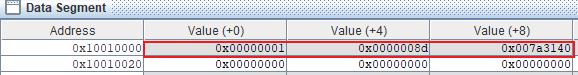
# Pràctica 4

## Activitat 4.A

### Exercici 4.1

| # === DESCOMPON ===  descompon:      slt $t0,$a0,$zero  # $t0 = cf<0      sw $t0,0($a1)  # \*s = $t0      sll $a0,$a0,1  # cf = cf<<1  descompon\_if:      bne $a0,$zero,descompon\_else  # cf!=0 goto descompon\_else      sw $zero,0($a2)  # \*e = 0      sw $a0,0($a3)  # \*m = cf      jr $ra  descompon\_else:      li $t0,18  # $t0 = 18 (Fare servir com "exp")        # EXPLICACIONS      # 1.      # NO ENTENC PERQUE INICIALITZEM EN 18      # ---      # 2.      # Portem el primer bit '1' de la mantissa a la pos 31 (Donat que quan arribi el num sera negatiu)      # Ho fem perque la mantisa es de la forma "1,[...]" (Tot i que haurem d'eliminar aquest 1 posteriorment)      # ---      # 3.      # NO ENTENC PERQUE BAIXEM L'EXPONENT      # ---      # 4.      # Fem cf>>8 per deixar en la posicio que pertoca la mantissa      # ---      # 5.      # 0x7FFFFF que aixo son "0111 1111 [...]" perque aquest primer 1 conserva el bit num 22 i treu el 23-31      # Recordem que la mantissa ocupa 23 bits així que interessa que estigui del 0 al 22.  descompon\_while:      sll $a0,$a0,1  # cf = cf<<1  # Jo se que la primera vegada s'executa (sino hagues saltat if)      addiu $t0,$t0,-1  # exp--      bge $a0,$zero,descompon\_while  # cf>=0 goto descompon\_while      srl $a0,$a0,8  #cf = cf>>8 # No sra perque volem afegir 0      li $t1,0x007FFFFF      and $a0, $a0, $t1  # cf = cf & 0x7FFFFF      addiu $t0,$t0,127  # exp += 127      sw $t0,0($a2)  # \*e = exp      sw $a0,0($a3)  # \*m = cf      jr $ra |
| --- |



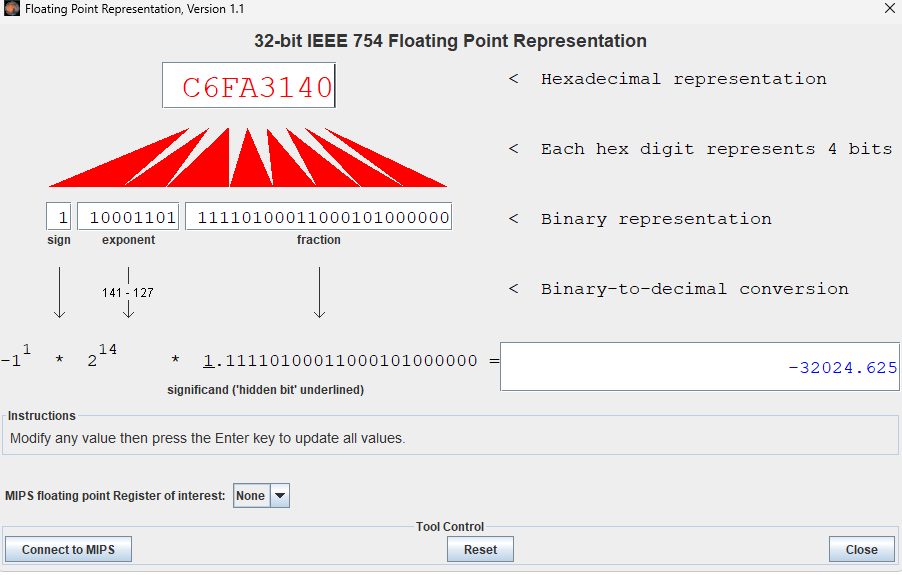
## Activitat 4.B

### Exercici 4.2

| # === COMPON ===  compon:      sll $a0,$a0,31  # signe<<31      sll $a1,$a1,23  # exponent<<23      or $t0,$a0,$a1  # (signe<<31)|(exponent<<23)      or $t0,$t0,$a2  # (signe<<31)|(exponent<<23)|mantissa      mtc1 $t0,$f0  # Fica l'enter $t0 en un registre de coma flotant      jr $ra |
| --- |

Donat que estem treballant amb reals i no enters, els Mars (en la memòria per defecte) mostra els valors enters, és per això que haurem d’anar a “Tools”>”Floating Point Representation” i introduir el número en hexa que hi ha a la posició de memòria corresponent a l’etiqueta “cflotant”

