ARITMETICA ENTERS I COMA FLOTANT

Operació	carry / borrow	ovf. mat	ovf. enters
$a + b = c$	$c < a$ (mat.)	carry	sig(a) == sig(b) i sig(a) ≠ sig(c)
$a - b = c$	$a < b$ (mat)	borrow	sig(a) ≠ sig(b) i sig(a) ≠ sig(c)

Multiplicació

Overflow hi haurà quan el resultat supera la grandària dels operands (Tots iguals) // En cas dels operands.

En cas dels $\mathbb{Z}$: 1. Ficar-los en positiu a tots

2. Multiplicar normal

3. Canviar el signe si pertoca.

if (LSB del MR == 1) Pau activa al MD
MD<<1
MR>>1 (< >)

Saber si hi ha ovf en $\mathbb{Z}$: → Bit més pes de \$lo ≠ Tots els bits de \$hi

1. Fes "sra \$lo, \$lo, 31" i "xor \$t0, \$hi, \$lo"

2. Si \$t0 == 0 sig que ezen no hi ha ovf.

\$t1
\$t0
\$hi □ lo

Penso que si MSB de \$lo és '1' sig que és negatiu i si \$hi no és tot 1's alguna cosa.

Divisió

1. Buscar potència de 2 més gran

2. Restar i repetir fins que quedi 1 o 0.

"div/divu Ra, Rb" → mflo Rd Quocient ; → mfhi Rd Residu

R-R-D R=R+D

DR . R Q (> <)
Q 1 1

D>>1 ← (Q<<1) 1

D[Hi] = Y D[Lo] = 0
R[Hi] = 0 R[Lo] = X

Penso que això són ~~...~~ que fem petit com dividir

▽ ▽ El negatiu més gran ÷ '-1' ⇒ OVERFLOW.

Representació Coma fixe

$23'42 \to 10111,0110|10 \to 2^{-2} + 2^{-3} = 0'25 + 0'125 = 0'375$ (Truncar)

$0'42 \times 2 = 0'84$
$0'84 \times 2 = 1'68$
$0'68 \times 2 = 1'36$
$0'36 \times 2 = 0'64$ (Truncar)
$0'64 \times 2 = 1'28$
$0'28 \times 2 = 0'56$

$\mathcal{E} = |23'42 - 23'375| = 0'045 = \mathcal{E}$

Representació Coma flotant

$\boxed{x = \pm m * b^e}$ (m) mantissa : fraccionària i normalitzada (1 digit abans de la coma).

$23'375 \to 10111'0110 \Rightarrow 1'01110110$ (4) $m = 0111 0110$, $e = 4$. El representem en excés (-127)

$e = 4 - 127 = -3 \equiv 1000\,0011$

Single precission (32 b)

S	e	m
signe	exp	mantissa
1B	8B	23B

m serà normalitzat, bit ocult.

Exemples

$-23'375$

1) Signe: 1 pq. negatiu

2) mantissa: $10111'011$

3) normalitzar $1'0111011 \times 2^4$

4) Exp en excés: $4+127 = 131 = 128+2+1 =$
$= 1000\ 0011$

[1|100 0001 1|011 1011 0000 0000 0000 0000]

$-23'45$

1) Signe: 1 pq. és neg.

2) mantissa: $10111'011100\,110011\ldots$

$0'45 \times 2 = 0'90$
$0'90 \times 2 = 1'80$
$\boxed{0'80 \times 2 = 1'60}$
$0'60 \times 2 = 1'20$
$0'20 \times 2 = 0'40$
$0'40 \times 2 = \boxed{0'80}$

3) normalitzar:

$1,0111\ 1011\ 1001\ 1001\ 1001\ 1001 \ldots$ Arrodonir $\rightarrow 1010$

4) Expo $= 1000\ 0011$

[1|100 0001 1|011 1011 1001 1001 1001 1010] $\Rightarrow$ 0xC1BB999A

Arrodoniment

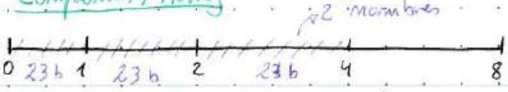
$100'1 \rightarrow 100'0$ $\quad 101'1 \rightarrow 110'0$
$4'5 \rightarrow 4$ $\quad 5'5 \rightarrow 6$

Anem al parell més pròxim.

