

Pràctica 6

Objectius:

- a) Completar l'escriptura de l'emulador de la màquina elemental.
- b) Complementar els coneixements sobre el funcionament d'un processador.
- c) Preparar l'alumne per a la pràctica a entregar.

A les dues pràctiques anteriors s'ha iniciat l'escriptura d'un emulador de la màquina elemental amb les fases de *fetch* i de descodificació de les instruccions. A aquesta pràctica heu de completar el vostre emulador centrant-vos a la fase d'execució de les instruccions.

Especificació del programa que s'ha d'escriure durant aquesta sessió

Tal com es va dir a la pràctica 4, el programa emulador que heu de completar amb aquesta pràctica haurà de ser una iteració que faci les següents passes per a cada instrucció de l'eprograma.

1. Realitzar el *fetch* de la següent instrucció a executar.
2. Descodificar l'instrucció.
3. Executar l'instrucció.

Tenint en compte que ja hem implementat les dues primeres fases a les sessions anteriors, en aquesta pràctica completarem l'emulació de les instruccions. Quan la subrutina torna el control al programa principal, aquest inicia la fase d'execució de l'instrucció fent servir el valor numèric que ha rebut de la subrutina. En concret, aquest valor s'introduirà dins un registre, per exemple D1, i es modificarà 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 EJMI
JMP EJMZ
JMP EJMN
JMP ESTORE
JMP ELOAD
JMP ECMP
JMP EADD
```

```
JMP ESUB  
JMP ER1R0  
JMP ER0R1  
JMP EHALT
```

A aquest llistat les etiquetes EJMI, ..., EHALT indiquen les adreces on s'inicia la fase d'execució de les corresponents instruccions. Així l'únic que queda és escriure a continuació de cadascuna d'aquestes etiquetes el codi necessari perquè es facin les operacions que implica l'execució de cada instrucció (mireu de quines operacions es tracta a la taula que descriu el repertori de la màquina elemental a la pràctica 4).

Per a la correcta realització d'aquesta pràctica hi ha una sèrie d'aspectes que s'han de tenir presents:

- Per tal de fer l'emulació correctament, a cada passa (*fetch* de l'instrucció, descodificació i execució) els registres s'han d'actualitzar amb el valor que prendrien a la màquina elemental.
- Noteu que les dades de la màquina elemental són de 12 bits mentre que les posicions del 68K que farem servir per "emular" aquestes dades (per exemple, les posicions que emulen els registres com ara EIR) són de 16 bits. En conseqüència els 4 bits més significatius de la paraula de 16 bits a on emmagatzemeu una dada de 12 bits hauran de ser 0 al final de l'execució de cada instrucció.
- La vostra subrutina DESCO ha de ser de llibreria: ha de poder ser usada per qualsevol usuari si sap com passar-li els paràmetres i d'on extreure el resultat. A més, aquesta subrutina ha de figurar al final del fitxer del programa emulador.
- A la implementació de la fase d'execució de les diverses instruccions s'ha de dedicar una atenció especial a l'obtenció dels flags correctes. Tot el necessari està indicat a la taula de que resumeix el conjunt de instruccions a la pràctica 4. No heu de copiar directament els flags que genera l'emulador del 68K. Si confieu en els flags que genera l'emulador del 68K podeu tenir sorpreses degut a diferències entre les aritmètiques de 12 i 16 bits (el 68K fa les operacions amb 8-16-32 bits mentre que la màquina elemental les faria amb 12).
- Per tal d'actualitzar els valors dels flags, necessitareu fer servir noves instruccions de maneig de dades a nivell de bits. Consulteu el funcionament de les instruccions `bset`, `bclr` i `bchg` al repertori d'instruccions del 68K.

Obviament el vostre programa emulador no ha de dependre del valor concret de les adreces de les dades (per exemple de l'adreça d'ER0) a la memòria del 68K. Això vol dir que sempre heu de fer referència a les dades a través de la seva etiqueta i no a través de l'etiqueta d'una altra variable. En conseqüència s'ha de poder modificar l'ordre de les variables a la capçalera, o introduir-hi directives de tipus `DS.W 0` i que el vostre emulador funcioni correctament.

Evidentment el vostre emulador també ha de funcionar perfectament malgrat es modifiquin els valors continguts dins el vector `EPROG` (ha de funcionar per a qualsevol altre eprograma).

De fet heu de provar d'emular un conjunt d'eprogrames suficientment nombrós i variat com per poder tenir seguretat de que el vostre emulador funciona correctament. Per poder modificar l'eprograma no ha de ser necessari modificar cap altre valor introduït per vosaltres que no siguin els valors de les components del vector **EPROG**.