

Maquinari Mapat en Memòria

Estructura de Computadors II - Primera Pràctica

1 Introducció

En aquesta pràctica programareu en el llenguatge ensamblador del MC68000 emprant l'ensamblador i el simulador EASy68K per tal d'accedir a maquinari mapat en memòria.

L'objectiu de la pràctica és doble. D'una banda, s'espera que trebal·leu els conceptes propis de maquinari mapat en memòria i compregueu el funcionament, avantatges i limitacions d'aquesta forma d'accés a la perifèria. D'altra banda, s'espera que amplieu els vostres coneixements de programació en ensamblador emprant elements com ara etiquetes locals, constants, etcètera.

Per tal d'assolir els objectius esmentats haureu de dur a terme tres tasques. En primer lloc, haureu de programar quatre subrutines anomenades MAPINIT, MAPPRBIT, STR2SEG i BITPOS. En segon lloc, haureu d'organitzar aquestes subrutines dins d'uns arxius que se us proporcionen, configurant així un petit projecte. Finalment, haureu de comprovar el correcte funcionament del projecte emprant un programa que també se us proporciona i que fa ús de les subrutines.

A continuació es descriuen amb detall aquestes tasques.

2 Subrutines

Heu de crear quatre subrutines, anomenades MAPINIT, MAPPRBIT, STR2SEG i BITPOS. A continuació s'especifica cada una d'elles.

2.1 Subrutina MAPINIT

Aquesta subrutina ha de rebre com a paràmetres d'entrada dos registres (D0.B i A0). Ja que els paràmetres són per registre, no és una subrutina de biblioteca. Tot i això, ha de restaurar tots els registres que modifiqui. El funcionament de la subrutina ha d'esser el següent.

Si el bit 0 (el bit de menys pes) de D0.B val 1 aleshores la subrutina ha de mostrar la finestra de maquinari. Si el bit 0 val 0 aleshores no ho ha de fer. Això darrer no significa que si la finestra de maquinari ja està visible l'hagi d'ocultar, ja que no és possible ocultar-la per programa. Simplement significa que si el bit 0 val 0 la subrutina no intentarà mostrar la finestra però si aquesta ja era visible abans aleshores romandrà visible.

Si el bit 1 de D0.B val 1 aleshores la subrutina ha d'obtenir les adreces de mapatge dels quatre dispositius mapats en memòria d'EASy68K i les ha de guardar

| | | | | | | | | |
|-----|-----|-----|-----|-----|-----|-----|-----|-------|
| | | | | | | | | |
| A,a | B,b | C,c | D,d | E,e | F,f | G,g | H,h | I,i |
| | | | | | | | | |
| J,j | K,k | L,l | M,m | N,n | O,o | P,p | Q,q | R,r |
| | | | | | | | | SPACE |
| S,s | T,t | U,u | V,v | W,w | X,x | Y,y | Z,z | SPACE |

Figura 1: Grafemes a emprar per a cada una de les lletres de l'alfabet i per l'espai. Notau que no es fa distinció entre majúscules i minúscules.

en memòria. Si el bit 1 de D0.B val 0, aleshores no ho ha de fer. Les adreces s'han de guardar a partir de la posició de memòria apuntada per A0 en aquest ordre exacte: visualitzadors de set segments, LEDs, interruptors i pulsadors.

La resta de bits de D0.B (del 2 al 7) no tenen cap efecte. Per tant, no han d'afectar al funcionament de la subrutina de cap manera.

Notau que el funcionament descrit permet quatre situacions possibles. Així, si el bit 0 de D0.B val 0 i el bit 1 val 0 la subrutina no ha de fer res. Si el bit 0 de D0.B val 0 i el bit 1 val 1 la subrutina ha de guardar les adreces de mapatge tal i com s'ha indicat. Si el bit 0 de D0.B val 1 i el bit 1 val 0 la subrutina només ha de mostrar la finestra de maquinari. I si el bit 0 de D0.B val 1 i el bit 1 val 1 la subrutina ha de fer les dues coses: mostrar la finestra i guardar les adreces de mapatge.

2.2 Subrutina STR2SEG

Aquesta subrutina ha de rebre com a paràmetres d'entrada dos registres (A0 i A1). Ja que els paràmetres són per registre, no és una subrutina de biblioteca. Tot i això, ha de restaurar tots els registres que modifiqui. El funcionament de la subrutina ha d'esser el següent.

La subrutina, que ha de suposar que A0 apunta a una seqüència de 8 caràcters codificats en ASCII i que A1 conté l'adreça de mapatge dels visualitzadors de set segments, mostrarà l'esmentada seqüència de caràcters pels visualitzadors de set segments. És molt important que per tal d'accedir als visualitzadors de set segments empreu l'adreça de mapatge proporcionada a la subrutina dins A1.

