

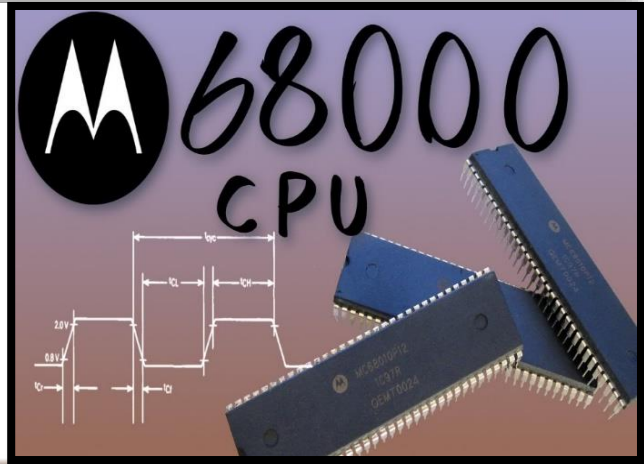
MAQUINARI MAPAT EN MEMÒRIA

Nom: Sergi Moreno Pérez

DNI: 45185320-A

Estructura de computadors II

Professor: Antoni Burguera Burguera



1. Introducció

L'objectiu de la feina era aprendre a treballar amb el maquinari mapat en memòria i ampliar els nostres coneixements en la programació en 68k, utilitzant algunes de les tècniques vistes enguany, com les etiquetes locals.

Per aconseguir-ho, havíem de programar 4 subrutines i situar-les a uns arxius que se'ns proporcionaven. MAPINIT i MAPPRBIT, situades al arxiu "MAP.X68"; i STR2SEG junt amb BITPOS a "AUXILIAR.X68". També l'enunciat ens proporcionava una arrel de programa "PRAC1.X68", que podíem utilitzar per comprovar el correcte funcionament de les subrutines i un arxiu de variables "VARS.X68" i un arxiu de constants "CONST.X68", necessàries per a l'execució del programa.

2. Subrutines

- MAPINIT

La subrutina rep per registre D0.B i A0 i al finalitzar haurà de restaurar tots els registres que hagi modificat.

Si el bit 0 de D0.B és 1, mostrarem la finestra de maquinari; en cas contrari, no la mostrarem. Si el bit 1 és 1, haurem d'obtenir les adreces de mapatge i guardar-les a la direcció de memòria apuntada per A0; en cas contrari, no les hauríem d'obtenir. Les opcions es poden combinar.

Primer salvem el valor que hi ha a D0 al registre D2. Així, si haguéssim de moure a D0 el valor de 32 decimal per mostrar la finestra de hardware amb la funció TRAP #15, el contingut inicial

no es veuria afectat, evitant possibles errors. Si la segona condició es compleix, mitjançant l'ús d'un vector local amb els valors necessaris per moure a D1, feim iteracions fins a trobar el valor de control del vector (0). Al bucle, obtindrem totes les adreces de mapatge a través de la funció TRAP #15 i les anirem emmagatzemant a la direcció de memòria guardada a A0, que s'anirà incrementant.

- STR2SEG

La subrutina rep per registre A0, que apuntarà a la primera char d'un vector de caràcters; i A1, que apuntarà a l'adreça de mapatge; i al finalitzar haurà de restaurar tots els registres que hagi modificat.

D0 comptabilitzarà les 8 iteracions que hem de fer per visualitzar l'output als displays de 7 segments. Carregam cada vegada la direcció del vector .VALUES a A2 i el codi ASCII del char apuntat en cada moment per A0 el mourem dintre de D1. A aquest valor li restarem 96 decimal, així si es tracta d'un caràcter minúscula ja sabrem a quina posició del vector de valors es troba i, en cas de que fos majúscula, posteriorment sumariem la diferència de codi entre minúscules i majúscules (32) per obtenir el mateix índex que en el cas de les minúscules. Pel cas del char ' ', també haurem fet la suma, però encara tindrem un valor negatiu al D1, així que no avançarem sobre el vector. Per la resta de caràcters, l'índex obtingut el sumam a l'adreça emmagatzemada a A2, avançant; i movem el nou valor al apunta A2 al byte menys significatiu de l'adreça de mapatge del display de set segments, emmagatzemada dintre A1. Seguidament s'augmenta la direcció d'A2 per així apuntar al següent display.