Nombres especials

	Exponent	Mantisa	
2^{127}	zero	zero	$\Rightarrow 0$
	zero	$\neq$ zero	$\Rightarrow$ denormals
2^{128}	tot uns	zero	$\Rightarrow \pm\infty$
	tot uns	$\neq$ zero	$\Rightarrow$ NaN

Compromís / Rang

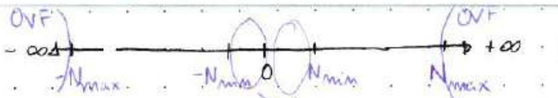


Precisió quan més a prop del 0.

L'error màxim és 2^{exp-24}

$N_{max} = 1'11\ldots1 \times 2^{127}$

$N_{min} = 1'00\ldots0 \times 2^{-126}$



Underflow i si fem servir un Denormal és "Gradual Underflow".

Denormals

$\rightarrow$ Exponent: $\emptyset$
Mantissa: $\neq \emptyset$ } DENORMAL $\rightarrow$ NO BIT OCULT, Exponent -126

NaN té -127 d'exponent

$0x00000001 \Rightarrow 0'\underbrace{0000\ 0000\ 0000\ 0000\ 001}_{23b} \times 2^{-126}$

Suma (i Resta)

1) ALINEAR EXPONENT: El més petit cap al més gran i ajustar mantissa

2) SUMAR O RESTAR MANTISSA: Com que són N hauen de fer més gran ± més petit.

3) NORMALITZAR: A vegades no cal

4) ARRODONIR

5) AJUSTAR SIGNE: A vegades no cal

Exemples

1) \$f2 = 0x4116 0000   \$f4 = 0x3F400000   (Suma)

⊕ \$f2 = 0|100 0001 0|001 0110 0000 0000 0000 0000 | exp = 128+2 = 130 − 127 = 3 = exp

⊕ \$f4 = 0|011 1111 0|100 0000 0000 0000 0000 0000 | exp = 64+32+16+8+4+2 = 126 − 127 = −1 = exp (+4)

• Hem d'alinear els exponents així que $-1 \xrightarrow{+4} 3$

$1'100\,0000\,0000\,0000\,0000\,0000 \times 2^{-1} \Rightarrow 0'0001\,1000\,0000\,0000\,0000\,000 \times 2^{3}$

• Ara fem la suma.

1'0010 110 0000 0000 0000 0000
⊕ 0'0001 100 0000 0000 0000 0000
1'0100 010 0000 0000 0000 0000

→ 0|1 00|000 1|0|01 0|00 10|0000|0000|0000
4 1 2 2 0 0 0

$\boxed{0x4122000}$

2) \$f2 = 0x417AC000   \$f4 = 0x3F140000   (Suma)

15'671875   0'578125

⊕ \$f2 = 0|100 0001 0|111 1010 1100 0000 0000 0000 | exp = 130 − 127 = 3 = exp

⊕ \$f4 = 0|011 1111 0|001 0100 0000 0000 0000 0000 | exp = 126 − 127 = −1 = exp (+4)

• Alineem i fem suma

1'1111 0101 1000 0000 0000 0000 $\times 2^3$
⊕ 0'0001 0010 100 0000 0000 0000 00 $\times 2^3$
10'0000 1000 0000 0000 0000 0000 $\times 2^3$

• Veiem que hem de normalitzar exp i perdem el LSB

1'0000 0100 0000 0000 0000 000 $\times 2^4$ ⇒ [x − 127 = 4 | = 128+3]

0|1000 001|1.0000 1 0000 0000 0000 0000 $\boxed{0x4182000}$
4 1 8 2 0 0 0 0

4) \$f2 = 0x40800000   \$f4 = 0xBE800009   (Resta)

⊕ \$f2 = 0|1000 0001|000 0000 0000 0000 0000 0000 | exp = 129 − 127 = 2 = exp

⊖ \$f4 = 1|0111 1101|000 0000 0000 0000 0000 1001 | exp = 125 − 127 = −2 = exp (+4)

