

Informe de la pràctica #3

Introducció

En aquesta tercera practica, que consta de tres exercicis, per primer cop es realitza un exercici no-guiat. Els objectius de la practica son els següents:

- Aprendre i dominar les sentencies de control de flux i els salts, tant condicionals com incondicionals.
- Consolidar els coneixements sobre ensamblador adquirits a les practiques anteriors.
- Per primer cop, realitzar un codi en ensamblador sense pautes o guies.

Exercicis a Realitzar

En el primer problema ens demanen que compilem un codi ja creat i comprovem la sea funcionalitat. Aquest codi es tracta de un programa que multiplica dos sencers, sumant X vegades un valor Y en un registre concret ($X \cdot Y = R(x)$).

```
Contador: .dW 5
Valor1: .dW 6
Resultat: .rW 1
.begin inici
Inici:
    ADD R0,R0,R1
    SUB R0,R0,R2
    LOAD Contador(R0), R3
    LOAD Valor1(R1), R2
Loop:
    ADD R1, R2,R1
    SUBI R3, #1,R3
    BG Loop
    STORE R1, Resultat(R0)
.end
```

Com en aquest cas el numero que serveix de iterador X és un 5 i el valor que es suma iterativament Y és un 6, el resultat que trobem a R1 al finalitzar el bucle és 30 ($5 \cdot 6 = 30$, o el que és el mateix $6 + 6 + 6 + 6 + 6 = 30$). Per tal de que el bucle funcioni, s'introdueix una linia de codi que activara el bucle novament sempre i quan el bit de negatiu N de la operació "*SUBI R3, #1, R3*" no salti, es a dir, fins que el comptador no arribi a zero.

En el segon problema, ens demanen que tot mirant el codi del primer exercici, el modifiquem per a que no hi hagi problemes per a multiplicar enters negatius o de diferent signe. Per a aquest exercici hem de tenir en compte problemes com el Carry i el Overflow. Desafortunadament, el SiMR 2.0 no funciona prou bé per a guardar els bits provocats per Overflow, així que és difícil poder tenir en compte aquest problema concret. Tot i aixó, per a fer aquest problema, tenim diferents casos possibles:

1. Que els dos sencers tinguin el mateix signe positiu.
2. Que els dos sencers tinguin el mateix signe negatiu.
3. Que els dos sencers tinguin diferents signes (positiu/negatiu o negatiu/positiu).

En el cas numero 1 no tenim problema algun, ja que simplement es fer la multiplicació de forma iterativa com havíem vist abans. En el cas numero 2, l'única complicació i diferencia respecte el cas 1 és que per a poder fer la multiplicació hem de calcular el Ca2 de cada un dels components de la operació per a poder calcular el resultat correctament amb instruccions del tipus "*SUB R0, R(x), R(x)*". Per al tercer i últim cas, primerament hem de calcular el Ca2 del factor que sigui negatiu, cosa que hem de trobar i comprovar amb una instrucció i un salt del tipus:

```
SUB R0, R(x), R0  
BL etiqLoopDesitjat
```

Si el bit de negatiu s'activa, la instrucció de salt portara el nostre punter del programa a la part de codi desitjada per a calcular el Ca2 de un factor de la multiplicació, o un altre. Un cop esbrinat quin és el negatiu i calculat el seu Ca2, podem operar tranquilament els dos sencers com si de positius es tractés, només canviant el signe del resultat al final del bucle ja que ($- \cdot + = -$).

Al tercer i últim problema, per primer cop hem de realitzar un codi no-guiat. En aquest cas ens demanen que fem un programa que calculi el Ca1 de un número donat. Ja que el propi simulador SiMR calcula el Ca2 de un valor quan tracta amb negatius, aprofitarem la propietat de que el Ca2 és el $\text{Ca1} + 1$, reduïnt el problema en gran dificultat, ja que no hem de invertir els bits del valor un per un. Mirant el codi, podria quedar tal que així:

```
Valor: .dw 15
Resultat: .rw 1
.begin inici
inici:
    SUB R1, R1, R1
    LOAD 0(R0), R1
loop:
    SUB R0, R1, R1    ;Amb això hem calcular el Ca2 del valor
    SUBI R1, #1, R1   ;Amb això realitzem el canvi a Ca1. (Ca1=Ca2-1)
    STORE R1, Resultat(R0)
.end
```

Després de obtenir el resultat, guardem el valor del resultat del Ca1 a la posició de memòria reservada.

Conclusions

- S'ha après i dominat els salts i bucles de ensamblador.
- S'ha vist com opera el simulador amb nombres negatius, tant com amb Ca2 com amb Ca1.
- S'ha realitzat un problema de forma independent per primer cop.