

INFORME DE LA PRÀCTICA 2

Introducció

Objectius

L'objectiu d'aquesta pràctica és familiaritzar-nos amb el funcionament dels registres que hi ha a la CPU. Per tal d'assolir aquest objectiu, farem servir el simulador SIMR.

Exercicis de la pràctica 2:

Exercicis guiats

Exercici 1: Carrega el següent programa en el simulador SIMR.

```
Dades: .dw 4, 5, 6, 7 ;dades per operar amb els registres
.begin inici
inci:
SUB R7, R7, R7 ;posem el registre a 0
LOAD Dades(R0), R1 ;carreguem de dades la posició 0 al
registre R1
ADDI R7, #1, R7 ;sumem 1 al contingut del registre R7, ara
tindrà el valor 1
LOAD 0(R7), R2 ;carreguem la posició de memòria 0 + 1 = 1h al
registre R2
ADDI R7, #1, R7 ;sumem 1 al contingut del registre R7, ara
tindrà el valor 2
LOAD 0(R7), R3 ;carreguem la posició de memòria 0 + 2 = 2h al
registre R3
ADDI R7, #1, R7 ;sumem 1 al contingut de R7, tindrà el valor 3
LOAD 0(R7), R4 ;carreguem al registre R4 la posició de memòria
0 + 3 = 3h
;en aquest moment els registres tenen el següents valors:
;R1 = 4, R2 = 5, R3 = 6, R4 = 7, R7 = 3.
loop:
ADD R1, R2, R5 ;sumem el contingut de R1 i el de R2 i el posem
a R5
ADD R3, R4, R6 ;sumem el contingut de R3 i R4 i el posem a R6
SUBI R3, #1, R3 ;li restem 1 al contingut de R3
BG loop ;tornem a loop mentre la operació ens doni major que
0.
.end
```

A l'exercici 1: (funcionament explicat en verd)

- i) Un cop acabat el programa quant val el contingut de R5? En R5 tindrem 9
- ii) Un cop acabat el programa quant val el contingut de R6? En R6 tindrem 8
- iii) Un cop acabat el programa quant val el contingut de R3? En R3 tindrem 0

Exercici 2: Feu un programa similar al de l'exercici 1.

- Carregueu al registre 1 el valor 0000110000001011b.
- Carregueu al registre 2 el valor 0000000000010001b.
- Feu l'operació $R7 \leq R7 + R1$ y decrementeu el valor de R2. Feu això en un bucle fins que el valor contingut en R2 sigui 0.

Aquest és el programa emprat. El programa realitza la multiplicació de dos nombres. Per introduir els nombres hem de passar de binari a decimal. Els nombres són respectivament 3083 i 17.

```
Dades: .dw 3083, 17 ;aquí tenim les dades del enunciat
.begin inici
inici:
SUB R7, R7, R7 ;inicialitzem R7 a 0.
LOAD Dades(R0), R1 ;seguint el model del programa anterior,
carreguem la primera dada a R1.
ADDI R7, #1, R7 ;incrementem en 1 el valor de R7
LOAD 0(R7), R2 ;carreguem la segona dada en el valor de R2.
SUBI R7, #1, R7 ;com que usarem R7, el tornem a posar a 0.
loop:
ADD R7, R1, R7 ;fem R[7] <= R[7] + R[1]
SUBI R2, #1, R2 ;decrementem el valor de R2.
BG loop ;al igual que en el programa anterior, mentre la
operació anterior ens doni major que 0 tornarem a
l'etiqueta loop.
STORE R7, 0(R2) ;després del bucle, demanem que ens guardi el
resultat en aquesta posició de memòria.
.end
```

A l'exercici 2:

- i) Quant val el contingut de R7 quan acaba el programa? R7 val CCBBh = 52411d que també és el resultat de fer 3083d * 17d.
- ii) Quantes vegades s'executa el codi? El codi del bucle s'executa 17 cops, fins que el contingut de R2 és 0.
- iii) Al final del programa, just abans del .end, poseu la instrucció STORE R7, 0(R2). En quina posició de memòria es guarda el resultat? Afegint la última línia de codi, veiem que el resultat l'emmagatzema a la posició de memòria 00h. Això és prou lògic tenint en compte que, al final del bucle, el contingut de R2 serà 0. Per tant, si busquem la posició de memòria 0 (offset) + R[2] = 0 + 0 = 00h.

Pràctica individual

Exercici 3: Ens demanen calcular un algoritme que ens faci el següent: Donades dues entrades enmagatzemades 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.

A l'exercici 3:

- i) Expliqueu el funcionament del programa
- ii) Feu que el resultat es guardi a la posició de memòria 22h (22h=34d)

Per a fer el programa suposo que guardar 0 es com no fer res. (ja que de per si en principi a 22h hi haurà un 0 i així no es modificaria).

Avans de començar, (abans del inici) posarem el programa que farà la suma, li direm **suma**.

Treballarem de la següent manera:

- restarem $B - A$.
 - si $B - A = 0 \rightarrow$ no faig res (guardo un 0)
 - si $B - A < 0 \rightarrow B < A$ llavors farem la **suma**
 - si $B - A > 0 \rightarrow B > A$ llavors farem només guardem la resta que hem fet per calcular quin cas ens hem trobat.