23 − 4 = 19 '0'

• Alineem i restem

1'0000 0000 0000 0000 0000 0000 000 $\times 2^2$
⊖ 0'000 1000 0000 0000 0000 0000 1001 $\times 2^2$
0'11 0111 1111 1111 1111 1111 0111 $\times 2^2$

→ Veiem que no ens queda '1' abans la mantissa així que ajustem

1'11 0111 1111 1111 1111 1111 0|111 $\times 2^{1}$ → exp = 1 + 127 = 128

• El nombre aquest s'ha d'arrodonir pq num post a LSB = 1 i el de després tmb. així que UP. 1'11...11 $\times 2^1$

• Analitzem que el terme "⊕ − ⊖" = "⊕ + ⊕" així que resultat positiu.

0|100 0000 0|110 1111 1111 1111 1111 1111 $\boxed{0x406FFFFF}$
4 0 6 F F F F F

3) \$f2 = 0xC076C000   \$f4 = 0x3ECA8000   (Suma)

⊖ f2 = 1|10000000|11101101100000000000000 | exp = 128 − 127 = $\underline{1 = exp}$

⊕ f4 = 0|01111101|10010101000000000000000 | exp = 125 − 127 = $\underline{-2 = exp}$ (+3)

∇ Veiem que exp 2 > exp 1 i que f2 és negatiu així que fem |f2| − |f4| = |f6| (f6 neg)
Com si fossin enters normals

• Alinien els exponents

$$
\begin{array}{r}
1'11101101100000000000000 \times 2^1 \\
\ominus\ 0'00110010101000000000000 \times 2^1 \\
\hline
1'10111010111000000000000 \times 2^1
\end{array}
$$

• Com que no cal arrodonir, ni ajustar expo ja podem ensamblar. Recordar negatiu.

1|100 0000 0|101 1101 0111 0000 0000 0000
C 0 5 D 7 0 0 0

$\boxed{0xC05D7000}$

Guard / Round / Sticky bits

Pot ser que la mantissa resultat després d'operar superi els 23 b. En IEE754 s'ha d'arrodonir així que farem ús de "3" bits especials (Posteriors als 23 de mantissa)

Guard Bit (G): Primer bit posterior als 23 de mantissa.

Round Bit (R): Segon bit posterior als 23 de mantissa.

Sticky Bit (S): Bit que s'aconsegueix de fer un OR amb la resta.

Si algun bit és '1' ⇒ S = 1. $1'\overbrace{11\ldots0}^{23\text{ b}}\,|\,10001001$ Guard, Round, Sticky (S)

G	R	S	Acció
0	*	*	Tallar
1	0	0	Arrodonir Parell
1	0	1	Arrodonir UP
1	1	0	Arrodonir UP
1	1	1	Arrodonir UP

Multiplicació

1) MULTIPLICAR MANTISSES

2) SUMAR EXPONENT

3) RE-NORMALITZAR

4) ARRODONIR

5) AJUSTAR SIGNE

Pau Bru Ribes

(5.11). a) X = 101000 ,, Y = 010011 Multiplicació.

iter	Passos	P	MD <<	MR >>
ini		000000 000000	000000 101000	010011
1		000000 101000	000001 010000	001001
2	Suma	0000 0111 1000	0000 10100000	000100
3		0000 1111000	00010100000	000010
4		0000 1111000	0010100000000	000001
5	Suma	0010 1111 1000	1010000000	000000

0101 0000
+ 101000
0111 1000

0000 0111 1000
+ 10 1000 0000
0010 1111 1000

(5.13). a) X = 101000 ,, Y = 010011 Divisió

iter	Passos	Q	Divisor >>	Dividend ; Residu <<
ini	Q = 0 $D_{11:6}$ = Y $D_{5:0}$ = 0 $R_{11:6}$ = 0 $R_{5:0}$ = X	000 000	010011 000000	000000 101000
	D = D>>1 R = R−D	000000	001001 100000	

Fet a OneNote explicat

(5.23). Num decimals a coma flotant simple precissió. Expresa en hexa. Calc error.

