Blai Ras Jimenez, Grup E (Dimarts 12-14)
DNI: 46980893N
Data: 13 de Març de 2016

Pràctica 1: Introducció a la Màquina SIMR (1 sessió)

Objectius

Familiaritzar-se amb el simulador de la màquina rudimentària (SiMR), entendre què fan les instruccions i comprendre l'analogia entre llenguatge màquina i el llenguatge assemblador.

Exercicis Guiats

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
a	LOAD R1(R0), R1	Incorrecte	El primer R1 hauria de ser un valor immediat, com per exemple #4, ja que sinó no estem indican que carregui res.
b	STORE R1,R1(3)	Incorrecte	El segon R1 està intercanviat amb el valor (3). La sintaxis correcte seria STORE R1,3(R1)
c	BG 6(R1)	Incorrecte	La instrucció BG (salta si el valor anterior és positiu) va seguida d'una posició de memòria, per tant, el registre no s'ha de posar, seria BG 6, sempre que 6 sigui una posició de memòria
d	ADDI R2, #11, 5(R3)	Incorrecte	El 5 davant del registre 3 no hi hauria de ser, ja que no indica res. Hauria de ser ADDI R2, #11, R3.
e	SUB R0,R2,R3	Correcte	
f	LOAD 3(R0),R1	Correcte	

Blai Ras Jimenez, Grup E (Dimarts 12-14)
DNI: 46980893N
Data: 13 de Març de 2016

2. Suposeu que la màquina rudimentària té els següents valors en alguns registres i posicions de memòria:
   - 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.

	Instruccions	R0	R1	R2	R3	N	Z	M[0Ch]	M[0Dh]	PC
A	LOAD 12(R0),R1	0000h	F45Ah	A5E3h	F412h	0	1	F45Ah	0033h	PC+1
B	STORE R1,1(R3)	0000h	F45Ah	A5E3h	F412h	X	X	F45Ah	0033h	PC+1
C	BR 33	0000h	F45Ah	A5E3h	F412h	X	X	F45Ah	0033h	PC=33
D	ADDI R2,#11,R3	0000h	F45Ah	A5E3h	A5EEh	0	0	F45Ah	0033h	PC+1
E	SUB R0,R2,R3	0000h	F45Ah	A5E3h	5A1Dh	1	0	F45Ah	0033h	PC+1
F	ASR R1,R1	0000h	7A2Dh	A5E3h	541Dh	0	0	F45Ah	0033h	PC+1

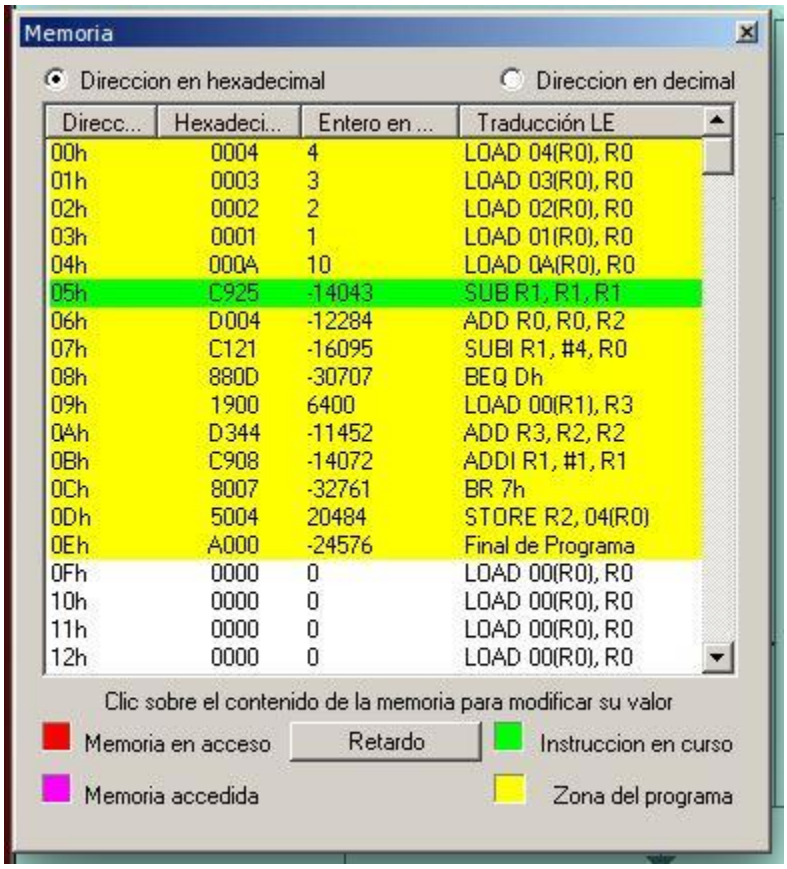
   - Breu descripció del que fa cada instrucció:

     a) Guarda al registre R1 el contingut de la posició de memòria 12 més el valor del registre R0 que sempre es 0 (12+0).
     b) Guarda el contingut del registre R1 en la posició de memòria 1 més el contingut del registre R3.
     c) Salt incondicional a la posició de memòria 33.
     d) Guarda en el registre R3 la suma del contingut del registre R2 més 11.
     e) Guarda en el registre R3 la resta del contingut del registre R0 (que sempre es 0) menys el contingut del registre R2.
     f) Guarda en el registre R1 el desplaçament d'1 bit cap a la dreta del registre R1.