Així quedaria el codi:

```
.dw 4,5 ; es un exemple

.begin inici

suma:
ADD R1,R2,R3 ; si  $B - A < 0 \rightarrow B < A$  llavors farem la suma
BR salt ; saltem a la part del programa que guarda el resultat

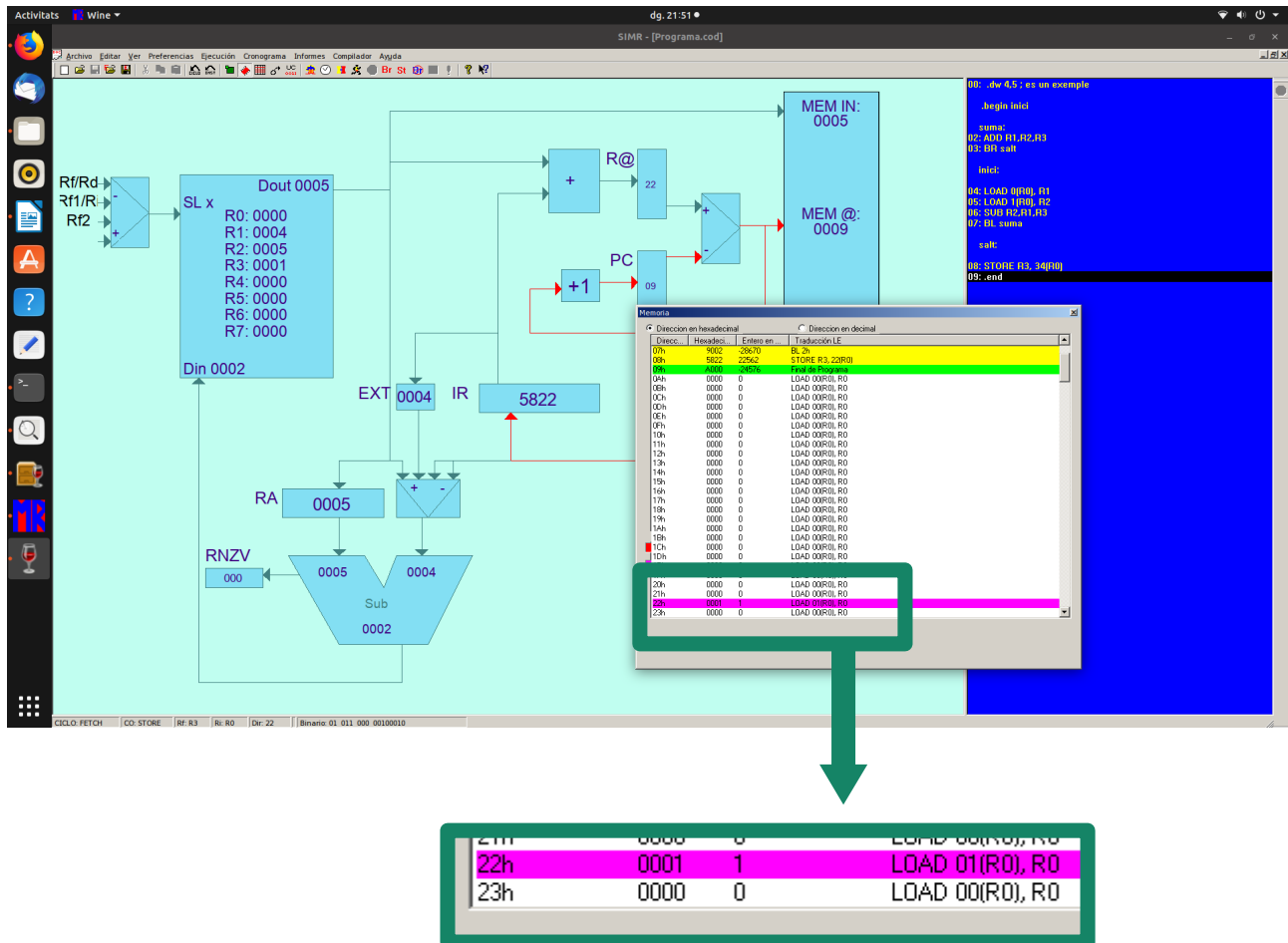
inici:

LOAD 0(R0), R1
LOAD 1(R0), R2
SUB R2,R1,R3 ; fem la resta per comprovar (si es el tercer cas
només haurem de guardar l contigut de R3)
BL suma ; si  $N=1 \rightarrow$  fem la suma (està adalt)

salt: ;aquí es on sortim després de fer el programa suma.

STORE R3, 34(R0)
.end
```

En aquest cas exemple, El resultat hauria de ser 1, ja que $4=A$ i $5=B$, llavors $B>A$ estem en el 3^r cas, i guardem la resta a 22h. ($5-4=1$) llavors a 22h hem de tenir 1. A continuació ho he comprovat amb el simr.



Conclusions:

En aquesta practica, hem fet els exercicis proposats i treball individual. A més hem recordat les comandes apreses a la practica 1 i hem treballat amb salts condicionals. Finalment també hem après el funcionament dels bucles senzills.