separado o incluido en el signo de la mantisa.
- Solo se deben 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 están representados en alguno de los sistemas en punto fijo que ya conocíamos como BSS, BCS, Ca2, Ca1, Exceso.

  La figura siguiente muestra un formato típico de representación en punto flotante.

Números en punto flotante Ejemplo 1

- Supongamos el siguiente formato en punto flotante:

Mantisa	BSS 4 bits entera	Exponente	BSS 4 bits entero

Determinar el rango y resolución?

Exponente (4)	Mantisa (4)

Números en punto flotante 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.

Números en punto flotante Ejemplo 1

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

  $R = (1 - 0)x2^{0} = 1$ ←

Números en punto flotante Ejemplo 1

Si hubiésemos usado los 8 bits en representación BSS, resultaría:

- Rango = [0,..,255]
- Resolución en el extremo superior

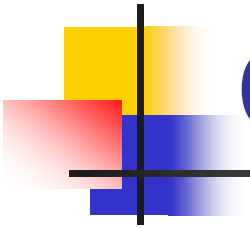
  R = 255 – 254 = 1
- Resolución en el extremo inferior

  R = 1 – 0 = 1

Comparación 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).
- ✓ 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).



Conclusión

- ✓ 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 Nºs no están igualmente espaciados, como en punto fijo. Es decir perdemos resolución.

Ejemplo 2

- Ejemplo: supongamos el siguiente formato en punto flotante

Mantisa	Ca2 4 bits entera	Exponente	Ca2 4 bits entero

Determinar el rango y resolución:

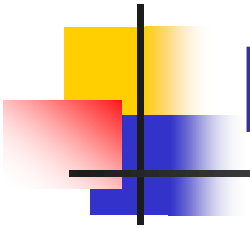
Exponente Ca2 (4 bits)	Mantisa Ca2 (4 bits)

Ejemplo 2 (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

  $R = (7 - 6) \times 2^7 = 1 \times 2^7 = 128$
- Resolución en el origen

  $R = (1 \times 2^{-8} - 0) = 1 \times 2^{-8} = 1/256$



Ejemplo 2 (3)

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

  $R = 1 - 0 = 1$

Comparación 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 (hay puntos repetidos).
- ✓ en punto flotante la resolución no es constante a lo largo del intervalo, como lo es en el segundo ejemplo (la distribución de puntos no es constante). Pero la resolución cerca del 0 es mayor.

Ejemplo 3

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

Mantisa
- BCS (ó MyS)
- 23 bits
- fraccionaria
- 1 bit signo

Exponente
- Ca2
- 8 bits
- entero

Determinar el rango y resolución

S	Exp. (8)	Mantisa (23)

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

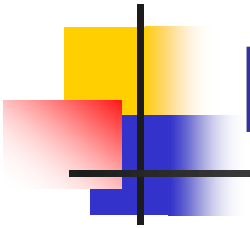
1 $0{,}111..111 \times 2^{01111111} = -(1-2^{-23}).2^{+127}$

Formato final

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

- Ejemplo: el mínimo negativo es

1	01111111	1111.....11

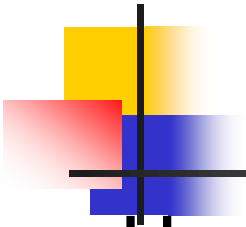


Normalización

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

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

- 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: 0,1ddddddd.....ddd
  - donde d es un dígito binario que vale 0 ó 1.
- Para este ejemplo todas las mantisas deben empezar con 0,1...
- También se puede pensar que la coma debe quedar a la izquierda del primer bit en 1.

Normalización

- Ejemplo: formato en punto flotante

Mantisa
- BCS
- 23 bits fraccionaria
- 1 bit de signo
- Normalizada

Exponente
- Exceso 128
- 8 bits
- entero

Determinar el rango y resolución

S	Exp. (8)	Mantisa (23)

Normalización

- ✓ Máximo positivo

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

- ✓ Mínimo positivo ($\neq 0$)

  0 0,100..000 x $2^{00000000} = +(0,5).2^{-128}$

- ✓ Máximo negativo ($\neq 0$)

  1 0,100..000 x $2^{00000000} = -(0,5).2^{-128}$

- ✓ Mínimo negativo

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

- ✓ Cero: no existe!

