



Ε.Μ.Π. - ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧ. ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ
 ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
 ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ
 ΑΚΑΔ. ΕΤΟΣ 2020-2021

ΑΘΗΝΑ 23 - 4 - 2021

3^η ΟΜΑΔΑ ΑΣΚΗΣΕΩΝ
ΓΙΑ ΤΟ ΜΑΘΗΜΑ "Συστήματα Μικροϋπολογιστών"

Βικέντιος Βιτάλης el18803 και Αριάδνη Καζδάγλη el18838

Σειρά Ασκήσεων 3 Συστήματα Μικροϋπολογιστών 16/5/2021

Άσκηση 1:

```
ask1 - TSIK
START:
    IN 10H          ;Epitrepei thn prosbash sthn mnhmh.
    LXI H,0A00H     ;Oi theseis 0A00H-0A05H periexun ton
    MVI M,10H       ;kwdiko ektypwshs tu antistoixu pshfioy
    INX H           ;sthn othonh. Emeis thelume na emfanizume
    MVI M,10H       ;kati mono sta 2 mesaia pshfia enw ola ta
    INX H           ;ypoloipa einai sbhsta kai den emfanizun kati.
    LXI H,0A04H     ;Etsi apo8hkeyoyme stis 8eseis 0A00H-0A02H kai 0A04H-0A05H
    MVI M,10H       ;ths mnhmhs to kwdiko 10H pou shmainei ta antistoixa
    INX H           ;7-segments na mhn emfanizun tipota
    MVI M,10H
    MVI A,0DH
    SIM             ;Arxikopoihsh maskas diakopwn
    EI              ;Energopoihsh diakpwn

WAIT:
    JMP WAIT        ;Anamonh gia diakoph

BCD:
    PUSH PSW        ;Routina metatrophs ari8mou twn ypoloipomenwn
    PUSH B          ;deyteroleptwn apo dekaeksadiko se dekadiko kai
    PUSH D          ;emfanish toy sthn o8onh.
    PUSH H
    MVI B,FFH
    MOV A,E

DECA:
    ;Fortwse to metrhth deyteroleptwn E ston A
    INR B           ;Aykshsh kata 1 toy metrhth dekadwn ston B
    SUI 0AH         ;Meiwsh toy A kata 10
    JNC DECA        ;An einai 8etikos synexise
    ADI 0AH         ;Dior8wsh arnhtikou ypoloipoy
    LXI H,0A02H
    MOV M,A         ;Apothikeyse tis monades sth thesh 0A00H
    INX H
    MOV M,B         ;Apothikeyse tis dekades sth thesh 0A01H
    LXI D,0A00H
    CALL STDM       ;Fortwse sto B thn kathisterisi 25ms
    LXI B,0019H     ;Arxikopoihsh metrhth 40 epanalhpsewn
    MVI A,28H

ONESECLoop:
    ;Broxos synolikhs kathisterisis 25ms * 40 = 1sec
    CALL DCD        ;pou emfanizei sthn o8onh
    CALL DELB       ;thn idia timh deyteroleptwn
    DCR A           ;gia synoliko diasthma 1 deyteroleptou
    CPI 00H
    JNZ ONESECLoop
    POP H
    POP D
    POP B
    POP PSW
    RET

INTR_ROUTINE:
    ;Routina eksyphrethshs diakophs
    POP H
    MVI A,FFH       ;Anapse ola ta LED(negative logic)
    CMA
    STA 3000H
    CALL MEM_INIT
    MVI E,3CH       ;Arxikopoihsh metrhth 60 seconds
    EI              ;Energopoihsh diakopwn

LOOP1:
    CALL BCD
    CALL BLINKING   ;Klsh ypouroutinas pou anabosbhnei ta LED
    DCR E           ;Meiwsh tu metrhth deyteroleptwn kata 1
```

```

MOV A,E      ;An o metrhts den einai 0, epanelabe
CPI 00H
JNZ LOOP1
MVI A,00H    ;Alliws sbhse ola ta LED(arnhtikh logikh)
CMA
STA 3000H
JMP WAIT    ;Alma sthn wait gia anamonh neas diakophs
MEM_INIT:   ;Routina gia apothikeysh sth thesi mnhmhs 0900H
PUSH H      ;ths arxikhs katastash tw n LED
LXI H,0900H
MOV M,A
POP H
RET

BLINKING:
PUSH PSW
PUSH B
PUSH D
PUSH H
LXI B,01F4H ;Fortwsh ston BC ths kathisterisis 01F4H = 500ms
LXI H,0900H ;Fortwsh ston HL th 8esh mnhmhs 0900H, h opoia
MOV A,M     ;periexei thn parousa katastash tw n LED. Fortwse to
CMA         ;periexomeno ayto ston A kai ystera apo 500ms
CALL DELB   ;emfanise to sthn eksodo(negative logic)
STA 3000H   ;Apothikeyse th nea twrinh katastash(antistrofh thw prohgumenhs)
MOV M,A     ;sth thesi 0900H
POP H       ;Etsi eptiygxanetai to anama/sbhsimο tw n LEDse kathe klhsh
POP D       ;ths blinking
POP B
POP PSW
RET

END

