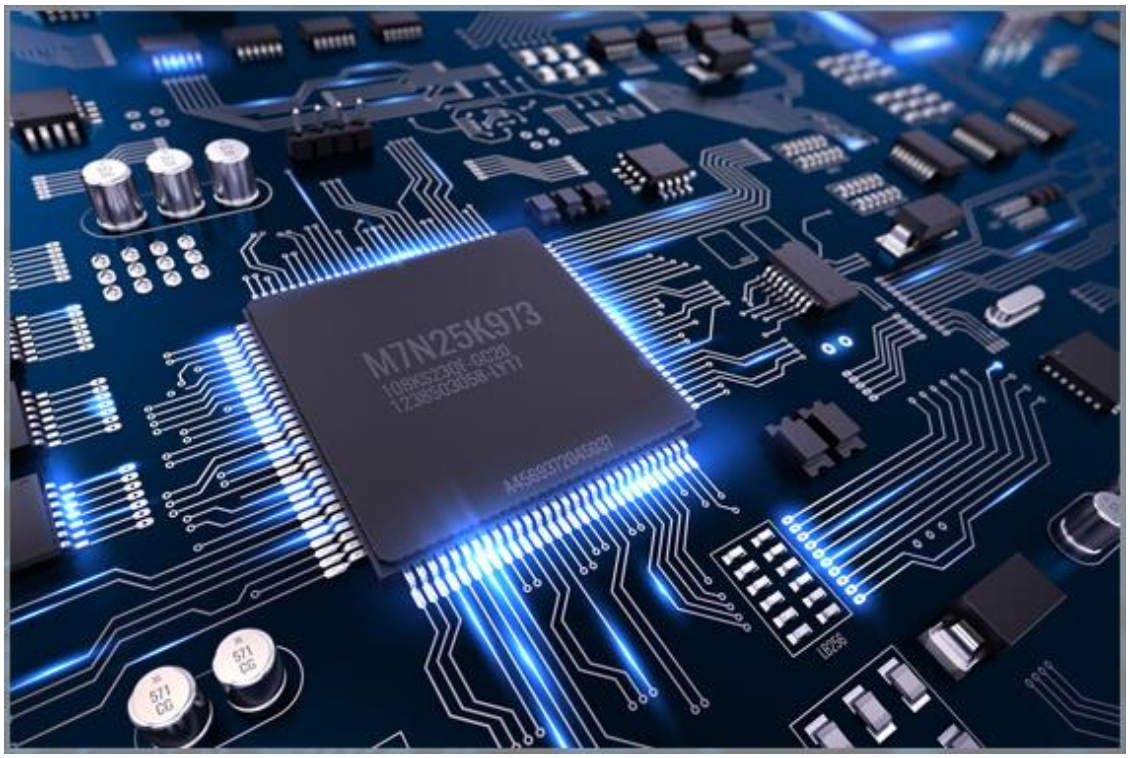
ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

3Η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ



ΜΑΥ 16, 2021

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028
ΚΡΙΣ ΚΟΥΤΣΗ – EL18905

ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

Άσκηση 1

```
     IN 10H

     MVI A,10H          ;Set up Display
     STA 0B00H
     STA 0B01H
     STA 0B02H
     STA 0B03H
     STA 0B04H
     STA 0B05H

     MVI A,0DH          ;Initialization of Interupt mask
     SIM
     EI
WAIT:
     JMP WAIT
INTR_ROUTINE:
     POP H              ;POP return address so that the stack doesn't fill up
     EI                 ;Enable interrupts inside interrupt routine
     MVI A,00H          ;Turn on LEDs
     STA 3000H
     MVI H,06H          ;Counter for 6 iterations
     MOV A,H
     DCR A              ;Set up tens
     STA 0B03H          ;Store tens in the 4th segment display
SECONDS:
     MVI A,09H          ;Set up 9 secs (units)
LIGHTS_ON:
     STA 0B02H          ;Store units in the 3rd segment display
     CALL DISPLAY
     DCR A
     CPI 00H            ;Compare with zero
     JNZ LIGHTS_ON      ;If Z=0 then 9 seconds passed
     CALL ZERO          ;Display zero unit (1 sec)
     DCR H              ;Decrease counter
     JZ EXIT            ;If Z=0 end timer
     MOV A,H
     DCR A
     STA 0B03H
     JMP SECONDS        ;Reapeat for 60 seconds
EXIT:
     MVI A,FFH          ;Turn off LEDs
     STA 3000H
     JMP WAIT           ;Return to wait (main program)
DISPLAY:
     LXI B,0064H        ;100 msec delay
     LXI D,0B00H        ;For STDM
     PUSH PSW
     PUSH H
     PUSH D
     PUSH B
     CALL STDM
     MVI A,0AH          ;10*100msec=1sec
```

```
1SEC:
     CALL DCD
     CALL DELB
     DCR A
     CPI 00H
     JNZ 1SEC
     POP B
     POP D
     POP H
     POP PSW
     RET
ZERO:                    ;Display zero in the 3rd segment display
     MVI A,00H
     STA 0B02H
     CALL DISPLAY
     CALL DELB
     RET

     END
```

Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση_1.8085*

Άσκηση 2

```
     IN 10H
     MVI A,10H      ;Setup display
     STA 0B00H
     STA 0B01H
     STA 0B02H
     STA 0B03H
     STA 0B04H
     STA 0B05H
     MVI A,0DH      ;Initialization of Interupt mask
     SIM
     EI

WAIT:
     PUSH D
     CALL DCD       ;Keep the display on
     POP D
     JMP WAIT
INTR_ROUTINE:
     CALL KIND      ;Insert units from keyboard
     STA 0B00H      ;Store units in the 1st segment display
     MOV L,A        ;Store A temporarily until we get the tens digit
     CALL KIND      ;Insert tens from keyboard
     STA 0B01H      ;Store tens in the 2nd segment display
     RLC            ;Rotate left 4 times to move the tens' digit to the 4 MSBs
     RLC
     RLC
     RLC
     ORA L          ;Store the whole number in one register
     MOV L,A        ;Save the number to L
     EI
     MOV A,L
     CALL DISPLAY
     CMP D          ;Compare with K1
     JC RANGE1      ;A<K1?
     JZ RANGE1      ;A=K1?
     CMP E          ;Compare with K2
     JC RANGE2      ;A<K2?
     JZ RANGE2      ;A=K2?
     MVI A,FBH      ;[0...K1] => 1st LED
     STA 3000H      ;Turn on the 1st LED
     RET
RANGE1:
     MVI A,FEH      ;(K1...K2] => 2nd LED
     STA 3000H      ;Turn on the 2nd LED
     RET
RANGE2:
     MVI A,FDH      ;(K2...FFH] => 3ï LED
     STA 3000H      ;Turn on the 3rd LED
     RET
```

```
DISPLAY:
     LXI D,0B00H    ;For STDM
     PUSH PSW
     PUSH H
     PUSH D
     PUSH B
     CALL STDM
     CALL DCD
     POP B
     POP D
     POP H
     POP PSW
     MVI D,20H      ;K1
     MVI E,40H      ;K2
     RET

     END
```

Το πρόγραμμα βρίσκεται στο αρχείο *Άσκηση_2.8085*

Άσκηση 3

α)

```
SWAP Nible MACRO Q
     PUSH PSW
     MOV A,M
     RRC
     RRC
     RRC
     RRC
     MOV M,A
     MOV A,Q
     RLC
     RLC
     RLC
     RLC
     MOV Q,A
     POP PSW

     ENDM
```

β)

```
FILL MACRO RP, X, K
     PUSH PSW
     PUSH H
     MOV H,R
     MOV L,P
LOOP:
     MOV M,K
     INX H
     DCR X
     JNZ LOOP
     POP H
     POP PSW

     ENDM
```

γ)

```
RHLR MACRO n
     PUSH PSW
     PUSH H
LOOP:
     MOV A,n
     CPI 00H
     JZ END
     MOV A,H
     RAR
     MOV A,L
     RAR
     DCR n
     JMP LOOP
END:
     POP H
     POP PSW

     ENDM
```

Άσκηση 4

Αρχικά ο μετρητής προγράμματος είναι (PC) = 0800H και ο δείκτης σωρού (SP) = 3000H. Κατά την εκτέλεση της εντολής **CALL 0880H**, προκαλείται hardware διακοπή RST 7.5, οπότε θα πάρει προτεραιότητα η εξυπηρέτησή της σε σχέση με το υπόλοιπο πρόγραμμα. Θα έχουμε, λοιπόν, τις ακόλουθες λειτουργίες:

