

Tema 5. Aritmètica d'enters i coma flotant

Estructura de Computadors (EC)

Rubèn Tous

rtous@ac.upc.edu

Computer Architecture Department
Universitat Politècnica de Catalunya



UNIVERSITAT POLITÈCNICA
DE CATALUNYA
BARCELONATECH

Índex

1 5.4 Coma flotant: representació

Índex

1 5.4 Coma flotant: representació

5.4.1 Representació en coma fixa

- Extensió del sistema numèric posicional amb potències negatives de la base (dreta de la coma).
- Nombre fix de dígit (part entera i part fraccionària).
- En binari utilitzarem n bits per a la part entera i m bits per a la fraccionària.

eeee eeee eeee eeee eeee eeee eeee.ffff

5.4.1 Representació en coma fixa

$$X = X_{n-1} \dots X_1 X_0 X_{-1} X_{-2} \dots X_{-m}$$

$$x = \sum_{i=-m}^{n-1} X_i * 2^i$$

Exponents part fraccionària:

3	2	1	0	, -1	-2	-3
e	e	e	e	, f	f	f
8	4	2	1	, 0,5	0,25	0,125

$$2^{(-3)} = 1/2^3$$

5.4.1 Representació en coma fixa

Exemple 2,125 amb eeee ... eeee eeee.ffff:

$$2 = \dots 0000 \ 0000 \ 0010$$

$$0,125 = 1/8 = 1/2^3 = 2^{-3} = ,001$$

Resultat exemple: $\dots 0000 \ 0000 \ 0010,001$

(exacte, sense error)

5.4.1 Representació en coma fixa

- Inconvenient: S'ha de fer un compromís entre precisió i rang.
- Avantatge: Podem aplicar directament aritmètica entera.

5.4.1 Representació en coma fixa

No sempre un nombre fraccionari decimal exacte es pot representar sense error en binari. Exemple:

Exemple 23,42 amb ...eeee eeee eeee.ffff:

23 = 1 0111

0,42 * 2 = 0,84 -> 0

0,84 * 2 = 1,68 -> 1

0,68 * 2 = 1,36 -> 1

0,36 * 2 = 0,72 -> 0

0,72 * 2 = 1,44 -> 1

0,44 * 2 = 0,88 -> 0

...

0,011010... Trunquem (tenim 4 bits):

0,0110

5.4.2 Representació en coma fixa

Què hem codificat exactament?

$$1\ 0111 = 23$$

$$0,0110 = 0 \star (1/2^1) + 1 \star (1/2^2) + 1 \star (1/2^3) + 0 \star (1/2^4) = 0 + 0,25 + 0,125 + 0 = 0,375$$

Hem codificat el 23,375

Hi ha un error ($|23,42 - 23,375| = 0,045$)

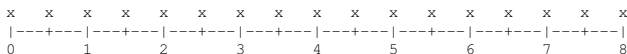
5.4.2 Error

- Error de codificació = $|\text{valor real} - \text{valor codificat}|$
- En l'exemple anterior: $|23,42 - 23,375| = 0,045$
- Més endavant parlarem d'arrodoniments.

5.4.2 Error

Error uniforme, independent de la magnitud del nombre:

Amb un bit de part fraccionaria:



5.4.3 Aritmètica en coma fixa

Igual que en aritmètica entera:

$$2,5 + 2,5 = 5$$

$$\begin{array}{r} 10,1 \\ +10,1 \\ \hline 101,0 \end{array}$$

5.4.3 Aritmètica en coma fixa

$$2,5 * 2,5 = 6,25$$

$$\begin{array}{r}
 10,1 \\
 *10,1 \\
 \hline
 10\ 1 \\
 000 \\
 101 \\
 \hline
 1100\ 1 = 110,01 = 6,25
 \end{array}$$

5.4.4 Representació en coma flotant

Mètode de representació d'un subconjunt dels reals amb un compromís entre rang precisió:

$$x = + / - m * b^e$$

mantissa(m): fraccionària i normalitzada (un dígit significatiu abans de la coma)

base(b)

exponent(e)

Exemple: $2,3375 \times 10^1$ representa el nombre 23,375.

5.4.4 Representació en coma flotant

En binari:

signe(+/-): Signe i magnitud. 1 bit (1, negatiu; 0, positiu)

mantissa(m): fraccionària i normalitzada, amb bit ocult

base(b): 2

exponent(e): representat en excés $2^{e-1} - 1$.

Exemple: $1,01110110 * 2^4$ representa el nombre 10111,0110

5.4.5 Formats IEEE-754 (simple/doble)

(IEEE = Institute of Electrical and Electronics Engineers)

- IEEE 754 standard (1985). Utilitzat a la majoria de computadors (hi ha una versió 2008).
- 2 formats molt usats: single-precision i double-precision.
- Single (32 bits): 1 bit de signe, 23 bits de mantissa + bit ocult, i 8 bits d'exponent excés 127.
- Double (64 bits): 1 bit de signe, 52 bits de mantissa + bit ocult, i 11 bits d'exponent excés 1023.
- Existeixen més formats.
- Quatre formes d'arrodoniment. Recomanat al més proper o parell.
- Cinc excepcions: Divisió per zero, overflow, underflow, invàlid i inexacte

5.4.5 Formats IEEE-754 (simple/doble)

Single-precision:

s | eee eeee e | mmm mmmm mmmm mmmm mmmm mmmm

signe(+/-): 1 bit (1, negatiu; 0, positiu)

mantissa(m): 23 bits. Fraccionària, normalitzada amb bit ocult

base(b): 2.

exponent(e): 8 bits representat en excés $2^{e-1} - 1 = 127$.

