

## INFORME DE LA PRÀCTICA 1

### Introducció

#### *Objectius*

- *Familiaritzar-se amb el simulador de la màquina rudimentària (SiMR).*
- *Entendre què fan les instruccions.*
- *Comprendre l'analogia entre llenguatge màquina i el llenguatge ensamblador.*

### Exercicis de la pràctica 1:

#### Exercicis guiats

**Exercici 1:** Indiqueu quines de les següents instruccions són incorrectes segons les especificacions de la Màquina Rudimentària (MR), expliqueu el perquè en cada cas:

Instruccions	Correcte/ Incorrecte	Motiu
LOAD R1(R0), R1	Incorrecte	El primer R1 ha de ser un número no un registre.
STORE R1,R1(3)	Incorrecte	El segon R1 s'ha de posar un número i on el 3 s'ha de posar un registre.
BG 6(R1)	Incorrecte	En aquestes instruccions sempre un número, no un registre.
ADDI R2, #11, 5(R3)	Incorrecte	El 5 i els () s'han de treure.
SUB R0,R2,R3	Correcte	Suma R0 a R2 i ho guarda a R3.
LOAD 3(R0),R1	Correcte	Càrrega de la posició de memòria $3 + 0 = 03h$ al registre R1.

**Exercici 2:** Supposeu que la màquina rudimentària té els següents valors en alguns registres i posicions de memòria (recordeu que amb la lletra “h” s’indica que el número està en format hexadecimal):

-Registres: R0=0000h, R1=0002h, R2=A5E3h, R3=F412h

-Bits de condició (recordeu que N és el bit de Negatiu, Z és el bit de resultat igual a zero): N=0, Z=1

-Memòria: M[0Ch]=F45Ah i M[0Dh]=0033h

Indiqueu què fa cadascuna de les instruccions següents, cal explicitar totes les alteracions que es produeixen a la memòria, bits de condició, registres i valor final. Per cada una de les instruccions sempre partiu de les condicions inicials descrites més a dalt.

	Instruccions	R0	R1	R2	R3	N	Z	M[0Ch]	M[13h]	PC
a	<b>LOAD 12(R0),R1</b>	0000h	F45Ah	A5E3h	F412h	0	0	F45Ah	0033h	1
b	<b>STORE R1,1(R3)</b>	0000h	F45Ah	A5E3h	F412h	0	0	F45Ah	F45Ah	2
c	<b>BR 33</b>	0000h	F45Ah	A5E3h	F412h	0	0	F45Ah	F45Ah	3 (33)
d	<b>ADDI R2,#11,R3</b>	0000h	F45Ah	A5E3h	A5EEh	1	0	F45Ah	F45Ah	34
e	<b>SUB R0,R2,R3</b>	0000h	F45Ah	A5E3h	5A1Dh	0	0	F45Ah	F45Ah	35
f	<b>ASR R1,R1</b>	0000h	7A2Dh	A5E3h	5A1Dh	0	0	F45Ah	F45Ah	36

Breu descripció del que fa cada instrucció:

**a.** Carrega el contingut de fer  $CR[0] + 12 = 0 + 12 = 0Ch$  posició de memòria a R1.  $CR[1] \leq Mem[0Ch]$

**b.** Guarda el contingut de R1 a la posició  $CR[3] + 1 = F412h + 1 = F413h$ . Només agafem els dos darrers bits per accedir a memòria, per tant, posició 13h.  $Mem[13h] \leq CR[1]$ .

**c.** Es mou a la posició de memòria 33 (afecta al PC).

**d.** Suma al contingut de R2 el número 11d.  $CR[3] \leq CR[2] + 11 = A5EBh + 11d = A5EEh$ . Salta el bit de signe perquè Ah = 1010(2) i, per tant, el bit més significatiu és un 1, que marca el signe negatiu.

**e.** Restem al contingut de R0 el contingut de R2 i ho posam a R3. Com que el contingut de R0 sempre és 0, estem fent el negatiu del contingut de R2.

