

Completa l'exercici 4.1 abans de continuar:

Exercici 4.1: Tradueix a ensamblador MIPS la subrutina `descompon`.

<pre>descompon: slti \$t0, \$a0, 0 sw \$t0, 0(\$a1) sll \$a0, \$a0, 1 bne \$a0, \$zero, else move \$t0, \$zero b fi else: addiu \$t0, \$zero, 18 #exp = 18 while: blt \$a0, \$zero, fi_while sll \$a0, \$a0, 1 addiu \$t0, \$t0, -1 b while fi_while: addiu \$t0, \$t0, 127 #exp += 127 srl \$a0, \$a0, 8 li \$t1, 0x7FFFFFFF and \$a0, \$a0, \$t1</pre>	<pre>fi: sw \$t0, 0(\$a2) sw \$a0, 0(\$a3) ir \$ra</pre>
--	--

Al fitxer `s4a.s` estan ja programats el programa principal (`main`) i les declaracions de variables globals en ensamblador MIPS. Afegeix-hi el codi de l'exercici 4.1. A continuació, carrega'l al simulador, assembla'l i executa'l.

Comprova que al final de l'execució del programa `signe = 1`, que `exponent = 0x0000008D` (141 en decimal), i que `matissa = 0x007A3140`.

Activitat 4.B: Funció `compon`

```
float compon(int signe, int exponent, int mantissa)
{
    return (signe << 31) | (exponent << 23) | mantissa;
}
```

Figura 4.3: Codi en alt nivell de la subrutina `compon`

La subrutina `compon` (figura 4.3) rep com a paràmetres els enters `signe`, `exponent` i `mantissa` passats per valor i retorna el número en coma flotant de simple precisió equivalent. El `signe` s'ha de desplaçar a la posició 31, i l'`exponent` a la posició 23. Els tres camps de bits resultants es poden combinar amb operacions `or` lògiques, ja que sabem que cada camp ocupa sols els bits que li són propis, essent la resta zeros.

Completa l'exercici 4.2 abans de continuar:

Exercici 4.2: Programa en ensamblador la subrutina `compon`.

```
compon:

    sll $a0, $a0, 31
    sll $a1, $a1, 23
    or $v0, $a1, $a0
    or $v0, $v0, $a2
    mtc1 $v0, $f0
    jr $ra
```

Afegeix el codi de l'exercici 4.2 al fitxer **s4a.s**. Veuràs que el fitxer conté una subrutina `compon` provisional amb 1 sola instrucció, escrita per permetre fer proves a l'activitat 4.A: esborra-la. A continuació, assembla'l i executa'l.

Comprova que el número resultant, en coma flotant, val `cfloatant=0xC6FA3140`. També pots comprovar el seu valor en decimal en el simulador. De quina manera ho consultaràs?

Settings -> Values displayed in hexadecimal -> Coproc 1 -> \$f0 -> float

Quin és el valor final de `cfloatant`, en decimal, que dona Mars?

Valor decimal de `cfloatant` = **-32024.625**

Completa el següent exercici abans de continuar:

Exercici 4.3: Codifica en coma fixa i en coma flotant (en hexadecimal) els següents números:

Decimal	cfixa (valor inicial)	cfloatant (valor final)
0.0	0x 00000000	0x 00000000
- 0.0	0x 80000000	0x 80000000
12.75	0x 0000C733	0x 41473300

Verifica que el teu programa també converteix correctament els números de l'exercici 4.3, inicialitzant `cfixa` amb els valors de la segona columna i observant al simulador si el resultat en `cfloatant` és el de la tercera columna.

Activitat 4.C: Errors de precisió en la conversió

Un cop llegida la pràctica i resolts els anteriors exercicis hauries de ser capaç de reflexionar sobre els errors de precisió que es poden cometre fent la conversió proposada, i respondre les preguntes del següent exercici:

Exercici 4.4: Contesta les següents preguntes

- 1) Quina condició ha de complir el valor inicial de `cfixa` perquè es produeixi pèrdua de precisió en la conversió que proposa aquesta pràctica?

Es produirà una pèrdua de precisió en la conversió quan la mantissa estigui composta per més de 23 bits significatius

- 2) Indica un valor de `cfixa` per al qual es produiria pèrdua de precisió al convertir-lo, i el corresponent valor en coma flotant:

cfixa	cflotant
0x008F22FF	0x 0x450F22FF

- 3) En quina sentència concreta del programa en alt nivell es pot produir la pèrdua de precisió?

`cf = (cf >> 8) & 0x7FFFFFFF`

- 4) Quin dels 4 modes d'arrodoniment que coneixes està portant a la pràctica aquest programa de conversió?

Truncament

- 5) El format de coma fixa explicat en aquesta pràctica permet codificar un rang de valors bastant limitat. Indica un número positiu que estigui DINS el rang del format de coma flotant de simple precisió (en decimal) però que estigui FORA del rang del format de coma fixa. Indica també quin és el MENOR número potència de 2 que compleixi aquesta condició.

$2^{19} = 524.288$, ja que el màx en coma fixa es $2^{19} - 1$, el menor serà 2^{-126} (1 és el menor nombre que pot haver en l'exponent per tant, $1 - 127 = -126$, si fos zero seria una indeterminació) i esta fora del rang de la part fraccionaria de la coma fixa (son només 12 bits i el numero a representar es $1,17 \cdot 10^{-38}$)

