

# Συστήματα Μικροπολογιστών

## 3<sup>η</sup> Ομάδα Ασκήσεων

Μοίρας Αλέξανδρος Α.Μ.: el18081

Παπαδημητρίου Ευθύμιος Α.Μ.: el18129

### Ασκήσεις Προσομοίωσης

#### 1<sup>η</sup> Άσκηση:

```
IN 10H ; Apenergopoihsh prostasias mnhmhs
LXI B,0064H ; 100msec kathysterhsh gia thn DELB
LXI D,0B00H ; Dieythynsh pou 8a vrei h STDM ta pshfia pros apeikonis
MVI H,00H ; Arxikopoihsh xronometrou sto 0
MVI A,0DH ; Energopoihsh diakophs RST6.5
SIM
EI

LOOP1: MOV A,H ; Sygkrish xronometrou me to 0
CPI 00H ; An einai 0 svhse ta fwta
JZ TURN_OFF

LXI B,0064H ; Ftiaxe pali tous BC kai DE pou xalane apo thn STDM
LXI D,0B00H

MVI A,00H ; Anapse ta fwta
STA 3000H
MVI A,10H ; Mhn ektypwseis tipota sta 7-segment 1,2,5,6
STA 0B00H
STA 0B01H
STA 0B04H
STA 0B05H
MOV A,H ; Xronometro ston A
ANI 0FH ; Krata to prwto dekadiko pshfio kai apeikonise to sto
STA 0B02H ; 7-segment display 3
MOV A,H ; Krata to prwto dekadiko pshfio kai apeikonise to sto
RAR ; 7-segment display 4
RAR
RAR
RAR
ANI 0FH
STA 0B03H

MOV A,H ; Apothikeush xronometrou sth mnhmh gia na mh xalasei apo
STA 0B06H ; thn klhsh ths STDM. (oxi stoiva gt den kanoume RET sth
CALL STDM ; sth rutina exyphrethshs kai mporei na mh ginoun ta
LDA 0B06H ; pop meta. Epanafora xronometrou
MOV H,A
MVI L,0AH ; 10 fores klhsh ths routines apeikonishs gia na fainetai
DISPLAY: CALL DCD ; to 7-segment synexomena. Kathe fora 100ms
CALL DELB ; Ara synolo 1sec
```

```
DCR L
JNZ DISPLAY
DCR H ; Meiwsh xronometrou
MOV A,H ; Gia na kratame dekadikh metrhsh kathe fora pou meiwnontas
ANI 0FH ; ftanoume se arithmo XFH afairoume 6 gia na ginei
CPI 0FH ; dekadikos
JNZ LOOP1
MOV A,H
SUI 06H
MOV H,A
JMP LOOP1

TURN_OFF: MVI A,FFH ; Svhsimo fwtwn
STA 3000H
STUCK: JMP LOOP1 ; Xana elegxos xronometrou

INTR_ROUTINE:
MVI H,60H ; An ginei diakoph vale 60 sec sto xronometro
INX SP ; Kane pop to return address afou den tha kanoume RET
INX SP ; Gia na mh synexisei to programma apo lathos shmeio
        ; kai xasoume kapoio deuterolepto
EI ; Energopoihse pali tis diakopes
JMP LOOP1 ; Phgaine sto kyriws programma

END
```

