

## Introducció.

L'objectiu d'aquesta pràctica és iniciar-nos en la manipulació de les eines de desenvolupament i programació de sistemes basats en microprocessadors o microcomputadors. En el transcurs de les sessions pràctiques ens centrarem bàsicament en el microcontrolador 8031/8051 d'Intel.

Les eines de les quals disposem són:

**Assemblador A8051:** Aplicació que ens permet codificar (traduir) instruccions escrites en format text (mnemònics i operands) en instruccions màquina. Aquestes instruccions són la "imatge" del codi que hi haurà ubicat a la memòria per ser executat. També ofereix la possibilitat de manegar símbols o etiquetes (que posteriorment seran traduïts a adreces físiques) i "macros" (agrupació de seqüències d'operacions i/o instruccions que seran "replicades" cada cop que aquesta sigui referenciada). Anomenarem mòdul a cadascuna de les "porcions" de codi descrites en llenguatge assemblador.

**Enllaçador XLINK:** Programa que permet enllaçar els diferents mòduls de programa i/o de llibreria a fi de generar un codi únic "executable". Cal remarcar que, a partir de diferents mòduls (objectes) generem un únic programa. Aquest programa és universal (a diferència del anterior), en el sentit que permet enllaçar programes per a diferents tipus de microprocessadors.

**Simulador SIM51:** Programa que permet la simulació de la execució de programes escrits en assemblador del 8051. Ens permet la visualització i la interacció amb les diferents àrees de memòria, registres interns, ports, així com l'execució de programes pas a pas amb els resultats d'execució de cada instrucció.

### 1-L'Assemblador A8051.

Cal escriure el codi font en llenguatge assemblador (format ASCII utilitzant un editor de textos del tipus EDIT del DOS, WordPad de Windows...). Per tal de simplificar el procediment d'assemblatge en aquesta primera pràctica, cal donar a aquest fitxer l'extensió ".S03" (S per "source", que vol dir font). Aquesta aplicació funciona sobre MS-DOS, per això, cal donar noms curts a tots els fitxers (8 caràcters pel nom i 3 per a la extensió, sense símbols "extranys"). Per executar-lo, el cridarem des de la línia del intèrpret de comandes (des de Windows "Inicio, Ejecutar, Command" o bé fer click sobre la icona del MS-DOS).

La comanda bàsica per assemblar en l'A8051 és:

*a8051 nomfitxer.s03 (retorn)*

En aquest cas, es genera el codi objecte amb extensió ".R03" (R per "relocatable" que vol dir re-ubicable). El format (què conté) d'aquest fitxer és "propietat" de IAR Systems i per això no és "accessible al usuari". "Funcionalment" conté el codi font "traduït" (codificat) a instruccions màquina amb les referències (adreces) relatives l'inici del mòdul.

També es pot executar aquest programa, escrivint simplement:

*a8051 (return)*

En aquest cas, el programa demana línia a línia:

El fitxer amb el codi font, el nom del fitxer que es vol donar al codi objecte, el nom del fitxer de llistat (si es desitja) i les opcions d'assemblatge. A la qüestió sobre les opcions d'assemblatge es pot contestar amb ? i el programa ens informarà sobre les opcions disponibles.

Les opcions, són:

-X: Per a incloure la taula de símbols (noms d'etiquetes, per exemple).

-P: Nombre de línies per pàgina, en el llistat.

-E: Si en el llistat només es desitgen els missatges d'error.

-F,-W: Fan referència al format del llistat.

-S: Si cal incloure informació simbòlica en l'objecte (per tal que pugui ser utilitzada en etapes posteriors).

En aquesta pràctica es pot utilitzar la comanda bàsica descrita anteriorment. Si no hi ha errors en el codi font, el programa genera un fitxer objecte amb el mateix nom que el fitxer del codi font amb l'extensió .r03.

## **2- L'enllaçador XLINK.**

Aquest programa presenta una gran quantitat d'opcions que permeten generar executables optimitzats segons diferents criteris. No oblidem que aquesta aplicació és per "agrupar" diferents mòduls objecte (que poden ser descrits en diferents llenguatges; ASM, C,...). A continuació es descriuen solament les comandes mínimes que permeten generar el codi executable a partir d'un fitxer objecte amb extensió ".R03". Tot i així, les opcions poden ser consultades escrivint:

*xlink (return)*

La comanda bàsica per generar el codi executable a partir del fitxer objecte, és:

