

# Εθνικό Μετσόβιο Πολυτεχνείο

## Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Συστήματα Μικροϋπολογιστών

Σειρά Ασκήσεων 4	Θανάσουλας Γρηγόριος   AM: 03114131 Μόνου Σταματίνα   AM: 03114077
------------------	---

### Άσκηση 1

```
IN 10H
MVI A,0CH
SIM
EI
CALL INIT
```

```
START:    CALL LIGHTS_OFF ; Απενεργοποίηση LEDs
          JMP START
```

```
INTR_ROUTINE: ; Ρουτίνα διακοπής, επαναφέρει στο 60
              MVI B,06H      ; Δεκάδες B = 6
              MVI C,00H      ; Μονάδες C = 0
              EI
              CALL LIGHTS_ON
              CALL COUNT
              RET
```

```
INIT:        ; Αρχικοποίηση καταχωρητών και μνήμης
              MVI A,00H
              LXI B,0000H
              LXI D,0A00H
              LXI H,0A00H    ; Αρχική διεύθυνση segments
              MVI M,10H      ; Κενό σύμβολο στο segment
              INX H
              MVI M,10H
              INX H
              MVI M,10H
              INX H
              MVI M,10H
```

```
INX H
MVI M,10H
INX H
MVI M,10H
RET
```

```
LIGHTS_OFF:      MVI A,FFH          ; Ρουτίνα απενεργοποίησης LEDS
                  STA 3000H
                  RET
```

```
LIGHTS_ON: MVI A,00H
            STA 3000H
            RET
```

```
DELAY_1S:      ; Καθυστέρηση 1000*1ms = 1 sec
                PUSH PSW
                PUSH B
                LXI B,03E7H ; B = 999
```

```
DELAY_1mS: ; Καθυστέρηση 1ms
            CALL DELA      ; Σταθερή χρονοκαθυστέρηση 1ms
            CALL DCD       ; Εμφάνιση μέτρησης
            DCX B
            MOV A,B
            CPI 00H        ; B == 0?
            JNZ DELAY_1mS
            MOV A,C
            CPI 00H        ; C == 0?
            JNZ DELAY_1mS
            POP B           ; Αποκατάσταση καταχωρητών
            POP PSW
            RET
```

```
DISPLAY:      LXI D,0A00H
                LXI H,0A04H
                MOV M,C
                INX H
                MOV M,B

                PUSH PSW
                PUSH B
                PUSH H
                PUSH D
                CALL STDM   ; Μετακίνηση συμβόλων στη θέση μνήμης των segments
                POP D
                POP H
```

```

        POP B
        POP PSW
        RET

COUNT:                ; Μέτρηση απο 60 έως 00
REDUCE_B: CALL DISPLAY
        CALL DELAY_1S
        DCR B           ; Μείωση δεκάδων
        MVI C,09H       ; C = 9

REDUCE_C: CALL DISPLAY
        CALL DELAY_1S
        DCR C           ; Μείωση μονάδων
        MOV A,C
        CPI 00H         ; C = 0
        JNZ REDUCE_C    ; Αν C != 0, συνέχιστε τη μείωση του

        MOV A,B
        CPI 00H         ; B = 0?
        JNZ REDUCE_B
        CALL DISPLAY    ; Εμφάνιση τελικού 00
        CALL DELAY_1S
        JMP START

END

```

## Άσκηση 2

```

        IN 10H
        MVI A,0CH
        SIM
        EI
        CALL INIT

        EQU LIM_B,0FH
        EQU LIM_C,43H

START:   LXI D,0A00H
        CALL STDM
        CALL DCD

JMP START

INTR_ROUTINE:
        EI

