

Practica 7

Objectius

Programar en ensamblador del 8085 varies aplicacions, demostrant els coneixements adquirits en teoria.

Explicació de la practica

En aquesta practica s'havien de fer 3 exercicis, el primer consistia en un programa que fes una suma, el segon que fes una resta i l'últim que fes les operacions aritmètiques i lògiques suma , resta, AND i OR.

Teoricopràctic 7:

1) Una memòria SDRAM és una memòria

Trieu-ne una:

- ☐ a. Una memòria reconfigurable, que pot treballar com estàtica o dinàmica
- ☒ b. és una memòria volàtil
- ☐ c. Totes les opcions són certes
- ☐ d. és una memòria que no requereix refresc

2) Quina de les següents memòries no requereix refresc

Trieu-ne una:

- ☐ a. DDR4
- ☒ b. DRAM
- ☐ c. SRAM
- ☐ d. SDRAM

3) La unitat mínima de informació al disc dur utilitzada pel disc dur és

Trieu-ne una:

- ☐ a. el cilindre
- ☒ b. el sector
- ☐ c. la pista
- ☐ d. el cluster

4) Una memòria cache

Trieu-ne una:

- ☐ a. és una memòria DRAM amb etiquetes
- ☒ b. és una memòria de tipo flash integrada en la CPU
- ☐ c. és una memòria associativa
- ☐ d. és una memòria ROM

5) Quin dels següents tipus de memòria es fa servir en una cache

Trieu-ne una:

- ☐ a. DDR
- ☐ b. SDRAM
- ☒ c. SRAM
- ☐ d. DRAM

6) Una memòria cache té una eficiència del 80% i un temps d'accés de 1 nseg. La memòria principal té un temps d'accés de 10 nseg. Quin és el temps d'accés mitjà?

Trieu-ne una:

- ☐ a. 10 nseg
- ☐ b. 1,8 nseg
- ☐ c. 11nseg
- ☒ d. 3 nseg

$$Ta_{mitja} = 1 \cdot 0.8 + (1 + 10) \cdot (1 - 0.8) = 3 \text{ ns}$$

7) Una memòria cache té un temps d'accés de 0,5 nseg amb una probabilitat d'encert del 75%. Com a memòria principal tenim una DRAM de 8GB amb un temps de 20 nseg. Quin és el temps d'accés mitjà?

Trieu-ne una:

- ☐ a. 0,5 nseg
- ☒ b. 5,5 nseg
- ☐ c. 20,5 nseg
- ☐ d. 25 nseg

$$Ta_{mitja} = 0.5 \cdot 0.75 + (0.5 + 20) \cdot (1 - 0.75) = 5.5 \text{ ns}$$

8) Una memòria cache té una taxa de error del 20%. El temps d'accés és de 0.85 nseg. La memòria principal corresponent al següent nivell té un temps d'accés de 10 nseg. Quin serà el temps mitjà d'accés a la memòria?

Trieu-ne una:

- ☐ a. 0,85 nseg
- ☐ b. 10 nseg
- ☐ c. 98,85 nseg
- ☒ d. 2,85 nseg

$$Ta_{mitja} = 0.85 \cdot (1 - 0.2) + (0.85 + 10) \cdot 0.2 = 2.85 \text{ ns}$$

9) Una memòria RAM té un decodificador de files de 10 bits i un decodificador de columnes de 10 bits. Si la unitat mínima de memòria és de 32 bits, quina és la capacitat de la memòria?

Trieu-ne una:

- ☐ a. 1 MByte
- ☐ b. 32 kbits
- ☐ c. Totes les respostes són incorrectes
- ☒ d. 4 MBytes

$$M = 2^{10+10} \cdot 2^5 = 2^{25} = 2^2 \cdot 2^{20} \cdot 2^3 = 4 \text{ MByte}$$

10) Calcula la mida de una memòria que té un descodificador de files de 10 bits, un descodificador de columnes de 8 bits i una unitat mínima de memòria de 16 bits.

Trieu-ne una:

- ☐ a. No es pot calcular. Desconec la mida del bus d'adreces i la mida del bus de dades
- ☐ b. 32 kBytes
- ☒ c. 4Mbits
- ☐ d. 4 kBytes

$$M = 2^{10+8} \cdot 2^4 = 2^{22} = 2^2 \cdot 2^{20} \cdot 2^0 = 4Mbits$$

Informe:

Part Guiada

1. Introducció de dades per consola i mostra de les dades pantalla

Quina és la funció del parell de registres BC en aquest codi?