*xlink nom1 -c8051 -o nom2 (return)*

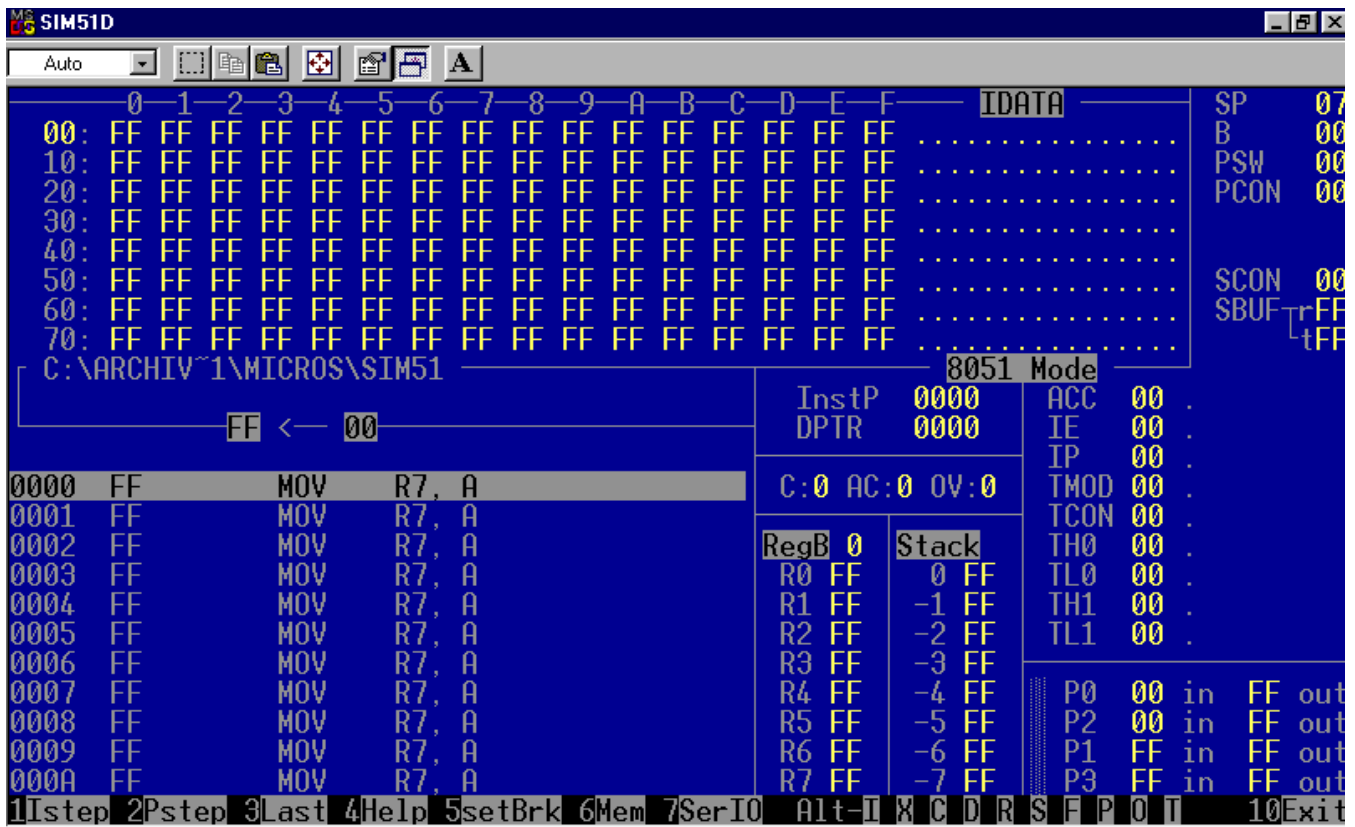
nom1 correspon al fitxer objecte i no cal que porti extensió; -c8051 permet escollir el tipus de microprocessador pel qual s'està realitzant el procés d'enllaçat; nom2 és el fitxer executable de sortida (que tindrà extensió .A03; A vé de "absolute"... ) i que per defecte el format és INTEL STANDART.

### 3-Simulador SIM51D.

Aquest programa permet la depuració del codi executable (format INTEL STANDART) sense necessitat de disposar del prototipus hardware basat en microcontrolador. Per cridar aquest programa cal escriure:

`sim51d (return)` (o bé `sim51` si el primer no carregués)

En la següent figura es mostra la pantalla principal del simulador. Tal com es pot veure, hi ha diferents finestres que es corresponen amb els recursos del microcontrolador 8051. En la part central de la pantalla hi apareix el prompt des del qual l'usuari pot entrar les comandes del programa.



Les diferents finestres que apareixen en el programa principal són:

*Finestra de memòria (IDATA):* Pot mostrar el contingut de la memòria de dades interna, memòria de dades externa (es suposa que es disposa de 64Kb) i memòria de programa.

*Finestra de codi:* Mostra una porció del codi que s'està executant. En vídeo invers apareix la instrucció que s'executa en cada moment.

*Finestra del àrea del SFR:* Es mostra el contingut dels registres de l'àrea del SFR del 8051.

