Pràctica n° 7 - FINAL

Activitat avaluable - 25 % (sobre la nota final de l'assignatura)
Data d'entrega: 24 de Maig de 2015 (fins a les 23:55h)

Objectius:

a) Modificar l'emulador que s'ha desenvolupat a les pràctiques anteriors per a que emuli una màquina lleugerament diferent.

b) Usar totes les tècniques de programació apreses a les pràctiques anteriors.

c) Avaluar els coneixements de programació adquirits (25 % de la nota final de l'assignatura).

Introducció

La màquina que heu de realitzar consisteix en modificar l'emulador que heu escrit i verificat a les pràctiques anteriors, per tal d'emular una nova màquina, molt semblant a la PEPAr, que anomeramem PEPA4r (PEPAr amb 4 registres). Aquesta nova màquina és més flexible que l'original en el sentit de que:

- Incorpora 4 registres: R0, que s'utilitza com a registre acumulador a la majoria d'instruccions de tipus ALU; R1, que s'utilitza com a interfície amb la memòria; R2 i R3 que només poden intervenir en les operacions de tipus ALU.
- Implementa 4 instruccions noves: ADQ, NAN, TRA i SET.
- Totes les instruccions de tipus ALU poden fer servir els 4 registres com a operand font. L'operand destí és majoritàriament l'R0.

Noteu que aquests canvis impliquen modificacions del *datapath* de la màquina (noves connexions i nou *hardware*). Assumirem que aquests canvis s'han incorporat, però no ens entretendrem vegent de quins canvis es tracta.

La nova màquina: la PEPA4r

A la taula 1 s'indica la informació més rellevant del conjunt d'instruccions de la PEPA4r. Noteu que, en general, la codificació de la PEPA4r canvia respecte de la PEPAr. Noteu també les funcionalitats de les instruccions de tipus ALU.

Id	Mnemònic	Codificació	Acció	Flags
0	STO M	000xmmmmmmmm	M ← [R1]	n.s'a.
1	LOA M	001xmmmmmmmm	R1 ← [M]	n.s'a.
2	CMP Ra,R0	0100xx00aa00	[R0] − [Ra]	C, Z i N = s.v.r.
3	ADD Ra,R0	0100xx01aa00	R0 ← [R0] + [Ra]	C, Z i N = s.v.R0
4	SUB Ra,R0	0100xx10aa00	R0 ← [R0] − [Ra]	C, Z i N = s.v.R0
5	NAN Ra,R0	0100xx11aa00	R0 ← [R0] nand [Ra]	C = n.s'a., Z i N = s.v.R0
6	ADQ #d,Rb	0101xxxxddbb	Rb ← [Rb] + d	C, Z i N = s.v.Rb
7	TRA Ra,Rb	0110xxxxaabb	Rb ← [Ra]	C = n.s'a., Z i N = s.v.Rb
8	SET #c,Rb	0111cccccbb	Rb ← c	n.s'a.
9	JMZ M	1100mmmmmmmm	si Z = 1, PC ← M	n.s'a.
10	JMN M	1101mmmmmmmm	si N = 1, PC ← M	n.s'a.
11	JMI M	1110mmmmmmmm	PC ← M	n.s'a.
12	HLT	1111xxxxxxxx	Atura la màquina	n.s'a.

Ra: Qualsevol registre, veure aa.
Rb: Qualsevol registre, veure bb.
aa: Index del registre *src*: 00 - R0, 01 - R1, 10 - R2, 11 - R3
bb: Index del registre *dst*: 00 - R0, 01 - R1, 10 - R2, 11 - R3
cccccc: Constant de 6 bits en complement a 2, c $\in \{-32, \ldots, +31\}$.
dd: Constant de 2 bits en complement a 2, d $\in \{-2, -1, 0, +1\}$.
mmmmmmmm: Adreça de memòria de 8 bits.
x: Bit no utilitzat.
n.s'a.: No s'actualitzen.
s.v.r.: Segons el valor del resultat de l'operació.
s.v.R0: Segons el valor del registre R0 després d'executar l'operació.
s.v.Rb: Segons el valor del registre Rb després d'executar l'operació.

Taula 1: Conjunt d'instruccions de la PEPA4r. (NOTA: enrecordau-vos de fer **l'extensió de signe** de les constants **c**/**d** de 6/2 a 12 bits)

Encara que quan s'escriu en assemblador s'utilitzen noms simbòlics per denotar variables i adreces de bot, els programes s'han de traduir finalment a llenguatge màquina. En el cas de la PEPA4r, això vol dir que els programes que finalment haurà d'emular la màquina hauran de contenir adreces numèriques absolutes tant per especificar els bots com per especificar els operands de les instruccions LOA i STO. Així, a la figura 1, el programa de l'esquerra no es podria executar a l'emulador, sino que s'hauria de transformar en el programa de la dreta de la figura perquè l'emulador el poguès interpretar. Tal com es fa a aquest exemple, per passar a adreces absolutes suposarem que **el programa s'emmagatzema a partir de la posició 0 de la memòria** de la màquina emulada. Observa que en aquest exemple les dades s'emmagatzemen després de l'HLT.

