Universisdad de Los Andes
Facultad de Ingeniería
Escuela de Sistemas

Aritmética
Punto Flotante

Basada en: What Every Computer Scientist Should Know About Floating-Point Arithmetic
Por: David Goldberg

Prof. Gilberto Díaz
gilberto@ula.ve

Departamento de Computación, Escuela de Sistemas, Facultad de Ingeniería
Universidad de Los Andes, Mérida 5101 Venezuela
Programación Paralela y Distribuida

Motivación

- ¿Los programas que desarrollamos están bien hechos?
- ¿Son correctos los resultados que obtenemos?
- ¿Estamos haciendo cosas peligrosas?

Introducción

Casi todos los lenguajes de programación proporcionan el tipo de dato punto flotante

Desde los PCs hasta los supercomputadores tienen aceleradores de operaciones punto flotantes

Muchos compiladores son ejecutados para procesar algoritmos punto flotantes y

Virtualmente todo sistema operativo debe responder a excepciones punto flotantes (overflows)

Problema de la Representación

Existen infinitos números enteros

Sin embargo, en la mayoría de los problemas la representación de los datos y resultados es suficiente utilizar 32 bits.

$$N = \{1, 2, 3, 4, 5, \ldots\ldots\ldots\ \infty\}$$

Problema de la Representación

También existen infinitos números reales

Pero su representación es más compleja y se tiene que realizar una aproximación para realizar esta tarea.

$$R = \{1, \ldots \infty \ldots\ldots 2, \ldots\ldots \infty \}$$

Problema de la Representación

Dado una cantidad finita de bits, las operaciones con números reales producen resultados que no pueden ser representadas en esa cantidad de bits.

Problema de la Representación

Por esto los resultados de operaciones con números reales deben ser redondeados para que puedan ser representados en una cantidad finita de bits.

El error de redondeo resultante es una característica de las operaciones con tipos de datos punto flotante.

Historia

- En los 60s y 70s las operaciones con números reales se manejaban de forma distinta en cada computador
  - Formato
  - Precisión
  - Redondeo
  - Gestión de excepciones
  - Etc.
- De esta manera era muy difícil escribir código portátil

Historia

- En 1982 la IEEE definió el estándar IEEE-754 y lo implantó por primera vez en los Intel 8087.
- En 1985 este formato fue aceptado como el estándar universal.
- Esto aseguró al menos que las máquinas que lo utilizaban podian correr programas que obtuvieran los mismos resultados

Historia

- Para el 2000 el estándar IEEE-754 es implantado universalmente en todos los computadores de propósito general.

Representación de Números Reales

Hay varias técnicas para representar los números reales en un computador:

- Binary Coded Decimal (BDC)
- Números Racionales
- Punto Fijo
- Punto Flotante

Binary Coded Decimal

Cada dígito es representado por 4 bits, ejemplo:

38.57 = 0011 1000 0101 0111

- Es lento
- No se adecua a ninguna plataforma de hardware

Números Racionales

Los números son almacenados como números racionales, ejemplo

0.5 = ½ 0.3333 = 1/3

- Excente precisión
- Se necesitan números racionales muy grandes

Punto Fijo

Cada número real es representado como un entero utilizando un número fijo de lugares decimales, ejemplo

1234,56 = 123456

- Reales con un número grande de dígitos decimales no pueden ser representados

Punto Flotante

Se utiliza la notación científica:

mantisa/base/exponente

- Un amplio número de valores se pueden representar cuando se utiliza una notación de longitud fija.
- La diferencia entre dos números punto flotante no es constante.

Punto Flotante

- Los números punto flotante son finitos
- Muchos números reales se mapean en el mismo número punto flotante.

