

Informe de la pràctica #6

Introducció

En aquesta practica aprofundirem en el simulador i8085 mitjançant diversos exercicis diferents dividits en tres parts diferents.

Objectius de la primera part:

- Estudiar el espai de memòries del simulador.
- Comprendre el funcionament de la pila.
- Veure un exemple de funcionament de les subrutines.

Objectius de la segona part:

- Comprendre el direccionament a ports E/S.

Objectius de la tercera part:

- Dissenyar un programa en ensamblador de forma autònoma.

Exercicis

Primera Part

En aquesta primera part, es proporciona un codi i es demana que l'executem:

```
.define  
    num 02h  
  
.data 0b  
    mat1: db 1,2  
    mat2: db 3,4  
    mat3: db 0,0  
  
.data 40  
    pila: db 9,8
```

```
.org 600H
    LXI H, pila
    SPHL
    MVI A,23
    SUI 42
    MVI B, num
    LXI D, mat1
    LXI H, mat2
suma:
    call sumar
    DCR B
    jnz suma
NOP
HLT

sumar:
    PUSH PSW
    LDAX D
    ADD M
    STAX D
    INX H
    INX D
    POP PSW
ret
```

Pregunta 1: L'adreçament de la instrucció LXI és: *Inmediat*.

Pregunta 2: Quina instrucció guarda el PC en la pila? *PUSH PC*.

Pregunta 3: Quin espai en memòria ocupa la subrutina sumar?

Ocupa des de la direcció 0619h fins a la direcció 0620h (amb RET inclòs).

Pregunta 4: Quants cicles triga la subrutina sumar en executar-se?

Triga uns 52 cicles de rellotge en executar-se.

Segona Part

En aquesta segona part es proporciona un codi curt que conté una subrutina "puertos":

```
.data 30

pila:

.org 500H
    LXI H, pila
    SPHL
    call puertos
NOP
HLT
```

```
puertos:
    PUSH PSW
    in 00h
    ANI 00000001
    out 05h
    POP PSW
ret
```

Pregunta 1: Que fa la subrutina “puertos”?

Primerament, i després de guardar el estat dels registres a la pila amb PUSH PSW, agafa el port IN 00h (es a dir, el teclat) i el guarda al acumulador, fà la operació lògica ANI entre el valor del Acc i el bit que s'introdueix 01h, i finalment, es passa al port de sortida 05h i es guarda.

Tercera Part

A la tercera part es demana que creem un codi en ensamblador que representi els numeros del zero al cinc que premem en el teclat en una pantalla de set segments, amb la condició que, si premem la tecla “c”, es faci un CLEAR al display, netejant la pantalla.

```
.define

    cero 77h
    uno 44h
    dos 3Eh
    tres 6Eh
    cuatro 4Dh
    cinco 6Bh

.data 40h
    buffer: db 00h, 07h

.org 100h
inici:
    JMP inici

borrado:
    MVI A, 00h
    OUT 00h
    SHLD buffer
    JMP inici

ncero:
    MVI A, cero
    OUT 00h
    SHLD buffer
    JMP inici
```

```
nuno:
    MVI A, uno
    OUT 00h
    SHLD buffer
    JMP inici
ndos:
    MVI A, dos
    OUT 00h
    SHLD buffer
    JMP inici
ntres:
    MVI A, tres
    OUT 00h
    SHLD buffer
    JMP inici
ncuatro:
    MVI A, cuatro
    OUT 00h
    SHLD buffer
    JMP inici
ncinco:
    MVI A, cinco
    OUT 00h
    SHLD buffer
    JMP inici

interrupcio:
    LHLD buffer
    IN 00h
    MOV M, A
    SUI 30h
    JZ ncero
    ADI 30h
    SUI 31h
    JZ nuno
    ADI 31h
    SUI 32h
    JZ ndos
    ADI 32h
    SUI 33h
    JZ ntres
    ADI 33h
    SUI 34h
    JZ ncuatro
    ADI 34h
    SUI 35h
    JZ ncinco
    ADI 35h
    SUI 63h ;VALOR HEXADECIMAL PER LA TECLA C
    JZ borrado

ret

HLT

.org 24h
    JMP interrupcio
```

En aquest codi guardem en un “.define” els valors hexadecimals per al visor de set segments dels dígit que volem representar. Tenint un bucle infinit “inici” només s'interrompeix mitjançant una interrupció TRAP, quan es prem una tecla pel teclat. En aquest punt, la interrupció, ubicada a la posició 24h, salta al punt de codi “interrupcio”, on s'evalua mitjançant comparacions, el valor ASCII del dígit introduït, i salta a la seva subrutina corresponent, on es carrega al acumulador els valors previament definits (“cero”, “uno”, “dos”, “tres”, “cuatro” i “cinco”) i els mostra a la pantalla de set segments amb el “OUT 00h”. Finalment, si es prem la tecla “c”, el display fa un CLEAR, deixant la pantalla buida.

Una versió diferent de aquest programa, fent servir com a display la pantalla de text, mostra qualssevol tecla del teclat:

```
.data 40h
    buffer: db 00h, E0h

.org 100h
inici:
    JMP inici

interrupcio:
    LHLD buffer
    IN 00h
    MOV M, A
    INX H
    SHLD buffer
ret
HLT

.org 24h
    JMP interrupcio
```

Conclusions

En aquesta practica de tres parts hem après i dominat el funcionament dels diferents elements del simulador i8085:

- Hem après el funcionament total de les subrutines en ensamblador.
- Hem après el funcionament de la pila i com guarda els valors dels registres (PC entre altres).
- Hem après a fer servir i comunicar-nos amb dispositius i interfícies de E/S en ensamblador.
- Hem aprofundit el coneixement dels espais de memòria dels processadors i direccionaments.