

Εργαστήριο Μικροϋπολογιστών

8η εργαστηριακή άσκηση

Τμήμα: Β Ομάδα: 15

Συνεργάτες: Μαρουφίδης Ιωάννης (03113506),

Περράκης Γεώργιος (03113511) ,

Σοφιανίδης Γεώργιος (03113179)

1^η Άσκηση

```
.include "m16def.inc"

.def temp = r16
.def led1 = r17
.def led2 = r18
.def g1 = r19
.def g2 = r20
.def g3 = r21
.def g4 = r22
.def g5 = r23

start:
    ldi temp, 0x0f
    out DDRC, temp
    ;PORTC for output
    clr temp
    out DDRB, temp
    ;PORTB for input
    out DDRA, temp
    ;PORTA for input

    out PORTC, temp
    ;close all leds

gate1:
    in led1, PINB
    ;led1 = input
    mov led2, led1
    andi led1, 01
    ;mask to keep bit0
    andi led2, 02
    ;mask to keep bit1
    lsr led2
    ;right shift
    or led2, led1
    mov g1, led2

gate2:
```

```

in led1, PINB
;led1 = input
mov led2, led1
andi led1, 04
;mask to keep bit2
andi led2, 0x08
;mask to keep bit3
lsr led2

and led2, led1
lsr led2
;right shift to move result to bit1 place
mov g2, led2

```

gate3:

```

in led1, PINB
;led1 = input
mov led2, led1
andi led1, 16
;mask to keep bit4
andi led2, 32
;mask to keep bit5
lsr led2
eor led2, led1

lsr led2
lsr led2
;right shift to move result to bit2 place
mov g3, led2

```

gate4:

```

in led1, PINB
;led1 = input
mov led2, led1
andi led1, 64
;mask to keep bit6
andi led2, 128
;mask to keep bit7
lsr led2
eor led2, led1

lsr led2
lsr led2
lsr led2
;right shift to move result to bit3 place
mov g4, led2

```

gate5:

```

mov led1, g3
lsl led1
;left shift to move the result of gate3 to bit3 place
eor led2, led1
mov g5, led2

```

result:

```

clr temp
or temp, g1
or temp, g2
or temp, g3
or temp, g4

```

check:

```

mov led1, temp
;read PC buttons 0-7 and check if any PC buttons is pressed
in temp, PINA
andi temp,0x0f
eor led1, temp
out PORTC, led1
rjmp start

```

Στην παρούσα άσκηση υλοποιούμε μερικές λογικές πύλες, οι είσοδοι των οποίων βρίσκονται στα dip switches PB7-0 ενώ οι έξοδοι στα leds PC3-0. Πατώντας κάποιο από τα dip switches PA3-0 αντιστρέφεται η τιμή της αντίστοιχης εξόδου των συναρτήσεων.

2^η Άσκηση

```

#include <avr/io.h>
#include <stdio.h>

int main(void)
{
    unsigned char A,B,C,D,E,F0,F1,F2;
    DDRA = 0xFF;          /* PA00,PA01,PA02 as output */
    DDRC = 0x00;          /* PC0,PC1,PC2,PC3,PC4 as input */
    PORTA = 0x00;
    while (1)
    {
        A = PINC & 0x01;    /* we take each of the 5 LSBs of PIND by
applying masks */
        B = PINC & 0x02;
        B = B >> 1;
        C = PINC & 0x04;
        C = C >> 2;
        D = PINC & 0x08;
        D = D >> 3;
        E = PINC & 0x10;
        E = E >> 4;
        F0 = ~ ( (A&B) | (B&C) | (C&D) | (D&E) );
        F0 = F0 & 0x01;
        F1 = ((A & B & C & D) | ((~D) & (~E)) );
        F1 = F1 & 0x01;
        F2 = F0 | F1;
        F1 = F1 << 1;
        F2 = F2 << 2;
        F1= F1 | F2 | F0;
        PORTA = F1;
    }
    return 0;
}

```

Στην παρούσα άσκηση υλοποιούμε δύο απλές λογικές συναρτήσεις σε γλώσσα C. Χρησιμοποιούμε ολισθήσεις και μάσκες για να πάρουμε τα αποτελέσματα μας.