```

Άσκηση 2:

```
ask2 - TSIK
IN 10H
START:
    LXI H,0A02H      ;load HL with the start of the memory
    MVI M,10H        ;I will store the output byte
    INX H            ;10H means print nothing
    MVI M,10H        ;We point to the block 0A02H...0A05H
    INX H            ;
    MVI M,10H        ;
    INX H            ;0A02H to 0A05 to have code 10H, to
    MVI M,10H        ;print nothing.

    ;here user gives threshold constants K1, K2
    MVI D,5BH        ;D = K1 = 5BH (USER GIVEN)
    MVI E,7FH        ;E = K2 = 7FH (USER GIVEN)
    INR D            ;for <= and >= K2
    INR E
    MVI A,0DH        ;enable interrupts
    SIM             ;in RST 6.5
    EI

WAIT:
    JMP WAIT

INTR_ROUTINE:
    POP H            ;read input from keyboard
    CALL KIND        ;reads keyboard input
    MOV B,A          ;B has the least significant hex digit
    CALL KIND        ;reads keyboard input
    ;E has the most significant hex digit

    LXI H,0A00H
    MOV M,B          ;0A04H<--least significant hex digit
    INX H
    MOV M,A          ;0A05H<--most significant hex digit

    RLC              ;compare with K1, K2 and light LED
    RLC
    RLC
    RLC
    ADD B
    MOV B,A          ;now B has the full number

    CMP D
    JC FIRST
    CMP E
    JC SECOND

    MVI A,04H
    CMA
    STA 3000H
    JMP LCD

FIRST:
    MVI A,01H
    CMA
    STA 3000H        ;light 1st LED
    JMP LCD

SECOND:
    MVI A,02H
    CMA
    STA 3000H        ;light 2nd LED
    JMP LCD
;LCD output
```

```

LCD:      LXI D,0A00H      |
          CALL STDM
          MVI D,5CH
          MVI E,80H
          EI                ;enable interrupts again
OUT:      CALL DCD
          JMP OUT
END
    
```

Άσκηση 3:

α)SWAP Nible 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
    
```

β)FILL MACRO RP, X, K

```

        PUSH PSW
        PUSH H

        LDAXR
        MOV H,A
        MVI A,X

START:
        MVI M,K
        INR M
    
```

```
DCR A
JNZ START
POP H
POP PSW
ENDM
```

```
γ)RHLL MACRO n
    PUSH PSW
    PUSH B

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

```
START:
    MOV A,L
    RAR
    MOV L,A
    MOV A,H
    RAR
    MOV H,A
    DCR B
    JNZ START
```

```
FINISH:
    POP B
    POP PSW
ENDM
```

Άσκηση 4:

Η RST 7.5 είναι hardware διακοπή, που σημαίνει ότι θα προηγηθεί της εκτέλεσης του υπόλοιπου προγράμματος. Συμβαίνει στο μέσο της εντολής CALL 0880H, άρα θα ολοκληρωθεί η εκτέλεση της τρέχουσας εντολής: η τρέχουσα τιμή του μετρητή προγράμματος (0800H) αποθηκεύεται στην στοίβα(θέσεις μνήμης (SP-1) & (SP-2)), ο δείκτης στοίβας ανεβαίνει 2 θέσεις πάνω και στον μετρητή προγράμματος καταχωρείται η διεύθυνση 0880H. Έπειτα σώζεται η τιμή του μετρητή προγράμματος και η κατάσταση του 8085 και εκτελείται η ρουτίνα εξυπηρέτησης της διακοπής RST 7.5. Η τιμή του μετρητή προγράμματος αποθηκεύεται ξανά στην στοίβα, ο δείκτης στοίβας ανεβαίνει άλλες 2 θέσεις πάνω και στον μετρητή προγράμματος καταχωρείται η διεύθυνση διακοπής η διεύθυνση της διακοπής, ώστε να εκτελεστεί η σχετική ρουτίνα. Μετά την εκτέλεση της ρουτίνας εξυπηρέτησης διακοπής, η διεύθυνση που βρίσκεται στην κορυφή της στοίβας(0880H) επανέρχεται στον μετρητή προγράμματος, ο δείκτης κατεβαίνει 2 θέσεις κάτω και εκτελείται η ρουτίνα που αρχίζει από τη διεύθυνση που ορίζει η εντολή CALL 0880H. Αφού ολοκληρωθεί η εκτέλεση και της τελευταίας ρουτίνας, η διεύθυνση στην κορυφή της στοίβας επαναφέρεται στον μετρητή προγράμματος, ο δείκτης της στοίβας κατεβαίνει άλλες 2 θέσεις κάτω και συνεχίζεται η εκτέλεση του προγράμματος από τη διεύθυνση 0801H.