La funció del parell de registres BC es apuntar a la pantalla que esta en la posició de memòria E000h, d'aquesta manera si es vol mostrar alguna lletra o numero per pantalla nomes s'haurà de fer un stax B i es mostrarà el contingut de l'acumulador per pantalla.

Part no guiada:

1. Suma de dos valors introduïts per consola (cal entregar)

Dissenyeu una subrutina que a partir de dos nombres (base 10) introduïts pel teclat del simulador i8085 faci la suma i presenti el resultat en la pantalla de text del i8085. Feu servir els adreçaments directe i indirecte i indiqueu al codi on tenim aquests adreçaments.

Tasca 1. Pugeu el codi. Com gestioneu el problema del signe? Com gestioneu el problema del overflow?

Quan es fa la suma, pot ser que el resultat sigui més d'una xifra, per això al fer la suma es resta Ah (10) al resultat, si dona negatiu significa que només tenim una xifra, per tant s'haurà de convertir en resultat en codi ASCII sumant-li 30h més Ah que hem restat prèviament, 3Ah i es mostrarà per pantalla. Si després de restar Ah dona positiu significarà que hi ha overflow y tenim un carry, si això succeeix es crida a la subrutina carry on es gestionarà i es mostrarà el resultat amb 2 xifres.

```

.org 50h
    pila:
.org 100h
    lxi H, pila
    sphl

    LXI H, E000h
    JMP loop

.org 600h
    loop:
    JMP loop

.org 24h
    IN 00h
    MOV M, A
    INX H
    CPI 3Dh
    JZ guardar
    JMP loop
    JMP loop

;Subrutina que guarda el valores de memoria als registres i escull l'operacio que es
;realitzara amb els valors dels registres
.org 200h
guardar:
    MOV B, H
    MOV C, L

    DCX B
    DCX B

    LDAX B
    SUI 30h
    MOV E, A

    DCX B
    DCX B
    LDAX B
    SUI 30h
    MOV D, A
    INX B

    LDAX B
    CPI 2Bh
    JZ suma

```

```

;Subrutina que realitza la suma
.org 250h
suma:
    MOV A, D
    ADD E

    SUI Ah
    JP carry

    ADI 3Ah
    MOV M, A
    INX H

    MVI A, 3Bh
    MOV M, A
    INX H

    JMP loop

;Subrutina que gestiona el resultat si hi ha un carry en la suma
.org 300h
carry:
    MVI M, 31h
    INX H
    ADI 30h
    MOV M, A
    INX H

    MVI A, 3Bh
    MOV M, A
    INX H

    JMP loop

```

2. Resta de dos valors introduïts per consola (cal entregar)

Dissenyeu una subrutina que a partir de dos nombres (base 10) introduïts pel teclat del simulador i8085 faci la resta i presenti el resultat en la pantalla de text del i8085. Feu servir els adreçaments directe i indirecte i indiqueu al codi on tenim aquests adreçaments.

Tasca 2. Pugeu el codi de la resta. Com gestioneu el problema del signe? I el problema del carry?

Si el resultat de la resta dona positiu només s'imprimeix, però si dona negatiu, es crida a la subrutina signe, que posa el signe primer i després el resultat numèric amb 1 xifra. En aquest cas no ens hem de preocupar per l'overflow ja que el resultat no donarà 2 xifres.

```

.org 50h
    pila:
.org 100h
    lxi H, pila
    sphl

    LXI H, E000h
    JMP loop

.org 600h
    loop:
    JMP loop

.org 24h
    IN 00h
    MOV M, A
    INX H
    CPI 3Dh
    JZ guardar
    JMP loop
    JMP loop

;Subrutina que guarda el valores de memoria als registres i escull l'operacio que es
;realitzara amb els valors dels registres
.org 200h
guardar:
    MOV B, H
    MOV C, L

    DCX B
    DCX B

    LDAX B
    SUI 30h
    MOV E, A

    DCX B
    DCX B
    LDAX B
    SUI 30h
    MOV D, A
    INX B

    LDAX B
    CPI 2Dh
    JZ resta