a) 340

1) Signe: 0

2) Mantissa: 101010100'0 (340, 8)

3) Normalitzar:
$1'01010100 \times 2^8$

4) Cod. expo:
8 + 127 = 135 = 10000111

5) Cod mantissa: 23−8 = 15 zeros
01010100'000000000000000

6) Ensamblar
0|1000 0111|0101 0100 0000 0000 0000 0000
4 3 A A 0 0 0 0

$\boxed{0x43AA0000}$

b) 44'4

1) Signe: 0

2) Mantissa: 101100'0110
0'4x2 = 0'8
0'8x2 = 1'6
0'6x2 = 1'2
0'2x2 = 0'4

3) Normalitzar:
$1'01100 \times 2^5$

4) Cod Expo
5+127 = 132 = 1000 0100

5) Cod. Mantissa: 23−6 = 18 num
Periode de 4.

01100'0110 0110 0110 0110 01|10 0110
Arrodonir

6) Ensamblar:
0|100 0010 0|011 0011 0011 0011 0011 0|010
4 2 3 1 9 9 9 A

$\boxed{0x4231999A}$

▽ Cuidado que en aquest s'ha d'arrodonir i com al tall de bits vàlids queda com > 0'5 s'arrodoneix al pròxim parell.

c) −255'125

1) Signe: 1

2) Mantissa: 11111111'001 (7, a 0'1)

3) Normalitzar:
$1'1111111 \times 2^7$

4) Cod. Expo:
7 + 127 = 134 = 10000110

5) Cod Mantissa: 8+3 = 11 dig. 6 23−11 = 12
11111110010000000000000

6) Ensamblar:
1100 0011 0111 1111 1001 0000 0000 0000
C 3 7 F 2 0 0 0

$\boxed{0xC37F2000}$

d) -10'75

1) Signe: 1

2) Mantissa:

1010'11

3) Normalitzar:

$1'01011 \times 2^{3}$

4) Cod. Expo:

3+127=130= 1000 0010

5) Cod. Mantissa: 23-3=20

01011000000000000000000

6) Ensamblar:

1|1000 0010|0101 1000 0000 0000 0000 0000

C 1 2 C 0 0 0 0

$\boxed{0\times C12C0000}$

e) 6'5

1) Signe: 0

2) Mantissa:

110'1

3) Normalitzar:

$1'101 \times 2^{2}$

4) Cod. Expo:

2+127=129= 1000 0001

5) Cod. Mantissa: 23-3=20

101 0000 0000 0000 0000 0000

6) Ensamblar:

0|100 0000 1|101 0000 0000 0000 0000 0000

4 0 D 0 0 0 0 0

$\boxed{0\times 40D00000}$

∇ b) 44'4 Calc $\mathcal{E}$

0

$1'011.0001\ 1001\ 1001\ 1001\ 1010 \times 2^{5} = N$

$N - 44'4 = \mathcal{E}$

$(1+2^{-2}+2^{-3}+2^{-7}+2^{-8}+2^{-11}+2^{-12}+2^{-15}+2^{-16}+2^{-19}+2^{-20}+2^{-22}) \times 2^{5}$

$N - 44'4 = \boxed{0'000001526 = \mathcal{E}}$

5.27. $f2 = 0x417AC000 , $f4 = 0x3F140000 i fem add.s (Arrodonir resultat.

a) Calcula a mà el resultat de $f6.

⊕ f2 = 0|1000 0010|111 1010 1100 0000 0000 0000 | exp = 128+2 = 130−127 = 3 = exp

⊕ f4 = 0|0111 1110|001 0100 0000 0000 0000 0000 | exp = 126−127 = −1 = exp (+4) 0'0001'001

$$
\begin{array}{r}
1'11110101100000000000000 \\
+\ 0'00010010100000000000000 \\
\hline
10'00000000000000000000000
\end{array}
$$

→ Veiem que hauem d'ajustar l'exponent (4)
Donat que ex. $10'0\ldots \times 2^{3} \Rightarrow 1'00\ldots \times 2^{4}$
$4+127 = 131 = (1000\ 0011)_2$

