

3η Σειρά Ασκήσεων

Ειρήνη Στρουμπάκου 03121183

Χαράλαμπος Παπαδόπουλος 03120199

1) Το πρόγραμμα σε γλώσσα assembly είναι το εξής:

```
SETUP_DISPLAY:
    LXI B,0032H
    MVI A,00H
    STA 0B00H
    MVI A,00H
    STA 0B01H

START:
    MVI A,0DH
    SIM
    EI

MAIN:
    JMP MAIN

INTR_ROUTINE:
    POP H
    MVI D,2DH
    MVI E,00H
    MVI H,00H
    MVI L,FFH
    EI

LIGHTS:
    INR E
    MOV A,E
    CPI 05H
    JNZ SKIP

    MOV A,L
    CMA
    MOV L,A
    STA 3000H
    INR H
    MVI E,00H
    MOV A,H
    CPI 04H
    JNZ SKIP

    DCR D
    MVI H,00H
    MOV A,D
    CPI 00H
    JZ START

SKIP:
    CALL DELB
    PUSH H
    MOV A,D
    MVI B,00H
```

```

BASE10:
    CPI 0AH
    JC FINISH
    SUI 0AH
    INR B
    JMP BASE10

FINISH:
    STA 0B00H
    MOV A,B
    STA 0B01H
    PUSH D
    LXI D,0B00H
    CALL STDM
    CALL DCD
    MOV D,B
    MOV E,C
    LXI B,0032H
    POP D
    POP H
    JMP LIGHTS

END

```

Τρέχοντας το πρόγραμμα παρατηρούμε ότι πράγματι με το πλήκτρο INTR, που αντιστοιχεί στην διακοπή RST6.5 τα leds στις εξόδους πραγματοποιούν τις επιθυμητές λειτουργίες.

2) Το πρόγραμμα σε γλώσσα assembly είναι το εξής:

```

MVI C,28H
MVI D,46H
MVI E,64H

MVI A,10H ;We empty the screen
STA 0B00H
STA 0B01H
STA 0B02H
STA 0B03H
STA 0B04H
STA 0B05H
MVI A,0DH ;We allow the RST6.5
SIM
EI
WAIT: ;The main program makes a non ending loop
    JMP WAIT

INTR_ROUTINE:
    CALL KIND
    MOV B,A ;Store the first hex digit, which we have assumed
            ;symbolizes the units
    STA 0B02H ;Show in the middle display
    CALL KIND
    STA 0B03H

```

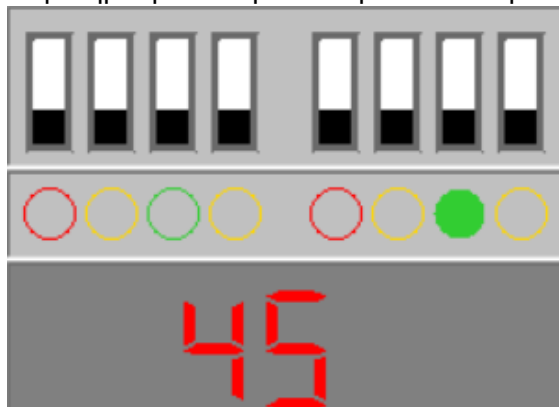
```

RLC
RLC
RLC
RLC
ADD B ;Add the units to the tens
MOV B,A
PUSH D ;temporarily save D and E
LXI D,0B00H ;address for STDM
CALL STDM
CALL DCD
POP D ;restore D and E
MOV A,B
CMP C
JC LED0 ;K1
CMP D
JC LED1 ;K2
CMP E
JC LED2 ;K3
JMP LED3 ;Else go to the final range
LED0:
MVI A,FEH
JMP ENDING
LED1:
MVI A,FDH
JMP ENDING
LED2:
MVI A,FBH
JMP ENDING
LED3:
MVI A,F7H
JMP ENDING
ENDING:
STA 3000H ;Turn on the proper led
EI
JMP WAIT
END

```

Τρέχουμε το πρόγραμμα και αφού πατήσουμε την διακοπή INTR, πληκτρολογούμε τα ψηφία 5 και 4. Έχουμε υποθέσει ότι το 1^ο ψηφίο αντιστοιχεί στις μονάδες και το 2^ο στις δεκάδες και βλέπουμε ότι πράγματι παίρνουμε στην οθόνη τον αριθμό 45 στις 2 μεσαίες θέσεις.

