Συστήματα Μικρουπολογιστών - 3ή Σειρά Ασκήσεων

Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών
Ακαδημαϊκό έτος : 2018 – 2019 Εξάμηνο : 6ό
Μέλη ομάδας : Βόσινας Κωνσταντίνος ΑΜ : 03116435
Ανδριόπουλος Κωνσταντίνος ΑΜ : 03116023

Άσκηση 1η

Δίνεται το πρόγραμμα σε 8085. Τα προγράμματα ελέγχθηκαν στον προσομοιωτή TSIK 8085.

```
START:
    IN 10H
    MVI A,0Dh   ;Set appropriate mask for RST5.5
    SIM
    EI          ;Enable Interrupts

INPUT_LOOP: JMP INPUT_LOOP

INTR_ROUTINE:
    EI
    MVI B,03H       ;Set (BC) = 1000, 1 sec delay
    MVI C,E8H
    CALL TURNON_LED ;Turn on leds
    MVI A,3CH       ;Set A=60, it is a timer

CHECK1:
    PUSH PSW
                ;First seperate digits of A to print on 7-segment display
    MOV D,A     ;Mov to D temporarily
    ANI 0FH     ;Keep only 4 LSB's
    STA 0900H   ;Store to adress 0900, from where it will be displayed

    MOV A,D
    ANI F0H     ;Keep only 4 MSB's
    RRC         ;Shift 4 times, so they are in the place of 4 LSB's
    RRC
    RRC
    RRC
    STA 0901H   ;Store in next memory location
    LXI D,0900H ;Point DE to memory location 0900

    CALL STDM   ;Display remaining time
    CALL DCD
    CALL DELB

    POP PSW
    DCR A             ;Decrement timer,
    CPI 00H           ;Check if it has reached zero
    JNZ CHECK1        ;If not, repeat
    CALL TURNOFF_LED
    EI                ;Else, wait for another interrupt
    JMP INPUT_LOOP

TURNON_LED:
    MVI A,00H
    STA 3000H
    RET

TURNOFF_LED:
    MVI A,FFH
    STA 3000H
    RET
END
```

Άσκηση 2η

```
START:
    IN 10H      ;Disable Memory protection
    MVI A,0Dh   ;Set appropriate mask for RST5.5
    SIM
    EI

INPUT_LOOP: JMP INPUT_LOOP

INTR_ROUTINE:
    CALL TURNOFF_LED    ;Turn off leds
    CALL BLNK           ;Set display to blank

READ1:
    CALL KIND   ;Get input
    MOV D,A     ;Save it to D
    CPI 10H     ;Read until you get number between [00,0F]
    JNC READ1
READ2:
    CALL KIND   ;Get second input
    MOV E,A     ;Save it to E
    CPI 10H
    JNC READ1
    EI

DISPLAY:
    MOV A,D     ;Move first number (LSB) to accumulator
    STA 0900H   ;Store to adress 0900, from where it will be displayed

    MOV A,E     ;Move next number (MSB) to accumulator
    STA 0901H   ;Store in next memory location

    MOV A,E     ;Move MSB to accumulator and rotate 4 times
    RRC
    RRC
    RRC
    RRC
    ADD D       ;Add LSB
    PUSH PSW

    LXI D,0900H ;Point DE to memory location 0900

    CALL STDM   ;Call 7-segment display processes

    POP PSW     ;Check which LED will turn on
    CMP C       ;If (num)>C then turn on led 3
    JNC CHECK2
    MVI A,FBH
    JMP L1
CHECK2:
    CMP B       ;Else if (num)>B
    JNC CHECK3
    MVI A,FDH   ;Turn on led 2
    JMP L1
CHECK3:
    MVI A,FEH   ;Else turn on led 1
L1:
    STA 3000H   ;Loop until new interrupt occurs
    CALL DCD
    JMP L1
```

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

TURNOFF_LED:
    MVI A,FFH
    STA 3000H
    RET
END
```

Άσκηση 3η

α)

```
SWAP MACRO Nible Q
PUSH H                    ; STORING H ,L
PUSH L
MOV A,Q                   ; SWAPING Q BITS BY USING 4
RLC                       ; LEFT ROTATIONS
RLC
RLC
RLC
MOV Q,A                   ; RENEWING Q VALUE
POP H                     ; POPING H AND L FROM STACK
POP L                     ; SO THAT H VALUE SWAPS WITH L
ENDM
```

β)

```
FILL MACRO ADDR, LENGTH, K

PUSH L              ;STORE L, H
PUSH H              ;BECAUSE THEY ARE GOING TO CHANGE
PUSH C              ;STORE C FOR THE SAME REASON
LXI H,ADDR          ;POINT TO THE FIRST “ARRAY” ELEMENT
MVI C,LENGTH        ;STORE “ARRAY” LENGTH

FILL_ARRAY:

MVI M,K             ;WRITE TO “ARRAY”
INX H               ;POINT TO THE NEXT “ARRAY” CELL
DCR C
JNZ FILL_ARRAY      ;IF WHOLE “ARRAY” IS FILLED RETURN
POP C               ;RESTORE REGISTERS
POP H
POP L

ENDM
```

γ)