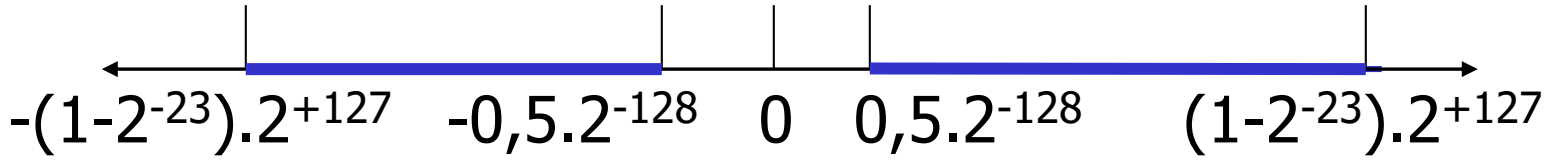
Normalización

- El formato anterior se puede representar

- Ejemplo: el máximo negativo ($\neq 0$) es

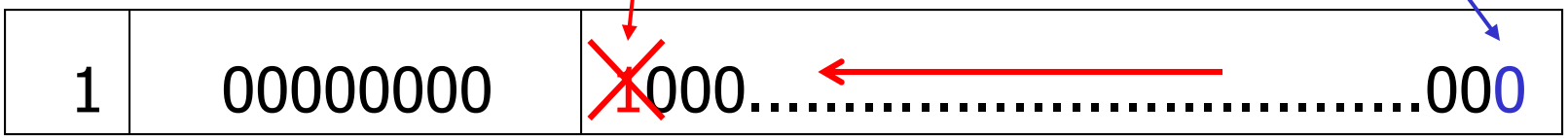
1	00000000	1000.....00

- Distribución del rango en la recta



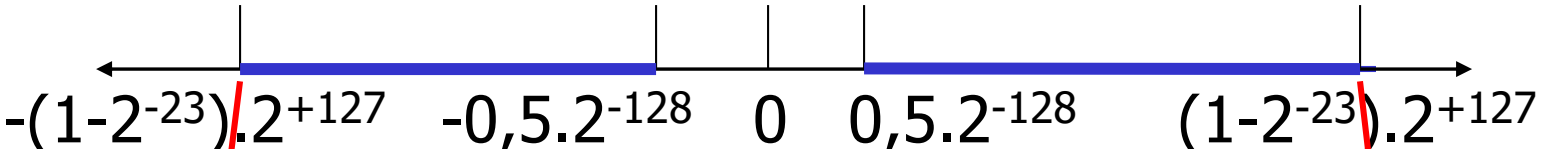
Bit implícito

- 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 lo almaceno, puedo "adicionar" un bit más a la mantisa. El bit en 1 no almacenado se conoce como *bit implícito.*

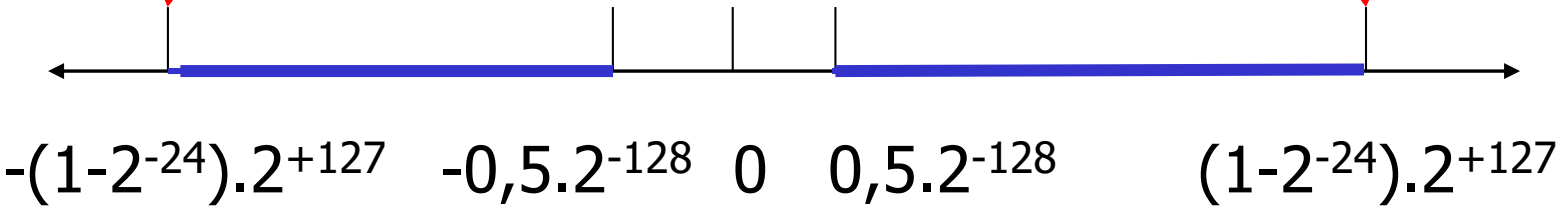


Recta numérica

- Sin bit implícito



- Con bit implícito



¿Cómo se escribe un Nº en punto flotante normalizado?

