

Συστήματα Μικροϋπολογιστών 2018-2019

3η Ομάδα Ασκήσεων

1η Άσκηση

Δίνεται το πρόγραμμα. Όλη η λειτουργία του προγράμματος γίνεται ουσιαστικά μέσα στη ρουτίνα εξυπηρέτησης της διακοπής RST 6.5. Ο υπολογισμός του χρόνου του 1 λεπτού της ώρας χωρίστηκε σε 6 δεκάδες δευτερολέπτων.

```
IN 10H
LXI B,0064H      ;Καθυστέρηση 100 msec
LXI D,0B00H      ;Για την STDM

MVI A,10H        ;Καθαρισμός της οθόνης
STA 0B00H
STA 0B01H
STA 0B02H
STA 0B03H
STA 0B04H
STA 0B05H

MVI A,0DH        ;Ενεργοποίηση διακοπής RST 6.5
SIM
EI

START:
JMP START

INTR_ROUTINE:
EI

MVI A,00H        ;Ανάμμα των LEDs
STA 3000H

MVI A,05H        ;59...50 secs
STA 0B05H
MVI A,09H
LIGHTON1:
STA 0B04H
CALL SCREEN
DCR A            ;Χρόνος
CPI 00H          ;Εκπνοή χρόνου ?
JNZ LIGHTON1     ;Έλεγχος εκπνοής χρόνου
CALL ZERO

MVI A,04H        ;49...40 secs
STA 0B05H
MVI A,09H
LIGHTON2:
STA 0B04H
CALL SCREEN
DCR A
CPI 00H
JNZ LIGHTON2
CALL ZERO

MVI A,03H        ;39...30 secs
STA 0B05H
MVI A,09H
LIGHTON3:
STA 0B04H
CALL SCREEN
DCR A
CPI 00H
JNZ LIGHTON3
CALL ZERO

MVI A,02H        ;29...20 secs
STA 0B05H
MVI A,09H
LIGHTON4:
STA 0B04H
CALL SCREEN
```

```

DCR A
CPI 00H
JNZ LIGHTON4
CALL ZERO

MVI A,01H           ;19...10 secs
STA 0B05H
MVI A,09H
LIGHTON5:
STA 0B04H
CALL SCREEN
DCR A
CPI 00H
JNZ LIGHTON5
CALL ZERO

MVI A,00H           ;09...00 secs
STA 0B05H
MVI A,09H
LIGHTON6:
STA 0B04H
CALL SCREEN
DCR A
CPI 00H
JNZ LIGHTON6
CALL ZERO

MVI A,FFH           ;Σβήσιμο των LEDs
STA 3000H           ;

RET

SCREEN:             ;Ανανέωση της οθόνης
PUSH PSW
PUSH H
PUSH D
PUSH B
CALL STDM
MVI A,0AH           ;10*100msec=1sec
CONT:
CALL DCD
CALL DELB
DCR A
CPI 00H
JNZ CONT
POP B
POP D
POP H
POP PSW
RET

ZERO:              ;Εμφάνιση μηδενικού στη 2η θέση της οθόνης
MVI A,00H
STA 0B04H
CALL SCREEN
CALL DELB
RET

END

