Εργαστήριο Μικροϋπολογιστών
Θοδωρής Παπαρρηγόπουλος
el18040
Ομάδα 21

1ο Εργαστήριο στον 8085

Άσκηση 1

```
; x is C
; My counter is on D

LXI B,1000H ;1000 miliseconds delay

START:
     MVI D,00H

GET_X:
     LDA 2000H ; Read input
     MOV H,A ; Store H
     ANI 80H

     JZ GET_X ; If msb is not on read again
     ;; Split C into X

     MOV A,H
     ANI 0FH ; 0000 1111 To get lsb's
     MOV H,A

     CPI 00H ; If input was 0 then jump to start
     JZ START

     INR H
     JMP INCREASE

INCREASE:
;; check for msb
     LDA 2000H
     ANI 80H
     JZ INCREASE

     MOV A,D ; Get counter
     CMA
     OUT 30H
     CALL DELB

     INR D     ; increase counter
     MOV A,D
     CMP H ; compare counter and x+1
```

```
     JNZ INCREASE ; If not x+1 continue
     DCR D ; get it to x

     JMP DECREASE ; else decrease

DECREASE:
;; check for msb
     LDA 2000H
     ANI 80H
     JZ DECREASE

     DCR D
     MOV A,D
     CMA
     OUT 30H
     CALL DELB

     MOV A,D
     CPI 00H
     JNZ DECREASE
     INR D
     JMP GET_X

END
```

Άσκηση 2

```
IN 10H

START: ;initializations
     MVI A,10H ;space to print
     LXI D,0B00H ; read dcd from here
     STA 0B03H ;space
     STA 0B04H ;space
     STA 0B05H ;space
     MVI A,00H
     STA 0B02H ;initialize to 0

     CALL KIND ;read 1st value
     RLC
     RLC
     RLC
     RLC ;16*x
     MOV C,A

     CALL KIND ;read 2nd value
     ADD C ;16*x+y
     MOV C,A
     CPI C8H ;compare with 200
     JC STATE_LESS_200 ;if smaller
     ;; We are in 2xx
     MVI A,02H ; else print 2 in 3rd digit
     STA 0B02H
     MOV A,C
     SUI C8H ;sub 200 to find decs -> 2xx - 200
     JMP DECA

STATE_LESS_200:
     CPI 64H ;compare with 100
     JC DECA ;if less than 100
     ;; 3rd digit should be 1
     MVI A,01H
     STA 0B02H

     MOV A,C
     SUI 64H ;subract 10

DECA: ;hex to bcd
     MVI B,FFH ;; 255 (MAX value)

DECS:
     INR B
     SUI 0AH
     JNC DECS
```

```
     ADI 0AH
     STA 0B00H
     MOV A,B
     STA 0B01H


     CALL STDM
     CALL DCD ;print 3 digit decimal number
     JMP START
END
```

Άσκηση 3

```
;; We have a counter that we are gonna
;; rotate left and right accordingly
;; It's important to notice that when lsb gets off
;; in 1000 0000 since there is a double switch, it should
;; continue with 0000 0001
;; same thing applies to 0000 0001 when you lsb gets off

LXI B,1000H ;1000 miliseconds delay

START:
     MVI D,01H ; initialize B with 1
     MVI L,00H ; L is the previus position of lsb

LEFT_CHECK:
     IN 20H
     MOV H,A
     ANI 80H
     JZ LEFT_CHECK ;; If msb is not on stay where you are

     MOV A,H ; Move back H
     ANI 01H ; Keep only lsb
     CMP L
     JZ LEFT
     MOV L,A
     CPI 00H
     JZ RIGHT_CHECK

     JMP LEFT

LEFT:
     ;; Rotate left, print
     ;; from 1000 0000 -> 0000 0001 we jump back to right
     CALL DELB
     MOV A,D
     CMA
     OUT 30H
     CMA
     RLC
     MOV D,A
     CPI 01H
     JNZ LEFT_CHECK
     RRC
     JMP RIGHT_CHECK

RIGHT_CHECK:
     IN 20H
     MOV H,A
```

```
     ANI 80H
     JZ RIGHT_CHECK ;; If msb is not on stay where you are

     MOV A,H ; Move back H
     ANI 01H ; Keep only lsb
     CMP L
     JZ RIGHT
     MOV L,A
     CPI 00H
     JZ LEFT_CHECK

     JMP RIGHT

RIGHT:
     CALL DELB
     MOV A,D
     RRC
     CMA
     OUT 30H
     CMA
     MOV D,A
     CPI 01H
     JNZ RIGHT_CHECK
     RLC
     JMP LEFT
END
```