

Συστήματα Μικροϋπολογιστών  
4η σειρά ασκήσεων

Αλέξανδρος Λιαροκάπης (03114860)  
Απόστολος Μάνος (03114855)



## Άσκηση 1

Στο μLAB να γραφτεί πρόγραμμα Assembly που να ελέγχει μέσω της διακοπής τύπου RST 6.5 τα φώτα ενός χώρου. Όταν προκαλείται διακοπή RST 6.5 να ανάβουν όλα τα LED της πόρτας εξόδου. Αυτό να παραμένει για περίπου ένα λεπτό της ώρας και μετά να σβήνει. Αν όμως ενδιάμεσα ξαναενεργοποιηθεί η διακοπή να ανανεώνεται ο χρόνος του ενός λεπτού. Ο χρόνος που παραμένει στην κατάσταση αυτή να απεικονίζεται σε sec συνεχώς στα 2 αριστερότερα δεκαεξадικά ψηφία των 7-segment displays και σε δεκαεξадική μορφή.

Να γίνει χρήση των ρουτινών χρονοκαθυστέρησης του εκπαιδευτικού συστήματος μLAB.

### Λύση

```
IN 10H
LXI H,0B00H
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,10H
INX H
LXI B,03E8H

START:    NOP
          MVI A,0DH
          SIM
          EI
WAIT:     NOP
          JMP WAIT
INTR_ROUTINE: NOP
          POP H
          MVI A,00H
          STA 3000H
          EI
          MVI E,3CH
MYCOUNT: NOP
          MOV A,E
          DCR A
          MOV E,A
          CPI FFH
          JNZ MYDISPLAY
          MVI A,FFH
          STA 3000H
          JMP WAIT
MYDISPLAY: NOP
          LXI H,0B04H
          MVI D,00H
XD_DEC:   NOP
          CPI 0AH
          JC XD_CONT
          INR D
          SUI 0AH
          JMP XD_DEC
XD_CONT:  NOP
```

```
ANI OFH
MOV M,A
INX H
MOV A,D
ANI OFH
MOV M,A

PUSH B
PUSH D
PUSH PSW
LXI D,0B00H
CALL STDM
CALL DCD
DI
CALL DELB
EI
POP PSW
POP D
POP B
JMP MYCOUNT
```

```
END
```

## Ασκηση 2

Να υλοποιηθεί και να εκτελεστεί στο  $\mu$ LAB πρόγραμμα σε assembly που όταν προκαλείται διακοπή τύπου RST 6.5 να διαβάζει τα 2 διαδοχικά δεκαεξαδικά ψηφία ενός αριθμού (0-255) που δίνονται στη συνέχεια απο το πλκτρολόγιο και να τα απεικονίζει στα 2 δεξιότερα 7-segment display. Να συγκρίνει την τιμή αυτή με δύο κατώφλια K1 και K2 με  $K1 < K2$ , που οι τιμές τους βρίσκονται στους καταχωρητές B και C αντίστοιχα. Στη συνέχεια να ανάβει ένα απο τα τρία LED εξόδου που αντιστοιχούν στις περιοχές τιμών  $[0..K1]$ ,  $(K1..K2]$  και  $(K2..FFH)$ .

### Λύση

```
IN 10H
MVI B,07H
MVI C,70H

MVI L,00H
MVI H,00H

SHLD 0B00H
SHLD 0B02H
SHLD 0B04H

MVI A,0DH
SIM
EI

START: NOP
      JMP START

INTR_ROUTINE: NOP
      PUSH PSW
      PUSH H
      PUSH D
      CALL KIND
      MOV H,A
      RRC
      RRC
      RRC
      RRC
      MOV D,A
      CALL KIND
      MOV L,A
      ORA D
      SHLD 0B00H

CHECK: NOP
      CMP B
      JC FIRST_LSB
      CMP C
      JC SECOND_LSB
      JMP THIRD_LSB

FIRST_LSB: NOP
      MVI A,01H
      JMP EXIT
```

```
SECOND_LSB: NOP
            MVI A,02H
            JMP EXIT
THIRD_LSB: NOP
            MVI A,04H
```

```
EXIT: NOP
        CMA
        STA 3000H
        LXI D,0B00H

        CALL STDH
```

```
PRINT: NOP
        CALL DCD
        JMP PRINT

        POP D
        POP H
        POP PSW
        EI
```

```
END
```

### Άσκηση 3

Στο  $\mu\text{E } 8085$  εκτελείται η εντολή  $\text{CALL } 3000\text{H}$ . Ο μετρητής προγράμματος είναι  $\text{PC} = 2000\text{H}$  και ο δείκτης σωρού  $\text{SP} = 4000\text{H}$ . Στο μέσο της εκτέλεσης της εντολής συμβαίνει διακοπή  $\text{RST } 5.5$ . Δώστε τις νέες τιμές των  $\text{PC}$ ,  $\text{SP}$ , το περιεχόμενο του σωρού καθώς και τις λειτουργίες που συμβαίνουν στν αρχή και στην επιστροφή απο την ρουτίνα εξυπηρέτησης.

