

Εργαστήριο Μικροϋπολογιστών

2η εργαστηριακή άσκηση

Τμήμα: Β Ομάδα: 15

Συνεργάτες: Μαρουφίδης Ιωάννης (03113506),

Περράκης Γεώργιος (03113511) ,

Σοφιανίδης Γεώργιος (03113179)

1^η Άσκηση

```
MVI E,00H          ;counter for interrupts
LXI B,01FAH        ;BC=500 => delay is 0,5 sec
MVI D,01H          ;flag that tells if we have to
MVI A,0DH          ;move right or left
SIM                ;enable interrupts
EI
MVI H,FEH          ;register for leds
```

MOVE_LEFT:

```
MVI D,01H          ;if D is 1 means that we move left
LDA 2000H          ;while LSB of the switches is
RAR                ;off we dont count
JNC MOVE_LEFT
MOV A,H
STA 3000H          ;store the leds
MVI A,0DH
SIM                ;desable interrupt before we use DELB
DI
CALL DELB
EI                ;enable interrupt after
```

```

MOV A,H
RLC          ;move to the next bit
MOV H,A
CPI 7FH      ;check if we arrived to the last bit
JNZ MOVE_LEFT ;if we didn't we continue
STA 3000H    ;store the last bit

```

L1:

```

MVI A,0DH
SIM          ;enable interrupts
EI
JMP L1       ;stop the led and wait for an interrupt

```

MOVE_RIGHT:

```

MVI D,02H    ;if D is 1 means that we move left
LDA 2000H    ;while LSB of the switches is
RAR          ;off we dont count
JNC MOVE_RIGHT
MOV A,H
STA 3000H    ;store the leds
MVI A,0DH
SIM
DI           ;desible interrupt before we use DELB
CALL DELB
EI           ;enable interrupt after
MOV A,H
RRC          ;move to the next bit
MOV H,A
CPI FEH      ;check if we arrived to the last bit
JNZ MOVE_RIGHT ;if we didn't we continue
STA 3000H    ;store the last bit

```

L2:

```
MVI A,0DH
SIM          ;enable interrupts
EI
JMP L2      ;stop the led and wait for an interrupt
```

INTR_ROUTINE:

```
LDA 2000H    ;while LSB of the switches is
RAR          ;off we dont count
JNC L1
INR E        ;E gets values from 1-3
MOV A,E
CPI 03H      ;check if E is 3
JNZ M2       ;if it isn't we continue
MVI E,01H    ;otherwise we set E as 1
```

M2:

```
MOV A,E
CPI 01H      ;check if E is 1
JZ L1        ;if it is that means that we are in the first interrupt and we
;have to stop our led
MOV A,D      ;otherwise we have to move our led
CPI 01H      ;the flag D will show us which way to follow
JZ MOVE_RIGHT
JMP MOVE_LEFT
```

END

Σε αυτή την άσκηση έχουμε ένα led το οποίο ξεκινάει από το LSB και κινείται προς το MSB με καθυστέρηση 0,5 sec σε κάθε θέση. Όταν φτάσει στην άλλη μεριά σταματάει. Όσο το LSB της θύρας εισόδου είναι ON το led κινείται, ενώ όταν είναι OFF σταματάει. Κάθε φορά που προκαλείται διπλή διακοπή αναστρέφεται η κατεύθυνσή του, ενώ σε κάθε απλή διακοπή το led σταματάει εκεί που είναι και περιμένει διπλή διακοπή για να συνεχίσει την κίνησή του.

2ⁿ Άσκηση

```
MVI A,10H      ;code 10H is blank
STA 0B00H      ;and is stored to the 7
STA 0B01H      ;segment displays
STA 0B02H      ;as a starting condition
STA 0B03H
STA 0B04H
STA 0B05H
LXI D,0B00H    ;register D has the starting
CALL STDM      ;address for routine STDM
CALL DCD
CALL KIND      ;while routine KIND is waiting
               ;for a value, routine DCD
               ;is executed by default
               ;Code of the button to
               ;register A
               ;KIND routine recognises
               ;only buttons 0-F
```

LOOP1:

```
MOV H,A
ANI 0FH
MOV C,A        ;right digit of the code
MOV A,H        ;to register C
ANI F0H        ;and left to register B
RRC
RRC
RRC
RRC
MOV B,A
MOV A,C        ;the code is stored to
```

```

    STA 0B00H      ;the 2 right digits
    MOV A,B        ;while the other 4
    STA 0B01H      ;are left as blank
    MVI A,10H
    STA 0B02H
    STA 0B03H
    STA 0B04H
    STA 0B05H
    LXI D,0B00H
    CALL STDMA
    CALL DCD
    CALL KIND
    JMP LOOP1      ;continuous operation
END

```

Σε αυτή την άσκηση το πρόγραμμά μας περιμένει να πατηθούν δύο αριθμοί και μετά με τη βοήθεια της συνάρτησης KIND εμφανίζει τους αριθμούς στα δύο δεξιότερα displays του προσομοιωτή. Οι συναρτήσεις STDMA και DCD βοηθούν στην απεικόνιση των αριθμών. Τέλος, οι θέσεις μνήμης στη RAM που χρησιμοποιήθηκαν είναι 0B00H-0B05H.

3^η Άσκηση

```

    IN 10H
START:
    MVI A,10H
    STA 0903H
    STA 0904H
    STA 0905H
    LDA 2000H      ;read the number
    MOV B,A
    RLC
    JNC POSITIVE   ; check for the sign of the number

```

JC NEGATIVE

POSITIVE:

```
MVI A,10H      ;set blank the 3rd digit
STA 0902H
LXI D,0902H
MOV A,B
CPI 64H        ;check for the boundary
JP HIGH
JMP CONVERT
```

NEGATIVE:

```
MOV A,B
CMA            ;negative number
ADI 01H        ;2's complement
MOV B,A
MVI A,1CH
STA 0902H      ;the third digit of 7-Segment is (-)
LXI D,0900H
CALL STDM
CALL DCD
MOV A,B
CPI 64H
JP LOW
JMP CONVERT
```

CONVERT:

```
MVI C,00H      ;convert HEX->DEC
```

TENS:

```
SUI 0AH
```

```
JC UNITS
INR C          ;tens = tens +1;
JMP TENS      ;C = tens
```

UNITS:

```
ADI 0AH        ;A = units
STA 0900H      ;A goes to the first digit
MOV A,C
STA 0901H      ;C goes to the second digit
LXI D,0900H    ;display in 7-Segment
CALL STDMM
CALL DCD
JMP START
```

HIGH:

```
MVI A,19H      ;set (I) to the first digit
STA 0900H
MVI A,11H      ;set (H) to the second digit
STA 0901H
LXI D,0900H
CALL STDMM
CALL DCD
JMP START
```

LOW:

```
MVI A,00H      ; set (O) to the first digit
STA 0900H
MVI A,12H      ; set (L) to the second digit
STA 0901H
LXI D,0900H
CALL STDMM
```

CALL DCD

JMP START

END

Σε αυτή την άσκηση το πρόγραμμα απεικονίζει στα τρία δεξιότερα 7-segment σε δεκαδική μορφή δύο ψηφίων την τιμή που διαβάστηκε από τη θύρα εισόδου. Αν είναι μεταξύ +99 και -99 τότε εμφανίζει τον αριθμό με δύο ψηφία αν είναι θετικός και με τρία αν είναι αρνητικός (πρόσημο + δύο ψηφία). Αν $x > 99$ ή $x < -99$ εμφανίζει HI ή -LO αντίστοιχα. Αντί για το πρόσημο + έχουμε βάλει κενό.