Errores de redondeo

Lección 02

Dr. Pablo Alvarado Moya

CE3102 Análisis Numérico para Ingeniería
Área de Ingeniería en Computadores
Tecnológico de Costa Rica

II Semestre 2017

Contenido

Representaciones numéricas

Coma fija

Números codificados con coma fija

Las representaciones con **coma fija** son posicionales, donde el peso de cada bit en la representación es constante.

Número de N bits:

b_{N-1}	$\ldots$	b_5	b_4	b_3	b_2	b_1	b_0
$\uparrow$		$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$	$\uparrow$
2^{N-1}	$\ldots$	2^5	2^4	2^3	2^2	2^1	2^0
MSB							LSB

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.

Enteros sin signo

- Sea x un número entero sin signo de N-bits

$$x = \sum_{n=0}^{N-1} b_n 2^n$$

  donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x.
- El rango representable es entonces desde 0 hasta $2^N - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-1} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-1}

Ejemplo (1)

Ejemplo

Encuentre el equivalente decimal del número binario

$$(10100101)_2$$

Ejemplo (2)

Solución: El número de 8 bits

$$x = (10100101)_2$$

es equivalente al número en base 10

$$\begin{aligned} x &= 1 \times 2^7 + 1 \times 2^5 + 1 \times 2^2 + 1 \times 2^0 \\ &= 128 + 32 + 4 + 1 \\ &= 165 \quad (= 1 \times 10^2 + 6 \times 10 + 5) \end{aligned}$$

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a $1/M$.

Coma fija sin signo

- Sea x un número sin signo de N-bits

$$x = \frac{1}{M} \sum_{n=0}^{N-1} b_n 2^n$$

donde $b_n \in \{0, 1\}$ es el n-ésimo dígito de x y M es una constante de normalización elegida usualmente como 2^m.

- El rango representable es entonces desde 0 hasta $(2^N - 1)/M$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a $1/M$.
- El dígito b_{N-1} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de $2^{N-1}/M$.

Ejemplo (1)

Ejemplo

Encuentre el equivalente decimal del número binario

$$(10,100101)_2$$

Ejemplo (2)

Solución: El número de 8 bits

$$x = (10, 100101)_2$$

es equivalente al número en base 10

$$\begin{aligned} x &= 1 \times 2^1 + 1 \times 2^{-1} + 1 \times 2^{-4} + 1 \times 2^{-6} \\ &= 2 + \frac{1}{2} + \frac{1}{16} + \frac{1}{64} \\ &= \frac{128 + 32 + 4 + 1}{64} = \frac{1}{64} 165 \\ &= 2,578125 \quad (= 2 \times 10^0 + 5 \times 10^{-1} + 7 \times 10^{-2} + \ldots) \end{aligned}$$

Ejemplo (3)

- En este caso se tiene $M = 64$, con dos bits en la parte entera y 6 en la parte fraccionaria
- Nóte que $M = 2^f$ con f el número de bits en la parte fraccionaria

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-2} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-2}.

Complemento a dos

- La representación de N bits de un número entero con signo en complemento a dos está dada por

$$x = -b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n$$

  lo que permite representar números en el rango desde -2^{N-1} hasta $2^{N-1} - 1$.
- El dígito b_0 es el menos significativo (LSB, *least significant bit*) y su peso relativo es igual a uno.
- El dígito b_{N-2} es el más significativo (MSB, *most significant bit*) y tiene un peso relativo de 2^{N-2}.
- El último bit b_{N-1} codifica al signo.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.
- Es posible sumar varios números con signo, y siempre que el resultado **final** se encuentre en el rango de representación, es irrelevante si resultados intermedios producen desbordamiento.

Sumas con complemento a dos

- El uso del complemento a dos es el más difundido de todas las representaciones de números con signo.
- Es posible sumar varios números con signo, y siempre que el resultado **final** se encuentre en el rango de representación, es irrelevante si resultados intermedios producen desbordamiento.
- Por ejemplo, supóngase que se debe hacer la secuencia de operaciones $2 + 3 - 2$ con números de 3 bits. La secuencia de adiciones es entonces

$(x_i)_{10}$	$(x_i)_2$	$(\sum x_i)_2$	$(\sum x_i)_{10}$
2_{10}	010	010	2_{10}
3_{10}	011	101	-3_{10}
-2_{10}	110	011	3_{10}

donde el resultado intermedio 5 fue representado por el número -3, sin afectar el resultado final.

Coma fija con signo

- La representación de N bits de un número con signo en complemento a dos está dada por

$$x = \frac{1}{M}\left(-b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n\right)$$

con la constante de normalización M elegida usualmente como 2^m.

