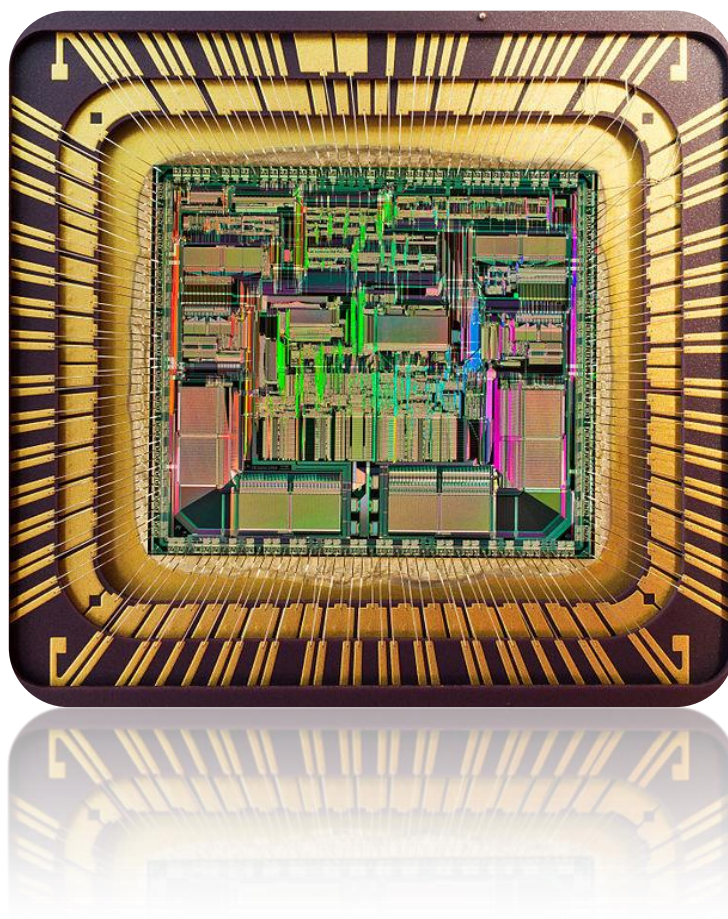


4η Ομάδα Ασκήσεων

Βαβουλιώτης Γεώργιος
ΑΜ : 03112083
6^ο Εξάμηνο

Γιαννόπουλος Αναστάσης
Α.Μ. : 03112176
6^ο Εξάμηνο



ΑΣΚΗΣΗ 1

Ο κώδικας σε assembly 8085 για την πρώτη άσκηση φαίνεται παρακάτω:

```
LXI B,0014H      ;DELAY IS 20MS
IN 10H
MVI A,0DH        ;SWITCH MASK 00001101
SIM
EI               ;SWITCH ENABLE
MVI A,00H
```

```
WAIT:
LXI H,0910H      ;IF CONTENT IS 2 THEN INTERRUPT
MOV A,M
CPI 02H          ;COMPARE WITH 2 DEC
JZ RST5
JMP WAIT
```

```
RST5:
LXI H,0910H      ;MAKE CONTENT ZERO AND WAIT FOR ANOTHER INTERRUPT
MVI M,00H
LXI H,0900H      ;START VALUE FOR 7 SEGMENTS
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,10H
INX H
MVI M,00H
INX H
MVI M,06H
MVI H,7AH        ;120 LOOPS
MVI A,FFH
```

```
SHOW1:
MVI L,19H        ;(L)=25(500MS DELAY)
MVI A,00H        ;(A)=0
```

```
SHOW2:
DI               ;TURN OFF THE SWITCHES
PUSH H           ;PUSH HL IN THE STACK
PUSH B           ;PUSH BC IN THE STACK
MOV B,A
LXI H,0910H
MOV A,M
CPI 01H          ;IF PUSH INTR GO TO LABEL WAIT
MOV A,B
POP B            ;POP OUT BC FROM THE STACK
POP H            ;POP OUT HL FROM THE STACK
EI
STA 3000H
JZ WAIT
CALL DELB        ;DELAY 20MS
JMP DISPLAY1     ;SHOW THE TIME LEFT
```

CMP1:

```

DCR L
JNZ SHOW2
INR D
MOV E,A          ; (E) <-- (A)
MOV A,D          ; 1sec
CPI 02H          ; COMPARE WITH 2 DEC
JZ CREATE

```

KEEP_ON:

```

MOV A,E
DCR H            ; DECREASE THE LOOPS
JNZ SHOW1
JMP WAIT        ; IF 1 MIN HAS PASSED GO TO WAIT

```

DISPLAY1:

```

DI              ; TURN OFF THE SWITCHES
PUSH B          ; PUSH HL IN THE STACK
PUSH D          ; PUSH DE IN THE STACK
PUSH H          ; PUSH HL IN THE STACK
LXI D,0900H     ; (DE) <-- CONTENT OF ALL 7-SEGMENTS
MOV B,A
CALL STD
CALL DCD
MOV A,B
POP H           ; POP OUT HL FROM THE STACK
POP D           ; POP OUT DE FROM THE STACK
POP B           ; POP OUT BC FROM THE STACK
EI
JMP CMP1

