Informe de la pràctica 3: Operacions i bucles amb SiMR

Introducció

En aquesta entrega, es pretén d'una vegada per totes la familiarització dels iteradors, bucles o loops. Per fer-ho, es plantegen problemes o petits algoritmes que tenen a veure amb senzilles operacions matemàtiques o electròniques

Base Teòrica

- Per realitzar la pràctica es fa servir el simulador SIMR.
- Primer de tot s'observa com funciona el programa donat i es respon el qüestionari. Posteriorment, es demana un programa capaç de multiplicar dos nombres tenint en compte el Carry.
- Finalment, es demana un programa capaç de passar a Complement a 1

Problemes

2. *A partir del programa que hi ha implementat al problema 1, feu un programa que permeti fer la multiplicació de dos números sencers, tant positius com negatius. Considereu la opció de tenir CARRY i OVERFLOW.*

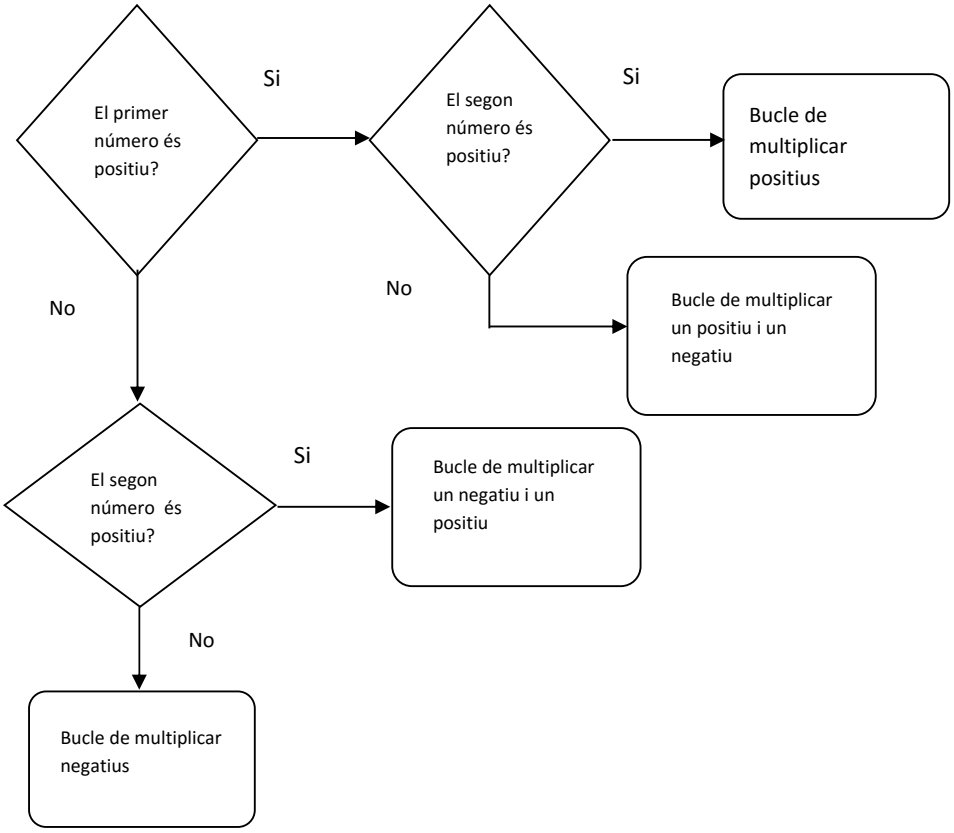
Codi + Explicació

```
valor1: .dW A ;Primer número a multiplicar
contador: .dW B ;Segon número a multiplicar i comptador
over: .DW 16384 ;Overflow
resultat: .rW 1 ;Posició de memòria on es guardarà el resultat
.begin inici
inici:
        ADD R0,R0,R1 ;Inicialitzo R1
        SUB R0,R0,R2 ;Inicialitzo R2
        ADD R0,R0,R3 ;Inicialitzo R3
        LOAD over(R0), R5 ;Es carrega l'overflow a R5
        LOAD valor1(R1), R2 ;Es carrega el primer valor a R2
        BL part2 ;Salto a la posició de memòria Part 2 on gestiono el primer número negatiu
        LOAD contador(R0), R3 ;Carrego el contador a R3
        BL loop2 ;Salta a la posició de memòria loop2 on gestiono el segon número negatiu
 loop: ;Si arribo aquí es perquè els dos números són positius
        ADD R1, R2,R1. Sumo R2 + R1. ;Si existeix el Carry, la suma serà negativa.
        BL fi ;Salto al final
        SUBI R3, #1,R3 ;Resto 1 al contador
        BG loop ;Si el contador no es zero,
        SUB R1, R5, R5 ;Miro l'overflow, per
        BEQ fi ;comprovar si he acabat
        STORE R1, resultat(R0) ;Guardo el resultat de la multiplicació a R1.
        BR fi ;Acabo el programa
 loop2: ;Si arribo aquí es perquè el segon número es negatiu
        SUB R1, R2,R1 ;Resto R2 – R1. El resultat serà negatiu, a no ser que hi hagi Carry.
        BG fi ;Si és així, el resultat serà positiu, i per tant salto al final
        ADDI R3, #1,R3 ;Sumo 1 al contador,
        BL loop2 ;Segueixo fins que sigui zero
        ADD R1, R5, R5 ;Comprovo l'overflow
        BEQ fi
        STORE R1, resultat(R0) ;Guardo el resultat de la multiplicació a R1
        BR fi

 part2: ;Si arribo aquí es perquè el primer valor es negatiu.
        LOAD contador(R0), R3 ;Carrego el contador a R3.
        BL loop4 ;Si es negatiu, salto a la posició de memòria loop 4.
 loop3: ;Si arribo aquí es perquè el valor 1 es negatiu i el comptador també ho és
        ADD R1, R2,R1 ;Sumo R2 + R1. Si hi ha carry, el resultat serà positiu,
```

```
        BG fi ;Per tant salto al final
        SUBI R3, #1,R3 ;Si no, Resto 1 del comptador,
        BG loop3 ;Fins que no sigui zero, continuaré
        ADD R1, R5, R5 ;Comprovo l'Overflow
        BEQ fi
        STORE R1, resultat(R0) ;Guardo el resultat de la multiplicació a R1
        BR fi
loop4: ;Si arribo aquí es perquè el primer valor i el comptador són negatius.
        SUB R1, R2,R1 ;Resto R2-R1. Si el resultat és positiu, es que hi ha Carry
        BL fi; Per tant, salto al final
        ADDI R3, #1,R3 ;Si no, sumo al comptador 1,
        BL loop4 ;Fins que sigui zero
        SUB R1, R5, R5 ;Comprovo l'overflow
        BEQ fi
        STORE R1, resultat(R0) ;Guardo el resultat de la multiplicació a R1
        BR fi
fi:
        .end
```