3^η Άσκηση

```
.include "m16def.inc"

.def temp1 = r17
.def temp2 = r16
.def leds = r18

start:
    ldi r24,LOW(RAMEND)
                                ; initialize stack
    out SPL,r24
    ldi r25,HIGH(RAMEND)
    out SPH,r25

    ldi r24 ,0xFF
                                ; PORTA as output
    out DDRA ,r24
    ldi r24 ,(1 << PC7) | (1 << PC6) | (1 << PC5) | (1 << PC4)    ; ??te?
    ?? e??d??? ta 4 MSB
    out DDRC ,r24
                                ; t?? ???a? PORTC

read_loop:
                                ; we wait the number of our team '15'
    call scan_keypad
    ldi temp1, 0x10
    and temp1,r25
    cpi temp1,0x10
                                ; we wait till number '1' is pressed
    brne read_loop
                                ; if 'no' we read again
    mov temp2,r25
                                ; if 'yes' we keep current state of buttons
    mov temp1,r24

iner_loop:
    call scan_keypad
    cp r24,temp1
                                ; if r24 is increased that means a new button is
pressed
    breq state1
                                ; that is not '5' so we jump to start
    brlo state1
    jmp read_loop

state1:
                                ; buttons on r24 have not changed
    mov temp1,r24
    cp r25,temp2
    breq state2
                                ; if r25 is increased that means a new button
is pressed
```

```

        brlo state2
        or r25,temp2
        andi r25,0x02
        cpi r25,0x02
        breq open_close_leds
        jmp read_loop
; if that button is '5' we found the combination!

; else we read again
state2:
        mov temp2,r25
; buttons on r25 have not changed
        jmp iner_loop

open_close_leds:
        ldi temp1,0x0A
; we turn on and off the leds 10 times
with delay
leds_loop:
        rcall on
        ldi r24 , low(250)
; 0.25 sec delay
        ldi r25 , high(250)
        rcall wait_msec
        rcall off
        ldi r24 , low(250)
; 0.25 sec delay
        ldi r25 , high(250)
        rcall wait_msec
        dec temp1
        cpi temp1,0x00
        breq read_loop
        jmp leds_loop

on:
        ser leds
;turn on leds
        out PORTA,leds
        ret

off:
;turn off leds
        clr leds
        out PORTA,leds
        ret

scan_row:
        ldi r25 ,0x08
; αρχικοποίηση με '0000 1000'
back_: lsl r25
; αριστερή ολίσθηση του '1' τόσες θέσεις
        dec r24
; όσος είναι ο αριθμός της γραμμής
        brne back_
        out PORTC ,r25
; η αντίστοιχη γραμμή τίθεται στο λογικό '1'
        nop

```

<code>nop</code>	<code>; καθυστέρηση για να προλάβει να γίνει</code>
<code>η αλλαγή κατάστασης</code>	
<code>in r24 ,PINC</code>	<code>; επιστρέφουν οι θέσεις (στήλες) των διακοπών που</code>
<code>είναι πιεσμένοι</code>	
<code>andi r24 ,0x0f</code>	<code>; απομονώνονται τα 4 LSB όπου τα '1' δείχνουν</code>
<code>που είναι πατημένοι</code>	
<code>ret</code>	<code>; οι διακόπτες.</code>
<code>scan_keypad:</code>	
<code>ldi r24 ,0x01</code>	<code>; έλεγξε την πρώτη γραμμή του πληκτρολογίου</code>
<code>rcall scan_row</code>	
<code>swap r24</code>	<code>; αποθήκευσε το αποτέλεσμα</code>
<code>mov r27 ,r24</code>	<code>; στα 4 msb του r27</code>
<code>ldi r24 ,0x02</code>	<code>; έλεγξε τη δεύτερη γραμμή του πληκτρολογίου</code>
<code>rcall scan_row</code>	
<code>add r27 ,r24</code>	<code>; αποθήκευσε το αποτέλεσμα στα 4 lsb του r27</code>
<code>ldi r24 ,0x03</code>	<code>; έλεγξε την τρίτη γραμμή του πληκτρολογίου</code>
<code>rcall scan_row</code>	
<code>swap r24</code>	<code>; αποθήκευσε το αποτέλεσμα</code>
<code>mov r26 ,r24</code>	<code>; στα 4 msb του r26</code>
<code>ldi r24 ,0x04</code>	<code>; έλεγξε την τέταρτη γραμμή του πληκτρολογίου</code>
<code>rcall scan_row</code>	
<code>add r26 ,r24</code>	<code>; αποθήκευσε το αποτέλεσμα στα 4 lsb του r26</code>
<code>movw r24 ,r26</code>	<code>; μετέφερε το αποτέλεσμα στους καταχωρητές r25:r24</code>
<code>ret</code>	
<code>wait_usec:</code>	
<code>sbiw r24 ,1</code>	<code>; 2 κύκλοι (0.250 msec)</code>
<code>nop</code>	<code>; 1 κύκλος (0.125 msec)</code>
<code>nop</code>	<code>; 1 κύκλος (0.125 msec)</code>
<code>nop</code>	<code>; 1 κύκλος (0.125 msec)</code>
<code>nop</code>	<code>; 1 κύκλος (0.125 msec)</code>
<code>nop</code>	<code>; 1 κύκλος (0.125 msec)</code>
<code>brne wait_usec</code>	<code>; 1 ή 2 κύκλοι (0.125 ή 0.250 msec)</code>
<code>ret</code>	<code>; 4 κύκλοι (0.500 msec)</code>
<code>wait_msec:</code>	
<code>push r24</code>	<code>; 2 κύκλοι (0.250 msec)</code>

```

push r25                                ; 2 κύκλοι
ldi r24 , low(998)                      ; φόρτωση του καταχ. r25:r24 με 998 (1 κύκλος -
0.125 μsec)
ldi r25 , high(998)                    ; 1 κύκλος (0.125 μsec)
rcall wait_usec                         ; 3 κύκλοι (0.375 μsec), προκαλεί συνολικά
καθυστερήση 998.375 μsec
pop r25                                ; 2 κύκλοι (0.250 μsec)
pop r24                                ; 2 κύκλοι
sbiw r24 , 1                            ; 2 κύκλοι
brne wait_msec                          ; 1 ή 2 κύκλοι (0.125 ή 0.250 μsec)
ret                                     ; 4 κύκλοι (0.500 μsec)

```

Στην παρούσα άσκηση υλοποιούμε μία ηλεκτρονική κλειδαριά με κωδικό τον αριθμό της ομάδας μας “15”. Για το σκοπό αυτό χρησιμοποιούμε τις έτοιμες συναρτήσεις για την επικοινωνία με το πληκτρολόγιο από όπου περιμένουμε τα δύο ψηφία με τη σειρά.