

Informe de la pràctica 5: Introducció al simulador i8085

Introducció

Després de veure el simulador SiMR, donem un pas endavant i entrem amb contacte per primera vegada amb el simulador i8085. En aquesta entrega, es pretén conèixer els modes d'adreçament del 8085 i familiaritzar-se amb les seves instruccions.

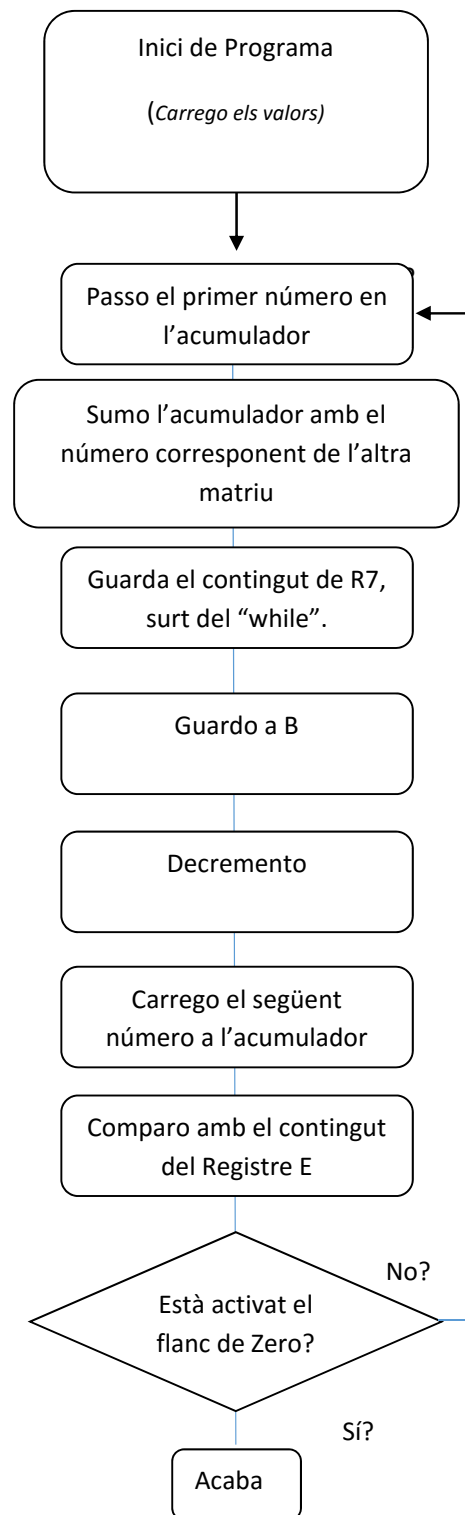
Base Teòrica

Per realitzar la pràctica el professor farà una petita introducció del Simulador 8085, es seguiran els passos per comprendre el funcionament del simulador. Es faran dos exercicis, el primer tracta de suma de matrius i el segon tracta de subrutines.

Exercicis

1. Escriviu dos programes diferents per sumar dues matrius de 5x1 i desar el resultat en una tercera, podeu fer servir adreçament indirecte, per parella de registres o per registres. El resultat el desem sobre una de les dues matrius d'entrada, mat3 és irrellevant. El programa hauria d'utilitzar un comptador, instruccions d'increment, comparació i salt condicional. Feu la suma de números hexadecimals i sense tenir en compte números que pugui donar overflow.

```
.data 00h
    mat1: db 1,2,3,4,5
    mat2: db 6,7,8,9,0
    mat3: db 0,0,0,0,0
.org 100h
    LXI D,mat1 ;carrego el primer valor de mat1 al registre D (10 cicles)
    LXI H,mat2 ;carrego el primer valor de mat2 al registre H (10 cicles)
    LXI B,mat3 ;carrego el primer valor de mat3 al registre B (10 cicles)
loop:
    LDAX D ;Guardo en acumulador el contingut del registre D (7 cicles)
    ADD M ;Sumes l'acumulador amb el que hi ha a B i ho guardes a l'acumulador (4 cicles)
    STAX B ;instrucció que guarda en l'acumulador la posició de memòria direccionalada pel parell
de registres BC (7 cicles)
    INX D ;incrementa D (6 cicles)
    INX H ;incrementa H (6 cicles)
    INX B ;incrementa B (6 cicles)
    MVI A,mat2 ;Carrego mat2 a l'acumulador (10 cicles)
    CMP E ;Comparo l'acumulador amb el contingut d'E (4 cicles)
    JNZ loop ;salta si no es zero (Entre 7 i 10 cicles)
    //JMP subrutina
HLT ;acaba (4 cicles)
```