Figura 1. Diagrama del Problema 2:

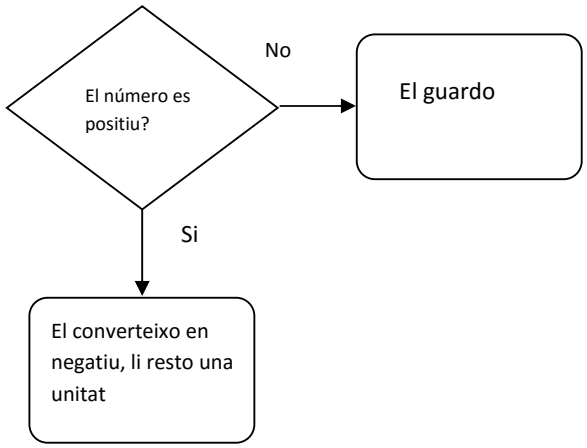


3. *Feu un programa que calculi el Ca1 del contingut d'una determinada posició de memòria*

```
valor1: .dW A ;Valor a convertir en Ca1
resultat: .rW 1 ;Posició de memòria on guardarem el valor en Ca1
.begin inici
inici:
        ADD R0,R0,R1 ;Inicialitzo R1
        LOAD valor1(R1), R2 ;Carrego el valor a R2
        BL fi ;Si la operació anterior es negativa, és a dir, si el valor es negatiu, salto al final
        SUB R0, R2,R2 ;Converteixo el valor en negatiu
        SUBI R2, #1,R2 ;Li resto 1 a aquest valor
        STORE R2, resultat(R0) ;Guardo el valor en R2.
fi:
        .end
```