5.4.6 Representació en coma flotant. Conversió a/de decimal

Exemple de codificació: -23,375

① $23 = 16 + 7 = 1\ 0111$ (part entera)

② $0,375 =$ (part fraccionària)

$$0,375 * 2 = 0,75$$

$$0,75 * 2 = 1,5$$

$$0,5 * 2 = 1,0$$

$$0,0 * 2 = 0$$

...

$$0,375 = 0,0110000\dots$$

5.4.6 Representació en coma flotant. Conversió a/de decimal

- 1 Normalitzem i calculem l'exponent:

10111,011000...

-> desplacem 4 posicions

= 1,0111011000... * 2⁴

- 2 Calculem l'exponent en Excés 127:

4+127 = 131 = 128 + 3 = 1000 0011

- 3 Juntem les parts:

1|100 0001 1|011 1011 0000 0000 0000 0000

= 0xC1BB0000

5.4.6 Representació en coma flotant. Conversió a/de decimal

Exemple de decodificació:

① 0xC1BB0000 =
 1|100 0001 1|011 1011 0000 0000 0000 0000

② Càlcul exponent:

$$131 - 127 = 4$$

③ Mantissa:

$$\begin{aligned} &1,01110110000\dots * 2^4 \\ &= 10111,011000\dots \end{aligned}$$

④ Part entera:

$$\text{Part entera: } 1\ 0111 = 23$$

⑤ Part fraccionària:

$$0,011 = 0,25 + 0,125 = 0,375$$

⑥ Error = 0

5.4.6 Representació en coma flotant. Conversió a/de decimal

Exemple amb error: -23,45

① $23 = 16 + 7 = 1\ 0111$ (part entera)

② $0,45 =$ (part fraccionària)

$$0,45 * 2 = 0,9$$

$$0,9 * 2 = 1,8$$

$$0,8 * 2 = 1,6$$

$$0,6 * 2 = 1,2$$

$$0,2 * 2 = 0,4$$

$$0,4 * 2 = 0,8$$

$$0,8 * 2 = 1,6$$

...

1 0111,01 1100 1100 1100 1100 1...

5.4.6 Representació en coma flotant. Conversió a/de decimal

1 Normalitzem i arrodonim:

1,011 1011 1001 1001 1001 1001 | 1001... * 2^4

1,011 1011 1001 1001 1001 1010 * 2^4
(al més proper o parell)

2 Calculem l'exponent en Excés 127:

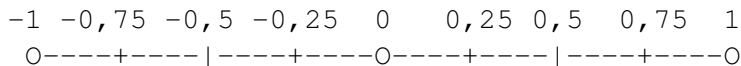
$4+127 = 131 = 128 + 3 = 1000\ 0011$

3 Juntem les parts:

1|100 0001 1|011 1011 1001 1001 1001 1010
= 0xC1BB999A = -23,449999

4 Error = 0.000001

5.4.7 Arrodoniments definits per l'IEEE 754



zero	->	->	->	<-	<-	<-
+inf	->	->	->	->	->	->
-inf	<-	<-	<-	<-	<-	<-
prox	<-	->	->	<-	<-	->

Per defecte arrodoniment al proper o parell (en cas d'empat aproxima al nombre parell).

5.4.7 Arrodoniments definits per l'IEEE 754

Exemples:

$0000,01 = 0000$ (al més proper)

$0000,11 = 0001$ (al més proper)

$0000,100\dots001 = 0001$ (al més proper)
(per això cal un bit extra, sticky bit)

$0000,1 = 0000$ (al parell)

$0001,1 = 0010$ (al parell)

5.4.8 Codificacions especials: Zero, Inf, Denorm, Nan

- Quin valor codifica 0x00000000 en base al que hem explicat fins ara?
- Exponent 0 codifica el $0 - 127 = -127$.
- Bit ocult: 1,...
- Per tant $0x00000000 = 2^{-127}$.
- Aleshores com codifiquem el 0?

5.4.8 Codificacions especials: Zero, Inf, Denorm, Nan

- Fixarem que el zero es representa com 0x00000000 (+0) o 0x80000000 (-0).
- Exponent i mantissa = zero.
- Perdem la possibilitat d'utilitzar l'exponent -127.
- El 0x80000000 (-0) sempre serà a conseqüència d'haver-se produït un *underflow*.

