



**Εθνικό Μετσόβιο Πολυτεχνείο**  
**Σχολή: Ηλεκτρολόγων Μηχανικών & Μηχανικών**  
**Υπολογιστών**  
**Συστήματα Μικροϋπολογιστών (6<sup>ο</sup> εξάμηνο)**  
**4<sup>η</sup> Ομάδα Ασκήσεων**

Δημήτριος Καλαθάς - el18016  
Δημήτριος Καλέμης - el18152

## Ασκήσεις Προσομοίωσης

### Άσκηση 1

#### Κώδικας σε Assembly

```
.include "m16def.inc"
.def temp=r24
.def led=r20

stack:
    ldi r24, low(RAMEND)      ;initialize stack pointer
    out SPL, r24
    ldi r24, high(RAMEND)
    out SPH, r24

set_ports:
    ser temp                  ;initialize PORTA for output - puts 1s
    out DDRA,temp
    clr temp                  ;initialize PINB for input - puts 0s

    out DDRB,temp

main:
    ldi led,01                ;initialize the led that will open
    rcall left                ;go left
    nop
    rcall right               ;go right
    rjmp main                 ;go back to main

left:
    in temp,PINB              ;temp gets input
    andi temp,01              ;isolate PBO
    cpi temp,01               ;while PBO=1 loop here
    brcc left
    out PORTA,led             ;show current led
    cpi led,80                ;when it reaches left end
    brcc return               ;return
```

lsl led	;if it hasn't reached it yet, move led left
rjmp left	
right:	
in temp,PINB	;temp gets input
andi temp,01	;isolate PBO
cpi temp,01	;while PBO=1 loop here
brcc right	
out PORTA,led	;show current led
cpi led,01	;when it reaches right end
breq return	;return
lsl led	;if it hasn't reached it yet, move led right
rjmp right	
return:	
ret	

## Άσκηση 2

Κώδικας σε C	
#include <avr/io.h>	
unsigned char a, b, c, notc, d, f0, f1;	
int main(void)	
{	
DDRB=0xFF;	//αρχικοποίηση ως portB output
DDRA=0x00;	//αρχικοποίηση ως portA input
while (1)	
{	
a = PINA & 0x01;	//διαβάζει απο το portA το 1ο LSB
b = PINA & 0x02;	//διαβάζει απο το portA το 2ο LSB
b = b >> 1;	//ολισθηση δεξια κατα 1 θέση (το βάζει στο 1ο LSB)
c = PINA & 0x04;	//διαβάζει απο το portA το 3ο LSB
c = c >> 2;	//ολισθηση δεξια κατα 2 θέση (το βάζει στο 1ο LSB)
d = PINA & 0x08;	//διαβάζει απο το portA το 4ο LSB
d = d >> 3;	//ολισθηση δεξια κατα 3 θέση (το βάζει στο 1ο LSB)
f1 = (a b) & (c d);	//φτιαχνω την f1
f1 = f1 << 1;	//ολισθηση αριστερα μια θέση για να παει στο 2ο LSB
notc = c^0x01;	//σημπληρωμα c με XOR

```

        f0 = ((a & b & notc) | (c & d));
        f0 = f0^0x01;           //φτιαχνω την f1

        f0 = f0|f1;             //φτιαχνω την portB
        PORTB = f0;             //γρᾶφω στην portB

    }
    return 0;
}

```

### Άσκηση 3

#### Κώδικας σε C

```

#include <avr/io.h>

char x;

int main(void){
    DDRA = 0xFF;                // αρχικοποίηση PORTA ως output
    DDRC = 0x00;                // αρχικοποίηση PORTC ως input

    x = 0x01;                    // αρχικοποίηση μεταβλητής για αρχικά αναμμένο led
    PORTA = x;                   //εξοδος σε PORTA

    while(1){
        if((PINC & 0x01) == 1){ // ελεγχος πατήματος push-button SW0
            while((PINC & 0x01) == 1){ // ελεγχος επαναφοράς push-button
                if(x == 0x80){        //αν το αναμμένο led είναι στο MSB
                    x = 0x01;         //μετακίνησέ το στο LSB
                }
            }
            else{
                x = x<<1;             // ολίσθηση αριστερά
            }
        }
        if((PINC & 0x02) == 2){       // ελεγχος πατήματος push-button SW1
            while((PINC & 0x02) == 2){ // ελεγχος επαναφοράς push-button
                if(x == 0x01){        //αν το αναμμένο led είναι στο LSB
                    x = 0x80;         //μετακίνησέ το στο MSB
                }
            }
            else{
                x = x>>1;             // ολίσθηση δεξιά
            }
        }
        if((PINC & 0x04) == 4){       // ελεγχος πατήματος push-button SW2
            while((PINC & 0x04) == 4){ // ελεγχος επαναφοράς push-button
                x = 0x80;             //μετακίνηση αναμμένου led στην θέση MSB
            }
        }
        if((PINC & 0x08) == 8){       // ελεγχος πατήματος push-button SW3
            while((PINC & 0x08) == 8){ // ελεγχος επαναφοράς push-button
                x = 0x01;             //μετακίνηση αναμμένου led στην αρχική του θέση LSB
            }
        }

        PORTA = x;                   //εξοδος σε PORTA
    }
    return 0;
}

```