1. Se escribe el Nº en el sistema numérico propuesto para la mantisa.
2. Se desplaza la coma y se ajusta el exponente hasta obtener la forma normalizada.
3. Se convierte el exponente al sistema propuesto para él.

¿Cómo……? (2)

Ej. - 13,5

Pasarlo a punto flotante normalizado según:

Mantisa
- BCS
- 23 bits fraccionaria
- 1 bit de signo
- Normalizada

Exponente
- Exceso 128
- 8 bits
- entero

¿Cómo……? (2)

- 1) Mantisa en MYS

  1 1101,100..0 = 1 1101,100..0x2^0

- 2) Normalización de mantisa y exponente

  1 0,110110..0 x 2^4

- 3) Conversión del exponente

  4 en Ca2=00000100

  4 en Exceso 128 = 10000100

¿Cómo..... ? (3)

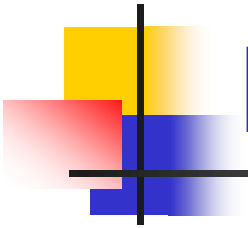
El número con mantisa en MYS y exponente en exceso 128 queda:

- Sin bit implícito

1	10000100	1101100000.....00

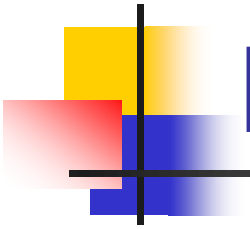
- Con bit implícito

1	10000100	101100000.....00



Resolución – Error absoluto

- 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.
- Error relativo: error absoluto referido al número a representar.



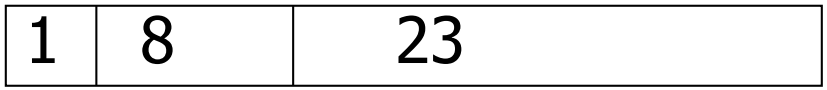
Error absoluto y relativo

- Error Absoluto máximo $\leq$ Resolución/2
- Error Relativo = EA/Número a representar

Estándar IEEE 754

Estandar IEEE 754 de punto flotante normalizado:

- 2 precisiones: 32 ó 64 bits.
- Formato Simple precisión 32 bits:



- Formato Doble precisión 64 bits

Estándar punto flotante IEEE 754

- Mantisa:
  - M y S
  - Mantisa 23 bits (en s-p) o 52 bits (en d-p)
  - 1 bit de signo
  - fraccionaria normalizada, con la coma después del primer bit en uno (es decir del tipo 1,…).
- Exponente:
  - Exceso $2^{n-1} - 1$
  - 8 bits (en s-p) o 11 bits (en d-p)

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} a ~2^{128}	2^{-1022} a ~2^{1024}

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

Ejemplo 1 en simple precisión

¿Qué valor representa el hexadecimal $3F800000_{16}$?

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 x 2^0 = 1

Ejemplo 2 en simple precisión

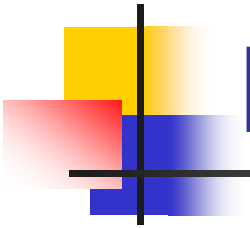
¿Qué valor representa el hexadecimal C0066666?

1100 0000 0000 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



Estándar IEEE 754

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

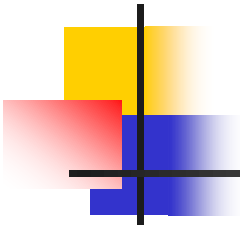
Sumar y restar

- Alinear mantisas (ajuste de exponentes).

Pensar: $123x10^{0} + 456 \times 10^{-2} =$

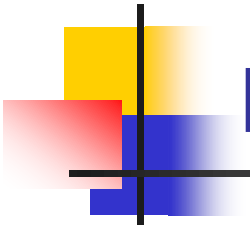
$$123x10^{0} + 4{,}56x10^{0} = 127{,}56 \times 10^{0}$$

- 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... (2)

- Multiplicar y dividir
  - Comprobar valores cero.
  - Sumar o restar exponentes.
  - Multiplicar o dividir mantisas
    - tener en cuenta el signo
  - 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/