```

```
CALL DISPLAY
JMP START
```

DISPLAY:

```
MVI B,LIM_B      ; Φόρτωση ορίων σταθερών
MVI C,LIM_C
CALL KIND        ; Διάβασμα input
MOV E,A
LXI H,0A00H      ; Τοποθέτηση περιεχομένου 2ου 7Segment
MOV M,A
CALL KIND
LXI H,0A01H      ; Τοποθέτηση περιεχομένου 1ου 7Segment
MOV M,A
RLC              ; Μετακίνηση 4 θέσεις αριστερά
RLC
RLC
RLC
ADD E            ; Ενοποίηση αριθμών
MOV E,A
```

```
PUSH D           ; Εμφάνιση αριθμών
LXI D,0A00H
PUSH PSW
PUSH D
PUSH B
PUSH H
CALL STDH
POP H
POP B
POP D
POP PSW
CALL DCD
POP D
```

```
MOV A,E
INR B
CMP B
JC INTERVAL_1    ; A < B + 1 ==> A <= B
INR C
CMP C
JC INTERVAL_2    ; A < C + 1 ==> A <= C
```

```
INTERVAL_3: MVI A,FEH      ; Ενεργοποίηση 1ου LED απο δεξιά
STA 3000H
JMP START
```

```

INTERVAL_2: MVI A,FDH          ; Ενεργοποίηση 2ου LED απο δεξιά
             STA 3000H
             JMP START

INTERVAL_1: MVI A,FBH          ; Ενεργοποίηση 3ου LED απο δεξιά
             STA 3000H
             JMP START

INIT:        ; Αρχικοποίηση καταχωρητών και μνήμης
             MVI A,00H
             LXI B,0000H
             LXI D,0A00H
             LXI H,0A00H      ; Αρχική διεύθυνση segments
             MVI M,10H        ; Κενό σύμβολο στο segment
             INX H
             MVI M,10H
             INX H
             MVI M,10H
             INX H
             MVI M,10H
             INX H
             MVI M,10H
             INX H
             MVI M,10H
             RET

END