Heu de suposar que la cadena de text només conté caràcters dins dels rangs A-Z (majúscules) i a-z (minúscules) a més de l'espai. Els grafemes concrets a mostrar a cada visualitzador de set segments en funció de la lletra es mostren a la Figura 1. Notau que, degut a les limitacions dels visualitzadors de set segments, alguns grafemes són difícils d'interpretar (com ara el corresponent a la lletra R), d'altres són idèntics per varies lletres (com ara els grafemes de la U, la V i la W o de la H, la K i la X) i d'altres es poden confondre amb altres lletres (com la M i la N o la O i la



Figura 2: Exemple de visualització de la cadena de text HOLAhola.

Q). Ara bé, això no és un problema que hageu de resoldre. Els grafemes a emprar han d'esser els indicats.

A mode d'exemple, una vegada hageu implementat la subrutina, el següent programa hauria de mostrar el que es veu a la Figura 2 pels visualitzadors de set segments. Notau que aquest programa pressuposa que els visualitzadors es troben mapats a \$E00000, cosa que no té per què ser així en un cas general.

```

                ORG        $1000
START
                LEA        .STR,A0
                MOVE.L     #$E00000,A1
                JSR        STR2SEG
                SIMHALT
.STR            DC.B      'HOLAhola'
                DS.W       0

```

2.3 Subrutina BITPOS

Aquesta subrutina ha de rebre com a paràmetre d'entrada un registre (D0.B) i retorna la seva sortida dins del mateix registre. Ja que els paràmetres són per registre, no és una subrutina de biblioteca. Tot i això, ha de restaurar tots els registres que modifiqui. El funcionament de la subrutina ha d'esser el següent.

La subrutina ha de tornar dins D0.B (per tant, sobreescrivint la pròpia entrada) un número entre 0 i 7 que indica la posició del primer bit a 1 de D0.B. Recordau que els bits es numeren de dreta a esquerra i de 0 a 7 i que, per tant, el bit 0 és el de més a la dreta.

Per exemple, si D0.B val 1 aleshores la subrutina ha de posar un 0 dins D0.B ja que 1 en binari és %00000001 i el bit que està a 1 és el bit 0. Si D0.B valgués 240, la subrutina hauria de posar dins D0.B un 4 ja que 240 en binari és %11110000 i el primer bit que està a 1 és el bit 4 (comptant a partir de la dreta).

2.4 Subrutina MAPPRBIT

Aquesta subrutina ha de rebre com a paràmetres d'entrada dos registres (A0 i A1). Ja que els paràmetres són per registre, no és una subrutina de biblioteca. Tot i això, ha de restaurar tots els registres que modifiqui. El funcionament de la subrutina ha d'esser el següent.

La subrutina ha de suposar que A0 i A1 contenen les adreces de mapatge dels botons i dels visualitzadors de set segments respectivament. La subrutina ha d'es-

| Botó pitjat | Text a mostrar |
|-------------|----------------|
| CAP | NONE |
| 0 | ZERO |
| 1 | ONE |
| 2 | TWO |
| 3 | THREE |
| 4 | FOUR |
| 5 | FIVE |
| 6 | SIX |
| 7 | SEVEN |

Taula 1: Textos a mostrar en funció del botó pitjat. El botó 0 és el de més a la dreta i el botó 7 és el de més a l'esquerra.

criure un text pels visualitzadors de set segments en funció de quin botó estigui pitjat tal i com indica la Taula 1.

El text s'ha de mostrar alineat a l'esquerra i s'han d'omplir amb espais els visualitzadors que no hagin de contenir cap lletra. La Figura 3 mostra alguns exemples del que ha de fer la subrutina.

Notau que la subrutina només comprova l'estat dels botons i escriu als visualitzadors si, abans de cridar-la, s'han posat dins A0 i A1 els valors pertinents i, en el moment de la cridada, el botó pertinent està pitjat. Si voleu fer proves haureu de cridar la subrutina dins d'un bucle infinit. Ara bé, la subrutina no ha de contenir el bucle infinit.

Notau també que estam parlant dels botons. Per tant, només n'hi pot haver un de pitjat a la vegada i no cal considerar els casos en que n'hi hagués més d'un.

Per tal de programar aquesta heu d'emprar BITPOS i STR2SEG. És a dir, MAPPRBIT ha de cridar, internament, a BITPOS i STR2SEG.

3 El projecte

Juntament amb aquest enunciat se us ha proporcionat un arxiu anomenat MATERIAL.ZIP. Si descomprimiu l'arxiu ho trobareu cinc arxius: PRAC1.X68, AUXILIAR.X68, MAP.X68, VARS.X68 i CONST.X68.

MAP.X68 ha de contenir les subrutines directament relacionades amb el maquinari mapat en memòria. És a dir, heu d'ubicar convenientment les subrutines MAPINIT i MAPPRBIT que heu fet dins d'aquest arxiu. Veureu que l'arxiu proporcionat ja conté les etiquetes i unes capçaleres amb comentaris. Podeu modificar la capçalera principal per indicar la vostra autoria i els canvis fets (HISTORY i COMMENTS).