Όσον αφορά τα leds, τα όρια k, που έχουμε ορίσει, είναι 28H, 46H, 64H. Παρατηρούμε ότι ορθά ανάβει το δεύτερο led, αφού $28H < 45H < 46H$.



3) α)

```
INR16 ADDR:
    PUSH B
    PUSH H
    LXI H, ADDR
    MOV C, M
    INX H
    MOV B, M
    INX B
    MOV M, B
    DCX H
    MOV M, C
    POP H
    POP B
    ENDM
```

b)

```
FILL ADDR K:
    PUSH PSW
    PUSH H
    LXI H, ADDR
    MOV A, K
LOOP:
    MOV M, A
    INX H
    DCR A
    JNZ LOOP
    POP H
    POP PSW
    ENDM
```

c)

```
RHLL Q, R:
    MOV A, R
    RAL
    MOV R, A
    MOV A, Q
    RAL
    MOV Q, A
```

4) Η εντολή RST6.5 είναι hardware εντολή, οπότε πρώτα θα ολοκληρωθεί η εντολή JMP 2200H και μετά θα εκτελεστεί η ρουτίνα διακοπής. Άρα (PC)<-2200H.

Έπειτα, εκτελείται η ρουτίνα διακοπής:

Αρχικά, αποθηκεύεται το PCH στο (SP)-1, δηλαδή

2FFFH<-22H

και το PCL στο (SP)-2, δηλαδή

2FFEH<-00H

Έπειτα, το μέγεθος της στοίβας αλλάζει και γίνεται

(SP)<-(SP)-2 => (SP)<-2FFEH

Στον PC αποθηκεύεται η διεύθυνση της διακοπής ως εξής:

$n=6.5 \Rightarrow 8 \cdot n=52$ και το 52 στο δεκαεξαδικό σύστημα είναι το 34H, άρα (PC)<-0034.

Αφού ολοκληρωθεί η ρουτίνα εξυπηρέτησης, ο PC ξαναπαίρνει την παλιά του τιμή, δηλαδή (PC)<-2200H και ο SP αυξάνεται κατά 2, οπότε πάλι (SP)<-3000H.

5) a)

```
RST5.5:
JMP INTRPT
START: MVI D,20H
MVI H,00H
MVI L,00H
MVI B,00H
MVI A,0EH
SIM
RETURN: MOV A,D
CPI 00H
JZ FINISH
EI
WAIT:
JMP WAIT
INTRPT:
INX SP
INX SP
DCR D
MOV A,D
ANI 01H
CPI 00H
JZ LSBS
IN 20H
ANI 0FH
RLC
RLC
RLC
RLC
MOV C,A
JMP RETURN
LSBS:
IN 20H
ANI 0FH
ADD C
MOV C,A
DAD B
JMP RETURN
FINISH: DI
DAD H
DAD H
DAD H
DAD H
END
```

b)

```
LXI H,00H
MVI C,32d
START:
IN 20H
ANI 01H
```

```
JZ START
MOV A,C
ANI 00000001b
MOV B,A
JMP LSBS
MSBS:
IN 20H
ANI 00001111b
RLC
RLC
RLC
RLC
ORA B
MVI D,00H
MOV E,A
DAD D
LSBS:
DCR C
JZ ADDR
CHECK:
IN 20H
ANI 01H
JNZ CHECK
JMP START
ADDR:
DAD H
DAD H
DAD H
MOV A,L
ANI 01H
MVI L,00H
JNZ ROUND
ENDING:
HLT
ROUND:
INR H
JMP ENDING
```