```

```

;Subrutina que fa la resta i es comprova si el resultat es positiu o negatiu
.org 250h
resta:
    MOV A, D
    SUB E
    JM signe

    ADI 30h
    MOV M, A
    INX H

    MVI A, 3Bh
    MOV M, A
    INX H

    JMP loop

;Subrutina que gestiona el resultat si dona negatiu
.org 300h
signe:
    MOV D, A
    MVI A, 0h
    SUB D
    ADI 30h
    MVI M, 2Dh
    INX H

    MOV M, A
    INX H

    MVI A, 3Bh
    MOV M, A
    INX H

    JMP loop

```

3. Ensamblant el codi (cal entregar)

A partir dels codis generats en els apartats 1 i 2, feu un programa capaç de fer sumes, restes, AND's i OR's

Tasca 3. Pugeu el codi final.

Qüestió 1:

Quina diferència hi ha entre la suma i la OR?

i) són iguals

ii) la OR és una operació lògica i la suma és una operació aritmètica.

iii) La OR és una operació aritmètica i la suma és una operació lògica

iv) cap de les anteriors és correcta.

Qüestió 2:

La instrucció STA 1234h

i) és una operació que carrega el contingut de la posició de memòria 1234h en l'acumulador

ii) fa servir adreçament directe

iii) fa servir adreçament immediat

iv) totes són certes

Codi:

```
.org 50h
    pila:
.org 100h
    lxi H, pila
    sphl

    LXI H, E000h
    JMP loop

.org 600h
    loop:
    JMP loop

.org 24h
    IN 00h
    MOV M, A
    INX H
    CPI 3Dh
    JZ guardar
    JMP loop

;Subrutina que guarda el valors de memoria als registres i escull
;l'operacio que es realitzara amb els valors dels registres
.org 200h
guardar:
    MOV B, H
    MOV C, L

    DCX B
    DCX B

    LDAX B
    SUI 30h
    MOV E, A

    DCX B
    DCX B
    LDAX B
    SUI 30h
    MOV D, A
    INX B
```

```
LDAX B
CPI 2Bh
JZ suma
CPI 2Dh
JZ resta
CPI 26h
JZ and
CPI 7Ch
JZ or
```

```
;Subrutina que realitza la suma i comprova si hi ha overflow
.org 250h
suma:
```

```
MOV A, D
ADD E
SUI Ah
JP carry
ADI 3Ah
MOV M, A
INX H

MVI A, 3Bh
MOV M, A
INX H

JMP loop
```

```
;Subrutina que gestiona el resultat si hi ha overflow
.org 300h
carry:
```

```
MVI M, 31h
INX H
ADI 30h
MOV M, A
INX H

MVI A, 3Bh
MOV M, A
INX H

JMP loop
```

```
;Subrutina que fa la resta i es comprova si el resultat es positiu o
negatiu
.org 350h
resta:
```

```
MOV A, D
SUB E
JM signe

ADI 30h
MOV M, A
INX H

MVI A, 3Bh
MOV M, A
INX H
```

```

        JMP loop

;Subrutina que gestiona el resultat si dona negatiu
.org 400h
signe:
        MOV D, A
        MVI A, 0h
        SUB D
        ADI 30h
        MVI M, 2Dh
        INX H

        MOV M, A
        INX H

        MVI A, 3Bh
        MOV M, A
        INX H

        JMP loop

;Subrutina que realitza la operacio logica OR
.org 450h
or:
        MOV A, D
        ORA E
        SUI Ah
        JP carryor
        ADI 3Ah
        MOV M, A
        INX H

        MVI A, 3Bh
        MOV M, A
        INX H

        JMP loop

;Subrutina que gestiona el resultat de l'OR quan hi ha overflow
.org 500h
carryor:
        MVI M, 31h
        INX H
        ADI 30h
        MOV M, A
        INX H

        MVI A, 3Bh
        MOV M, A
        INX H

        JMP loop

;Subrutina que realitza la operacio logica AND
.org 550h
and:
        MOV A, D
        ANA E
        ADI 30h
        MOV M, A

```

```
INX H

MVI A, 3Bh
MOV M, A
INX H

JMP loop
```

Conclusió

En aquesta practica he m'he he familiaritzat mes amb el codi del i8085 i amb el funcionament de les interrupcions. A més de treballar i aprendre a utilitzar els dispositius d'entrada/sortida.