Organización de Computadoras



Clase 3

# 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.

La diferencia principal entre la representación en el papel y su almacenamiento en la computadora, es que no se guarda coma alguna, se supone que está en un lugar determinado.

# Rango y Resolución

* Rango: diferencia entre el número mayor y el menor
* Resolución: diferencia entre dos números consecutivos

# Error en punto fijo (1)

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

|  |  |
| --- | --- |
|  |  |
| 5,01 | 5,02 |

5,015

* 5,01  No  5,015 se representa por 5,01
* 5,015 < No  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

* Que corresponden a los No marcados en rojo ó azul.

# Números en punto flotante

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”.

# Números en punto flotante (2)

Un número decimal “muy grande”: 976.000.000.000.000 se puede representar como:

9,76 x 10 14

Un número decimal “muy pequeño”:

0,0000000000000976

9,76 x 10 -14

Números en punto flotante (3)

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.

Esto permite tener un rango de números desde “muy pequeños” a “muy grandes” y pueden ser representados con pocos dígitos.

Números en punto flotante (4)

Veamos este mismo enfoque con números binarios:

Un número se puede representar de la forma:

± M x B E

Este número se puede almacenar en una palabra binaria con dos campos:

Mantisa M

Exponente E

Números en punto flotante (5)

La base B es implícita y no necesita almacenarse ya que es la misma para todos los números. Debemos almacenar M y E.

Se necesitan menos bits para almacenar M y E, que para almacenar el “número completo” en la base correspondiente.

Números en punto flotante (6)

M y E están representados en alguno de los sistemas en punto fijo que ya conocíamos como BSS, BCS, Ca2, Ca1, Exceso.

|  |  |
| --- | --- |
| exponente | mantisa |

La figura muestra un formato típico

# Ejemplo

Supongamos el siguiente formato en punto flotante

BSS BSS Mantisa 4 bits Exponente 4 bits entera entero

Determinar el rango y resolución

# Ejemplo 1

Máximo = 1111 x 21111 = 15 x 215

Mínimo = 0

Rango = [ 0,..,15x215]=[ 0,..,491520]

Resolución en el extremo superior

R = (15 – 14)x215 = 1 x 215

Resolución en el extremo inferior

R = (1 – 0)x20 = 1

# Ejemplo 2

Consideremos enteros de 8 bits y en BSS Calcular el rango y resolución:

Rango = [ 0,..,255 ]

Resolución en el extremo superior

R = 255 – 254 = 1

Resolución en el extremo inferior

R = 1 – 0 = 1

# Comparación

Si comparamos ambos ejemplos vemos:

el rango en punto flotante es mayor

la cantidad de combinaciones binarias distintas es la misma en ambos sistemas

28 =256

en punto flotante la resolución no es constante a lo largo del intervalo, como lo es en el segundo ejemplo.

# Conclusión

En el sistema de punto flotante el rango es mayor. Podemos representar números más grandes ó 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.

# Mantisa y exponente en Ca2

Ejemplo: supongamos el siguiente formato en punto flotante

Ca2 Ca2 Mantisa 4 bits Exponente 4 bits entera entero

Determinar el rango y resolución

# Mantisa y exponente en Ca2

Máximo = 0111 x 20111 = +7 x 2+7 Mínimo = 1000 x 20111 = -8 x 2+7

Rango = [-8 x 2+7,...,+7 x 2+7]

Resolución en el extremo superior

R = (7 – 6) x 27 = 1 x 27

Resolución en el origen

R = (1 x 2-8 – 0 ) = 1 x 2-8

# Mantisa fraccionaria

Ejemplo: supongamos el siguiente formato en punto flotante

BCS (ó MyS) Ca2

Mantisa 23 bits Exponente 8 bits fraccionaria entero

1 bit signo

Determinar el rango y resolución

# Mantisa fraccionaria

Máximo positivo

0 0,111..111 x 201111111=+(1-2-23).2+127

Mínimo positivo (0)

1. 0,000..001 x 210000000=+(2-23).2-128

Máximo negativo (0)

1. 0,000..001 x 210000000= - (2-23).2-128

Mínimo negativo

1 0,111..111 x 201111111= -(1-2-23).2+127

# Formato final

El formato anterior se puede representar

0 1 8 9 31

|  |  |  |
| --- | --- | --- |
| S | Exponente | Mantisa |

El mínimo negativo es

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

Veamos el siguiente ejemplo:

40x100 = 4x101 = 0,4x102 = 400x10-1

* Existen distintos valores de mantisa y exponente para representar un mismo número.
* 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 la normalización.
* Con el objetivo anterior, las mantisas fraccionarias se definen de la forma:

## 0,1dddddd.....ddd

* donde d es un dígito binario que vale 0 ó 1.
* Todas las mantisas empiezan con 0,1... Ejemplo: formato en punto flotante

BCS Exceso

23 bits Exponente 8 bits

Mantisa fraccionaria entero

1 bit signo

Normalizada

Determinar el rango y resolución

Máximo positivo

0 0,111..111 x 211111111=+(1-2-23).2+127

Mínimo positivo (0)

1. 0,100..000 x 200000000=+(0,5).2-128

Máximo negativo (0)

1. 0,100..000 x 200000000= - (0,5).2-128

Mínimo negativo

1 0,111..111 x 211111111= -(1-2-23).2+127

# Normalización

* El formato anterior se puede representar

0 1 8 9 31

|  |  |  |
| --- | --- | --- |
| S | Exponente | Mantisa |

* El máximo negativo (0) es

|  |  |  |
| --- | --- | --- |
| 1 | 00000000 | 1000…………………………………………00 |

# Bit implícito

* Como todos los números comienzan con 0,1 ¿es necesario almacenar el 1?
* siempre está presente !!!
* Si no lo almaceno, puedo “adicionar” un bit más a la mantisa. El bit no almacenado se conoce como bit implícito.

1 00000000 1000....................................000

# Recta numérica

* Sin bit implícito

-(1-2-23).2+127 -0,5.2-128 0 0,5.2-128 (1-2-23).2+127

* Con bit implícito

-(1-2-24).2+127 -0,5.2-128 0 0,5.2-128 (1-2-24).2+127

¿Cómo se escribe un No en punto flotante normalizado?

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

# ¿Cómo......? (2)

Ej. - 13,5 . Formato anterior

* 1) 1 1101,100..0=1 1101,100..0x20
* 2) 1 0,110110..0 x 24
* 3) 4 en Ca2=00000100

4 en Exceso=10000100

* Finalmente

# ¿Cómo...... ? (3)

* Sin bit implícito

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

* Con bit implícito

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

# Resolución – Error absoluto

* Resolución: es la diferencia entre dos representaciones sucesivas, y 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 absoluto y relativo

* Error Absoluto máximo  Resolución/2

* Error Relativo = EA/Número a representar

Simple precisión

1 8 23

|  |  |  |
| --- | --- | --- |
|  |  |  |

Doble precisión

1 11 52

|  |  |  |
| --- | --- | --- |
|  |  |  |

Mantisa: fraccionaria normalizada, con la coma después del primer bit que es siempre uno (1,) en M y S.

Exponente: representado en exceso

2n-1 - 1

|  |  |
| --- | --- |
|  | Simple Doble precisión |
| 1 | 1 |
| 8 | 11 |
| 23 | 52 |
| 32 | 64 |
| 127 | 1023 |
| 126 a +127 | –1022 a +1023 |
| -126 a ~2128 | 2-1022 a ~21024 |

Bits en signo

Bits en exponente

Bits en fracción

Total de bits

Exponente en exceso

Rango de exponente - Rango de números 2

# Ejemplo 1 en simple precisión

¿Qué valor representa el hexadecimal 3F800000?

0011 1111 1000 0000 0000 0000 0000 0000

01111111=127 en exceso 127 representa 0

00000000000000000000000=0

+ 1,0 x 20 = 1

# Ejemplo 2 en simple precisión

¿Qué valor representa el hexadecimal C0066666?

1100 0000 0000 0110 0110 0110 0110 0110

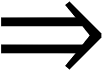
10000000=128 en exceso 127 representa 1

00001100110011001100110=0,05

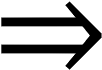
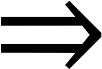
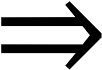
- 1,05 x 21 = -2,1

# Estándar IEEE 754

Casos especiales:

* E = 255/2047, M **≠** 0  NaN -Not a

Number-

* E = 255/2047, M = 0  Infinito
* E = 0, M = 0  Cero
* E = 0, M **≠** 0  Denormalizado
* ± 0,mantisa\_s-p 2–126
* ± 0,mantisa\_d-p 2–1022

# Operaciones aritméticas en pf

Sumar y restar

* Comprobar valores cero.
* Ajuste de mantisas (ajuste de exponentes).
* Sumar o restar las mantisas.
* Normalizar el resultado.

# Operaciones aritméticas… (2)

Multiplicar y dividir

* Comprobar valores cero.
* Sumar y restar exponentes.
* Multiplicar y 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
* PFI-PFO. Software en Descargas del sitio de cátedra Capítulo 8: Aritmética del computador (8.4., 8.5.)
* Stallings, W., 5º Ed.

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