NOTA: per facilitar la distinció entre tot el relatiu a la PEPA4r i el relatiu al 68K s'afegirà a partir d'ara el prefix "e" a tot el que pertany a la primera. Així, el registre R0 de la màquina emulada el denotarem ER0, els programes de la màquina emulada es diran eprogrames, etc.

Etiqueta	Assemblador amb etiquetes	Adreça	Assemblador sense etiquetes	Instruccions codificades	Hex
	SET #0,R0	0:	SET 0,R0	0111 000000 00	700
	LOA B	1:	LOA 15	0010 0000 1111	20F
	TRA R1,R2	2:	TRA R1,R2	0110 0000 0110	606
	JMZ EXIT	3:	JMZ 11	1100 0000 1011	C0B
	LOA A	4:	LOA 14	0010 0000 1110	20E
	TRA R1,R1	5:	TRA R1,R1	0110 0000 0101	605
	JMZ EXIT	6:	JMZ 11	1100 0000 1011	C0B
LOOP:	ADD R2,R0	7:	ADD R2,R0	0100 0001 1000	418
	ADQ #-1,R1	8:	ADQ -1,R1	0101 0000 1101	50D
	JMZ EXIT	9:	JMZ 11	1100 0000 1011	C0B
	JMI LOOP	10:	JMI 7	1110 0000 0111	E07
EXIT:	TRA R0,R1	11:	TRA R0,R1	0110 0000 0001	601
	STO C	12:	STO 16	0000 0001 0000	010
	HLT	13:	HLT	1111 0000 0000	F00
A:	8	14:	8	0000 0000 1000	008
B:	4	15:	4	0000 0000 0100	004
C:	0	16:	0	0000 0000 0000	000

Figura 1: Exemple de programa per la PEPA4r. Observa que aquest programa implementa l'operació $C = A \times B$, per $A \geq 0$ i $B \geq 0$. Per tant, després de la seva execució, C = $32_{(10}$.

Estructura del programa emulador

El programa que heu de dissenyar i escriure ha de començar amb una capçalera com la següent:

```
        ORG $1000
EPROG:  DC.W $0700, $020F, $0606, $0C0B, $020E, $0605, $0C0B, $0418, $050D
        DC.W $0C0B, $0E07, $0601, $0010, $0F00, $0008, $0004, $0000
EIR:    DC.W 0       ;eregistre d'instrucció
EPC:    DC.W 0       ;ecomptador de programa
ER0:    DC.W 0       ;eregistre R0
ER1:    DC.W 0       ;eregistre R1
ER2:    DC.W 0       ;eregistre R2
ER3:    DC.W 0       ;eregistre R3
ESR:    DC.W 0       ;eregistre d'estat (00000000 00000ZNC)
```

L'eprograma que es vulgui emular en cada cas s'introduirà com un vector de paraules (16 bits) a partir de l'etiqueta `EPROG`. Les einstruccions es codificaran en els 12 bits menys significatius de cada paraula, afegint zeros davant per completar fins als 16 bits. Comproveu que els *words* a partir de l'etiqueta `EPROG` de la capçalera es corresponen amb les einstruccions de l'eprograma d'exemple que hem posat abans tant amb etiquetes com sense. El vostre programa ha de ser capaç d'emular l'execució de qualsevol eprograma que s'escrigui codificat dins el vector

`EPROG`.

A més del vector que conté l'eprograma i les eposicions de memòria reservades per a dades i resultats, a la capçalera es reservaran una sèrie de *words* que al final de l'execució de cada einstrucció de l'eprograma contindran als seus 12 bits menys significatius el valor dels eregistres corresponents (la resta de bits han de ser 0). Així aquestes posicions s'anomenaran `EIR` (eregistre d'einstrucció), `EPC` (ecomptador d'eprograma), `ER0` (eregistre R0), `ER1` (eregistre R1), `ER2` (eregistre R2), `ER3` (eregistre R3) i `ESR` (eregistre d'estat). Aquest darrer tindrà als seus 3 bits menys significatius els *eflags* amb l'ordre indicat (`ZNC`).