```
RHLL MACRO N
PUSH B              ; STORING B
MVI B, N            ; N- >B

LOOP:
MOV A,L             ; ROTATING CY,H AND L BITS TO THE LEFT
RAL
MOV L,A
MOV A,H
RAL
MOV H,A
DCR B
JNZ LOOP            ; ROTATING N TIMES
```

POP B
ENDM

Άσκηση 4η

Στην αρχή, η τιμή του Program Counter (PC) ισούται με 2000H όπως μας δίνεται στην εκφώνηση. Επίσης, ο δείκτης της στοίβας (Stack Pointer=SP) έχει την τιμή 4000H.

Η διακοπή πραγματοποιείται στη μέση της εντολής CALL 3000H, δηλαδή έχει ήδη αρχίσει να εκτελείται η συγκεκριμένη εντολή,οπότε η διακοπή στην ουσία θα αναγνωριστεί από το σύστημα αφού ολοκληρωθεί και ο τελευταίος κύκλος της εντολής CALL. Επομένως, ο PC θα πάρει πρώτα την τιμή 3000H όπως του ορίζει η εντολή CALL 3000H. Αμέσως μετά θα αναγνωριστεί η διακοπή, θα απενεργοποιηθούν αυτόματα οι διακοπές και θα αποθηκευτεί η τιμή του PC στη στοίβα. Η τιμή του PC αποτελείται από 16 Bits επομένως για να αποθηκευτεί θέλει 2 θέσεις της στοίβας.
Άρα οι μεταβολές που θα πραγματοποιηθούν στη στοίβα είναι οι ακόλουθες:

1) Στη θέση (SP)-1 θα αποθηκευτεί η τιμή 30H (τα 8 σημαντικότερα bits του PC)
2) Στη θέση (SP)-2 θα αποθηκευτεί η τιμή 00H (τα 8 λιγότερο σημαντικά bits του PC)
3) Η τιμή του δείκτη στοίβας (SP) θα μειωθεί κατά δύο για να δείχνει την επόμενη θέση που τυχόν θα εισαχθεί το επόμενο στοιχείο στη στοίβα

Πιο συγκεκριμένα θα πραγματοποιηθούν τα ακόλουθα:

1) ((SP)-1) ← 30H (PCHigh)
2) ((SP)-2) ← 00H (PCLow)
3) (SP) = (SP)-2 (δηλαδή (SP)=3FFEH)

Τέλος, ο PC θα πάρει την διεύθυνση της ρουτίνας RST 5.5 και θα εκτελεστεί η ρουτίνα εξυπηρέτησης της διακοπής.

Άσκηση 5η

Σε αυτήν την άσκηση, ζητείται να γράψουμε ένα πρόγραμμα σε Assembly 8085 το οποίο θα λαμβάνει 32 δεδομένα των 8 Bit από μία συσκευή και καθένα από αυτά να το μεταφέρει με 2 βήματα, πρώτα τα 4 LSB και έπειτα τα 4 MSB, μέσω των Bits (X_0-X_3) της θύρας PORT_IN. Τα υπόλοιπα 4 MSBits δε χρησιμοποιούνται. Στο σύστημα αυτό, μετά από κάθε αποστολή, η συσκευή προκαλεί διακοπή τύπου RST 5.5. Τέλος, ζητείται να υπολογιστεί ο μέσος όρος των 32 δεδομένων με ακρίβεια των 8 Bit.

```
MVI A,0EH   ; μάσκα διακοπών = 00001110
SIM
LXI H,0000H   ; (H)(L)<-0
MVI C,40H   ; μεταφέρω στον C την τιμή 64 για τις επαναλήψεις
EI       ;ενεργοποίηση διακοπών

FIRST:          ; αναμονή μέχρι C=0
MVI A,C         ; εκτέλεση 64 φορές
CPI 00H
JNZ FIRST
DI              ; απενεργοποίηση διακοπών
DAD H           ; ολίσθηση 3 θέσεις αριστερά
DAD H
DAD H
HLT

RST6.5:
PUSH PSW    ; κρατάω το περιεχόμενο του A και των flags
MOV A,C
ANI 01H
CPI 00H         ; έλεγχος αν πρόκειται για LSB ή MSB
JZ SECOND
IN PORT_IN  ; ή IN 80H
ANI 0FH       ; Αν διαβάζεται LSB μηδενίζονται τα 4 MSB
MVI D,00H
MOV E,A
JMP FINISH

SECOND:
IN PORT_IN  ; ή IN 80H
RRC             ; δεδομένα στα 4 MSB
RRC
RRC
RRC
ANI F0H         ; μηδενισμός των 4 LSB
MVI D,00H
MOV E,A

FINISH:
DAD D           ; sum (HL) += D
DCR C
```

```
POP PSW     ;επαναφορά του A και των flags
EI          ;ενεργοποίηση διακοπών
RET
```

Για το μέσο όρο των 32 δεδομένων, κάνουμε 3 ολισθήσεις προς τα αριστερά του HL. Αυτό έχει ως αποτέλεσμα το ακέραιο μέρος του Μ.Ο. να βρίσκεται στον καταχωρητή H και το κλασματικό μέρος στον καταχωρητή L.