Coma fija con signo

- La representación de N bits de un número con signo en complemento a dos está dada por

$$x = \frac{1}{M}\left(-b_{N-1}2^{N-1} + \sum_{n=0}^{N-2} b_n 2^n\right)$$

con la constante de normalización M elegida usualmente como 2^m.

- El rango representable será entonces desde $-2^{N-1}/M$ hasta $(2^{N-1}-1)/M$.

Ejemplo (1)

Ejemplo

Encuentre la representación binaria del número decimal $x = -3,125$ con cinco bits para la parte fraccionaria y tres bits para la parte entera utilizando coma fija con complemento a dos.

Ejemplo (2)

Solución:
Con $f = 5$ bits para la parte fraccionaria se obtiene

$$M = 2^f = 32$$

por lo que el número entero a convertir es

$$32 \times -3,125 = -100$$

y finalmente $-100 = -128 + (16 + 8 + 4) = (10011100)_2$

El caso fraccionario

- Un caso frecuentemente utilizado permite representar números de valor absoluto igual o inferior a uno empleando $M = 2^{N-1}$ con lo que se obtiene

$$x = -b_{N-1} + \sum_{n=0}^{N-2} b_n 2^{n-N+1}$$

Coma flotante

Números codificados con coma flotante

- La representación en coma flotante permite ampliar el rango de representación numérica.
- La separación entre dos números adyacentes es variable: pequeña para números pequeños, grande para números grandes.
- Las representaciones de 32 y 64 bits más frecuentemente utilizadas han sido estandarizadas por la IEEE (estándar 754 en su versión original de 1985 y su más reciente versión de 2008).

Codificación según IEEE 754

Un número codificado con el estándar consiste en

- un bit de signo s,
- el exponente e con E bits y
- la mantisa m normalizada (fraccionaria) de M bits,

y se codifica como

s	Exponente e	Mantisa m

Codificación según IEEE 754

Un número codificado con el estándar consiste en

- un bit de signo s,
- el exponente e con E bits y
- la mantisa m normalizada (fraccionaria) de M bits,

y se codifica como

s	Exponente e	Mantisa m

De forma algebraica, el número representado es

$$x = (-1)^s \times (1, m) \times 2^{e-\text{bias}}$$

con

$$\text{bias} = 2^{E-1} - 1$$

Equivalencia decimal de número en coma flotante

Nótese que la mantisa se completa con un 1 bit *oculto* (en el sentido de que no se indica explícitamente en la representación), mientras que los bits especificados en la mantisa representan solo la parte fraccionaria.

Ejemplo (1)

Ejemplo

Indique cuál es la representación en coma flotante del número $10,125_{10}$ en un formato de 14 bits que utiliza $E = 6$ bits y $M = 7$ bits.

Ejemplo (2)

Solución:

Primero, el bias está dado por

$$\text{bias} = 2^{E-1} - 1 = 2^5 - 1 = 31$$

y para la mantisa

$$10,125_{10} = 1010,0010_2 = 1,0100010_2 \times 2^3$$

El exponente corregido se obtiene con

$$e = 3 + \text{bias} = 34_{10} = 100010_2$$

Finalmente, la representación del número es:

s	Exponente e	Mantisa m
0	100010_2	0100010_2

Ejemplo (1)

Ejemplo

Encuentre qué número decimal es representado por el código de coma flotante con $E = 6$ bits y $M = 7$ bits:

s	Exponente e	Mantisa m
1	011110_2	1000000_2

Ejemplo (2)

Solución:
El número representado está dado por

$$-1 \times 1,1000000_2 \times 2^{30-\text{bias}} = -1,1_2 \times 2^{-1} = -0,11_2 = -0,75_{10}$$

Estándar de coma flotante IEEE 754-2008

	Simple	Doble
Ancho de palabra	32	64
Mantisa	23	52
Exponente	8	11
Bias	127	1023
Rango	$2^{128} \approx 3,4 \times 10^{38}$	$2^{1024} \approx 1,8 \times 10^{308}$

Algunos números especiales en precisión simple

Nombre	s	e	m	Hex
			$11\ldots11$	$\texttt{FFFFFFFF}_H$
-NaN (Quiet)	1	$11\ldots11$	$\vdots$	$\vdots$
			$10\ldots01$	$\texttt{FFC00001}_H$
			$01\ldots11$	$\texttt{FFBFFFFF}_H$
-NaN (Signal)	1	$11\ldots11$	$\vdots$	$\vdots$
			$00\ldots01$	$\texttt{FF800001}_H$
$-\infty$	1	$11\ldots11$	$00\ldots00$	$\texttt{FF800000}_H$
-0	1	$00\ldots00$	$00\ldots00$	$\texttt{80000000}_H$
$+0$	0	$00\ldots00$	$00\ldots00$	$\texttt{00000000}_H$
$+\infty$	0	$11\ldots11$	$00\ldots00$	$\texttt{7F800000}_H$
			$00\ldots01$	$\texttt{7F800001}_H$
+NaN (Signal)	0	$11\ldots11$	$\vdots$	$\vdots$
			$01\ldots11$	$\texttt{7FBFFFFF}_H$
			$10\ldots01$	$\texttt{7FC00000}_H$
+NaN (Quiet)	0	$11\ldots11$	$\vdots$	$\vdots$
			$11\ldots11$	$\texttt{7FFFFFFF}_H$