El programa emulador que heu d'escriure haurà de ser una iteració que faci les següents passes per a cada einstrucció de l'eprograma:

1. Realitzar el *fetch* de la següent einstrucció a executar.
   - Fer servir el valor contingut a l'EPC per accedir a la següent einstrucció.
   - Emmagatzemar l'einstrucció a l'eregistre EIR.
   - Incrementar l'eregistre EPC.
2. Descodificar l'einstrucció.
   - Cridar una **subrutina de llibreria** que, a partir del codi de l'einstrucció, analitzarà de quina einstrucció es tracta i retornarà un valor numèric que la identifiqui.
3. Executar l'einstrucció.
   - Amb el valor numèric retornat per la subrutina de descodificació, botar a la posició on es duguin a terme les operacions sobre els eregistres i/o eposicions de memòria corresponents a la fase d'execució de l'einstrucció.
   - Retornar al principi del programa després d'actualitzar el valor dels eregistres i/o les eposicions de memòria pertinents.

Per tal de fer l'emulació correctament, a cada passa (1,2,3) els eregistres i les eposicions de memòria s'han d'actualitzar amb el valor que prendrien a la PEPA4r.

Subrutina de descodificació

La subrutina de descodificació ha d'**acomplir tots els requisits d'una subrutina de llibreria** indicats tant a les classes de teoria com a les pràctiques anteriors.

El pas de paràmetres s'ha de fer de la següent manera: en primer lloc, el programa principal ha de reservar una paraula (16 bits) a la pila per a que la subrutina pugui deixar el resultat; a continuació, el programa principal ha d'insertar a la pila el contingut de l'EIR; i finalment s'ha d'invocar la subrutina. La subrutina no ha de fer servir cap altre informació externa que

el paràmetre que se li passa. En cas de necessitar posicions de memòria extra per càlculs, les haurà de **reservar a la pila**.

Recordeu que, pel fet de ser de llibreria, la subrutina ha de salvar a la pila els registres que faci servir, per tal de poder recuperar el seu valor abans de tornar el control al programa principal. Després, ha d'analitzar el codi de l'einstrucció i retornar **un valor comprés entre 0 i 12**, d'acord amb la columna *id* de la taula 1: 0 en cas d'un STO, 1 en cas d'un LOA, ..., 12 en cas d'un HLT.

En acabar la descodificació, la subrutina ha de recuperar el valor dels registres modificats, deixar la pila tal com estava al principi de l'execució de la subrutina, posar el resultat de la descodificació al lloc corresponent de la pila, i finalment retornar al programa principal. Des del programa principal, s'ha d'eliminar de la pila el codi de l'einstrucció que s'ha descodificat (el paràmetre), i s'ha de recuperar el resultat. L'*stack pointer* ha de quedar tal com estava abans d'iniciar el pas de paràmetres abans d'invocar la subrutina. **Es recomana que tot el que fiqueu a la pila siguin paraules (16 bits) o paraules llargues (32 bits).**

Per iniciar la fase d'execució de l'einstrucció descodificada se farà servir el valor numèric retornat per la subrutina. Aquest valor s'ha d'introduir dins un registre, per exemple D1, i s'ha de modificar per tal de servir com a índex a la taula de bots que es mostra al llistat següent:

```
        MULU #6,D1
        MOVEA.L D1,A1
        JMP JMPLIST(A1)
JMPLIST:
        JMP ESTO
        JMP ELOA
        JMP ECMP
        JMP EADD
        JMP ESUB
        JMP ENAN
        JMP EADQ
        JMP ETRA
        JMP ESET
        JMP EJMZ
        JMP EJMN
        JMP EJMI
        JMP EHLT
```

A aquest llistat, les etiquetes `ESTO`, ..., `EHLT` indiquen les adreces on s'inicia la fase d'execució de les corresponents einstruccions. Així, l'únic que queda és escriure a continuació de cadascuna d'aquestes etiquetes el necessari perquè es duguin a terme les operacions que implica l'execució de cada einstrucció (taula 1). Observa que cada fase d'execució ha d'acabar amb un bot al principi de la iteració que executa les einstruccions d'EPROG, excepte quan s'executa la instrucció HLT. La fase d'execució d'aquesta einstrucció ha d'aturar la màquina, el qual és equivalent a finalitzar el programa emulador.

Especificació del treball a realitzar

- Heu d'implementar la descodificació de les einstruccions mitjançant una subrutina de llibreria **que faci el pas de paràmetres de la manera indicada i que s'anomeni DESPEPA4R**. Naturalment, la vostra subrutina ha de poder ser usada per qualsevol usuari si sap com es diu la subrutina (DESPEPA4R), com passar-li els paràmetres i d'on extreure el resultat. A més, **la subrutina ha de figurar al final del fitxer on entregueu el programa emulador**.

- Una vegada dissenyada la subrutina, heu d'implementar el programa emulador d'acord amb les especificacions de la PEPA4r (taula 1). Heu de dedicar una atenció especial a l'obtenció dels *eflags* correctes generats per la fase d'execució de les einstruccions. Se recomana no copiar directament els *flags* que genera l'emulador del 68K. Si feu això, podeu tenir sorpreses degut a les diferències entre les aritmètiques de 12 (PEPA4r) i 16 bits (68K).