```

2η Άσκηση

Δίνεται το πρόγραμμα, με το οποίο όμως λειτουργούν σωστά είτε τα LEDs εξόδου είτε η οθόνη των 7-segments. Με το πλήρες πρόγραμμα, όπως δίνεται παρακάτω, λειτουργεί σωστά η οθόνη αλλά δε λειτουργούν τα LEDs. Τοποθετώντας σε σχόλιο ή σβήνοντας την εντολή CALL SCREEN, απενεργοποιείται η οθόνη και λειτουργούν τα LEDs.

```
IN 10H
MVI B,0FH      ;K1
MVI C,EFH      ;K2
MVI L,FFH      ;Βοηθητικός
LXI D,0B00H    ;Για την STDM

MVI A,10H      ;Καθαρισμός της οθόνης
STA 0B00H      ;
STA 0B01H      ;
STA 0B02H      ;
STA 0B03H      ;
STA 0B04H      ;
STA 0B05H      ;

MVI A,0DH      ;Ενεργοποίηση διακοπής RST 6.5
SIM            ;
EI            ;
START:
STA 3000H      ;Έξοδος στα LEDs
MOV A,L
CALL SCREEN    ;Έξοδος στην οθόνη

CMP B          ;Σύγκριση με K1
JC RANGE1     ;A<K1 ?
JZ RANGE1     ;A=K1 ?
CMP C          ;Σύγκριση με K2
JC RANGE2     ;A<K2 ?
JZ RANGE2     ;A=K2 ?
MVI A,FBH     ;[0...K1] => 1ο LED
JMP START

RANGE1:
MVI A,FEH     ;(K1...K2] => 2ο LED
JMP START

RANGE2:
MVI A,FDH     ;(K2...FFH] => 3ο LED
JMP START

SCREEN:        ;Ανανέωση της οθόνης
PUSH PSW
PUSH H
PUSH D
PUSH B
CALL STDM
CALL DCD
POP B
POP D
POP H
POP PSW
RET

INTR_ROUTINE:
PUSH PSW

CALL KIND      ;Είσοδος από πληκτρολόγιο
STA 0B00H      ;LSB
MOV L,A        ;Προσωρινή αποθήκευση μέχρι να λάβουμε το MSB
CALL KIND
STA 0B01H      ;MSB

RLC            ;4 φορές ολίσθηση για να τοποθετηθεί στο MSB
RLC            ;
RLC            ;
RLC            ;
ORA L          ;Ένωση LSB, MSB
MOV L,A

POP PSW
EI
RET

END
```

3η Άσκηση

Σημείωση: στη μακροεντολή FILL ονομάσαμε το μέγεθος του τμήματος len για αποφυγή σύγχυσης με τον καταχωρητή L. Επίσης, υποθέσαμε ότι το len είναι 16-bit, αφού το μέγεθος του τμήματος είναι από 1 έως 256. Αν η εκφώνηση είναι λάθος και το μέγεθος του τμήματος είναι από 1 έως 255, τότε το len είναι 8-bit και οι εντολές LXI H, len και MOV A, L μπορούν να αντικατασταθούν από την MVI A, len.

```
SWAP Nibble MACRO Q
    PUSH PSW
    MOV A,Q
    RLC
    RLC
    RLC
    RLC
    MOV Q,A

    MOV A,M
    RRC
    RRC
    RRC
    RRC
    MOV M,A
    POP PSW
ENDM
```

