Εθνικό Μετσόβιο Πολυτεχνείο
Σχολή Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών

Συστήματα Μικροϋπολογιστών

Σειρά Ασκήσεων 6	Θανάσουλας Γρηγόριος | ΑΜ: 03114131 Μόνου Σταματίνα | ΑΜ: 03114077

Άσκηση 1

```
.def delay1=150
.def delay2=50
.def tmp=R16
.def counterON=R9
.def counterOFF=R10


clr tmp
out DDRD, tmp

ser tmp
out DDRB, tmp
clear


out PORTD, tmp
read:
in tmp, PIND
sbrs tmp, 7             ;Αν το 7ο bit ισούται με 1 κάνε skip την επόμενη εντολή
jmp ON_150_OFF_50       ; Αν το bit ελέγχου ισούται με 0
ON_50__OFF_150:         ; Εκτελείται αν bit = 1
        ldi counterOFF, 150
        ldi counterON, 50
        jmp setLEDS
ON_150_OFF_50:          ; Εκτελείται αν bit = 0
        ldi counterOFF, 50
        ldi counterON, 150
setLEDs:
        ser tmp
        out PORTB, tmp      ; Άναμμα LEDs
count_1:
        call Delay10    ; Καθυστέρηση 10 ms
        dec counterON
        brne count_1

        clr tmp
```

```
        out PORTB, tmp
count_2:
        call Delay10
        dec counterOFF
        brne count_2

        jmp read
```

Άσκηση 2

```
        .def tmp=R19
        .def input=r6
        .def result=r5 ;Για τα LED μας υποθέσαμε θετική λογική, δηλαδδή ανάβει όταν 1
                                ;_ _ _ _ _ 0 X1 X2


start:
        ;A έξοδος, C είσοδος
        clr tmp
        out DDRC, tmp
        ser tmp
        out DDRA, tmp
        out PORTC, tmp
        in tmp, PORTC

        ; Υπολογισμό X0

        ; AB
        mov input, tmp
        andi tmp, 0b00001100
        cpi tmp, 0b00001100
        breq ONE_X0

        ; CD'E'F
        mov tmp, input          ; Επαναφορά τιμής
        andi tmp, 0b11110000
        cpi tmp, 0b10010000
        brne CALC_X1            ; Αν το X0 = 0

ONE_XO:
        ldi result, 0x05; Αν είναι X0 = 1 θα είναι και η X2 = 1

        ; Υπολογισμός X1
        ; ABC'D
        mov tmp, input          ; Επαναφορά τιμής
        andi tmp, 0b00111100
        cpi tmp, 0b00101100
```

```
        breq ONE_X1

        ; D'EF'
        mov tmp, input                  ; Επαναφορά τιμής
        andi tmp, 0b11100000
        cpi tmp, 0b01000000
        brne set_leds           ; Αν το X1 = 0
        ori result, 0x03        ; Αν το X1=1 τότε και X2=1

set_leds:
        out PORTC, result
        jmp start
```

```
#include <avr/io.h>
unsigned char A, B, C, D, E, F, X0, X1, X2, temp;

int main (void) {
   DDRC = 0x00;
   PORTC = 0XFF;
   DDRA = 0XFF;
   A = PINC;
   B = PINC;
   C = PINC;
   D = PINC;
   E = PINC;
   F = PINC;
   A >>= 7;
   B >>= 6;
   C >>= 5;
   D >>= 4;
   E >>= 3;
   F >>= 2;

   X0 = (A & B) | (C & (~D) & (~E) & F);
   X1 = (A & B & (~C) & D) | ((~D) & E & (~F));
   X2 = X0 | X1;
   X2 &= 1;
   X1 &= 1;
   X0 &= 1;
   X2 <<= 2;
   X1 <<= 1;
   X2 = X2 | X1 | X0;
   PORTA = X2;
   while (1);
}
```

Άσκηση 3

```
        .def tmp=R19
        .def input=r6
        .def output=r5

        ;B έξοδος, D είσοδος
        clr tmp
        out DDRD, tmp

        ser tmp
        out DDRB, tmp
        out PORTB, tmp

        ldi output, 0x01                ; Άναμμα μόνο ενός LED
        out PORTB, output

REPEAT:
        in input, PORTD                         ; Διάβασμα εισόδου
        out tmp, input


SW_0:
        sbrs input, 0                   ; Skip αν είναι bit0 = 1
        jmp SW_1
        ldi output, 0x00001111          ; Άναμμα 4 LSB Leds
        out PORTB, output
        jmp REPEAT

SW_1:
        sbrs input, 1                   ; Skip αν είναι bit1 = 1
        jmp SW_2
        ldi output, 0x11110000          ; Άναμμα 4 MSB Leds
        out PORTB, output
        jmp REPEAT

SW_2:
        sbrs input, 2                   ; Skip αν είναι bit2 = 1
        jmp SW_3
        rol output                      ; Περιστροφή αριστερά
        out PORTB, output
        CALL Delay100
        jmp REPEAT

SW_3:
        sbrs input, 3                   ; Skip αν είναι bit3 = 1
        jmp SW_4
        ror output                              ; Περιστροφή δεξιά
        out PORTB, output
```

```
        CALL Delay100
        jmp REPEAT

SW_4:
        sbrs input, 4                   ; Skip αν είναι bit4 = 1
        jmp REPEAT
        ldi output, 0x01                ; Άναμμα MSB LED
        out PORTB, output
        jmp REPEAT
```

Άσκηση 4

```
.def Temp=r16
.def leds=r17
.def count=r18
.def Delay1=r19
.def Delay2=r20
.def Delay3=r21

        jmp reset
        jmp interrupt1
        reti
reset:
        ldi temp,high(RAMEND)
        out SPH,temp
        ldi temp,low(RAMEND)
        out SPL,temp

        ldi temp,0xFf
        out DDRB,temp
        ldi temp,1<<INT1            ;Ενεργοποίηση εξωτερικής διακοπής INT1
        out GIMSK,temp

        ldi temp,0b00000010         ; Ενεργοποίηση διακοπής στην ακμή καθόδου
        out MCUCR,temp
        sei                         ; Global interrupt
WAIT:
        rjmp WAIT                   ; Αναμονή για διακοπή

interrupt1:
        ldi temp,1<<INT1
        out GIMSK,temp
        sei                             ;Ενεργοποίηση διακοπών
        ser leds
        out PORTB,leds
        ldi temp,30
```

```
intloop:
        dec temp                        ; Total Delay = 30 x 1 sec
        call DELAY
        brne intloop
        clr leds
        out PORTB,leds
        reti

DELAY:                                  ; Delay 1 sec
        ldi Delay1, 6
        ldi Delay2, 19
        ldi Delay3, 174
d1:
        dec Delay1
        brne d1
        dec Delay2
        breq d1
        dec Delay3
        breq d1
        ret
```

```
#include <avr/io.h>
unsigned char counter, leds;

interrupt[ext_int1] void ext_into_isr(void) { // Διακοπή INT 1 Routine
   counter = 30;
   GIMSK = (1<<INT1);
   #ASM("SEI")
   leds = 0xFF;
   PORTA = leds;
   while (counter != 0) { // Υλοποίηση καθυστέρησης 30 x 1sec
      delay(1);
   }
   leds = 0x00;
   PORTA = leds;
}

int main(void) {
   DDRA = 0xFF;
   PORTA = 0x00;
   GIMSK = (1<<INT1);      //Ενεργοποίηση INT1
   MCUCR = 0x02;           // Διακοπή στην ακμή καθόδου
   #ASM("SEI")

   while (1);
}
```