Aquest codi funciona ja que convertir el valor en negatiu i després restar 1 és l'equivalent a convertir un número en Ca2 a Ca1. Si m'introdueixen un número negatiu, el SiMR mel guarda en Ca2, per tant, ja he acabat, salto al final. Tindria el següent diagrama:

Figura 2: Diagrama del Problema 3:



Preguntes

1. *Problema 1*

    a. *Quin és el valor final d'R1?*

    El valor final és 001Eh, que equival a 30 en decimal

    b. *Quin bit s'activa per sortir del bucle?*

    El Z, és a dir, és 1 l'operació anterior dóna un resultat positiu

    c. *Quina operació estem fent en aquest algoritme?*

    Una multiplicació de 6 x 5, però sumant "6" cinc cops

    d. *A que equival valor 1?*

    Equival a la posició de memòria zero, on en aquest cas tinc un 6.

Conclusions

He aconseguit utilitzar sentències de control de flux, generació de bucles, salts condicionals i incondicionals i he aconseguit solidificat els coneixements adquirits.

```
Dades: .DW 3083,17 ;valors a carregar a les posicions de memòria 1 i 2.
.begin inici ;directiva d'inici de programa
inici:
LOAD 0(R0), R1 ;inicialitza R1 al contingut de la posició de memòria 0.
LOAD 1(R0), R2 ;carrega a R2 el contingut de la posició de memòria 1.
loop: ;while
ADD R7, R1, R7 ;Suma R7 + R1 i guarda-ho en R7
SUBI R2,#1, R2 ;Resta el contingut de R2 menys 1
BG loop ;si la operació anterior dona positiu, salta a la posició de memòria "loop"
STORE R7, 0(R2) ;Guarda el contingut de R7 en la posició de memòria del contingut de R2.
.end
```

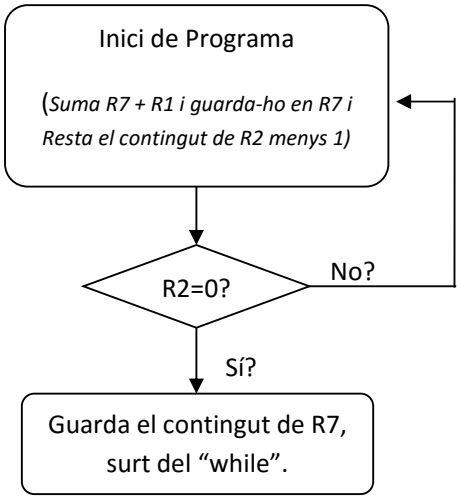


Diagrama de flux 1: Programa de l'exercici 2

2. *Ens demanen calcular un algoritme que ens faci el següent: Donades dues entrades emmagatzemades en les posicions de memòria A i B, fer la comparació. Si A>B calculem la suma. Si B>A fem la diferència (B-A) i si són iguals que finalitzi l'algoritme.*

Per fer-ho, està clar que s'haurà de fer us de les instruccions de tipus salt condicional i incondicional. Les nostres condicions en aquest cas són 3: B més gran que A (BL) A i B iguals (BEQ) i A més gran que B.

El que faig és el següent: carrego les dades a R1 i R2, les resta i les guarda a R3. Si el resultat és negatiu salta a guardar R3 a la posició de memòria 22h i acaba, sinó, torno a fer la mateixa resta, per comprovar si són iguals. En conseqüència, si el resultat dona zero, salta al final i acaba, sinó, suma R1 i R2 i ho guarda a R3, ja que voldrà dir que A és més gran que B. Després, guarda-ho a la posició de memòria 22h.

```
Dades: .DW 4,3
.begin inici ;directiva d'inici de programa
inici:
        LOAD 0(R0), R1 ;carrega el primer valor a R1
        LOAD 1(R0), R2 ;carrega el segon valor a R2
        SUB R1,R2,R3 ;resta R1 menys R2
        BL guarda ;Si el resultat de la operació anterior és negatiu, ja he acabat (b és mes gran que a)
        SUB R1,R2,R3 ;Torna a restar R1 i R2 per comprovar si són iguals
        BEQ fi ;Si la operació anterior ha donat zero, ja he acabat (són iguals)
        ADD R1,R2,R3 ;Si he arribat fins aquí vol dir que A és més gran que B. Suma.
        BR guarda ;Salt incondicional a guardar el resultat
guarda:
        STORE R3, 22(R0) ;guarda el resultat en la posició de memòria 22h.
fi:
```