---

## 2<sup>η</sup> Άσκηση:

```
IN 10H ;Xekleidwma mnhmhs
MVI A,0DH ; Energopoihsh diakophs Reset6.5
SIM
EI

MVI A,10H ;Arxikopoihsh tw 7-segment displays sto keno
STA 0A00H
STA 0A01H
STA 0A02H
STA 0A03H
STA 0A04H
STA 0A05H

MVI D,05H ;Arxikopoihsh
MVI E,10H

MOV A,D ; Katwflia se theseis mnhmhs giati xreiazontai oi
STA 0A06H ; kataxwrhtes D,E gia thn STDM
MOV A,E
STA 0A07H

MAIN: LXI D,0A00H ;Theseis mnmhs pou tha xrhsimopoihsei h STDM
CALL STDM
CALL DCD ;Ektypwsh sta 7-segment displays
EI ; Energopoihsh diakopwn afou ginei prohgoumenh diakoph
JMP MAIN ;Infinite loop

INTR_ROUTINE:

INX SP ; Pop dieuthynshs epistrofhs giati de xrhsimopoioume RET
INX SP

CALL KIND ;Anagnwsh least significant digit
STA 0A00H ;Apothikeush tou sth thesh dexiotrou 7-segment display
MOV H,A ;kai ston kataxwrhth H
CALL KIND ;Anagnwsh most significant digit
STA 0A01H ;Apothikeush tou sth thesh 2ou 7-segment display
RAL ;Metakinsh tou sta 4 most significant bits
RAL
RAL
RAL
```

---

```

ANI F0H ;Maska gia exaleipsh tyxon kratoumeno sta LSBits
ORA H ;Dhmiourgia oloklhrou tou hex arithmou pou diavasthke
MOV H,A ;Apothikeush tou ston H
LDA 0A06H ;Epanafora katwfliwn stous D kai E
MOV D,A
LDA 0A07H
MOV E,A
MOV A,H ;Epanafora arithmou pou diavasthke ston A

CMP D ;Sygkrish me to katw katwfli kai anama tou prwtou LED an
JC LED_1 ; o arithmos einai mikroteros h isos autou
JZ LED_1

CMP E ;Alliws sygkrish me to anw katwfli kai anama tou deuteron
JC LED_2 ;LED an einai mikroteros h isos autou efoson einai
JZ LED_2 ;megalyteros tou katw katwfliou

JMP LED_3 ;Alliws anama tou 3ou LED

LED_1: MVI A,FEH ; Anama prwtou LED kai epistrofh sth MAIN
STA 3000H
JMP MAIN

LED_2: MVI A,FDH ; Anama deuteron LED kai epistrofh sth MAIN
STA 3000H
JMP MAIN

LED_3: MVI A,FBH ; Anama tritou LED kai epistrofh sth MAIN
STA 3000H
JMP MAIN

END

```

---

## Θεωρητικές Ασκήσεις

### 3<sup>η</sup> Άσκηση:

α)

```

SWAP Nible MACRO Q ; Το Q μπορεί να είναι κάποιος καταχωρητής γενικού σκοπού από
                     τους B, C, D, E ή M για τη θέση μνήμης που δείχνει ο H-L

    PUSH PSW ; Αποθήκευση A
    MOV A,Q ; Εναλλαγή τεσσάρων LSB και τεσσάρων MSB του Q μέσω του A
    RRC
    RRC
    RRC
    RRC
    MOV Q,A ; Αποθήκευση αποτελέσματος στον Q
    POP PSW ;Επαναφορά A

ENDM

```

β)

```
FILL MACRO RP,X,K
    PUSH H ; Αποθήκευση αρχικής τιμής του ζεύγους HL
    PUSH PSW ; Αποθήκευση accumulator
    PUSH RP ; Μετακίνηση της διευθυνσής από το αρχικό ζεύγος καταχωρητών στον HL
    με push και pop
    POP H
    MVI A,X ; Μετακίνηση μήκους στον A
    LOOP1:
        MVI M,K ; Αποθήκευση του K στη θέση μνήμης
        INX H ; Επόμενη θέση μνήμης
        DCR A ; Μείωση υπολειπόμενων θέσεων μνήμης προς αποθήκευση
        JNZ LOOP1 ; Αν μήκος όχι 0 συνέχισε να αποθηκεύεις
    POP PSW ; Επαναφορά A
    POP H ; Επαναφορά H
ENDM
```