1. Ολοκληρώνεται η εκτέλεση της εντολής **CALL** και αποθηκεύεται στην στοίβα διεύθυνση της εντολής που βρίσκεται μετά το **CALL** (η 0803H), καθώς η εντολή αυτή έχει μέγεθος 3 bytes. Ο PC παίρνει την τιμή 0880H. (SP) = (SP) – 2 = 2FFEH.
2. Στη συνέχεια, ενεργοποιείται η ρουτίνα εξυπηρέτησης διακοπής RST 7.5 και ο PC παίρνει ως τιμή την διεύθυνση 003CH. Ακόμη, αποθηκεύεται στη στοίβα η διεύθυνση 0880H. (SP) = (SP) – 2 = 2FFCH
3. Αφού τελειώσει η ρουτίνα εξυπηρέτησης της RST 7.5, γίνεται POP από την στοίβα η προηγούμενη διεύθυνση του PC (δηλαδή η 0880H) και ο PC παίρνει την τιμή αυτή. (SP) = (SP) + 2 = 2FFEH.
4. Τέλος, ολοκληρώνεται η εκτέλεση της τελευταίας ρουτίνας στην διεύθυνση 0880H και γίνεται POP η διεύθυνση 0803H από την στοίβα και ο PC λαμβάνει την τιμή αυτή. (SP) = (SP) + 2 = 3000H
5. Συνεχίζεται η εκτέλεση του προγράμματος από την διεύθυνση 0803H

2FFCH					SP-4	80H				
2FFDH					SP-3	08H				
2FFEH			SP-2	03H	SP-2	03H	SP-2	00H		
2FFFH			SP-1	08H	SP-1	08H	SP-1	08H		
SP	PC	0800H	PC	0880H	PC	003CH	PC	0880H	PC	0803H
	1		**2**		**3**		**4**		**5**	

Άσκηση 5

α)

```
     MVI A,0DH      ;Μάσκα διακοπών
     SIM
     LXI H,00H      ;Συσσωρευτής δεδομένων
     MVI C,64d      ;64 δεκαδικό στον μετρητή δεδομένων
     EI             ;Ενεργοποίηση διακοπών
ADDR:               ;Αναμονή δεδομένων
     MVI A,C
     CPI 00H
     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
     MOV A,L
     ANI 80H
     MVI L,00H      ;Θέτουμε L=00H για να έχουμε 8bit ακρίβεια
     CPI 00H
     JNZ ROUNDING   ;Αν το MSB του L είναι 1 να γίνει άνω στρογγυλοποίηση
BACK:
     HLT
ROUNDING:           ;Άνω στρογγυλοποίηση
     INR H
     JMP BACK

0034:
     JMP RST6.5
RST6.5:
     PUSH PSW
     MOV A,C
     ANI 00000001b  ;00000001 δυαδικό για το LSB
     JPO 4MSB       ;Έλεγχος αν λάβαμε τα LSB ή τα MSB του δεδομένου
     IN 20H         ;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 4LSB       ;Επιστροφή στο πρόγραμμα μέχρι να ληφθούν τα MSB του δεδομένου
4MSB:               ;Επεξεργασία των MSB του δεδομένου
     IN 20H         ;Είσοδος των 4 MSB του δεδομένου
     ANI 00001111b
     RLC            ;4 φορές ολίσθηση για να τοποθετηθεί το τμήμα του δεδομένου στα MSB
     RLC
     RLC
     RLC
     ORA B          ;Ένωση με τα LSB του δεδομένου
     MVI D,00H
     MOV E,A
     DAD D          ;Πρόσθεση δεδομένων
4LSB:
     POP PSW
     DCR C          ;Μείωση μετρητή
     EI
     RET
```

β)

```
     LXI H,00H      ;Συσσωρευτής δεδομένων
     MVI C,64d      ;64 δεκαδικό στον μετρητή δεδομένων
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 τα LSB ή τα MSB του δεδομένο
     IN 20H         ;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 4LSB       ;Επιστροφή στο πρόγραμμα μέχρι να ληφθούν τα MSB του δεδομένου
4MSB:               ;Επεξεργασία των MSB του δεδομένου
     IN 20H         ;Είσοδος των 4 MSB του δεδομένου
     ANI 00001111b
     RLC            ;4 φορές ολίσθηση για να τοποθετηθεί το τμήμα του δεδομένου στα MSB
     RLC
     RLC
     RLC
     ORA B          ;Ένωση με τα LSB του δεδομένου
     MVI D,00H
     MOV E,A
     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 του H-L στον εαυτό του για ολίσθηση 3 φορές αριστερά
     DAD H
     DAD H
     MOV A,L
     ANI 80H
     MVI L,00H      ;Θέτουμε L=00H για να έχουμε 8bit ακρίβεια
     CPI 00H
     JNZ ROUNDING   ;Αν το MSB του L είναι 1 να γίνει άνω στρογγυλοποίηση
BACK:
     HLT
ROUNDING:           ;Άνω στρογγυλοποίηση
     INR H
     JMP BACK
```