*Finestra del InstP* (no confondre amb IP. En la família '51 IP NO és l'Instruction Pointer): Mostra el contingut del registre punter d'instruccions i del punter de dades DPTR (ocupa dos bytes).

*Finestra de registres:* Mostra quin banc de registres és actiu (RegB) i el seu contingut (R0..R7).

*Finestra de pila (Stack):* Mostra el contingut de la pila.

*Finestra de Flags:* Mostra l'estat en que es troben alguns dels flags del registre PSW.

*Finestra de ports:* Mostra el contingut dels registres associats als ports P0, P1, P2 i P3.

En la part inferior de la pantalla apareixen les tecles clau (hotkeys) i les funcions associades a la tecla Alt.

**Hotkeys:** *F1*-execució pas a pas; *F2*- execució pas a pas (tractant les subrutines com si fossin instruccions); *F3*-repetició de la última comanda; *F4*-Help (en alemany); *F5*-inicialització dels punts de ruptura del programa; *F6*-definició del mapa de memòria; *F7*-simulació d'operacions d'E/S utilitzant el port sèrie i *F10*-sortida.

**Funcions de la tecla Alt:** *Alt-I* mostra la memòria de dades interna en la finestra de memòria; *Alt-X* mostra la memòria de dades externa; *Alt-C* mostra la memòria de codi; *Alt-D* commuta el control de la finestra SFR; *Alt-R* commuta a la finestra de registres Ri; *Alt-S* commuta a la finestra de la pila; *Alt-F* commuta a la finestra de flags; *Alt-P* commuta a la finestra de ports; *Alt-O* commuta a la finestra del punter d'instruccions; *Alt-T* commuta a la finestra d'ajut per a la simulació de la transmissió/recepció sèrie de dades.

Les comandes bàsiques que poden ser entrades des del prompt del programa, són:

A: Permet entrar noves línies de codi en assemblador.

L: Permet carregar un arxiu de codi executable. Cal indicar l'extensió.

Amb aquest programa es pot canviar el contingut de qualsevol registre, posició de memòria o port només reescrivint el nou valor. En aquest cas es pot modificar fins i tot el registre contador de programa (normalment no accessible per part de l'usuari).

## Enunciat de la pràctica.

A partir de les eines descrites anteriorment es crearan i executaran (simulació) diferents programes escrits en llenguatge ensamblador del 8051. Per a tots ells caldrà ensamblar-los amb el "A8051", enllaçar-los amb el "Xlink" i simular (verificar) la seva correcta execució amb el "Sim51".

És convenient familiaritzar-se amb les diferents opcions de les utilitats, veure (i entendre) el procés que va des de la descripció d'un algorisme fins a la seva execució en un processador així com la representació (en memòria) i evolució dels elements que hi intervenen (codi, dades, variables, registres...).

També cal revisar els reports (l·listats) generats al llarg del procés.

Aconsellem (i així ho fem) seguir un nivell "ascendent" en quant a complexitat i mida dels programes a avaluar i per això podeu començar amb programes com:

```
                ORG 0                ;Vector de reset, aquí comença. 'Un bucle'.
Inici:          MOV  A,#10            ; Inicialitzo l'acumulador a 10
Bucle:          DEC  A                ;
                NOP                    ; "El que calgui"
                JNZ  Bucle            ; Itero
                END
```

Per seguir, podeu "ampliar l'àmbit" a registres interns i ports amb:

```
                ORG 0                ; 'Doble bucle aniuat'.
Inici:          MOV  R0,#4            ; Inicialitzo comptadors a 4 i 3
ExtLoop:        MOV  R1,#3
IntLoop:        MOV  A,R1             ; Posaré cada comptador en els "nibbles"
                SWAP A                 ; del port P1
                ANL  A,#0f0H           ; 'esborro' la part baixa
                ORL  A,R0              ; i 'afegeixo' el contingut de R0
                MOV  P1,A
                DJNZ R1,IntLoop        ; 3 iteracions
                DJNZ R0,ExtLoop        ; 4 iteracions
                END
```

Per acabar de "dominar" el '51 ens cal accedir a les diferents zones de memòria:

```
                ORG 0                ;Vector de RESET. 'Moguda trencaclosques'.
                AJMP inici            ;això és el que s'executa quan engeguem

                ORG 100
Inici:          MOV  DPTR,#0F000h     ;"apuntem" a la part alta de la memòria externa
                MOV  A,#0A5h
                MOVX @DPTR,A         ;hi escrivim tres "sentinelles"
                INC  A                ;recordeu que "A5" NO és OpCode
                INC  DPTR
                MOVX @DPTR,A
                INC  A
                INC  DPTR
                MOVX @DPTR,A
```

```

MOV  DPTR,#0F000h           ;Recollim els valors emmagatzemats anteriorment
MOVX  A,@DPTR
MOV   R0,A                 ;i els "desem" (copiem) a diferents llocs
INC   DPTR

MOV   R1,#90h              ;Preparo R1 per adreçar mode indirecte
MOVX  A,@DPTR
MOV   @R1,A                ;S'ha escrit el "??" a P1?
INC   DPTR

MOVX  A,@DPTR
MOV   80h,A                ;P0 és a la pos. 80h del SFR

AJMP  Inici                 ;Bucle sense fi per poder tornar a veure-ho.
END.