γ)

```
RHLR MACRO n
    PUSH B ; Αποθήκευση ζεύγους BC
    MOV C,A ; Αποθήκευση accumulator στον C. Δεν κάνουμε PUSH PSW γιατί με το pop
    ; θα επαναφέρονταν
    ; και τα flags άρα και το CY
    MVI B,n ; Μετρητής στον B
    LOOP1:
        MOV A,B ; Έλεγχος αν ολοκληρώθηκαν οι περιστροφές
        CPI 00H
        JZ EXIT
        MOV A,H ; Δεξιά περιστροφή H, αρχικό κρατούμενο στο MSB του H, αρχικό LSB
        ; του H στο CY
        RAR
        MOV H,A
        MOV A,L ; Δεξιά περιστροφή L, αρχικό LSB του H στο MSB του L, LSB του L
        ; στο CY
        RAR
        MOV L,A
        DCR B ; Μείωση μετρητή περιστροφών
        JMP LOOP1 ; Loop
    EXIT:
        MOV A,C ; Επαναφορά A
        POP B ; Επαναφορά ζεύγους BC
ENDM
```

#### 4<sup>η</sup> Άσκηση:

Αφού η διακοπή συμβαίνει στο μέσον της εκτέλεσης της εντολής CALL 0880H θα ολοκληρωθεί η εκτέλεση της και μετά το πρόγραμμα θα μεταβεί στη ρουτίνα εξυπηρέτησης της διακοπής. Όταν εκτελεστεί η CALL 0880H στη στοίβα θα αποθηκευτεί η διεύθυνση επιστροφής που θα είναι η διεύθυνση της επόμενης εντολής άρα η  $0800H + 0003H = 0803H$  και ο SP θα μειωθεί κατά 2 και θα γίνει 2FFDH και ο PC θα γίνει 0880H. Όταν κληθεί η ρουτίνα εξυπηρέτησης μέσω της διακοπής θα αποθηκευτεί στη στοίβα ο τρέχων PC (0880H) και ο SP θα μειωθεί πάλι κατά 2 και θα γίνει 2FFBH και ο PC θα γίνει 003C (Εκεί οδηγεί η διακοπή RST7.5). Κατά την επιστροφή από τη ρουτίνα εξυπηρέτησης θα γίνει pop από τη στοίβα η διεύθυνση επιστροφής δηλαδή ο SP θα αυξηθεί κατά 2 και θα γίνει 2FFDH και ο PC θα γίνει 0880H.

#### 5<sup>η</sup> Άσκηση:

α)

START:

```
MVI A,0DH ; Ενεργοποίηση διακοπής RST6.5
SIM
MVI C,40H ; Μετρητής εισόδων αρχικοποιημένος στο δεκαδικό 64 (64 βήματα)
LXI H,0 ; Αρχικοποίηση αποτελέσματος στο 0
EI
```

LOOP1:

```
MOV A,C ; Έλεγχος αν έγιναν και τα 64 βήματα οπότε προχώρα στον υπολογισμό του Μ.Ο.
```

```
CPI 0H
JNZ LOOP1 ; Αλλιώς περίμενε έως ότου γίνουν
DI ; Απενεργοποίηση διακοπών
DAD H ; Τρεις αριστερές ολισθήσεις για να έρθει ο μέσος όρος στον H.
; Εξηγείται στο σχήμα παρακάτω
DAD H
DAD H

HLT
```

```
0034: JMP HANDLER_EVEN ; Μετάβαση στην κατάλληλη ρουτίνα εξυπηρέτησης αναλόγως αν
; ο αριθμός των διακοπών είναι άρτιος ή περιττός
```

```
HANDLER_EVEN: ; Αν είναι άρτιος ο αριθμός διακοπών η είσοδος είναι τα 4 LSB του
; αριθμού
```

```
PUSH PSW ; Προσωρινή αποθήκευση A
```

```

IN 20H ; Είσοδος
ANI 0FH ; Μάσκα για απαλοιφή 4 MSB ψηφίων που δε χρησιμοποιούνται
MVI D,0H ; Αριθμός που διαβάστηκε μεταφέρεται στον DE
MOV E,A
DAD D ; Πρόσθεση των 4 LSB που διαβάστηκαν στο αποτέλεσμα της πρόσθεσης
      ; που βρίσκεται στον HL
DCR C ; Μείωση μετρητή βημάτων
PUSH H ; Προσωρινή αποθήκευση αποτελέσματος
LXI H,HANDLER_ODD ; Αλλαγή της διεύθυνσης στην οποία στέλνει η διακοπή αφού η
      ; επόμενη διακοπή θα είναι περιττή

SHLD 0035H
POP H ; Επαναφορά H
POP PSW ; Επαναφορά A
EI ; Ενεργοποίηση διακοπών
RET ; Επιστροφή στο κυρίως πρόγραμμα