+ f6 = 0|1000 0011|000 0010 0000 0000 0000 0000 | $\boxed{0\times 41820000}$ ✓

Tau Bru Ribes

EC Examen de Problemes

Exercici 1 (Examen Parcial 2011/2012 Q2)

Suposant un circuit seqüencial per a la divisió de números naturals de 4 bits, anàleg a l'estudiat a classe per a 32 bits. Suposem que volem calcular la divisió (en base 2): 1011/0011. Omple la següent taula indicant quin és el valor final dels registres R, D i Q al final de cada iteració de l'algorisme que controla el circuit (omple tantes files com iteracions tingui l'algorisme).

iteració	R (Dividend/Residu)		D (Divisor)		Q (Quocient)
valor inicial	0000	1011	0011	0000	0000
1	"	"	0001	1000	0000
2	"	"	0000	1100	0000
3	0000	0101	0000	0110	0001
4	0000	0010	0000	0011	0011
		2			3

0000|0011
− 0000|0110
0000|0101
0000|0011
0000|0010

Primer faig desplaçant del D i si tot el D ≤ R faig resta
- fico 1 en quocient
-
00001011 > 00000010 Resta
Ja acabem pq. tenim D en l'altra part de línia

Exercici 2 (Examen Parcial 2012/2013 Q2)

Suposant un circuit seqüencial per a la multiplicació de números naturals de 4 bits, anàleg a l'estudiat a classe per a 32 bits. Suposem que amb aquest circuït multipliquem els números binaris de 4 bits 1010 (multiplicand) i 1101 (multiplicador). Completa la següent taula, que mostra els valors en binari dels registres P, MD, i MR després de la inicialització i després de cada iteració, afegint tantes iteracions com facin falta:

iteració	P		MD		MR
valor inicial	0000	0000	0000	1010	1101
1		1010	0001	0100	0110
2	0000	1010	0010	1000	0011
3	0011	0010	0101	0000	0001
4	1000	0010			

130

Miro si és 1 i sumo o desplaço
Si és 1 primer sumo i després desplaço
! El prod no el desplacem

Exercici 3 (Examen Parcial 2011/2012 Q1)

Considerant que els registres \$f0 = 0x3F800001 i \$f2 = 0x31880000 indica quin sera el contingut del registre \$f4 (en hexadecimal) d'esprés dexecutar la instrucció MIPS

127
− 99
28

```
add.s $f4, $f0, $f2
```

(+) f0 = 0|011.1111.|000.0000.0000.0000.0000.0001 | exp = 127 − 127 = 0
(+) f2 = 0|011.0001.|000.1000.0000.0000.0000.0000. | exp = 99 − 127 = −28

	G	R	S
1'000 0000 0000 0000 0000 0001	0	0	0
+ 0'000 0000 0000 0000 0000 000 0	0	0	1
1'000 0000 0000 0000 0000 0001	~~0~~	~~0~~	~~1~~

$\boxed{\$f4 = \$f0}$

Exercici 4 (problema 5.25 de la col.lecció))

La següent taula conté una llista de números binaris que representen nombres reals en coma flotant, en el format IEEE 754 de simple precisió. Marca amb una X la casella corresponent al tipus de valor de cada un d'ells, d'acord amb la notació:

```
NRM = normalitzat; DNRM = denormalitzat; 0 = zero;  INF = infinit
NAN = "Not a Number" (resultats d'operacions invalides)
```

signe	exponent	mantissa	NRM	DNRM	0	INF	NAN
0	0000 0000	110 0010 0000 1110 1110 1011		X			
0	0000 0000	000 0000 0000 0000 0000 0000			X		
0	0010 0100	000 0000 0000 0000 0000 0000	X				
1	1111 1111	000 0000 0000 0000 0000 0000				X	
0	0010 0100	110 0010 0000 1110 1110 1011	X				
1	0000 0000	000 0000 0000 0000 0000 0000			X		
0	1111 1111	101 0001 0001 0000 1001 0100					X

Exercici 5 (Examen Parcial 2012/2013 Q1)

Considera que el contingut dels registres \$f2 i \$f4 és 0x3FC00002 i 0x3F400005, respectivament i que s'executa la instrucció MIPS: add.s \$f0, \$f2, \$f4. Suposant que el sumador/restador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant) Quin és el contingut de \$f0 (en hexadecimal) després d'executar la instrucció ?