**f.** Desplacem el contingut de R1 un bit a la dreta.

**Exercici 3:** Escriviu en llenguatge màquina (segons les especificacions de la MR) el següent programa, feu-ho en hexadecimal i binari:

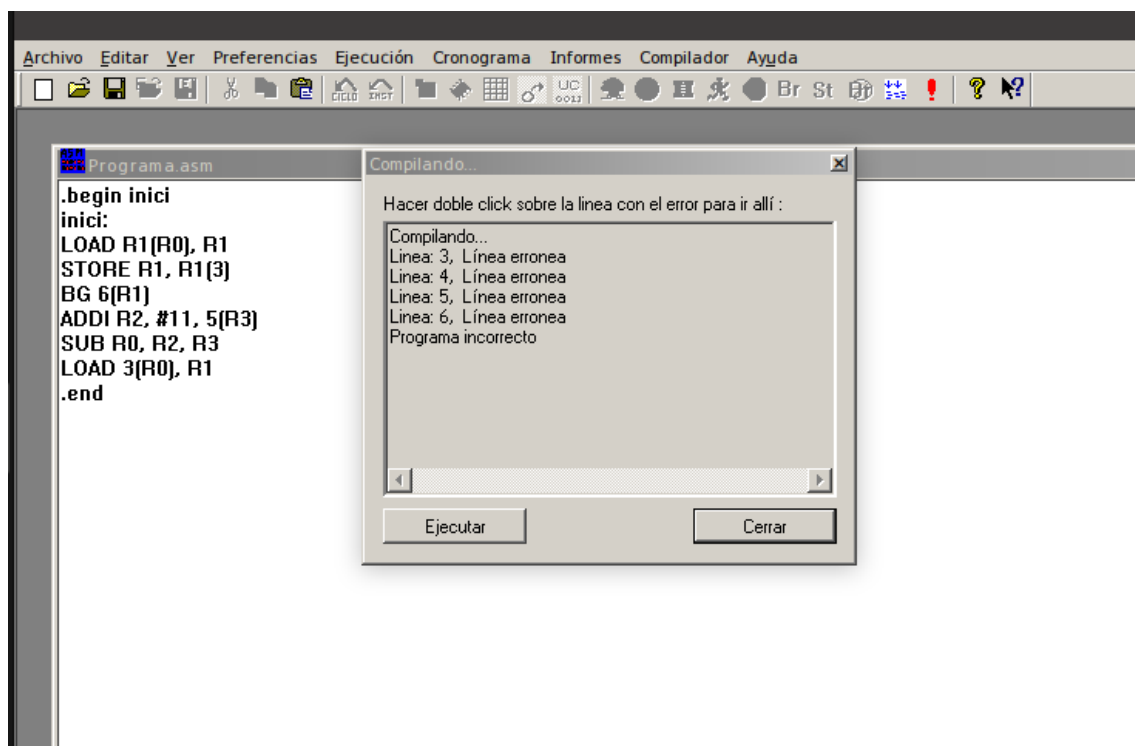
@	M[@]	LLenguatge Màquina	LLeng. Maq. (hex)
05h	SUB R1,R1,R1	11 001 001 001 00 101	C925h
06h	ADD R0,R0,R2	11 010 000 000 00 100	D004h
07h	SUBI R1, #4,R0	11 000 001 00100 001	C121h
08h	BEQ 13	10 001 000 00001101	880Dh
09h	LOAD 0(R1),R3	00 011 001 00000000	1900h
0Ah	ADD R3,R2,R2	11 010 011 010 00 100	D344h
0Bh	ADDI R1,#1,R1	11 001 001 00001 000	C908h
0Ch	BR 07	10 000 000 00000111	8007h
0Dh	STORE R2,4(R0)	01 010 000 00000100	5004h

### **Pràctica Guiada**

**Exercici 1:** Escriure les instruccions de l'apartat 1 de l'estudi previ en un programa al SiMR (no cal que implementi cap funcionalitat) i intenteu compilar-lo. Podreu comprovar com les instruccions incorrectes us donen error alhora de compilar. Feu les modificacions adequades per a que no us doni cap error en la compilació (recordeu que no es demana cap funcionalitat).

Un exemple de com escriure les instruccions anteriors en un programa és el següent:

```
.begin inici  
  
inici:  
  
LOAD R1(R0), R1 ;donarà error  
STORE R1, R1(3) ;donarà error  
BG 6(R1) ;donarà error  
ADDI R2, #11, 5(R3) ;donarà error  
SUB R0, R2, R3  
LOAD 3(R0), R1  
  
.end
```



**Exercici 2:** Pel programa de l'apartat 3 de l'estudi guiat, les posicions de memòria que van des de la 00h a la 03h, ambdues incloses, contenen una seqüència de quatre números 3, 5, 2, 8 emmagatzemats de forma consecutiva. Aquesta seqüència de valors no està especificada al programa anterior. La posició 00h conte el primer element, la posició 01h el segon element, etc.. El programa descrit anteriorment es troba emmagatzemat a partir de la posició de memòria 05h.