.end

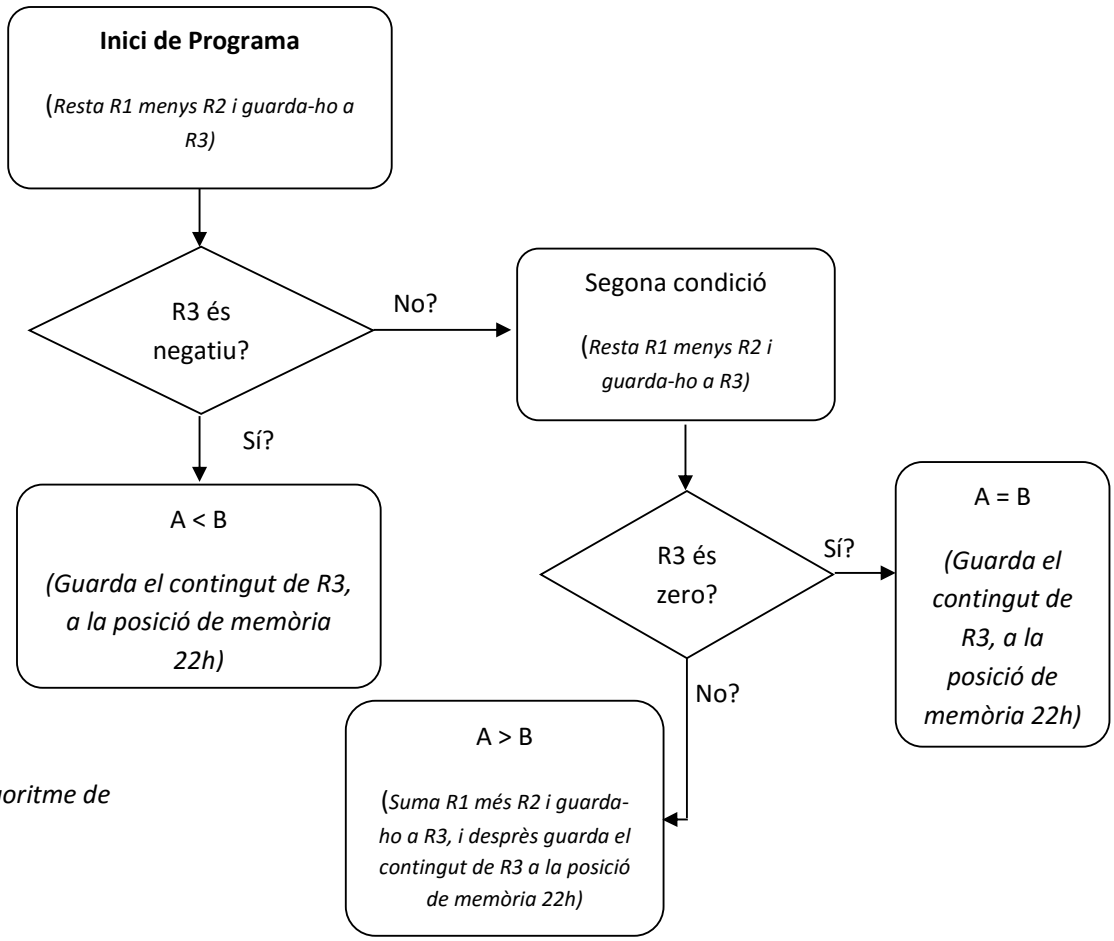


Diagrama de flux 2: Algoritme de l'exercici 3

Preguntes

1) Exercici 1

   a) Al acabar el programa, R5 val 0009h, R6 val 0008h i R3 val 0000h.

2) Exercici 2

a) R7 val CCBBh, que seria 1100 1100 1011 1011 en binari i 12249977906120007 85 en decimal.

b) El programa s'executa 17 vegades

c) Es guarda en la posició de memòria 0000h, ja que R2 val zero quan el programa acaba.

3) Exercici 3

a) Explicat a dalt.

b) Implementat a dalt.

Conclusions

En aquesta pràctica he après els coneixements mínims d'algorismes en llenguatge màquina amb les instruccions de tipus de salt de les que disposo, de manera que a la vegada he pres contacte amb els registres de la meva CPU i els seus valors.

- He aconseguit familiaritzar-me amb el funcionament dels registres de la CPU
- He aconseguit que l'algoritme realitzés el demanat.