#### Λύση

Έστω πως η διακοπή συμβαίνει στο μέσον της εκτέλεσης της  $\text{CALL } 3000\text{H}$ , τότε ο  $\text{C}$  θα δείχνει στη θέση μνήμης  $2000\text{H}$ . Άμα ενεργοποιηθεί η διακοπή τότε ο μικροεπεξεργαστής θα περιμένει για να ολοκληρωθεί η εκτέλεση της  $\text{CALL } 3000\text{H}$ . Μετέπειτα ο έλεγχος μεταφέρεται στην κατάλληλη ρουτίνα εξυπηρέτησης. Καθώς ο έλεγχος περνάει στη ρουτίνα εξυπηρέτησης, ο  $\text{PC}$  θα δείχνει στη διεύθυνση  $3000\text{H}$  η οποία και θα είναι αποθηκευμένη στο σωρό του συσσωρευτή με τα δύο περισσότερα σημαντικά ψηφία της στην προηγούμενη θέση του συσσωρευτή και με τα δύο λιγότερα σημαντικά στην προηγούμενη απο αυτή. Επομένως, η διεύθυνση στην οποία θα δείχνει ο δείκτης στοίβας θα είναι η παραπάνω διεύθυνση των δύο λιγότερα σημαντικών ψηφίων. Τελικά ο  $\text{PC}$  θα δείχνει στη διεύθυνση  $0034\text{H}$ , μιας και πρόκειται για διακοπή  $\text{RST } 6.5$ .

Βάσει αυτών, έχουμε:

$\text{PC: } 0034\text{H}$

$\text{SP: } 3\text{FFE}\text{H}$

$(3\text{FFF}\text{H}) \leftarrow 30\text{H}$

$(3\text{FFE}\text{H}) \leftarrow 00\text{H}$

Μετά την την εκτέλεση της ρουτίνας εξυπηρέτησης, ο δείκτης σωρού θα δείχνει ξανά στη διεύθυνση  $4000\text{H}$ , ενώ ο  $\text{PC}$  στη διεύθυνση  $3000\text{H}$ .

## Άσκηση 4

Να γραφτεί πρόγραμμα Assembly σε  $\mu\text{T-8085}$  που να λαμβάνει 32 δεδομένα των 8 bit από μία συσκευή. Το καθένα μεταφέρεται σε 2 βήματα μέσω των ( $X_0 - X_3$ ) της θύρας PORT\_IN ενώ τα υπόλοιπα MSbit της θύρας δεν χρησιμοποιούνται. Η συσκευή για κάθε 4 bit που αποστέλλει προκαλεί πριν διακοπή RST 5.5. Να υπολογιστεί ο μέσος όρος των 32 δεδομένων με ακρίβεια 8 bit.

### Λύση

```
MVI A,0EH
```

```
SIM
```

```
LXI H,0000H
```

```
MVI B,40H
```

```
LXI D,0000H
```

```
MVI C,00H
```

```
EI
```

```
LOOP: NOP
```

```
CALL RST5.5_ROUTINE
```

```
MOV A,B
```

```
ANI A,01H
```

```
CMP 01H
```

```
JZ ODD
```

```
EVEN: NOP
```

```
MOV D,E
```

```
CALL RST5.5_ROUTINE
```

```
JMP CHECK_COUNTER
```

```
ODD: NOP
```

```
MOV A,E
```

```
RLC
```

```
RLC
```

```
RLC
```

```
RLC
```

```
ORA D
```

```
MOV E,A
```

```
MOV D,00H
```

```
DAD D
```

```
CHECK_COUNTER: NOP
```

```
MOV A,B
```

```
CPI 0
```

```
JNZ LOOP
```

```
DI
```

```
AVERAGE: NOP
```

```
        DAD H
        DAD H
HTL
```

```
RST5.5_ROUTINE: NOP
```

```
        PUSH PSW
        IN PORT-IN
        MOV D,00H
        MOV E,A
        DCR B
        POP PSW
        EI
        RET
```