# Nota: Això ho he tret de [link](https://faculty.kfupm.edu.sa/coe/mudawar/coe301/lab/COE301_Lab_9_Floating_Point.pdf)



### Exercici 4.3

# Nota: A l'inici de la pràctica es defineix el format de coma fixa.

| **Decimal** | **cfixa (valor inicial)** | **cflotant (valor final)** |
| --- | --- | --- |
| 0.0 | 0 | 000 0000 0000 0000 0000 | 0000 0000 0000 0x00000000 | 0|000 0000 0|000 0000 0000 0000 0000 0000 0x00000000 |
| -0.0 | 1 | 000 0000 0000 0000 0000 | 0000 0000 0000 0x80000000 | 1|000 0000 0|000 0000 0000 0000 0000 0000 0x80000000 |
| 12.75 | 0 | 000 0000 0000 0000 1100 | 1100 0000 0000  0x0000CC00 | 0|100 0001 0|100 1100 0000 0000 0000 0000  0x414C0000 |

Procediment:

1. El número és positiu així que el signe és 0.
2. Passo a binari el número 12.75 = (1100,11)
3. Moc la coma fins al primer 1 de la seqüència i el nombre de salts que he fet representarà l’exponent (1100,11) => [3 salts] => (1,10011)
4. Codifico l’exponent en excés: exp=3+127=130 (1000 0001)
5. Codifico el número donat que tinc tot el necessari.

## Activitat 4.C

### Exercici 4.4

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

La pràctica proposa convertir el número “-3.1527934e-34”.  
Segons el format de coma fixa només tenim 19 bits per a la part entera i 12 per la part fraccionària (més 1 per al símbol), això significa que qualsevol número menor que (0000 0000 0001) = 2^-12 no podrà ser representat (en valor absolut donat que el de símbol ens és igual).

Llavors el número que proposa la pràctica NO podrà ser representat en aquest format de coma fixa donat que 2^-12>|-3.1527934e-34| i la millor aproximació a aquest número serà de fet, el més petit representable: 0x00000001.

1. **Indica un valor de cfixa per al qual es produiria pèrdua de precisió al convertir-lo, i el corresponent valor en coma flotant:**

Com ja hem discutit en el punt anterior, qualsevol número (enter) representable amb 19 bits que tingui una part decimal més petita que 2^-12=0.00024414062 llavors per exemple 23.0001 ja no.

| **cfixa** | **cflotant** |
| --- | --- |
| 0x00017001 (23,00024414062) | 0|100 000 1|011 1000 0000 0000 0011 0100  0x41B80034 |

1. **En quina sentència concreta del programa en alt nivell es pot produir la pèrdua de precisió?**

Agafem un valor on, després d’haver tret el signe faci que el número a avaluar sigui negatiu fent que no entri en el bucle de “while (cf>=0)”.

Estat inicial:

0|100 0000 0|000 0000 0000 0000 0000 0001

Fem “cf<<1” per eliminar signe.

1|000 0000 0|000 0000 0000 0000 0000 0010

Fem cf>>8 i & 0x007FFFFF per codificar la mantissa.

0|000 0000 1|000 0000 0000 0000 0000 0000

De fet, hem perdut els 7 bits de menor per fer cf>>8 (no n'hem perdut 8 perquè prèviament havíem fet cf<<1 i això ens atorgava un LSB extra que ens redueix la quantitat de bits perduts).

Nota: Indirectament, el fet de només treballar amb 32 bits també fa que hi hagi pèrdua de precisió.

1. **Quin dels 4 modes d’arrodoniment que coneixes està portant a la pràctica aquest programa de conversió?**

El de truncament perquè simplement normalitza com calqui i ja, no mira els LSB per a res.

1. **El format de coma fixa explicat en aquesta pràctica permet codificar un rang de valors bastant limitat. Indica un nombre 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**.

Cfixa: 1b | 19b | 12b 🡺 (-1)^exp **\*** 0<=entera<=2^19 – 1 **+** 0’decimal on 0<=decimal<=2^12 - 1

Cflotant: 1b | 8b | 23b

Un número que estigui dins del coma flotant, però no dins de coma fixa és per exemple “1048576” (2^20). Aquest sí que es pot amb coma flotant donat que podem fer 1,0x2^20, però com que es necessiten 20 bits de part entera i aquest format de cfixa només te 19b, no es pot.

Recorda que el número més gran/petit és 2^126 perquè el 127 està reservat per als “denormals”.

Un número potència de 2 que no pugui ser representat en coma fixa és de fet 2^19.

NOTA Post-Correcció: Realment el més gran (Recorda que s’ha de mencionar amb valor absolut) és el 2^-127 perquè aquest sí que està dins del cflotant però no dins de cfixa.