Blai Ras Jimenez, Grup E (Dimarts 12-14)
DNI: 46980893N
Data: 13 de Març de 2016

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	1100100100100101	C925
06h	ADD R0,R0,R2	1101000000000100	D004
07h	SUBI R1, #4,R0	1100000100100001	C121
08h	BEQ 13	1000 1000 0000 1101	880D
09h	LOAD 0(R1),R3	0001100100000000	1900
0Ah	ADD R3,R2,R2	1101001101000100	D344
0Bh	ADDI R1,#1,R1	1100100100001000	C908
0Ch	BR 07	1000000000000111	8007
0Dh	STORE R2,4(R0)	0101000000000100	5004



Blai Ras Jimenez, Grup E (Dimarts 12-14)
DNI: 46980893N
Data: 13 de Març de 2016

Realització de la Pràctica Guiada

2. Partint d'aquesta descripció responeu a les següents preguntes:

   a) *En quina posició de memòria escriu aquest programa el resultat?*
   En la posició 04h

   b) *Què fa el programa?*
   Va sumant els elements del array format per 3, 4, 2 i 8 fins que salti la instrucció endloop. Perquè salti, es necessari que després d'anar restant l'array arribi a Zero, que activarà el Flag Z. La instrucció endloop doncs correspon a STORE R2,4(R0).

   c) *Quina sentència de control de flux (en llenguatge d'alt nivell) implementa el programa?*
   És un "while", i seria mes o menys

   ```
   While (R1 != 0)  {

       Instruccions...
   }
   ```

   d) *Quina és la funció d'R1 al programa? I la d'R0?*
   El registre R1 es fa servir com a índex que indica quin número del array has d'agafar en cada instrucció.
   El registre R0 sempre es zero, així que només s'utilitza per inicialitzar R2 i indicar que guardi (Store) en la posició de memòria 4 + R0 = 4.

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

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

   No entenc la pregunta, però el codi del pract1.cod i del pract1.asm és el mateix...

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

   Amb la instrucció .begin ini, essent ini la instrucció SUB R1, R1, R1

Blai Ras Jimenez, Grup E (Dimarts 12-14)
DNI: 46980893N
Data: 13 de Març de 2016

c) *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:*

- SUB R1, R1, R1: Inicialitzem R1, Program Counter (PC) és 6, flags, la memòria i els registres es queden igual
- ADD R0, R0, R2: Inicialitzem R2. PC es 7. Flags, memòria i registres es queden igual.
- SUBI R1, #4, R0: Resto 4 al contingut de R1, però ho guardo en R0, que es "reseteja" cada vegada. R0 per tant es -4, PC es 8, Flag de negatiu activat (1), memòria es queda igual.
- BEQ endloop: Com que la operació anterior no ha donat zero, no salta. PC és 9, flags, registres i memòria es queden igual.
- LOAD array(R1), R3. R3 ara es 4. PC és 0A, Flags i memòria no es toquen.
- ADD R3, R2, R2: Ara R2 és 4, PC és 0B, Flags i memòria no es toquen.
- ADDI R1, #1, R1: R1 ara és 1, Pc es 0C, Flags i memòria no es toquen.
- BR loop: Salt incondicional. PC és 0D, Flags, memòria i registres no es toquen.
- SUBI R1, #4, R0: Resto 4 a R1 i ho guardo en R0. R0 per tant es (1-4), Flag de negatiu activat, memòria no es toca. PC es 08.
- BEQ endloop: la operació anterior no m'ha donat zero, no salto. PC és 9, Flags, memòria i registres continuen igual.
- LOAD array(R1), R3: Ara R3 es 3, PC és A, Flags i memòria continuen igual.
- ADD R3,R2,R2: Ara R2 és 7, PC es B, Flags i memòria no es toquen.
- ADDI R1, #1,R1: Ara R1 és 2, PC és C i Flags i memòria continuen igual.
- BR loop: Salt incondicional. PC es 7, Flags, registres i memòria es mantenen.
- SUBI R1, #4, R0: Resto 4 a R1. R0 és -2. PC és 8, Flag de negatiu activat, memòria es queda igual.
- BEQ endloop: La operació anterior no m'ha donat zero, no salto. PC és 9, Flags, registres i memòria es queden igual.
- LOAD array(R1), R3: Ara R3 és 2, PC és A, Flags i memòria es queden igual.
- ADD R3, R2, R2: ara R2 és 9, PC és B i Flags i memòria es queden igual.
- ADDI R1, #1, R1: R1 ara és 3, PC és C i Flags i memòria es queden igual.
- BR loop: Salt incondicional. PC és 7, Flags, memòria i registres es queden igual.
- SUBI R1, #4, R0: Resto 4 a R1, així que R0 és -1. PC és 8, Flag de negatiu activat i memòria es queda igual.
- BEQ endloop: l'operació anterior no és zero, no salto. PC és 9, Flags, registres i memòria es queden igual.
- LOAD array(R1), R3: Ara R3 és 1, PC és A i Flags i memòria es queden igual.
- ADD R3, R2, R2: Ara R2 és A, PC és B i Flags i memòria es queden igual.
- ADDI R1, #1, R1: R1 és 4, PC és C, Flags i memòria es queden igual.
- BR loop: salt incondicional. PC és 7, Flags, registre i memòria es queden igual.
- SUBI R1, #4, R0: Resto 4 a R1 i ho guardo en R0. R0 és zero, PC és 8 i s'activa el Flag de zero.

- BEQ endloop: Com que el Flag de Zero s'ha activat, salto a la instrucció endloop. PC és D, Flags i memòria es queden igual.
- STORE R2, sum(R0): La posició de memòria (4+0) ara guarda el contingut de R2, per tant, A. PC és E, Flags es mantenen.

d) Comproveu que el resultar de l'execució del programa coincideix amb el que havíeu previst.

Com dic abans, es guarda el contingut de R2 en la posició de memòria 4.