5.4.8 Codificacions especials: Zero, Inf, Denorm, Nan

Altres codificacions especials:

- Denormals: Exponent tot 0's i la mantissa diferent de zero. En parlarem més endavant.
- +/- infinit: Exponent tot 1's i la mantissa tot zeros.
- NaNs (Not a Number): Exponent tot 1's i la mantissa diferent de zero (ex: Arrel quadrada d'un negatiu).
- Perdem la possibilitat d'utilitzar l'exponent 128.

5.4.8 Codificacions especials: Zero, Inf, Denorm, Nan

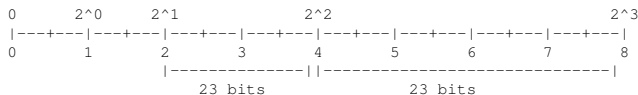
Taula resum:

IEEE 754

exponent	mantissa	significat
0s	0s	+/-zero
0s	$\neq 0$ s	denormals
1s	0s	+/- ∞
1s	$\neq 0$ s	NaN

5.4.9 Compromís rang/precisió

- No hi ha la mateixa 'quantitat' de números entre $2^1 - 2^2$ i $2^2 - 2^3$.



- Sempre el mateix nombre de bits (2^{23} números).
- Més *precisió* quan el número és proper a zero.
- Tots els nombres amb el mateix exponent estan a la mateixa distància.

5.4.9 Compromís rang/precisió

- Error màxim?
- Dependrà de l'exponent en concret.
- Distància més gran entre un nombre que no puguem representar i la seva representació?
- La meitat de la distància que hi ha entre dos nombres per aquell exponent:

$$\begin{array}{ccc}
 1,000000000 & & 1,000000001 \\
 | \text{-----} + \text{-----} | \\
 0,000000001 = 2^{-23}
 \end{array}$$

5.4.9 Compromís rang/precisió

$$\text{Error absolut} = \frac{2^{-23} * 2^{EXP}}{2} = 2^{-24} * 2^{EXP} = 2^{EXP-24}$$

5.4.10 Overflow i Underflow

Suposant només nombres normalitzats. Recta dels reals (rang):

$$-N_{max} \cdots -N_{min} \dots 0 \dots N_{min} \dots N_{max}$$

- Rang excés 127: -127..128
- Exponent més gran: 11111110 = 127 (11111111 reservat per a +inf/-inf i NaN)
- Exponent més petit: 00000001 = -126 (00000000 reservat per al zero i els denormals)

5.4.10 Overflow i Underflow

$$N_{max} = 1,1111...11111 * 2^{127} = (2^{24} - 1) * 2^{-23} * 2^{127} = (2 - 2^{-23}) * 2^{127}$$

$$N_{min} = 1,0000...00000 * 2^{-126} = 2^{-126}$$

5.4.10 Overflow i Underflow

- Si valor absolut $> N_{max}$: **overflow** (massa gran o petit).
- Si valor absolut $< N_{min}$: **underflow** (massa proper a zero).
- Els denormals permeten cobrir el 'underflow gap'.
- Al resultat d'una operació amb denormals l'anomenem *gradual underflow*

5.4.11 Denormals

- Nombres més propers a zero que els nombres normalitzats.
- No hi ha bit ocult i l'exponent és -126 (tot i que codifiquem com 00000000, que seria -127 en excés).

5.4.11 Denormals

Exemple denormal:

$0x00400000 = 0|000\ 0000\ 0|100\ 0000\ 0000\ 0000\ 0000\ 0000$

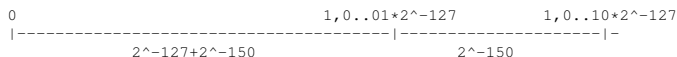
$= 0,1 * 2^{-126}$

5.4.11 Denormals

Denormal més petit: $2^{-23} * 2^{-126} = 2^{-149}$

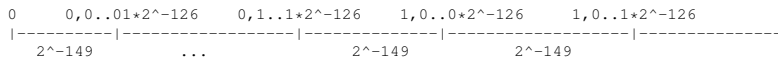
5.4.11 Denormals

- Per què els denormals? (EXPLICACIÓ OPCIONAL)
- Sense denormals queda un forat:



5.4.11 Denormals

- Amb denormals i exponent -126:



5.4.11 Denormals

- Distància amb els normalitzat més petit?

Normalitzat més petit: $1,0..0 * 2^{-126} = 10,0..00 * 2^{-127}$

Denormal més gran: $0,1..1 * 2^{-126} \quad - \quad 1,1..10 * 2^{-127}$

 $0,0..010 * 2^{-127} =$

$1,0..0 * 2^{-149}$

- A quina distància estan els dos normalitzats més petits?
 2^{-149} .

5.4.11 Denormals

- Què passaria si l'exponent fos -127?

Normalitzat més petit: $1,0..0 * 2^{-126} = 10,0..00 * 2^{-127}$
 Denormal més gran: $- 0,1..11 * 2^{-127}$

 $1,0..01 * 2^{-127}$

- Entre el denormal més gran i el normalitzat més petit queda un forat (aprox. 2^{-127}).
- Els denormals queden més comprimits i prop del zero (distància 2^{-150} entre ells)