Partint d'aquesta descripció responeu a les següents preguntes:

**a) En quina posició de memòria escriu aquest programa el resultat?** És el que fa la última línia de codi, escriu el resultat en la posició de memòria  $4 + CR[0] = 4d + 0d = 4d = 04h$ .

**b) Què fa el programa?** Suma tots els elements (3,5,2,8). El contingut de R1 es va incrementant d'1 en 1 (ja que quan operem *SUB R1, #4, R0* no ho estem guardant a R1). Per tant, en la operació *LOAD 0(R1), R3* anem carregant a R3 cada cop el contingut d'una posició de memòria una unitat més enllà. D'altra banda, quan hauríem d'accedir a la posició de memòria 04h, l'operació *SUBI R1, #4, R0* activa el bit de zero i per tant *BEQ 13* ens fa saltar a la posició de memòria 0Ch, guardem el resultat i acabem.

**c) Quina sentència de control de flux (en llenguatge d'alt nivell) implementa el programa?** Estem implementant el que és equivalent a un while o un bucle for. Tenim un control *SUBI R1, #4, R0* i *BEQ 13*, d'on podem observar que tenim una iteració que es repetirà 4 vegades (perquè R1 s'anava incrementant d'1 en 1 i quan operem  $CR[4] - 4 = 0$  sortirem del bucle).

**d) Quina és la funció d'R1 al programa? I la d'R0?** R1 ens actua com a comptador i control i també ens serveix per sortir del bucle, com he explicat en l'anterior pregunta. R0 ens serveix d'una banda per inicialitzar valors i d'altra per accedir a posicions de memòria. Com que sempre té el valor 0 i no es pot modificar, ens serveix de referència.

**Exercici 3:** Realitzeu les següents accions sobre el Simulador de la màquina rudimentària (SiMR) i expliqueu-ne els resultats:

**e) Arranqueu el simulador de la MR y carregueu el programa pract1.asm. Compileu-lo. Carregueu ara el programa pract1.cod. Establiu una relació entre el codi assemblador del programa presentat pel simulador i el de l'apartat anterior.**

La seva funció final serà la mateixa, sumar els elements de l'exercici anterior. La diferencia es que utilitza un sistema d'etiquetes per marcar l'inici del programa, el principi del bucle (loop) i el final del bucle (endloop).

**f) Observeu que inicialment PC=05h (el PC apunta a la primera instrucció executable del programa). Esbrineu com ho fa.**

Carrega les 5 primeres posicions de memòria, i finalment apunta a la 5<sup>a</sup>, que es on troba emmagatzemat el programa.

**g) Executeu el programa, instrucció per instrucció, observant el resultat d'executar cada una de les instruccions:**

Abans d'executar cada instrucció prediu les variacions que es produiran al banc de registres i a la memòria. Comproveu que el resultat de l'execució del programa coincideix amb el que havíeu previst. Per cada instruccions anoteu tots els canvis.

Instrucció	Resultat esperat
SUB R1,R1,R1	El contingut de R1 és zero. Cap canvi ja que $0 - 0 = 0$ .
ADD R0,R0,R2	R0 és 0, llavors R2 és 0, però ja ho era abans. Cap canvi.
SUBI R1, #4,R0	Hi ha canvis al registre aritmètic. Primer està a 0000, després 0001, després 0002, i després 0003. El registre d'estat té el bit N (negatiu) a 1, ja que el resultat de la resta és negatiu però en la última iteració el bit N es posa a 0 i el bit Z (zero) es posa a 1.
BEQ 13	El bit de zero del registre d'estat es posa a 1, saltem a la posició marcada pel final del loop.
LOAD 0(R1),R3	Carregarà la posició de memòria segons el valor que tingui R1. Es carrega aquest valor a R3.
ADD R3,R2,R2	Canvis al registre aritmetic, a R2 tindrem $R2 + R3$ .

ADDI R1,#1,R1	R1 incrementa en 1.
BR 07	El PC torna a apuntar a 07h.
STORE R2,4(R0)	Guarda el resultat a la posició 04h, i aquesta canvia.

### **Conclusions**

He assolit els objectius de la pràctica, m'he familiaritzat amb el simulador de la màquina rudimentària (SiMR), he après a entendre què fan les instruccions i he aconseguit comprendre l'analogia entre llenguatge màquina i el llenguatge ensamblador.