```

CREATE:

```

DI              ; TURN OFF THE SWITCHES
PUSH B          ; PUSH BC IN THE STACK
PUSH H          ; PUSH HL IN THE STACK
LXI H,0904H     ; DECREASE THE CONTENT OF THE LEFT DIGIT
DCR M
MOV B,A
MOV A,M         ; IF (M)=255 DEC DECREASE BY 1
CPI FFH         ; COMPARE WITH FFH=255 DECIMAL
JZ ZERO_END

```

NEXT_MOVE:

```

MOV A,B         ; (A) <-- (B)
POP H           ; POP OUT HL FROM THE STACK
POP B           ; POP OUT BC FROM THE STACK
EI
MVI D,00H       ; (D) <-- 0
JMP KEEP_ON

```

ZERO_END:

```

DI
PUSH B          ; PUSH BC IN THE STACK
PUSH H          ; PUSH HL IN THE STACK
LXI H,0905H     ; LOAD THE LEFTER 7-SEGMENT
DCR M           ; HAVE TO DECREASE THE DECADES
POP H           ; POP OUT HL FROM THE STACK
POP B           ; POP OUT BC FROM THE STACK
MVI M,09H       ; (M)=9 TO START COUNTING AGAIN
DCR H           ; DECREASE THE MEMORY POINTER
DCR M           ; DECREASE THE CONTENT OF THE MEMORY
JMP NEXT_MOVE

```

INTR ROUTINE:

```
PUSH H          ;PUSH H IN STACK TO SAVE IT
LXI H,0910H
INR M
POP H           ;POP OUT H FROM THE STACK
EI
RET             ;RESTORE
```

END:

END

ΑΣΚΗΣΗ 2

Στην άσκηση αυτή, υλοποιήσαμε σε γλώσσα assembly του 8085 ένα πρόγραμμα που εκτελεί την παρακάτω λειτουργία: Όταν προκαλείται διακοπή τύπου 6.5 το πρόγραμμα διαβάζει δύο διαδοχικά ψηφία ενός δεκαεξαδικού αριθμού από το πληκτρολόγιο και τα απεικονίζει στα δύο δεξιότερα 7 segment displays. Επιπλέον, ανιχνεύει το πεδίο τιμών στο οποίο ανήκει ο αριθμός που πατήθηκε και ανάβει κάποιο ενδεικτικό LED. Συνολικά, έχουμε 3 πεδία τιμών: [00H,K1], (K1,K2] , (K2,K3]. Οι τιμές των K1, K2 , K3 περιέχονται στους καταχωρητές B, C και D αντίστοιχα. Σημειώνουμε, ότι το πρώτο πεδίο αντιστοιχεί στο 3ο LED, το δεύτερο στο 2ο και το τρίτο στο 1ο πεδίο στο LSB LED. Κάθε φορά που καλείται η ρουτίνα INTR_ROUTINE (δύο φορές για κάθε πάτημα του INTRPT) διαβάζουμε το ψηφίο, το αποθηκεύουμε στη διεύθυνση που δείχνει ο HL και συμπληρώνουμε το τελευταίο bit του HL. Κατά συνέπεια, για κάθε ζευγάρι INTR_ROUTINE που θα εκτελεστεί τα δύο ψηφία που θα διαβαστούν αποθηκεύονται πάντα σε δύο κατάλληλες διαδοχικές θέσεις μνήμης (09A5H και 09A4H στο πρόγραμμά μας). Στη συνέχεια παρατίθεται ο κώδικας Assmbly που χρησιμοποιήθηκε:

```
IN 10H                ;ACCESS IN THE MEMORY
MVI B,56H             ; (B)=56H=86DEC
MVI C,ACH             ; (C)=172DEC
MVI A,0DH             ;SWITCH ON THE SWITCHES WITH MASK
SIM
EI                    ;ENABLE INTERRUPTS
MVI A,10H             ; (A)=10H
STA 09A5H             ;STORE THE EMPTY CHARACTER IN 7-SEGMENT
STA 09A4H             ;AT THE 4 LAST PLACES WHICH IS UNSED
STA 09A3H
STA 09A2H
LXI H,09A1H           ;2 FROM RIGTH DIGIT OF 7-SEGMENT
INR B                 ;INCREASE THE LIMITS
INR C
```

RELOAD:

```
LDA 09A0H             ;CREATE THE NUMBER TO SEE ONE OF THE FIELDS
ANI 0FH
MOV E,A
LDA 09A1H
RLC
RLC
RLC
RLC
ANI F0H              ;AND WITH 11110000 TO KEEP THE 4 MOST SIGNIFICANT BITS
ORA E                ;CREATE THE NUMBER
CMP B                ;CHECK IF IS AT [0,K1]
JNC CONTINUE         ;CHECK THE NEXT FIELD
MVI A,FBH            ; (A)=11111011 FOR THE THIRD LSB
JMP PUT_THIS
```