Exercici 6 (Examen Parcial 2012/2013 Q2)

Considera que el contingut dels registres \$f4 i \$f6 és 0x42000003 i 0xC0F00005, respectivament i que s'executa la instrucció MIPS: add.s \$f0,\$f4,\$f6. Suposant que el sumador/restador té 1 bit de guarda, un d'arrodoniment i un de "sticky", i que arrodoneix al més pròxim (al parell en el cas equidistant), quin és el valor de \$f0 en hexadecimal després d'executar la instrucció ?

(+) \$f2 = 0|011.1111.1|100.0000.0000.0000.0000.0010 | exp = 127-127 = exp=0 (+1)
(+) \$f4 = 0|011.1111.0|100.0000.0000.0000.0000.0001 | exp = 126-127 = exp=-1

GRS
1'100 0000 0000 0000 0000 0010 | 000 ×2^0
(+) 0'110 0000 0000 0000 0000 0010 | 100 ×2^0
10'010 0000 0000 0000 0000 0100 | 100 ×2^0 ⇒ (+1 exp) = 1

0|1000 0000|0010 0000 0000 0000 0000 0010
4 0 1 0 0 0 0 2

$\boxed{0x40100002}$

(+) \$f4 = 0|10000100|0000 00000 00000 00000 0011 | exp = 128+4 = 132 − 127 = 5 = exp (+3)
(−) \$f6 = 1|10000001|111 00000000 0000 0000 0101 | exp = 128+1 = 129−127 = 2 = exp

GRS
1'0000 0000 0000 0000 0000 011 | 000 ×2^5
(−) 0'0011 1100 0000 0000 0000 000 | 101 ×2^5
0'1100 0100 0000 0000 0000 010 | 011 ×2^5
(exp=4)

Veiem que |num_pos| > |num_neg| ⇒ (+) resultat

0|1000 0011|1000 1000 0000 0000 0000 101
4 1 C 4 0 0 0 5

$\boxed{0x41C40005}$

Pero al moure 6 ~~mantissa~~ exponent tmb caniem GRS ⇒ 1'11000100000000000000100 | 110 (GRS) ⇑ Round
11000100000010000000101

(Suma) $f2 = 0x01820003$ „ $f4 = 0x81700003$ (Cas denormals).

⊕ $f2 = 0|000\ 0001\ 1|000\ 0010\ 0000\ 0000\ 0000\ 0011$ | $exp = 3-127 = -124$ (+1)

⊖ $f4 = 1|000\ 0001\ 0|111\ 0000\ 0000\ 0000\ 0000\ 0011$ | $exp = 2-127 = -125$

$$\begin{array}{r} 1'000\ 0010\ 0000\ 0000\ 0011|000 \times 2^{-124} \\ \ominus\ 0'111\ 1000\ 0000\ 0000\ 0001|100 \times 2^{-124} \\ \hline 0'000\ 1010000\ 0000\ 000.1|1 \times 2^{-124} \end{array}$$

→ $0'0001\ 0000\ 0000\ 0000\ 000110 \times 2^{-126}$ Però realment encara hauríem de fer $\xrightarrow{\div 2}$ cosa que fa $exp = -128$ i no es pot representar.

#La cosa és que no podem fer més que -126 d'exponent fent que pugui fer $\xrightarrow{\div 2}$ només.

Com que no es pot fer -128 considerem el num. com un denormal ($exp = 0$ i mantissa $\neq 0$)

0|0000|0000|0|010|1000|0000|0000|0000|0110 ⇒ $\boxed{0x00280006}$

(Mult.) $f2 = 0x42F60000$ $f4 = 0xC1B00000$ $\boxed{0xC52A2000}$

⊕ $f2 = 0|100\ 0010\ 1|111\ 0110.0000.0000.0000.0000$ | $exp = 128+5 = 133-127 = \underline{6 = exp}$