- A les paraules de la memòria del 68K que es fan servir per representar eregistres o eposicions de memòria (12 bits) de la PEPA4r, hi ha 4 bits sobrants, els més significatius, que **hauran de ser 0 al final de l'execució de cada einstrucció per tal d'emular correctament la PEPA4r**.

- El fitxer, executable sobre l'emulador del 68K, tindrà per nom **PRAFIN15** i contindrà la subrutina i el programa amb les declaracions de variables que calguin, però **les primeres línies del fitxer hauran de ser inexcusablement les següents (substituïnt els noms dels autors), respectant el fet que les etiquetes estan en majúscules**.

```
*-----------------------------------------------------------
Title      : PRAFIN15
Written by : <noms complets dels autors>
Date       : 24/05/2015
Description: Emulador de la PEPA4r
*-----------------------------------------------------------
       ORG $1000
EPROG: DC.W $0700, $020F, $0606, $0C0B, $020E, $0605, $0C0B, $0418, $050D
       DC.W $0C0B, $0E07, $0601, $0010, $0F00, $0008, $0004, $0000
EIR:   DC.W 0       ;eregistre d'instrucció
EPC:   DC.W 0       ;ecomptador de programa
ER0:   DC.W 0       ;eregistre R0
ER1:   DC.W 0       ;eregistre R1
ER2:   DC.W 0       ;eregistre R2
ER3:   DC.W 0       ;eregistre R3
ESR:   DC.W 0       ;eregistre d'estat (00000000 00000ZNC)
```

Evidentment, el programa ha de funcionar correctament malgrat es modifiquin els valors continguts dins el vector EPROG (ha de funcionar per a qualsevol altre eprograma) i sense que sigui necessari modificar cap altre dada introduïda per vosaltres. Qualsevol referència a l'eprograma s'haurà de fer sempre mitjançant l'etiqueta EPROG.

Evidentment el programa tampoc ha de dependre de les adreces absolutes dels eregistres: sempre heu de fer referència a cada eregistre per la seva etiqueta. Això vol dir que sempre heu de fer referència a les variables a través de la seva etiqueta (EIR, EPC, etc.) i no a través de l'etiqueta d'una altra variable. En conseqüència, a la capçalera, **s'ha de poder modificar l'ordre en el qual apareixen els eregistres**, o introduir directives de tipus `DS.W 0`, i l'emulador ha de continuar funcionant correctament.

Lliurament i presentació de la pràctica

1. **La data límit de lliurament de la pràctica serà el dia 24 de Maig de 2015 (fins a les 23:55h)**. Cada grup de pràctiques ha de lliurar tant el programa com un informe de la feina feta. **Una còpia impresa de l'informe s'ha de lliurar al casiller del professor corresponent al grup** (Grup 1 [matí] - Emilio García, Grup 2 [horabaixa] - Javier Antich, Grup 3 [matí] - Alberto Ortiz; Consergeria A. Turmeda), **mentres que el fitxer electrònic de l'informe i el del programa executable s'han de lliurar a través de Campus Extens**. Les pràctiques lliurades després del 24 de Maig patiran una penalització a la seva nota del **10 %** per cada dia. Per tant, el **màxim retràs admissible és de 5 dies naturals**.

2. La còpia impresa de l'informe s'ha de presentar sense pàgines en blanc, enquadernacions, fundes, carpetes i sense ficar-lo en cap tipus de sobre. L'informe ha de contenir:

   - Una portada amb el nom de l'assignatura i el curs acadèmic, i els noms, DNIs i adreces de correu electrònic dels integrants del grup.
   - De tres a quatre pàgines explicatives sobre el treball realitzat, on se descrigui breument (per exemple, amb una o més taules): variables més importants; subrutines que s'han declarat, amb explicació de la interfície; registres que sempre se fan servir per la mateixa tasca, si hi ha qualcú; etc.
   - El llistat de la subrutina i del programa suficientment comentat (a alt nivell).

3. La pràctica quedarà automàticament suspesa en cas que no s'observin les restriccions anteriors.

4. Les **indicacions sobre estil de programació i documentació** incloses a la P3 s'haurien de respectar igualment en el cas d'aquesta pràctica: clarificar el codi font amb les corresponents indentacions, incloure comentaris útils (i no excessius), utilitzar noms d'etiquetes apropiats, evitar línees de codi i comentaris de més de 80 caràcters, etc.

5. D'acord amb la guia docent, **l'excessiva similitud entre pràctiques, o parts de pràctiques (p.e. la subrutina de descodificació), serà considerada còpia**, i ambdues pràctiques quedaran automàticament suspeses, així com, al manco, la present convocatòria de l'assignatura.

NOTA: Si hi ha canvis respecte d'aquest o altre apartat, es publicaran a la pàgina web de l'assignatura a Campus Extens.