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

    LXI H,len      ;ή MVI A,len
    MOV A,L        ;?
    LXI H,ADDR

START:
    MVI M,K
    INR M
    DCR A
    JNZ START

    POP H
    POP PSW
ENDM
```

```
RHLL MACRO n
    PUSH PSW
    PUSH B

    MVI A,n
    CPI 00H
    JZ FINISH
    MVI B,n

START:
    MOV A,L
    RAL
    MOV L,A
    MOV A,H
    RAL
    MOV H,A
    DCR B
    JNZ START

FINISH:
    POP B
    POP PSW
ENDM
```

4η Άσκηση

Η διακοπή συμβαίνει στο μέσο της εντολής CALL 3000H, άρα θα ολοκληρωθεί η εκτέλεση της τρέχουσας εντολής: η τρέχουσα τιμή του μετρητή προγράμματος (2000H) αποθηκεύεται στην στοίβα, ο δείκτης στοίβας ανεβαίνει 2 θέσεις πάνω και στον μετρητή προγράμματος καταχωρείται η διεύθυνση 3000H. Έπειτα σώζεται η τιμή του μετρητή προγράμματος και η κατάσταση του 8085 και εκτελείται η ρουτίνα εξυπηρέτησης της διακοπής RST 5.5. Αυτό σημαίνει ότι η τιμή του μετρητή προγράμματος αποθηκεύεται ξανά στην στοίβα, ο δείκτης στοίβας ανεβαίνει άλλες 2 θέσεις πάνω και στον μετρητή προγράμματος καταχωρείται η διεύθυνση της διακοπής για να εκτελεστεί η σχετική ρουτίνα. Όταν ολοκληρωθεί η εκτέλεση της ρουτίνας εξυπηρέτησης της διακοπής, η διεύθυνση που βρίσκεται στην κορυφή της στοίβας (3000H) επανέρχεται στον μετρητή προγράμματος, ο δείκτης στοίβας κατεβαίνει 2 θέσεις κάτω και εκτελείται η ρουτίνα που αρχίζει από τη διεύθυνση 3000H, σύμφωνα με την εντολή CALL 3000H. Όταν ολοκληρωθεί η εκτέλεση και της τελευταίας ρουτίνας, η διεύθυνση στην κορυφή της στοίβας (2000H) επαναφέρεται στον μετρητή προγράμματος, ο δείκτης στοίβας κατεβαίνει άλλες 2 θέσεις κάτω και συνεχίζεται η εκτέλεση του προγράμματος από τη διεύθυνση 2001H.

Η όλη διαδικασία φαίνεται σχηματικά στον παρακάτω πίνακα, όπου δίνονται τα περιεχόμενα του μετρητή προγράμματος και της στοίβας αρχικά (1), μετά την εκτέλεση της εντολής CALL 3000H (2), μετά την πραγματοποίηση της διακοπής RST 5.5 (3), μετά την εκτέλεση της ρουτίνας εξυπηρέτησης της διακοπής (4) και μετά την εκτέλεση της ρουτίνας που καλεί η εντολή CALL 3000H (5).

1		2		3		4		5	
PC	2000H	PC	3000H	PC	(RST 5.5)	PC	3000H	PC	2000H
SP	00H	SP	00H	SP	00H	SP	00H	SP	00H
SP+1	40H	SP+1	20H	SP+1	30H	SP+1	20H	SP+1	40H
		SP+2	00H	SP+2	00H	SP+2	00H		
		SP+3	40H	SP+3	20H	SP+3	40H		
				SP+4	00H				
				SP+5	40H				

5η Άσκηση

Δίνεται το παρακάτω πρόγραμμα μετά από τροποποίηση του σχετικού παραδείγματος των διαφανειών. Ο μετρητής δεδομένων ξεκινάει απ' το 64, αφού έχουμε 2 τμήματα για καθένα από τα 32 δεδομένα. Η ρουτίνα εξυπηρέτησης της διακοπής ελέγχει με κάθε είσοδο το LSB του μετρητή για να διαπιστώσει αν δέχεται τα 4 LSB ή τα 4 MSB του κάθε δεδομένου. Στην είσοδο εφαρμόζεται κατάλληλη μάσκα για να μηδενιστούν τα αδιάφορα 4 MSB αυτής. Τέλος, για να προκύψει ο μέσος όρος γίνεται 3 φορές ολίσθηση αριστερά του αθροίσματος των δεδομένων.

```
MVI A,0Eh      ;Μάσκα διακοπών
SIM
LXI H,0        ;Συσσωρευτής δεδομένων
MVI C,64d      ;64 δεκαδικό στον μετρητή δεδομένων
EI             ;Ενεργοποίηση διακοπών
ADDR:          ;Αναμονή δεδομένων
MVI A,C
CPI 0
JNZ ADDR      ;Έλεγχος εισόδου όλων των δεδομένων
DI            ;Απενεργοποίηση διακοπών
DAD H         ;3 φορές πρόσθεση του H-L στον εαυτό του για ολίσθηση 3 φορές αριστερά
DAD H
DAD H

HLT

002C: JMP RST5.5

RST5.5:
PUSH PSW

MOV A,C
ANI 00000001b ;00000001 δυαδικό για το LSB
JPO GET4MSB   ;Έλεγχος αν λάβαμε τα LSB ή τα MSB του δεδομένου
IN PORT_IN    ;Είσοδος των 4 LSB του δεδομένου
ANI 00001111b ;00001111 δυαδικό για τα 4 LSB της πόρτας
MOV B,A       ;Προσωρινή αποθήκευση μέχρι να λάβουμε τα MSB του δεδομένου
JMP GOT4LSB   ;Επιστροφή στο πρόγραμμα μέχρι να ληφθούν τα MSB του δεδομένου
GET4MSB:      ;Επεξεργασία των MSB του δεδομένου
IN PORT_IN    ;Είσοδος των 4 MSB του δεδομένου
ANI 00001111b
RLC           ;4 φορές ολίσθηση για να τοποθετηθεί το τμήμα του δεδομένου στα MSB
RLC
RLC
RLC
RLC
ORA B         ;Ένωση με τα LSB του δεδομένου

MVI D,0
MVI E,A
DAD D         ;Πρόσθεση δεδομένων
DCR C         ;Μείωση μετρητή
GOT4LSB:
POP PSW
EI
RET
```

Σημείωση: Η επεξεργασία του κώδικα των προγραμμάτων έγινε στο [Notepad++](#) 7.6.6 Rev 2 portable και η συγγραφή της παρούσας αναφοράς έγινε στο [LibreOffice](#) 6.3.1 portable.