```

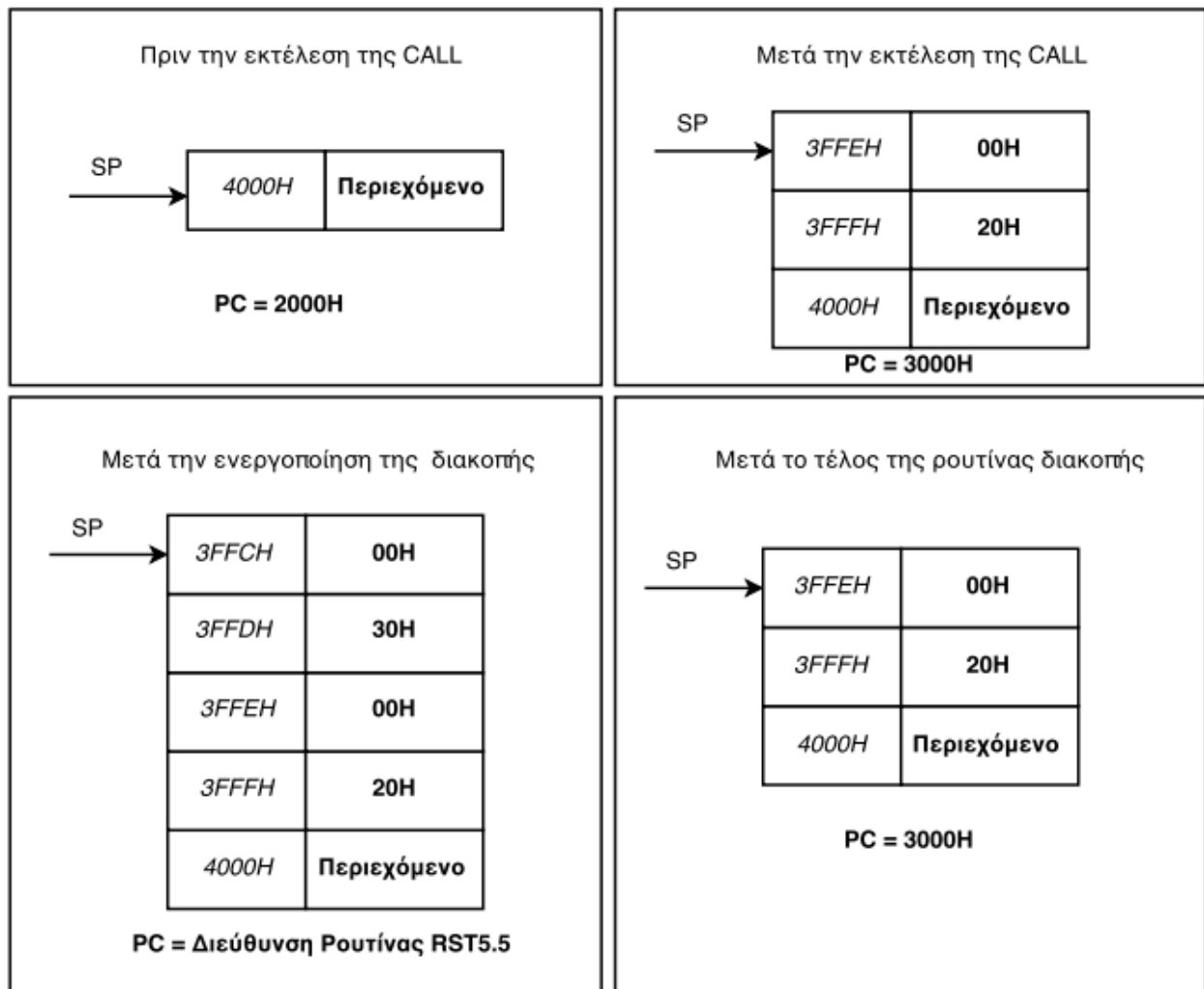
## Άσκηση 3

Η διακοπή RST 5.5 συμβαίνει στο μέσο της CALL 3000H, επομένως θα ολοκληρωθεί πρώτα και ο τελευταίος κύκλος της εντολής και στη συνέχεια θα μεταβούμε στη ρουτίνα της διακοπής, ενώ επίσης θα απενεργοποιηθούν και οι διακοπές. Έπειτα, θα γίνει push στη στοίβα ο μετρητής προγράμματος.

**Μόλις θα έχει ολοκληρωθεί η CALL 3000H**, θα έχει προστεθεί στη στοίβα η θέση επιστροφής μετά το τέλος της ρουτίνας, δηλαδή 2000H (Ο PC=2000H δείχνει στην επόμενη διεύθυνση μετά την CALL).

Κατα την εκτέλεση της ρουτίνας εξυπηρέτησης πρέπει να σωθεί η πλήρης κατάσταση του μικροεπεξεργαστή, δηλαδή οι τιμές των καταχωρητών του και των σημειών. Κατα την ολοκλήρωση της ρουτίνας εξυπηρέτησης, πρέπει να γίνουν POP από τη στοίβα τα δεδομένα που είχαν γίνει προηγουμένως POP, πραφανώς με αντίστροφη σειρά από αυτή της εισαγωγής στη στοίβα.

Τέλος, αφού ενεργοποιηθούν ξανά οι διακοπές, επιστρέφεται ο έλεγχος στο πρόγραμμα που διακόπηκε.



## Άσκηση 4

```

MVI A,0DH    ; A = 0000 1101
SIM
MVI D,00H    ; D=0, άρτια επανάληψη αλλιώς D=1 περιττή επανάληψη
MVI E,40H    ; E = 64, πλήθος 4-bit λέξεων
LXI H,0000H  ; HL = 0
EI
WAIT:        LOOP WAIT

INT_5.5:     MOV A,E      ; A = E
CPI 01H
JZ DISABLE_INTR
    
```

```

CONT:      MOV A,D
           CPI 00H      ; Αν CPI != 0 τότε διαβάσε τα MSB ψηφία
           JNZ READ_MSB

READ_LSB:  IN PORT_IN
           AND 0F        ; Απομόνωση 4 LSB

           MOV C,A      ; Αποθήκευση στον BC
           MVI B,00H

           DCR E        ; Μείωση μετρητή
           MOV A,01H
           XRA D        ; Αναστροφή σημαίας
           MOV D,A
           JMP SUM

READ_MSB:  IN PORT_IN
           AND 0F        ; Απομόνωση 4 LSB της πόρτας
           RLC          ; Μετακίνηση στη θέση των MSB
           RLC
           RLC
           RLC

           MOV C,A      ; Αποθήκευση στον BC
           MVI B,00H

           DCR E        ; Μείωση μετρητή
           MOV A,01H
           XRA D        ; Αναστροφή σημαίας
           MOV D,A

SUM:       DAD B        ; HL += BC
           MOV A,E
           CPI 00H
           JZ AVERAGE ; Αν έχει τελειώσει η είσοδος 64 δεδομένων
           RET

AVERAGE:  DAD H        ; Μετακίνηση συνολικά 3 θέσεις αριστερά
           DAD H
           DAD H
           HLT

DISABLE_INTR:
           DI          ; Απενεργοποίηση των διακοπών
           JMP CONT

```