- a. *En què simplificaria molt el codi del programa un dels modes d'adreçament del simulador de la MR?*

El programa creat i proposat pel professor fa ús d'un direccionament indirecte que és útil i simplifica el codi perquè es carrega segons el que hi ha en H i es treballa sobre aquesta.

- b. *Quants cicles de rellotge triga en executar-se una instrucció aritmètic – lògica del qualsevol?*

Una instrucció qualsevol pot arribar a trigar màxim 10 cicles i mínim 4 cicles de rellotge.

- c. *Pujeu el vostre codi i marqueu quina és la instrucció del vostre programa que triga més cicles en executar-se*

He marcat en negreta en l'explicació del codi els cicles de cada instrucció

2. *Fent ús del programa anterior, realitzeu una subrutina que codifiqui una zona de memòria. La zona de memòria s'indicarà posant al registre doble HL l'adreça de començament de la zona de dades a codificar. Aquestes dades es consideren com els paràmetres d'entrada de la subrutina. La codificació es farà mitjançant una XOR entre cada byte de la zona de dades i la clau. Els valors dels registres no han de quedar afectats per la crida a la subrutina. Feu un programa que faci ús d'aquesta subrutina per provar-la amb diferents combinacions de valors de la clau i les dades per codificar.*

El que faig es crido a la subrutina "subrutina" abans de que acabi el meu programa anterior (Posat en // en el codi anterior).

```
.define
    clau 8Bh
.data 00h
    mat1: db 1,2,3,4,5
    mat2: db 6,7,8,9,0
    mat3: db 0,0,0,0,0
.org 100h
    LXI D,mat1 ;carrego el primer valor de mat1 al registre D
    LXI H,mat2 ;carrego el primer valor de mat2 al registre H
    LXI B,mat3 ;carrego el primer valor de mat3 al registre B
loop:
    LDAX D ;Guardo en acumulador el contingut del registre D
    ADD M ;Sumes l'acumulador amb el que hi ha a B i ho guardes a l'acumulador
    STAX B ;instrucció que guarda en l'acumulador la posició de memòria
    direccionada pel parell de registres BC
    INX D ;incrementa D
```

INX H ;incrementa H
INX B ;incrementa B
MVI A,mat2 ;Carrego mat2 a l'acumulador
CMP E ;Comparo l'acumulador amb el contingut d'E
JNZ loop ;salta si no es zero
JMP subrutina

subrutina:

LXI H, mat1 ;carrego al registre H
MVI B, clau ;Carrego la clau a B
MVI C, 00h ;Inicialitzo el registre
MVI D, 15D ;Moc a D les posicions que es calcularan

calcul:

MOV A, M ;Poso el primer valor a l'acumulador
XRA B ;Faig l'operació XOR entre B i l'acumulador
INR C ;Incremento C
INX H ;Incremento H
MOV A, C ;Carrego el següent valor del registre C a l'acumulador
CMP D ;Comparo l'acumulador amb el valor del registre D