CONTINUE:

```
CMP C
JNC KEEP_GOING
MVI A,FDH            ; (A)=11111101 FOR THE SECOND LSB
JMP PUT_THIS
```

KEEP GOING:

```
MVI A,FEH            ; (A)=11111110 FOR THE FIRST LSB
```

PUT THIS:

```
STA 3000H      ;LED IS ON OF THE CERTAIN FIELD
PUSH B         ;PUSH IN THE STACK
PUSH H         ;PUSH AGAIN IN THE STACK
LXI D,09A0H    ;DISPLAY THE DIGITS IN 7-SEGMENT
CALL STDH
CALL DCD
POP H          ;POP OUT FROM THE STACK
POP B          ;POP OUT FROM THE STACK
JMP RELOAD
```

INTR ROUTINE:

```
PUSH PSW       ;STORE A AND FLAGS IN THE STACK
MVI A,00H      ;LED IS ON TO DECLARE AN INTERRUPT
STA 3000H
CALL KIND      ;READ FROM THE KEYBOARD
MOV M,A        ;STORE AT THE ADDRESS OF HL
MOV A,L
XRI 01H        ;XOR WITH 0000001 TO READ THE NEXT DIGIT
MOV L,A
POP PSW        ;POP OUT FROM THE STACK
EI             ;ENABLE THE INTERRUPTS
RET            ;RESTORE
```

END: END

ΑΣΚΗΣΗ 3

Αρχικά η τιμή του PC είναι 2000H ενώ η τιμή του δείκτη στοίβας SP είναι 4000H. Εφόσον εκτελείται ήδη η εντολή CALL 3000H, η διακοπή θα αναγνωριστεί αφού ολοκληρωθεί και ο τελευταίος κύκλος της εντολής CALL. Έτσι, αφού ο PC πάρει την τιμή 3000H, θα αναγνωριστεί η διακοπή, θα απενεργοποιηθούν αυτόματα οι διακοπές, και θα αποθηκευτεί η τιμή του PC στην στοίβα. Έτσι, θα είναι:

$((SP)-1) < 30H$ (PCHigh)

$((SP)-2) < 00H$ (PCLow)

$(SP) = (SP) - 2$ δηλαδή $(SP) = 3FFE H$.

Τέλος, ο PC θα πάρει την διεύθυνση της ρουτίνας RST 6.5 (0034H) και θα εκτελεστεί η ρουτίνα εξυπηρέτησης της διακοπής.

ΑΣΚΗΣΗ 4

Στην τελευταία άσκηση θα γράψουμε ένα πρόγραμμα σε assembly 8085 το οποίο θα παίρνει 32 δεδομένα των 8 bit και τα μεταφέρει σε 2 στάδια: αρχικά τα 4 LSB και μετά τα 4 MSB, μέσω των bits της θύρας PORT_IN (τα 4 MSB bits της θύρας δε χρησιμοποιούνται). Πρέπει να σημειώσουμε ότι μετά από κάθε αποστολή, η συσκευή προκαλεί διακοπή τύπου RST 5.5 και τελικά υπολογίζεται ο μέσος όρος των δεδομένων με ακρίβεια 8 bit.

```
MVI A,0EH      ;THE NUMBER OF INTERRUPTS IS 00001110
SIM
LXI H,0000H    ; (HL) ← 0
MVI B,40H      ; (B)=64 DEC
EI             ;ENABLE INTERRUPTS
```

```
LOOP1:         ;WAIT UNTIL (B)=0
MVI A,B        ;THIS COMMAND EXECUTE 64 TIMES
CPI 00H        ;COMPARE WITH 00 DEC
JNZ LOOP1      ;IF Z=0 GO TO LOOP1
DI             ;TURN OFF THE SWITCHES
DAD H          ;SHIFT LEFT THREE TIMES
DAD H
DAD H
HLT
```

```
INTRPT6.5:
PUSH PSW      ;PUSH A AND FLAGS IN THE STACK
MOV A,B
ANI 01H
CPI 00H        ;CHECK IF IS MSB OR LSB
JZ LABEL2     ;IF Z=1 GO TO LABEL2
IN PORT_IN
ANI 0FH       ;IF IS LSB THE 4 MSB BECAME 0
MVI D,00H
MOV E,A
JMP FINISH
```

```
LABEL2:
IN PORT_IN
RRC           ;SEND THE DATA AT 4 MSB DIGITS WITH 4 RRC
RRC
RRC
RRC
ANI F0H       ;4 LSB BECOME 0
MVI D,00H
MOV E,A
```

```
FINISH:
DAD D         ; (HL) = (HL) + D
DCR B
POP PSW       ;POP OUT A AND FLAGS FROM THE STACK
EI           ;ENABLE INTERRUPTS
RET          ;RESTORE
```

Παρατήρηση: Κάνουμε 3 ολισθήσεις αριστερά του διπλού καταχωρητή HL για τον μέσο όρο των 32 δεδομένων. Άρα το ακέραιο μέρος του μέσου όρου βρίσκεται στον καταχωρητή γενικού σκοπού H και το δεκαδικό μέρος στον καταχωρητή γενικού σκοπού L.