Pontificia Universidad Católica Madre Y Maestra

Campus Santiago

Facultad de Ciencias de la Ingeniería

Departamento de Ingeniería de Sistemas y Computación



Arquitectura Computacional

ST-ISC-359-T-001

Reporte Práctica #2, T2

*“Números flotantes IEEE 754”*

*Presentado por:*

Ing. Eric T. Núñez Chaves

*Matricula:*

2014-1329

A:

Ing. Álvaro A. Reyes P*.*

En fecha:

19 de Octubre del 2016

Índice

[Introducción 3](#_Toc464672684)

[Estándar IEEE 754 3](#_Toc464672685)

[Definición de las precisiones 4](#_Toc464672686)

[Valores 4](#_Toc464672687)

[Números desnormalizados 5](#_Toc464672688)

[Otros casos 7](#_Toc464672689)

[Ceros 7](#_Toc464672690)

[Infinitos 7](#_Toc464672691)

[NaN(Not a Number) 8](#_Toc464672692)

[Pasar de decimal a binario 8](#_Toc464672693)

Números flotantes IEEE 754

# Introducción

La memoria de una computadora no es ilimitada, es decir, no es posible almacenar números con una precisión infinita. Por esta razón, tanto los números decimales como las fracciones binarias se cortan en algún punto. Para satisfacer las necesidades de aquellos que necesitan un formato preciso para la realización de sus cálculos, se creó el formato de punto flotante.

# Estándar IEEE 754

El estándar IEEE 754 (IEEE Standard for Binary Floating-Point Arithmetic) fue definido por el Instituto de Ingenieros Eléctricos y Electrónicos. En este, se establecen dos formatos básicos con el que poder representar números en coma flotante (incluido el cero): la precisión simple y la precisión doble.

Cómo funcionan los números de punto flotante

Los números deben dividirse en tres partes:

Signo(s): 1 bit de signo (0 positivo,1 negativo). Determina si el valor será negativo o positivo.

Exponente(exp): Indica dónde se coloca el punto decimal (o binario) en relación con donde empieza la mantisa.

Mantisa(M): Contiene los dígitos del número. Existen también las mantisas negativas, que representan números negativos. La mantisa se calculará utilizando el BIAS(B), que determinará el desplazamiento del exponente. Si la parte fraccionaria es diferente de 000…00, el primer bit será 1. En cambio, si la parte fraccionaria es igual a 000…00 el primer bit será 0.

Exponente del valor (E): La fórmula varía según si el número es normalizado o desnormalizado. Si el valor del exponente es mayor que el BIAS, significa que el valor de E es positivo, si es más bajo que el BIAS, será negativo y si es igual, será cero.

# Definición de las precisiones

* Precisión Simple

En este caso, para escribir un número real se utilizan 32 bits: 1 bit para el signo del número (s),23 bits para la parte fraccionaria (frac) y 8 bits para el exponente (exp).

* Precisión doble

Por otro lado, tenemos la precisión doble, en el que se emplean 64 bits: 1 bit para el signo (s) del número, 52 bits para la fraccionaria (frac) y 11 bits para el exponente (exp).

# Valores

Los valores posibles pueden ser divididos en los siguientes:

Números normalizados: Si el exponente es diferente de 000...00 y diferente de 111...11 el número es normalizado

El BIAS se calcula con 2^(n-1)–1, siendo n el número del exponente

B = 2 ^ (8 – 1) – 1 = 127 para simple precisión

B = 2 ^ (11 – 1) – 1 = 1023 para doble precisión

Número= (-1) ^S\*M\*2^E

Ejemplo 1: Escribir el número en base binaria en el estándar IEEE 754 con precisión simple.

1. Dividir el numero en signo, exponente y parte fraccionaria

1 10000011 11100000000000000000000

S=1

Exp=10000011=131

Frac= 11100000000000000000000

1. Calcular BIAS y E

BIAS=2 ^ (8 – 1) – 1 = 127

E=exp – BIAS =131-127=4

1. Calcular la mantisa

Como el exponente es diferente de 0, la mantisa es:

M=1.frac=1.111=1+2^-1+2^-2+ 2^-3=15/8

1. Se obtiene el numero con la fórmula del valor

V=-1^ (1) \*(15/8) \*2^4=-30

Ejemplo 2: Escribir el número en base binaria en el estándar IEEE 754 con precisión doble.