Επομένως έχουμε:

ΑΡΧΙΚΑ		CALL 0880H		ΔΙΑΚΟΠΗ RST6.5		ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΡΟΥΤΙΝΑΣ ΔΙΑΚΟΠΗΣ		ΜΕΤΑ ΤΗΝ ΕΚΤΕΛΕΣΗ ΡΟΥΤΙΝΑΣ ΕΝΤΟΛΗΣ CALL 0880	
PC	0800H	PC	0880H	PC	003CH	PC	0880	PC	0800
SP	00H	SP	00H	SP	80H	SP	00H	SP	00H
SP+1	30H	SP+1	08H	SP+1	08H	SP+1	08H	SP+1	30H
		SP+2	00H	SP+2	00H	SP+2	00H		
		SP+3	30H	SP+3	08H	SP+3	30H		
				SP+4	00H				
				SP+5	30H				

Άσκηση 5:

Στο παρακάτω πρόγραμμα ο μετρητής δεδομένων ξεκινάει από το 64, εφόσον έχουμε δύο τμήματα για κάθε ένα από τα 32 δεδομένα. Η ρουτίνα εξυπηρέτησης διακοπής, ελέγχει με κάθε είσοδο το LSB του μετρητή για να διαπιστώσει αν δέχεται τα 4 MSB ή τα 4 LSB κάθε δεδομένου. Στην είσοδο εφαρμόζεται κατάλληλη μάσκα, ώστε να μηδενιστούν α αδιάφορα 4 MSB αυτής. Τέλος, γίνεται 3 φορές αριστερή ολίσθηση του αθροίσματος των δεδομένων(έτσι ώστε να προκύψει ο μέσος όρος).

MVI A,0DH ;Μάσκα διακοπών

SIM

LXI H,0000H ;Αρχικοποίηση συσσωρευτή

MVI C,40H ;C=40H (μετρητής)

EI ;Ενεργοποίηση διακοπών

WAIT_LOOP: ;Αναμονή ανάγνωσης όλων των δεδομένων

MOV A,C

CPI 00H

JNZ WAIT_LOOP

DI ;Απενεργοποίηση διακοπών

DAD H ;Υπολογισμός του μέσου όρου με αριστερή ολίσθηση του H-L

DAD H

DAD H

HLT

0034: **JMP RST6.5**

RST6.5:

PUSH PSW

MOV A,C

ANI 00000000b ;δυαδικό για το LSB

JPO GET4MSB ;Έλεγχος αν λάβαμε τα LSB ή τα MSB του δεδομένου

IN 20H ;είσοδος των 4ων LSB του δεδομένου

ANI 00001111b ;δυαδικό για τα 4 LSB της πόρτας

MOV B,A

JMP END

GOT4MSB: ;Επεξεργασία των MSB του δεδομένου

IN 20H ;Είσοδος των 4ων MSB του δεδομένου

ANI 00001111b

RLC ;4 φορές ολίσθηση

RLC

RLC

RLC

ORA B

MVI D,00H

MVI E,A

DAD D ;Πρόσθεση δεδομένων

DCR C

END:

POP PSW

EI

RET

β)

LXI H,00H ;Συσσωρευτής δεδομένων

MVI C,64d

MAIN:

IN 20H ;Αναμονή μέχρι να λάβουμε χ7=1

ANI 80H

JP MAIN

MOV A,C

ANI 00000001b ;00000001 δυαδικό για το LSB

JPO 4MSB ;Έλεγχος αν λάβαμε LSB or MSB

IN 20H ;Είσοδος των 4ων LSB

ANI 00001111b ;δυαδικό για τα 4 LSB της πόρτας

MOV B,A ;Προσωρινή αποθήκευση μέχρι τη λήψη του MSB του δεδομένου

JMP 4LSB ;Επιστροφή στο πρόγραμμα μέχρι να ληφθούν τα MSB

4MSB:

IN 20H ;Είσοδος των 4ων MSB

ANI 00001111b

RLC ;4 φορές ολίσθηση

RLC

RLC

RLC

ORA B ;Ένωση με τα LSB του δεδομένου

MVI D,00H

DAD D ;Πρόσθεση δεδομένων

4LSB:

DCR C ;Μείωση μετρητή

JZ ADDR

CHECK: ;Αναμονή για χ7=0

IN 20H

ANI 80H

JM CHECK

JMP MAIN

ADDR:

DAD H ;3 φορές πρόσθεση H-L στον εαυτό του για αριστερή ολίσθηση 3 φορές

DAD H

DAD H

MOV A,L

ANI 80H

MVI L,00H ;L=00H για 8 bit ακρίβεια

CPI 00H

JNZ ROUND ;Αν το MSB του L είναι ίσο με 1 να γίνει άνω στρογγυλοποίηση

BACK:

HLT

ROUNDING: ;Στρογγυλοποίηση

INR H

JMP BACK