Estructura de Computadores

Suposem que definim un format de coma flotant de 16 bits, similar a l'estàndard de simple precisió, excepte que té 7 bits de fracció en comptes de 23. La resta de camps es codifiquen igual.

a) Codifica el número x = -27,16 en el nou format de 16 bits aplicant l'arrodoniment al més pròxim, i expressa el resultat en hexadecimal:

x = 0x

b) Calcula l'error de precisió comès en l'anterior apartat, expressant-lo en base 10, amb 2 dígits significatius (per exemple: error = 0,000XX, o bé: 0,0XX, etc.):

error =

Considera la següent declaració MIPS de variables globals:

```
a:      .word  0xCC800000
b:      .word  0x4C800000
c:      .float 1.0
```

Suposant que s'executa el següent codi:

```
la      $t0, a
lwc1    $f0, 0($t0)
la      $t0, b
lwc1    $f2, 0($t0)
la      $t0, c
lwc1    $f4, 0($t0)
```

Es demana que contesteu quin serà el valor final a `$f6` en hexadecimal després de l'execució dels següent codis:

a) (0,6 pts)

```
add.s   $f6, $f0, $f2
add.s   $f6, $f6, $f4
```

$f6 = 0x

b) (0,6 pts)

```
add.s   $f6, $f2, $f4
add.s   $f6, $f0, $f6
```

$f6 = 0x

Problema

- Suponiendo que $f2=0xC076C000, $f4=0x3ECA8000, y que ejecutamos la instrucción: `add.s $f6, $f2, $f4`.
  Suponiendo que redondeamos al más próximo (al par en caso de equidistancia):
  1. Calcula (a mano) el valor final de $f6 en hexadecimal, siguiendo el algoritmo de suma de número en coma flotante.
  2. ¿Se produce algún error de precisión en el resultado?
  3. Convierte a decimal (base 10) el valor final de $f6.

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):
  1. Es produeix *underflow* quan els bits fraccionaris de la mantissa són tots zeros

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):
  1. Es produeix *underflow* quan els bits fraccionaris de la mantissa són tots zeros
  2. Es produeix *overflow* quan l'exponent del resultat d'una operació és major que $+126$

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):
  1. Es produeix *underflow* quan els bits fraccionaris de la mantissa són tots zeros
  2. Es produeix *overflow* quan l'exponent del resultat d'una operació és major que $+126$
  3. La codificació 0x7FFFFFFF representa el major número positiu no-nul

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):
  1. Es produeix *underflow* quan els bits fraccionaris de la mantissa són tots zeros
  2. Es produeix *overflow* quan l'exponent del resultat d'una operació és major que $+126$
  3. La codificació 0x7FFFFFFF representa el major número positiu no-nul
  4. La codificació 0x00800000 representa un número normalitzat

¿Verdadero o falso?

- Les següents afirmacions fan referència al format de simple precisió IEEE-754 (32 bits):
  1. Es produeix *underflow* quan els bits fraccionaris de la mantissa són tots zeros
  2. Es produeix *overflow* quan l'exponent del resultat d'una operació és major que $+126$
  3. La codificació 0x7FFFFFFF representa el major número positiu no-nul
  4. La codificació 0x00800000 representa un número normalitzat
  5. La codificació 0x00000001 representa un número denormal

Problema

- Suponiendo que f y g son variables de tipo float (coma flotante en simple precisión) almacenadas en $f12 y $f14, traduce a MIPS la siguiente sentencia en C:

```
f = g + 3.14;
```

Subrutinas

- Dibuja el bloque de activación de la subrutina `variancia`

```
float mitjana (float p[]);

float variancia (float vec[]) {
  int i;
  float m, q;
  float vquadrats[100];

  for (i=0; i<100; i++)
    vquadrats[i] = vec[i] * vec[i];
  q = mitjana(vquadrats);
  m = mitjana(vec);
  return q - m * m;
}
```