

Συστήματα Μικροϋπολογιστών

3^η ομάδα ασκήσεων

Νασοπούλου Ελένη – 03121087

Άσκηση 3

Ερώτημα 1

```
INR16 MACRO ADDR
    PUSH PSW
    PUSH B
    PUSH H

    LXI H,ADDR      ; HL= ADDR
    MOV C,M          ; C= (ADDR)= XLOW
    INX H             ; HL= ADDR+1
    MOV B,M          ; B= (ADDR+1)= XHIGH
    INX B             ; BC = BC+1
    MOV M,B          ; (ADDR+1) = XHIGH
    DCX H             ; HL= ADDR
    MOV M,C           ; (ADDR)= XLOW

    POP H
    POP B
    PUSH PSW
ENDM
```

Ερώτημα 2

```
FILL MACRO ADDR,K
    PUSH PSW
    PUSH H

    LXI H,ADDR      ; HL= ADDR
    MVI A,K          ; A= K
L1:  MOV M,A         ; (ADDR)= A
    INX H             ; ADDR++
    DCR A             ; A--
    JNZ L1            ; if (A!=0), loop

    POP H
    POP PSW
ENDM
```

Ερώτημα 3

RHLR MACRO Q,R

```
MOV A,R      ; A= R
RAL          ; left rotation of A, LSB(A)= CY, CY= MSB(A)
MOV R,A      ; update R
MOV A,Q      ; A= Q
RAL          ; left rotation of A, LSB(A)= CY, CY= MSB(A)
MOV Q,A      ; update Q
```

ENDM

Άσκηση 4

Έστω ότι στο μE 8085 εκτελείται η εντολή $\text{JMP } 2200\text{H}$. Ο μετρητής προγράμματος είναι στη θέση $(\text{PC})=2000\text{H}$ και ο δείκτης σωρού $(\text{SP})=3000\text{H}$. Στο μέσον της εκτέλεσης της εντολής αυτής συμβαίνει διακοπή $\text{RST } 6.5$.

Οι λειτουργίες που επιτελούνται ως απάντηση στην software διακοπή $\text{RST } 6.5$ είναι οι εξής:

1. Ολοκληρώνεται η τρέχουσα εντολή:

Πριν τη διακοπή:

PC	Address	Content
->	2000H	JMP 2200H

	2200H	<content>

SP	Address	Stack

	2FFEh	<data>
	2FFFh	<data>
->	3000H	<data>

Μετά τη διακοπή:

PC	Address	Content
	2000H	JMP 2200H

->	2200H	<content>

SP	Address	Stack

	2FFEh	<data>
	2FFFh	<data>
->	3000H	<data>

2. Σώζεται ο μετρητής προγράμματος και η κατάσταση του μE :

PC	Address	Content
	2000H	JMP 2200H

->	2200H	<content>

SP	Address	Stack

->	2FFEh	00
	2FFFh	22
	3000H	<data>

3. Εκτελείται η ρουτίνα εξυπηρέτησης

Ο PC γίνεται $8 \times 6.5 = 52(\text{dec}) = 0034(\text{hex})$:

PC	Address	Content
->	0034H	<content>
	0035H	<content>

SP	Address	Stack

->	2FFEh	02

	2FFFH	00
	3000H	<data>

4. Ανακτάται η κατάσταση του μE και επανέρχεται στην επόμενη εντολή από αυτήν όπου έγινε η διακοπή

PC	Address	Content
	2000H	JMP 2200H

->	2200H	<content>

SP	Address	Stack

	2FFEh	02
	2FFFH	00
->	3000H	<data>

Άσκηση 5

Ερώτημα 1

LXI H,0000H ; HL initialization
MVI C,20H ; C= 32, counter

MVI A,0EH ; mask for RST 5.5
SIM
EI ; enable interrupt

WAIT:
MOV A,C
CPI 00H ; (counter == 0)?
JNZ WAIT ; if no, loop

DI ; disable interrupt
DAD H
DAD H
DAD H
DAD H

HLT

; address-> 002CH

RST 5.5:
PUSH PSW

MOV A,C
ANI 01H ; A= LSB(C)
CPI 00H ; (LSB(C)==0)?
JZ LSBS ; if no, get MSBs

MSBS:

```

IN 20H          ; A= input
ANI 0FH         ; A= 0000MMMM
RLC
RLC
RLC
RLC             ; A= MMMM0000
MOV E,A         ; E= MMMM0000
JMP DONE

```

LSBS:

```

IN 20H          ; A= input
ANI 0FH         ; A= 0000LLLL
ORA E           ; A= MMMMLLLL
MOV E,A         ; E= MMMMLLLL

```

DONE:

```

MVI D,00H       ; DE= 00000000 MMMMLLLL
EI              ; enable interrupts
POP PSW
RET

```

Ο υπολογισμός του μέσου όρου προκύπτει από διαίρεση του H-L με τον αριθμό $16 = 2^4$. Επειδή ο διαιρέτης είναι δύναμη του 2, η διαίρεση μπορεί να αντικατασταθεί με την διαδικασία της ολίσθησης κατά 4 θέσεις.

Ερώτημα 2

```

LXI H,0000H     ; HL initialization
MVI C,20H       ; C= 32, counter

```

WAIT:

```

MOV A,C
CPI 00H         ; (counter == 0)?
JNZ FINISH      ; if no, loop

```

GET1:

```

IN 20H
ANI 01H
JZ GET1         ; if (LSB==0), loop

```

GET0:

```

IN 20H
ANI 20H
JNZ GET0        ; if (LSB==1), loop

```

```

MOV A,C
ANI 01H         ; A= LSB(C)
CPI 00H         ; (LSB(C)==0)?
JZ LSBS         ; if no, get MSBs

```

MSBS:

```
IN 20H          ; A= input
ANI 0FH         ; A= 0000MMMM
RLC
RLC
RLC
RLC             ; A= MMMM0000
MOV E,A         ; E= MMMM0000
JMP DONE
```

LSBS:

```
IN 20H          ; A= input
ANI 0FH         ; A= 0000LLLL
ORA E           ; A= MMMMLLLL
MOV E,A         ; E= MMMMLLLL
```

DONE:

```
MVI D,00H       ; DE= 00000000 MMMMLLLL
JMP WAIT
```

FINISH:

```
DAD H
DAD H
DAD H
DAD H
HLT
```

Το πρόγραμμα είναι ίδιο με πριν απλά τώρα δεν υπάρχουν τα interruptions οπότε ο έλεγχος γίνεται από το LSB. Για να λάβουμε υπόψιν και την επαναφορά του x_0 στην τιμή '1', φτιάχνουμε 2 loops. Το πρώτο loop υπάρχει για να σιγουρευτούμε ότι υπάρχει επαναφορά και το δεύτερο για την στιγμή που το x_0 θα γίνει '0'.