Εργαστήριο Μικροϋπολογιστών

5η Εργαστηριακή άσκηση

Ομάδα Γ04

Συνεργάτες:

- **Σκούλος Ραφαήλ Α.Μ: 03112404**
- **Αναστάσης Σταθόπουλος Α.Μ: 03112140**
- **Τζίνης Ευθύμιος Α.Μ: 03112007**

Άσκηση 1

Ο κώδικας της άσκησης με σχόλια είναι ο εξής:

```
.include "m16def.inc"

main:
     ldi r24,LOW(RAMEND)      ;initialize stack pointer
     out SPL,r24
     ldi r24,HIGH(RAMEND)
     out SPH,r24
     ser r24                  ;r24 = 11111111b
     out DDRB,r24             ;initialize PORTB for
                              ;output
     clr r24                       ;r24 = 0
     out DDRA,r24             ;initialize PORTA for input
     ldi r26,1                ;start from the 'most'
```

```
                              ;right LED


left:                         ;LED moving left
     inr27,PINA               ;PINA is the register
                              ;we use to read the input
                              ;from PORTA
     out PORTB,r26            ;show the led in PORTB
     ldi r24 , low(500)       ;r25:r24 = 500
     ldi r25 , high(500)      ;delay 0,5sec
     rcallwait_msec
     andi r27,0x80
     cpi r27,128              ;check if PA7 is
                              ; pushed
     breq left                ;if it is don't move the
                              ;LED
     lsl r26                  ;logical shift left
     cpi r26,128              ;check if the LED
                              ;reached the (left) end
     breq right               ;if so start moving it to
                              ;right
     jmp left                 ;else keep moving left



right:                        ;LED moving left
     in r27,PINA
     out PORTB,r26            ;show the led in PORTB
     ldi r24,low(500)
     ldi r25,high(500)        ;delay 0,5sec
     rcallwait_msec
     andi r27,0x80
     cpi r27,128                ;check if PA7 is pushed
     breq right
     lsr r26                    ;logical shift right
     cpi r26,1                  ;check if the LED reached
                                ;the (right) end
     breq left                  ;if so start moving it to
                                ;left
     jmp right                  ;else keep moving right

wait_usec:                      ;delay r25:r24 µsec
     sbiw r24,1
     nop
     nop
     nop
     nop
     brnewait_usec
     ret

wait_msec:                      ;delay r25:r24 msec
     push r24
     push r25
     ldi r24,low(998)
     ldi r25,high(998)
     rcallwait_usec
     pop r25
     pop r24
     sbiw r24,1
     brnewait_msec
     ret
```

Άσκηση 2

Ο κώδικας της άσκησης με σχόλια είναι ο εξής:

```
.include "m16def.inc"


     ldi r24,low(RAMEND)        ;initialize stack pointer
     out SPL,r24
     ldi r24,high(RAMEND)
     out SPH,r24
     ser r26
     out DDRA,r26               ;use PORTA for output
     clr r26
     out DDRB,r26               ;use PORTB for input

flash:
     in r26,PINB               ;read from PORTB (for
                               ;light-on)
     andi r26,0b11110000       ;mask the B4-B7 bits - for
                               ;light-on
     lsr r26                   ;ta fernwsta B0-B3
     lsr r26
     lsr r26
     lsr r26
     ldi r27,2
     mul r26,r27              ;2*x
     inc r26                  ;2*x+1
     ldi r27,50
     mul r26,r27              ;(2x+1)*50
                              ;r1:r0 <- r26*r27
     mov r25,r1               ;store the result in
                              ;r25:r24
     mov r24,r0               ;the correct delay
     ser r26                  ;light-on all the LEDs
     out PORTA,r26
     rcallwait_msec      ;light-on delay

     in r26,PINB              ;read from PORTB (for
                              ;light-off)
     andi r26,0b00001111      ;mask the B0-B3bits - for
                              ;light-off
     ldi r27,2
     mul r26,r27              ;2*x
     inc r26                  ;2*x+1
     ldi r27,50               ;(2x+1)*50
     mul r26,r27
     mov r25,r1
     mov r24,r0
```

```
     clr r26                  ;light-off all the LEDs
     out PORTA,r26
     rcallwait_msec      ;light-off delay
     jmp flash                ;continious functionality


wait_usec:                    ;delay r25:r24 usec
sbiw r24 ,1
nop
nop
nop
nop
brnewait_usec
ret

wait_msec:                    ;delay r25:r24 msec
push r24
push r25
ldi r24 , low(998)
ldi r25 , high(998)
rcallwait_usec
pop r25
pop r24
sbiw r24,1
brnewait_msec
ret
```

Άσκηση 3

Ο κώδικας της άσκησης με σχόλια είναι ο εξής:

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


unsigned char ror(unsigned char num,unsigned char n)
/*function to shift a number n spots right with curry*/

{

     unsigned char i,temp,j;

     for(j=0;j<n;j++){

     temp=num;

     i=(temp & 0x01);

     num=num>>1;

     if (i==0x01) num=num+ 0x80;

      }

     returnnum;

}


unsigned char rol(unsigned char num,unsigned char n)

/*function to shift a number n spots left with curry*/

{

     unsigned char i,temp,j;

     for(j=0;j<n;j++){

     temp=num;

     i=(temp & 0x80);

     num=num<<1;

     if(i==0x80) num=num+ 0x01;
```

```
       }
     returnnum;
}




int main(void)
{
unsigned char i,temp,prev=0,in=0;
DDRB=0xFF;     //set PortD as input
DDRD=0x00;     //set PortB as output
PORTB=0x80;    //turn on led7
temp=0x80;     //initialize temp which shows the led
               //which is on
while(1)
{
     prev=in;  //previous input
     i=PIND;   //check input
     in=i;
     if ((i& 0x10)==0&&  (prev& 0x10)==0x10)
     /*check if SW4 is pushed*/
     {
          temp=0x80;
          PORTB=temp;
          /*move led to his initial position*/


     }
```

```
        else if ((i& 0x08)==0  &&  (prev& 0x08)==0x08)
             /*check if SW3 is pushed*/

        {

             temp=rol(temp,2);
             /*move led 2 spots left*/

             PORTB=temp;


        }

        else if ((i& 0x04)==0  && (prev& 0x04)==0x04)
             /*check if SW2 is pushed*/

        {

             temp=ror(temp,2);
             /*move led 2 spots right*/

             PORTB=temp;

        }

        else if ((i& 0x02)==0  &&  (prev& 0x02)==0x02)
             /*check if SW1 is pushed*/

        {

             temp=rol(temp,1);
             /*move led 1 spot left*/

             PORTB=temp;

        }

        else if ((i& 0x01)==0  &&  (prev& 0x01)==0x01)
         /*check if SW0 is pushed*/

         {

             temp=ror(temp,1);
             /*move led 1 spot right*/

              PORTB=temp;

         }

   }
```

```
return 1;


}
```