AUXILIAR.X68 ha de contenir les subrutines auxiliars. En particular, heu d'ubicar convenientment les subrutines STR2SEG i BITPOS dins d'aquest arxiu. Veureu que l'arxiu proporcionat ja conté les etiquetes i unes capçaleres amb comentaris. Podeu modificar la capçalera principal per indicar la vostra autoria i els canvis fets (HISTORY i COMMENTS).



Figura 3: Exemple de funcionament de MAPPRBIT quan (a) no es pitja cap botó, (b) es pitja el botó 1, (c) es pitja el botó 3 i (d) es pitja el botó 7.

VAR.SX68 ha de contenir les variables (DS) que necessiteu. Com a mínim hi ha d'haver quatre variables on emmagatzemar les adreces de mapatge en l'ordre pertinent i, a més, una etiqueta anomenada MAPADDR que referencii l'inici d'aquest bloc de quatre variables. Si necessitau més variables, podeu declarar-les, però (en principi) no n'haurieu de necessitar més. Com de costum, podeu ampliar la capçalera de comentaris per a indicar la vostra autoria i els canvis fets.

CONST.SX68 ha de contenir les constants (EQU) que necessiteu. Com a mínim heu de definir MAPGETAD i MAPSHWHW amb els valors pertinents per a poder-los emprar abans de cridar MAPINIT. Mirau què es posa dins D0 abans de cridar MAPINIT a PRAC1.SX68.

PRAC1.SX68 és l'arrel del programa i no l'heu de modificar de cap forma. Observau com el programa té accés a tots els altres arxius ja que en fa INCLUDE.

Notau que aquest apartat de *Projecte* no heu de programar res. Simplement heu de copiar i enganxar les subrutines que ja heu programat dins dels arxius pertinents.

4 L'execució

Executau al programa PRAC1.SX68. Si ho feis una vegada heu ubicat subrutines, variables i constants correctament, veureu la finestra de maquinari i, quan pitjeu els distints botons, veureu els textos ZERO, ONE, ..., SEVEN o bé el text NONE si no se'n pitja cap.

Si no veis el que pertoca o teniu errors d'execució o d'assemblatge és que us heu

equivocat bé en la programació de les subrutines o bé quan les heu posat dins del projecte. Per tant, en aquest cas heu de revisar el que heu fet abans.

Si l'execució de PRAC1.X68 funciona correctament no significa que la pràctica estigui bé. Podria ser que un error n'emascaràs un altre. Ara bé, si PRAC1.X68 funciona correctament és molt probable que no tingueu cap error rellevant.

Notau que aquest apartat té per objectiu que pogueu comprovar si el que heu fet funciona correctament. No heu de programar res.

5 Entrega

L'entrega de la pràctica es farà a través de l'enllaç habilitat a tal efecte a l'Aula Digital de l'assignatura. Només un dels components del grup ha de fer l'entrega. L'entrega constarà d'un únic arxiu anomenat PRACTICA.ZIP. Descomprimir aquest arxiu ha de crear una única carpeta anomenada PRACTICA la qual haurà de contenir el següent al seu interior:

DOCUMENT.PDF La documentació de la pràctica en format PDF. Amb un màxim de tres pàgines heu d'explicar els trets més rellevants de cada una de les subrutines programades. La documentació ha de començar amb una introducció, on heu de descriure molt breument el problema a resoldre, i ha d'acabar amb unes conclusions, on he d'explicar els trets més rellevants i les dificultats més grans de la feina feta. A la primera pàgina del document també hi han d'aparèixer els vostre noms i DNIs.

SOLUCIO Dins la carpeta SOLUCIO hi ha d'haver els mateixos arxius que s'han adjuntat a MATERIAL.ZIP però ara ja complets. És a dir, hi ha d'haver el vostre codi. No s'acceptaran solucions amb més arxius dels que pertoqueu.

L'entrega es farà dins del termini establert. **Cada dia lectiu de retràs en l'entrega suposarà una penalització d'un punt en la nota de la pràctica.**

6 Avaluació

Aquesta pràctica *no recuperable* s'avaluarà en termes de funcionament general, compliment d'especificacions de cada una de les subrutines, qualitat i claredat del codi i documentació.

Si l'execució de PRAC1.X68 o d'alguna de les subrutines provoca errors d'assemblatge o d'execució (*Address or Bus Error* o similars) la pràctica no es corregirà i s'avaluarà amb 0 punts.

Un format d'entrega distint al descrit a la Secció 5 serà motiu de penalització.

Pel que fa a la documentació, s'avaluaran els continguts i l'estructura però també la correcció sintàctica, ortogràfica i gramatical. Recordau que la longitud màxima del document és de tres pàgines. Si superau aquesta extensió, les pàgines sobrants s'ignoraran.

Qualsevol còpia, total o parcial, de la pràctica suposarà una nota de 0.0 a tota l'assignatura sense opció a recuperació.