Algunos números especiales en precisión doble

Nombre	s	e	m	Hex
			$11\ldots11$	$\texttt{FFFFFFFFFFFFFFFF}_H$
-NaN (Quiet)	1	$11\ldots11$	$\vdots$	$\vdots$
			$10\ldots01$	$\texttt{FFC0000000000001}_H$
			$01\ldots11$	$\texttt{FFF7FFFFFFFFFFFF}_H$
-NaN (Signal)	1	$11\ldots11$	$\vdots$	$\vdots$
			$00\ldots01$	$\texttt{FFF8000000000001}_H$
$-\infty$	1	$11\ldots11$	$00\ldots00$	$\texttt{FFF0000000000000}_H$
-0	1	$00\ldots00$	$00\ldots00$	$\texttt{8000000000000000}_H$
$+0$	0	$00\ldots00$	$00\ldots00$	$\texttt{0000000000000000}_H$
$+\infty$	0	$11\ldots11$	$00\ldots00$	$\texttt{7FF0000000000000}_H$
			$00\ldots01$	$\texttt{7FF0000000000001}_H$
+NaN (Signal)	0	$11\ldots11$	$\vdots$	$\vdots$
			$01\ldots11$	$\texttt{7FF7FFFFFFFFFFFF}_H$
			$10\ldots01$	$\texttt{7FF8000000000000}_H$
+NaN (Quiet)	0	$11\ldots11$	$\vdots$	$\vdots$
			$11\ldots11$	$\texttt{7FFFFFFFFFFFFFFF}_H$

Error de redondeo

Se produce al utilizar representaciones numéricas incapaces de representar todas las cifras significativas del número a representar. Se produce porque

- El **rango** de cantidades representables es limitado. Fuera del rango representable ocurre el error de **desbordamiento** (*overflow*)
- Número **finito** de números representables en un rango. Al utilizar el número representable más cercano se produce el **error de cuantificación**. Este número se puede asignar por redondeo o por corte.
- Con coma flotante, intervalo Δx entre números aumenta conforme los números crecen en magnitud

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

- Si se utiliza redondeo

$$\frac{\mathscr{E}}{2} \geq \frac{|\Delta x|}{|x|}$$

Epsilon de formato

- En coma flotante, sea Δx el intervalo entre representaciones válidas alrededor de un valor x.
- Si se utiliza corte, el epsilon $\mathscr{E}$ del formato se define como el menor número que cumple con

$$\mathscr{E} \geq \frac{|\Delta x|}{|x|}$$

- Si se utiliza redondeo

$$\frac{\mathscr{E}}{2} \geq \frac{|\Delta x|}{|x|}$$

- En general se cumple $\mathscr{E} = 2^{1-M}$ donde M es el número de bits de la mantisa.

Información sobre tipos en C++

- STL (Standard Template Library)
- `<limits>`
- `std::numeric_limits<float>::epsilon()`
- `std::numeric_limits<double>::max()`
- Ver ejemplo `eps.cpp` ($M_{\texttt{float}} = 23$, $M_{\texttt{double}} = 52$)

Información sobre tipos en C++

- STL (Standard Template Library)
- `<limits>`
- `std::numeric_limits<float>::epsilon()`
- `std::numeric_limits<double>::max()`
- Ver ejemplo `eps.cpp` ($M_{\texttt{float}} = 23$, $M_{\texttt{double}} = 52$)
- ¿Por qué resultado da $\mathscr{E} = 2^{-M}$ y no $\mathscr{E} = 2^{1-M}$?

Manipulaciones aritméticas

Se brindarán ejemplos en base 10 por simplicidad, pero recuérdese que el computador utiliza números en base 2.

Suma
Coma flotante

- Se toma número con menor exponente
- Mantisa se modifica para igualar exponente del otro número (Alineación de la coma decimal)
- Se realiza la suma
- En caso necesario, se renormaliza número

Ejemplo: Suma (1)

Ejemplo

Sume los números $0,157 \times 10^{1}$ y $0,44 \times 10^{-1}$, asumiendo que el sistema numérico utilizado puede representar 3 cifras significativas.