- BITPOS

La subrutina rep pel registre D0.B el byte menys significatiu de l'adreça de mapatge dels botons i retorna la posició del primer bit a 1 del byte. Excepte per D0, recupera tots els valors dels registres que hagi modificat.

Utilitzarem D1 per emmagatzemar la posició del byte comprovada, augmentant en 1 el seu valor a cada iteració. Posteriorment mourem el que hi ha a D1 a D0 per retornar el valor obtingut. En cas de que no hagi cap 1 al byte, sortirem del bucle i es retornarà el valor 8, significat que no ha estat pitjat cap botó.

- MAPPRBIT

La subrutina rep per registre A0 l'adreça de mapatge dels buttons i A1 l'adreça de mapatge del display de 7 segments. Finalment, recuperarà el valor dels registres que ha modificat.

A D0 emmagatzemem el byte més baix de l'adreça de mapatge dels buttons, on apareix el valor del botó que ha estat pitjat. Després feim una operació NOT del contingut i entrem dins la subrutina BITPOS per així saber quin dels botons s'ha pitjat. Una vegada retorna la posició del primer bit a 1, carregarem a A0 el vector de strings a visualitzar, multiplicarem el valor de D0 per 8, ja que cada string està formada per 8 chars; i sumarem el resultat a la direcció emmagatzemada a A0, posicionant-nos en el string correcte a representar.

3. Conclusions

Aquesta pràctica va parèixer, en un primer moment, molt complexa, a la que els meus coneixements actuals no arribaven i que ni sabia plantejar. Durant les classes, el que vàrem fer sobre mapatge ho vaig entendre, però una altra cosa era posar-ho en pràctica.

Vaig decidir anar tranquil·lament, intentant comprendre el que se me demanava a cada subrutina per separat, seguint l'ordre en el que apareixien a l'enunciat i fora fixar-me molt en l'optimització. Crec que fer això me va ser profitós, ja que amb la subrutina MAPINIT vaig poder refrescar els programes realitzats a classe i, junt amb els exercicis resolts que hi ha penjats, va resultar en l'empenta que necessitava per entendre un poc tot el funcionament de la finestra de hardware i poder començar a programar.

En quant STR2SEG, se me va ocórrer que la millor manera de tenir els valors que necessitam per representar cada caràcter requerit era a través d'un vector. Fent la codificació en hexadecimal de les representacions i el conjunt d'operacions pertinents, el que volia era obtenir l'índex adequat per fer el desplaçament sobre el vector de valors. Al final vaig fer optimitzacions, com la suma directa de l'índex obtingut al registre d'adreces on hi havia carregada la direcció inicial del vector, enlloc de fer un bucle més costós en temps d'execució.

Dels errors més recurrents, hi ha el CLR, que si no el feia del tamany adequat, en utilitzar-lo als bucles hem podien quedar bits residuals als bytes superiors, provocant error en l'execució.

Algun pic, en haver estat el primer cop que les usava, tapava amb etiquetes globals els vectors locals de la instrucció on se'l cridava.

Vaig haver de revisar certs conceptes vists a Estructura de Computadors I dels quals no recordava massa alguns aspectes rellevants per a la seva implementació, com per exemple quan necessitava certs tipus de branches com BMI o BGE, que sabia que existien, però fora recordar les seves condicions exactes.

També hem va ocórrer que no vaig codificar bé els valors de representació de la 'M' i del ' '. Per la 'M' simplement vaig repetir els càlculs i vaig resoldre l'error, però pel ' ' no vaig entendre que necessitava com a valor a la fi de no encendre cap segment del display. Un dubte un poc incoherent ja que en aquell moment ja havia codificat tota la resta de caràcters, per això que va ser una error de ràpida solució.

En general, no he trobat la pràctica massa difícil, sobretot en termes de programació ja que les noves instruccions i funcions que he necessitat implementar, les vaig aconseguir entendre en gran part durant les sessions presencials. Al meu cas, el que hem va preocupar més va ser l'enunciat, però amb tranquil·litat i amb els documents i programes resolts adients al costat, tot va anar sortint i així com anava programant les subrutines, anava guanyant pràctica, fluïdesa i confiança per a les següents.