Gestión de Números Reales

- **Representación**: se utiliza la notación punto flotante

(mantisa/base/exponente)

Gestión de Números Reales

- **Operaciones**: indica como se deben manipular los números. Operaciones para
  - Aritmética: +, -, *, /, %, etc.
  - Comparación: <, >, ==, >=, <=
  - Conversión entre distintos formatos:
    Float → Double
    Float → Integer, etc.
  - Manejo de excepciones: ¿Qué hacer en caso de?
    NúmeroMásGrande * 2
  - Conversión binario/decimal (I/O)

Gestión de Números Reales

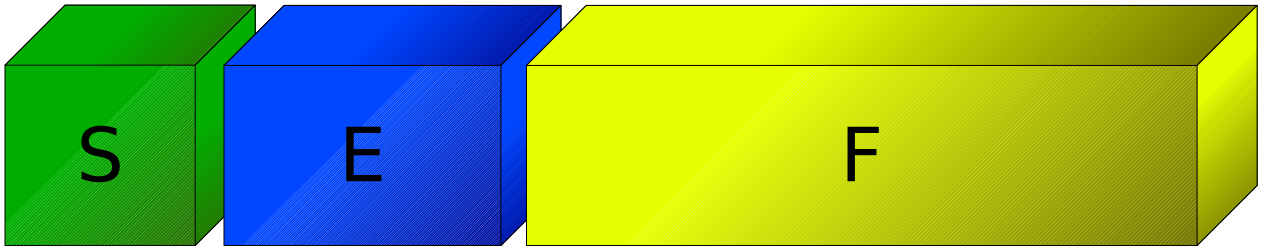
- **Lenguaje/Bibliotecas**: debe existir soporte para todas las operaciones anteriores para todas las operaciones anteriores

Formato de Punto Flotante

El formato de la notación punto flotante es una especie de notación científica:

$$\pm mantisa * 2^{exponente}$$

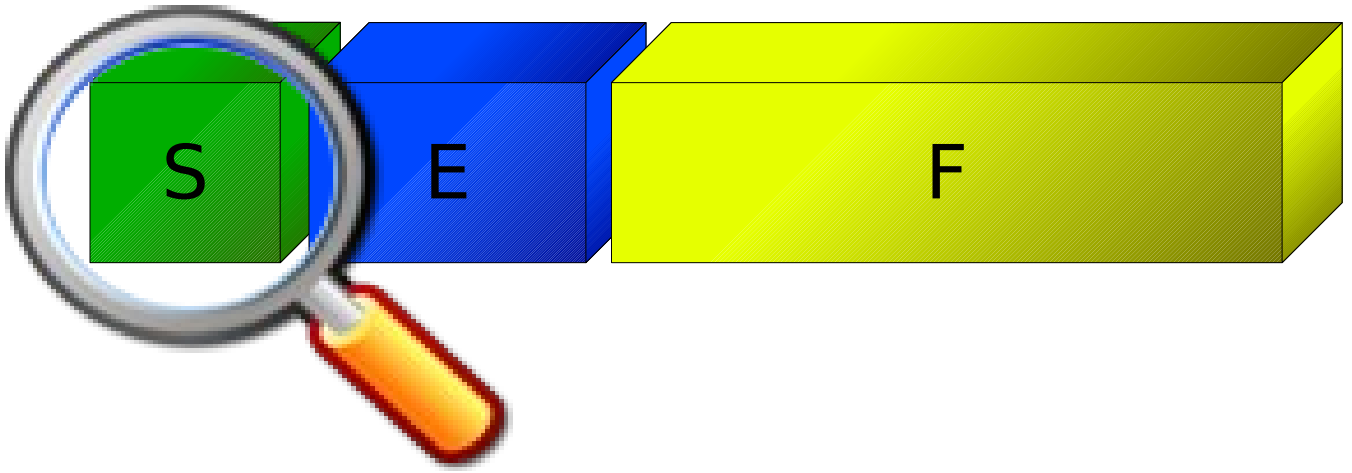
Podemos utilizar 3 campos binarios para la representación: signo (S), exponente (E) y fracción (F)



Formato de Punto Flotante

El **signo** se hace con la siguiente lógica:

0 para representar un número positivo
1 para representar un número negativo



Formato de Punto Flotante

El **exponente** es desplazado:

