

# Εργαστήριο Μικροϋπολογιστών

2η Σειρά Ασκήσεων.

Ιάσονας Νικολάου (03116129) - Χαρίτων Χαριτωνίδης (03116694)

## Άσκηση 1

C:

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

int main(void)
{
    DDRB = 0xFF;
    DDRC = 0x00;

    while (1)
    {
        int A = PINC & 0x01;
        int B = (PINC & 0x02) >> 1;
        int C = (PINC & 0x04) >> 2;
        int D = (PINC & 0x08) >> 3;

        int F0 = (~(A&B) | (~B&C&D)) & 0x01;
        int F1 = (((A&C) & (B | D)) & 0x01) << 1;

        PORTB = F0 | F1;
    }
}
```

Assembly:

```
start:
    clr r26
    out DDRC,r26 ; input
    ser r26
    out DDRB, r26 ; output

loop:
```

```

in r16, PINC
andi r16,0x0F
mov r17, r16 ; r17 = A
mov r18, r16 ; r18 = B
lsr r18
mov r19, r18 ; r19 = C
lsr r19
mov r20, r19 ; r20 = D
lsr r20

mov r21, r17 ; r21 = A
com r21 ; r21 = A'
and r21, r18 ; r21 = A'B

mov r22, r18; r22 = B'CD
com r22
and r22, r19
and r22, r20

or r21, r22 ; r21 = (A'B + B'CD)
com r21 ; r21 = (A'B + B'CD)'

mov r23, r17 ; r23 = AC
and r23, r19
mov r24, r18 ; r24 = B+D
or r24, r20
mov r25, r23 ; r25 = AC(B+D)
and r25, r24

lsl r25
andi r25, 0x02
andi r21, 0x01

or r21, r25
mov r26, r21

out PORTB , r26
rjmp loop

```

## Άσκηση 2

start:

```

.org 0x0
rjmp init
.org 0x4

```

```

        rjmp ISR1
init:
    ser r26
    out DDRC , r26 ; counter output
    out DDRB, r26 ; interrupt counter output
    clr r26
    out DDRA, r26 ; input
    clr r26 ; counter = 0
    clr r16 ; interrupt counter = 0

reset:
    ldi r24 ,( 1 << ISC11) | ( 1 << ISC10)
    out MCUCR , r24 ; enable at positive edge
    ldi r24 ,( 1 << INT1) ; enable interrupt INT1
    out GICR , r24
    sei

loop:
    out PORTC , r26
    inc r26 ; increase counter
    rjmp loop

ISR1:
    in r17, PINA ; read PA7-PA6
    andi r17, 0xC0
    cpi r17, 0xC0
    brne end_if
    inc r16 ; increase interrupt counter
    out PORTB, r16
end_if:
    rjmp reset

```

## Άσκηση 3

```

#include <avr/interrupt.h>
#include <avr/io.h>
#include <util/delay.h>

```

```

int countBits(int num)
{
    int c = 0;
    while (num)
    {
        c += num & 1;
        num >>= 1;
    }
}

```

```

        return c;
    }

ISR (INT0_vect)
{
    int numOfSetBits = countBits(PINB);
    if (PINA & 2)
    {
        PORTC = numOfSetBits;
    }
    else
    {
        int out = 0;
        while (numOfSetBits--)
        {
            out = (out << 1) | 1;
        }
        PORTC = out;
    }
}

int main()
{
    DDRA = 0x00;
    DDRB = 0x00;
    DDRC= 0xff;

    GICR = 0x40;
    MCUCR = 0x03;

    sei();

    while(42)
    {

    }
}

```