JNZ calcul ;Saltare quan l'operació anterior doni zero

HLT ;acaba

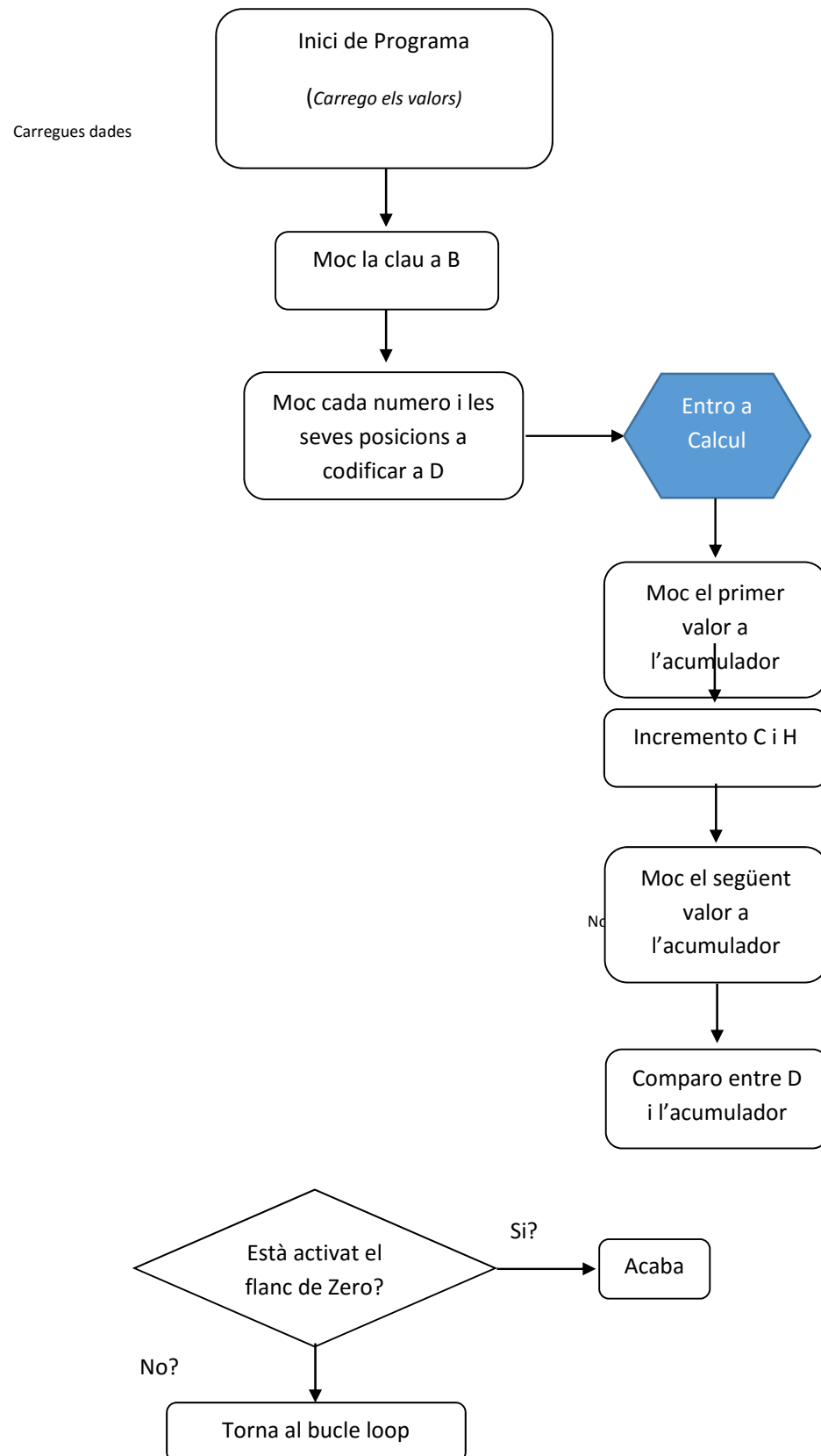
a. Quina instrucció fem servir per assignar la posició inicial al registre SP?

Fem servir la instrucció SPHL.

b. Quina es la instrucció utilitzada per guardar el PC en la pila quan treballem amb subrutines? I per recuperar de nou el valor del PC?

Per recuperar, en canvi, podem fer servir:

- RC: segueix el PC tal i com estava
- REC: s'utilitza amb CALL també
- RM: si es negatiu
- RNC, RNZ, RP, RPE, RPO, RZ...

Diagrama de Blocs:

Conclusions

He aconseguit conèixer els modes d'adreçament del 8085 i m'he familiaritzat-se amb les seves instruccions i subrutines.