```

HANDLER\_ODD: ; Αν είναι περιττός ο αριθμός διακοπών η είσοδος είναι τα 4 MSB του  
; αριθμού

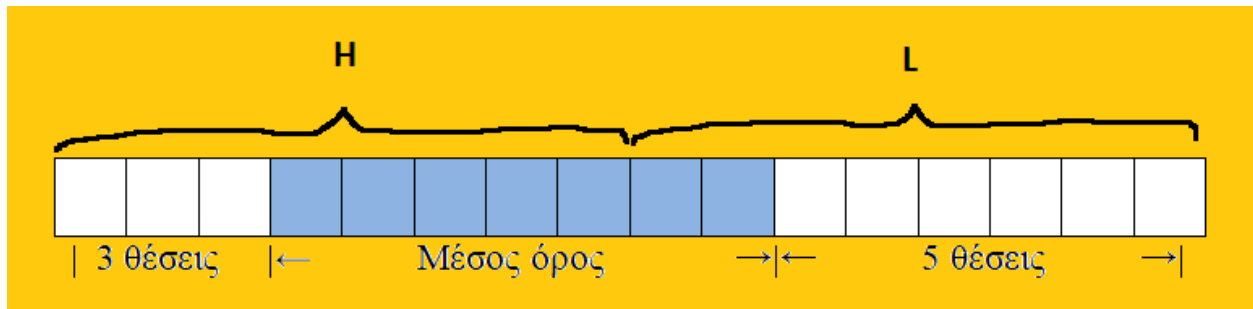
```

PUSH PSW
IN 20H
RLC ; Μετακίνηση εισόδου από τα 4 LSB στα 4 MSB
RLC
RLC
RLC
ANI F0H ; Απαλοιφή 4 LSB ψηφίων που προκύπτουν από τα 4 MSB της εισόδου που
      ; δεν χρησιμοποιούνται
MVI D,0H
MOV E,A
DAD D
DCR C
PUSH H
LXI H,HANDLER_EVEN ; Αλλαγή της διεύθυνσης στην οποία στέλνει η διακοπή αφού
      ; η επόμενη διακοπή θα είναι άρτια

SHLD 0035H
POP H
POP PSW
EI
RET

```

Το αποτέλεσμα της πρόσθεσης όλων των αριθμών θα βρίσκεται στον διπλό καταχωρητή HL. Για να πάρουμε τον μέσο όρο θα έπρεπε να διαιρέσουμε αυτό το αποτέλεσμα με το 32 αφού έχουμε 32 αριθμούς άρα θα έπρεπε να ολισθήσουμε το αποτέλεσμα 5 θέσεις αριστερά. Συνεπώς χωρίς ολίσθηση ο μέσος όρος θα βρίσκεται στα bits 6-12 του διπλού καταχωρητή HL άρα με 3 δεξιές ολισθήσεις μπορούμε να τον μεταφέρουμε στον καταχωρητή H.