Ejemplo: Suma (2)

Solución:

$$\begin{array}{r} 0,157 \times 10^1 \\ 0,0044 \times 10^1 \\ \hline 0,1614 \times 10^1 \end{array}$$

$$\downarrow$$

$$0,161 \times 10^1$$

Resta
Coma flotante

Idéntico al caso de la suma

- Se toma número con menor exponente
- Mantisa se modifica para igualar exponente del otro número (Alineación de la coma decimal)
- Se realiza la resta
- En caso necesario, se renormaliza número

Ejemplo: Resta (1)

Ejemplo

Reste los números 10 y $9,99$, asumiendo que el sistema numérico utilizado puede representar 3 cifras significativas.

Ejemplo: Resta (2)

Solución:

$$\begin{array}{r} 0,10 \quad \times 10^2 \\ 0,0999 \times 10^2 \\ \hline 0,0001 \times 10^2 \end{array}$$

$$\downarrow$$

$$0,100 \times 10^{-1}$$

Problema: si los dos números a restar son similares, en la representación numérica aprecen cifras que **no** son significativas.

Suma y resta
Coma fija

En representaciones de coma fija, las operaciones de suma y resta tienen la coma decimal alineada, lo mismo que el resultado, así que símplemente se realizan las operaciones sobre cada dígito con acarreo.

Multiplicación y división

Coma flotante

- Se suman (o restan) exponentes
- Se multiplican (o dividen) las mantisas de n dígitos
- Se normaliza el resultado (y se redondea si es necesario)

Nótese que el producto de dos mantisas de n dígitos produce $2n$ dígitos.

Ejemplo: Multiplicación (1)

Ejemplo

Multiplique los números 136,3 y 0,06423, asumiendo que el sistema numérico puede representar cuatro cifras significativas.

Ejemplo: Multiplicación (2)

Solución:

$$\begin{array}{ll} 0,1363 & \times 10^{3} \\ 0,6423 & \times 10^{-1} \\ \hline 0,08754549 & \times 10^{2} \end{array}$$

$$\downarrow$$

$$0,8754549 \times 10^{1} \approx 8,754 \text{ (corte)}$$

Procesos acumulativos

En procesos acumulativos de las formas:

$$y = \sum_i x_i \qquad\qquad y = \prod_i x_i$$

implementados iterativamente, los redondeos de cada resultado parcial introducen errores crecientes con cada paso.

Tarea

Ejemplo

Realice un programa en C++ que acumule (aditivamente) 100 000 veces los números 1 y $0,00001$ en precisiones simple y doble.

Suma de números grandes y pequeños

La suma de un número grande y otro pequeño, con una diferencia en órdenes de magnitud mayor al número de cifras significativas, no produce **ningún** efecto.

$$\begin{array}{r} 0,4000 \times 10^4 \\ 0,0000001 \times 10^4 \\ \hline 0,4000001 \times 10^4 \end{array}$$

$$\downarrow$$

$$0,4000 \times 10^4$$

¡Advertencia!

En sumas de gran número de términos, se debe procurar sumar primero los términos pequeños y por último los grandes, de modo que se minimice el efecto anterior.

Cancelación por resta

Redondeo inducido cuando se restan dos números de coma flotante casi iguales.

Cancelación por resta

Ejemplo

Cálculo de las soluciones de la ecuación cuadrática $ax^2 + bx + c = 0$:

$$x_{1,2} = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

con $b^2 \gg 4ac$.

En el ejemplo anterior el numerador tiende a cero.

Ejemplo

Verifique el problema anterior para $a = 1$, $b = 3000,001$ y $c = 3$, con un programa en C++, si se sabe que $x_{1,2} = -0{,}001| - 3000$.

Solución cuadrática alternativa

La ecuación cuadrática:

$$ax^2 + bx + c = 0$$

tiene una formulación alternativa de solución que minimiza error de cancelación por resta:

$$x_{1,2} = \frac{-2c}{b \pm \sqrt{b^2 - 4ac}}$$

Presencia de errores de redondeo

1. Series: p.ej. $\sum_{i=0}^{N} a_i x^i = a_0 + a_1 x + a_2 x^2 + \ldots$
2. Productos punto: $\sum_{i=0}^{N} x_i y_i = x_0 y_0 + x_1 y_1 + x_2 y_2 + \ldots$

- Reducción de error de redondeo: usar precisión extendida
- Existen técnicas de reducción de error, pero dependen de cada caso particular

Resumen

Este documento ha sido elaborado con software libre incluyendo LaTeX, Beamer, GNUPlot, GNU/Octave, XFig, Inkscape, LTI-Lib-2, GNU-Make y Subversion en GNU/Linux