1 10000000011 1110000000000000000000000000000000000000000000000000

1. Dividir el numero en signo, exponente y parte fraccionaria

1 10000011 11100000000000000000000

S=1

Exp=10000011=1,027

Frac= 11100000000000000000000

1. Calcular BIAS y E

BIAS=2 ^ (11 – 1) – 1 = 1,023

E=exp – BIAS =1027-1023=4

1. Calcular la mantisa

Como el exponente es diferente de 0, la mantisa es:

M=1.frac=1.111=1+2^-1+2^-2+ 2^-3=15/8

1. Se obtiene el número con la fórmula del valor

V=-1^ (1) \*(15/8) \*2^4=-30

# Números desnormalizados

Un número es desnormalizado si su exponente es igual a 00…000.

Ejemplo 1: Escribir el número en base binaria en el estándar IEEE 754 con precisión simple.

1 00000000 11010000000000000000000

1.Dividir el número en signo, exponente y parte fraccionaria

1 00000000 11010000000000000000000

Como la parte del exponente es 00…0000 el número es desnormalizado

S=1

Exp=00000000

Frac= 11010000000000000000000

2.Calcular BIAS y E

BIAS=2 ^ (8 – 1) – 1 = 127

Como el número es desnormalizado del valor del exponente cambia:

E=1-BIAS=1-127=-126

3.Calcular la mantisa

Como el exponente es 0, la mantisa es:

M=0.frac=0.1101=0+2^-1+2^-2+ 2^-4=13/16

4.Se obtiene el numero con la fórmula del valor

V=-1^ (1) \*(13/16) \*2^-126=-9.5508916\*10^-39

Ejemplo 2: Escribir el número en base binaria en el estándar IEEE 754 con precisión doble.

1 00000000000 1101000000000000000000000000000000000000000000000000

1.Dividir el numero en signo, exponente y parte fraccionaria

1 00000000000 1101000000000000000000000000000000000000000000000000

Como la parte del exponente es 00…0000 el número es desnormalizado

S=1

Exp=00000000000

Frac= 1101000000000000000000000000000000000000000000000000

2.Calcular BIAS y E

BIAS=2 ^ (11– 1) – 1 = 1,023

Como el número es desnormalizado del valor del exponente cambia:

E=1-BIAS=1-1,023=-1,022

3.Calcular la mantisa

Como el exponente es 0, la mantisa es:

M=0.frac=0.1101=0+2^-1+2^-2+ 2^-4=13/16

4.Se obtiene el numero con la fórmula del valor

V=-1^ (1) \*(13/16) \*2^-1,022=-0(La calculadora no llega a tal precisión)

# Otros casos

# Ceros

Existen dos tipos de 0:

Cero negativo (-0): Ocurre cuando el signo es 1(negativo). Su representación según el estándar seria la siguiente:

Ejemplo:

1 00000000 00000000000000000000000

Cero positivo (+0): Ocurre cuando el signo es 0(Positivo). Su representación según el estándar seria la siguiente:

Ejemplo:

0 00000000 00000000000000000000000

El formato de 64 bits seria el mismo caso, sólo que con 11 ceros en la parte del exponente y 52 ceros en la parte fraccionaria.

# Infinitos

Hay dos infinitos :

Infinito negativo (-∞): Ocurre cuando el signo es 1(Negativo). Su representación según el estándar seria la siguiente:

Ejemplo:

1 11111111 00000000000000000000000

Infinito positivo (+∞): Ocurre cuando el signo es 0(Positivo). Su representación según el estándar seria la siguiente:

Ejemplo:

0 11111111 00000000000000000000000

El formato de 64 bits seria el mismo caso, sólo que con 11 unos en la parte del exponente y 52 ceros en la parte fraccionaria.

# NaN(Not a Number)

Ocurre cuando el exponente es 111...11 y la parte fraccionaria es diferente de 000…00.

Ejemplo:

1 11111111 11010000000000000000000

# Pasar de decimal a binario

Para pasar de decimal a binario se necesita obtener en binario la parte entera y la parte decimal del número en cuestión.

Ejemplo:

Convierta el siguiente número decimal en binario según el estándar IEEE 754

5.5

1. Primero se debe de convertir la parte entera del número a binario

5=00000101

1. Posteriormente, se debe convertir la parte decimal del numero

0.5=2^-1=1/2=1

1. Se unen las dos partes y se rueda la coma hasta tener solo una cifra a la izquierda

00000101.100000000000000=1.01100000000000000

El exponente del valor(E) será pues las veces que se ha tenido que correr la coma:

E=2

1. Se calcula el valor

Exp= BIAS + E

El BIAS se calcula con 2^(n-1)–1, siendo n el número de bits de la parte exponencial:

BIAS=127

Exp=2+127=129= 100000001

El signo es positivo:

0 100000001 01100000000000000