β)

START:

```
MVI C,40H ; Μετρητής εισόδων αρχικοποιημένος στο δεκαδικό 64 (64 βήματα)
LXI H,0 ; Αρχικοποίηση αποτελέσματος στο 0
MVI B,0H ; Δείκτης αν ο αριθμός των εισόδων είναι περιττός (B=1) ή άρτιος
; (B=0)
```

WAITSTART: ; Στην αρχή περίμενε έως ότου το x7 γίνει 0

```
IN 20H ; γιατί μπορεί το x7 να είναι 1 όταν ξεκινήσει
RAL ; να εκτελείται το πρόγραμμα και να διαβαστεί
JC WAITSTART ; εσφαλμένα δεδομένο ενώ θέλουμε να διαβαστεί μόνο κατά
; το θετικό μέτωπο του x7 και όχι αν το x7 είναι 1 εξαρχής
```

LOOP1:

```
MOV A,C ; Έλεγχος αν έγιναν και τα 64 βήματα οπότε προχώρα στον υπολογισμό
; του M.O.
```

```
CPI 00H
```

```
JZ EXIT
```

```
IN 20H ; Αλλιώς διάβασε την είσοδο
```

```
PUSH PSW ; Αποθήκευσε την στη στοίβα
```

```
RAL ; Αν το x7 είναι 0 άρα όχι καινούριο δεδομένο κάνε loop
```

```
JNC LOOP1
```

```
MOV A,B ; Αλλιώς αν ο αριθμός των έως τώρα διακοπών είναι άρτιος
```

```
CPI 00H
```

```
JZ HANDLER_EVEN ; πήγαινε στη ρουτίνα HANDLER_EVEN
```

```
JMP HANDLER_ODD ; αλλιώς στην HANDLER_ODD
```

HANDLER\_EVEN: ; Αν είναι άρτιος ο αριθμός διακοπών η είσοδος είναι τα 4 LSB του  
; αριθμού

```
POP PSW ; Επαναφορά εισόδου στον A
```

```
ANI 0FH ; Μάσκα για απαλοιφή 4 MSB ψηφίων που δε χρησιμοποιούνται
```

```
MVI D,0H ; Αριθμός που διαβάστηκε μεταφέρεται στον DE
```

```
MOV E,A
```

```
DAD D ; Πρόσθεση των 4 LSB που διαβάστηκαν στο αποτέλεσμα που βρίσκεται στον
; HL
```



```

    DCR C ; Μείωση μετρητή βημάτων
WAIT1: ; Περίμενε έως ότου ξαναγίνει 0 το x7
    IN 20H
    RAL
    JC WAIT1
    INR B ; όταν ξαναγίνει κάνε το περιεχόμενο του B 1 για περιττό αριθμό
        ; περιστροφών
    JMP LOOP1 ; και γύρνα στην αρχή

HANDLER_ODD: ; Αν είναι περιττός ο αριθμός διακοπών η είσοδος είναι τα 4 MSB του
        ; αριθμού
    POP PSW
    RLC ; Μετακίνηση εισόδου από τα 4 LSB στα 4 MSB
    RLC
    RLC
    RLC
    ANI F0H ; Απαλοιφή 4 LSB ψηφίων που προκύπτουν από τα 4 MSB της εισόδου που
        ; δεν χρησιμοποιούνται
    MVI D,0H
    MOV E,A
    DAD D
    DCR C
WAIT2: ; Περίμενε έως ότου ξαναγίνει 0 το x7
    IN 20H
    RAL
    JC WAIT2
    DCR B ; όταν ξαναγίνει κάνε το περιεχόμενο του B 0 για άρτιο αριθμό
        ; περιστροφών
    JMP LOOP1 ; και γύρνα στην αρχή

EXIT:
    DAD H ; Τρεις αριστερές ολισθήσεις για να έρθει ο μέσος όρος στον H
    DAD H
    DAD H

    HLT

```