⊖ $f4 = 1|100\ 0001\ 1|011.0000.0000.0000.0000.0000$ | $exp = 128+3 = 131-127 = \underline{4 = exp}$

$$\begin{array}{r} 1'111\ 011|0\ 0000.0000\ 0000\ 0000 \times 2^{6} \\ \otimes\ 1'011|0000.0000.0000\ 0000.0000 \times 2^{4} \\ \hline 1111011 \\ 1111011 \\ 0000000 \\ \oplus\ 1111011 \\ \hline 10'1010\ 01001 \times 2^{6+4=10} \end{array}$$

Normalitzem

→ $1'0101001001\ 00,0000000000000 \times 2^{11}$

$exp = 11 + 127 = 138 = (1000\ 1010)_2$

$\oplus \cdot \ominus = \ominus$

1|100|0101|0|010|100|100|0|000|0000|0000|0.00

C 5 2 A 2 0 0 0

(P-16/17-Q2) $f2 = 0xBE80000C$ $f4 = 0x40800000$ suma i error.

⊖ $f2 = 1|011.1110.1|000.0\ 0000\ 0000\ 000.0000.1100$ | $exp = 125-127 = \underline{-2} = exp$ (+4)

⊕ $f4 = 0|100\ 0000\ 1|000\ 0000\ 0000\ 0000\ 0000\ 0000$ | $exp = 129-127 = \underline{2} = exp$

$\boxed{0x406FFFFE}$

$$\begin{array}{r} 1'000\ 0000\ 0000\ 0000\ 0000\ 0000|000 \times 2^{2} \\ \ominus\ 0'0001000\ 0000\ 0000\ 0000\ 0000|110 \times 2^{2} \\ \hline 0'1110111111111111111111|010 \times 2^{2} \end{array}$$

→ $1'11011111111111111111110 \times 2^{1}$ (1)

0|100|0000|0|110|1111|1111|1111|1111|1110

4 0 6 F F F F E

~~El error absolut~~ Si ens fixem el numero després de normalitzar queda $1'11011111111111111111110|10$ GRS però com que és "100" fem ~~truncat~~ arrodoniment a parell i aquest ja és parell així que simplement traiem GRS.

$\mathcal{E} = |N^o\ sense\ arrodonir - N^o\ arrodonit| = 0'000\ 0000\ 0000\ 0000\ 0000\ 0000|10 \times 2^{1}$

$\mathcal{E} = 2^{-24} \cdot 2^{1} = \boxed{2^{-23}}$

F-15/16 $ \$f2 = 0xBF000009$, $f4 = 0x41000000$ Suma. -a+b amb a>b   f2-f4

⊖ f2 = |0|111 1111 0|000 0000 0000 0000 0000 1001 | $exp = 126 - 127 = -1 = exp$ (f4)

⊕ f4 = 0|1000 0010|000 0000 0000 0000 0000 0000 | $exp = 128+2 = 130 - 127 = 3$ exp

~~0'0000100 0000 0000 0000 0000 | 101 ×2³~~
~~1'000 0000 0000 0000 0000 0000~~
~~1 000 0100 0000 0000 0000 0000 | 101 ×2³~~

		GRS	
	1'000 0000 0000 0000 0000 0000	000	$\times 2^3$
⊖	0'000 1000 0000 0000 0000 0000	101	$\times 2^3$
	0'1110 1111 1111 1111 1111 1111	011	$\times 2^3$
	1'110 1111 1111 1111 1111 1110	$\boxed{110}$	$\times 2^2$

Round UP

$1'110\,1111\,1111.1111.1111.1111 \Rightarrow 2+127 = 129 = (10000001)_2 = exp \Rightarrow$ ⊕ Signe

0 | 100 | 0000 | 1 | 110 | 1111 | 1111 | 1111 | 1111 | 1111 $\rightarrow$ $\boxed{0x40EFFFFF}$
4 0 E F F F F F

$\varepsilon = 1'110.1111.1111.1111.1110.110$ n° sum $1 \times 2^{24} \cdot (-2) \times 2^{-2} = -2 \times 2^{22}$
⊖ $1'110.1111.1111.1111.1111.$ n° red
$1111\,1111\,1111\,1111\,1111.110$ ¿-2?