Se le suma 127 en precisión simple o
Se le suma 1023 en precisión doble

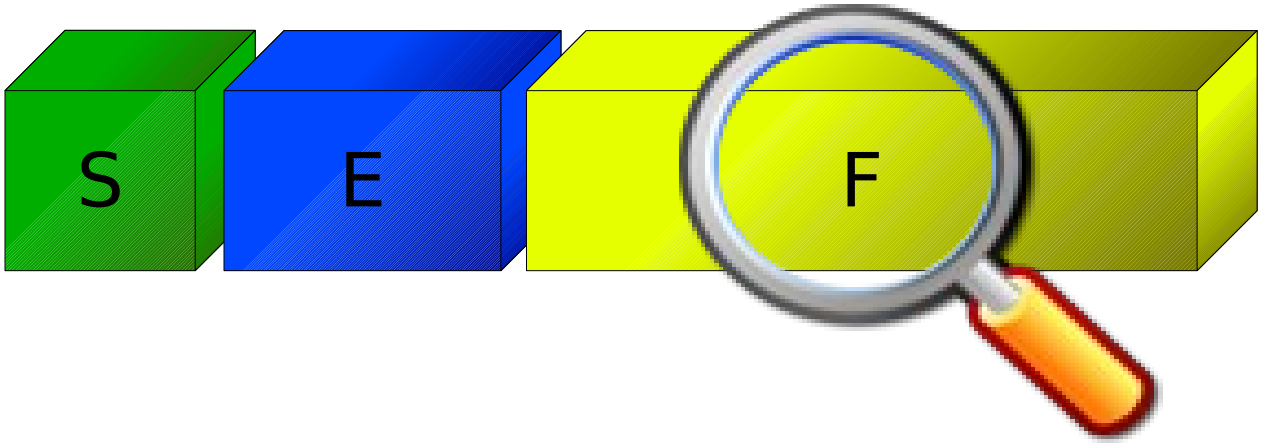
Ejemplos:

- Si el exponente es 4 el campo E será 131 (10000011)
- Si el campo es 01011101 (93) el exponente será -34.

Formato de Punto Flotante

La **fracción** tiene un número uno (1) implícito a la izquierda, ejemplo:

Si el campo es 01101 la mantisa real será 1.01101



Se utiliza una notación científica normalizada para escribir un número punto flotante

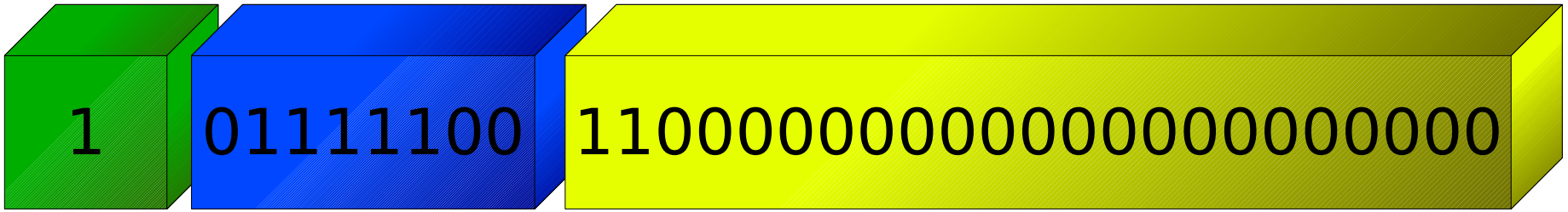
De Punto Flotante a Decimal

Para convertir un número de notación punto flotante a notación decimal utilizamos la siguiente relación

$$(1 - 2S) * (1 + F) * 2^{E\text{-}Desplazamiento}$$

De Punto Flotante a Decimal

Ejemplo: convertir a decimal el siguiente punto flotante:



$$(1 - 2S) * (1 + F) * 2^{E-Desplazamiento}$$

$S = 1$

$E = 01111100 = 124$

$F = 110000.... = 0.75$

$(1-2*1) * (1+0.75) * 2^{124-127} = (-1.75*2^{-3}) = -0.21875$

Valores Especiales

Tenemos que considerar algunos valores particulares

Clase	Exp	Fracción
2 Ceros (+ y -)	0	0
No. desnormalizados	0	≠0
No. normalizados	1-254	cualquiera
2 Infinitos (+ y -)	255	0
Not a Number (NaN)	255	≠0

Rangos

El número positivo más pequeño (distinto de cero) que podemos representar es

$$1 * 2^{-126}$$

El exponente más pequeño es 00000001 (1)

La fracción más pequeña es

00000000000000000000000 (0)

Rangos

El número positivo más grande (normal)que podemos representar es

$$(2 - 2^{23}) * 2^{127} = 2^{128} - 2^{104}$$

El exponente más grande es 11111110 (254)

La fracción más grande es

11111111111111111111111 ($1 - 2^{23}$)