```

Arribats a aquest punt ja s'ou prou familiaritzats amb l'entorn. Coneixeu els recursos del processador, sabeu escriure programes i depurar-los. Enuncieu un problema i resoleu-lo. Cal que entregueu l'enunciat (de collita pròpia), el codi (ben comentat) i els problemes que heu tingut (i com els heu resolt).

## Recomanacions:

Treballeu en el directori "temp" del disc dur. Al començar la sessió feu:

"CD C:\TEMP(retorn)", "DEL \*.\*(retorn)" (no patiu) i

"COPY A:\.\* C:\TEMP\.\*(retorn)" (suposem que porteu les pràctiques en un disket).

Abans de marxar només caldrà fer "XCOPY C:\TEMP\.\* A:\.\*". Amb això evitarem problemes de "desconfiguracions" (que ningú alteri el sistema sota cap concepte).

Construiu-vos els vostres fitxers de "procés per lots" (i guardeu-los al directori temp). Podeu fer un fitxer de text (amb l'edit o el Wordpad) anomenat "ASM.BAT" que contingui:

```
C:\ARCHIV~1\ICCA\A8051.EXE C:\TEMP\%1.S03,%1.LST,X,F
```

Aquest fitxer cridat amb "ASM PRAC1" executa la línia anterior; això és, l'assemblador cridat amb el 'path' absolut, substituint la referència "%1" pel primer paràmetre (PRAC1). El resultat és el mateix que 'situar-se' ("CD [C:\Archivos](#) de programa\ICC") al directori on hi ha l'assemblador i executar 'A8051 PRAC1.S03'. Construiu tants fitxers com calgui per tal d'agilitzar les tasques d'assemblatge i enllaçat (en successives pràctiques seran imprescindibles) fent servir 'paths' absoluts. Sota cap concepte heu d'escriure al directori on hi ha les eines.

A la redacció d'aquest document encara no podem assegurar en quin directori es trobaran les eines instal·lades, el professor us ho indicarà en el seu moment.

Porteu les pràctiques "preparades" (llegides, enteses i potser "picades") i proveïu-vos de la documentació necessària (és difícil recordar tot el repertori d'instruccions).

Davant de qualsevol anomalia en el funcionament del material del laboratori, notifiqueu-ho al professor (no tingueu por en notificar "accidents", ens passen a tots) d'aquesta manera serà arranjat en el menor espai de temps possible.

## Annex I.

### Manual Simulador SIM51.

<b>A</b> <SSEMBLE> <addr><inst>	Assemblador propi.
<b>AE</b>	Assemblador i informació de registres.
<b>B</b> <REAKPOINT> <OFF>	Punts d'aturada.
<b>DC</b> <addr>	Visualitzar àrea memòria codi.
<b>DD</b> <addr>	Visualitzar àrea memòria dades interna.
<b>DX</b> <addr>	Visualitzar àrea memòria dades externa.
<b>DB</b> <addr>	Visualitzar àrea memòria dades bit.
<b>DR</b>	Visualitzar registre recepció sèrie.
<b>DT</b>	Visualitzar registre emissió sèrie.
<b>EC</b>	Editar valors àrea codi.
<b>ED</b>	Editar valors àrea dades interna.
<b>EX</b>	Editar valors àrea dades externa.
<b>EB</b>	Editar valors àrea dades bit.
<b>ER</b>	Editar valor registre recepció sèrie.
<b>ET</b>	Editar valor registre emissió sèrie.
<b>G</b> <O> <addr1> <addr2>	Executar el programa.
<b>L</b> <IST><addr1> <addr2>	Desassemblar.
<b>L</b> <OAD> nomfitxer<.ext>	Carregar programa.
<b>M</b> <AP>	
<b>PD</b> port valor	Estat dels ports.
<b>Q</b> <UIT>	Sortir.
<b>R</b> <EAD> nomfitxer<.ext>	Carregar programa.
<b>R</b> <EGISTERS>	Valors dels registres.
<b>RESET</b>	
<b>SAVE</b> nomfitxer<.ext>	Salvar programa.
<b>S</b> <TEP>	Executar pas a pas.
<b>W</b> <RITE> nomfitxer<.ext>	Salvar programa.