Συστήματα Μικροϋπολογιστών – 4η Σειρά Ασκήσεων

Κυριακόπουλος Γιώργος – el18153
Τζελέπης Σεραφείμ – el18849

1η Άσκηση:

```
.include "m16def.inc"

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

IO_set: ser r24          ; initialize PORTA
        out DDRA, r24    ; for output
        clr r24          ; initialize PORTB
        out DDRB, r24    ; for input

main:   ldi r26, 01      ; initialize output
        ldi r22, 07      ; initialize counter
        out PORTA,r26

left:   in r24, PINB     ; check input
        andi r24, 01     ; repeat till it's not 1
        cpi r24, 01
        breq left
        lsl r26          ; shift output 1 left
        out PORTA, r26   ; send it to output
        dec r22          ; decrease the counter
        cpi r22, 00      ; check if it is 0
        brne left        ; if it is then don't loop

right:  in r24, PINB     ; check input
        andi r24, 01     ; repeat till it's not 1
        cpi r24, 01
        breq right
        lsr r26          ; shift output 1 right
        out PORTA, r26   ; send it to output
```

```
inc r22          ; increase the counter
cpi r22, 07      ; check if it is 7
brne right       ; if it is then don't loop
rjmp left
```

2η Άσκηση:

Assembly version

```
.include "m16def.inc"


.DEF A = r16
.DEF B = r17
.DEF C = r18
.DEF D = r19
.DEF F0 = r20
.DEF F1 = r21
.DEF T = r22

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

IO_set: ser r24         ; initialize PORTB
        out DDRB, r24   ; for output
        clr r24         ; initialize PORTA
        out DDRA, r24   ; for input

main:   in T, PINA      ; T <- input
        andi T, 0x0F    ; keep only 4 LSBs

        mov A, T        ; LSB(A) = A
        lsr T
        mov B, T        ; LSB(B) = B
        lsr T
        mov C, T        ; LSB(C) = C
        lsr T
        mov D, T        ; LSB(D) = D

        mov T, B        ; save B in T
        or T, A         ; T = A + B
        mov F1, C       ; F1 = C
        or F1, D        ; F1 = C + D
        and F1, T       ; LSB(F1) = (A + B) & (C + D)

        mov T, C        ; T = C
```

```
com T           ; T = C'
and T, B        ; T = BC'
and T, A        ; T = ABC'
mov F0, C       ; F0 = C
and F0, D       ; F0 = CD
or F0, T        ; F0 = ABC' + CD
com F0          ; LSB(F0) = (ABC' + CD)'

lsl F1          ; F1 is moved to 2nd LSB
andi F1, 0x02   ; keep only 2nd LSB
andi F0, 0x01   ; keep only 1st LSB
mov T, F1       ; T = 000000(F1)0
or T, F0        ; T = 000000(F1)(F0)
out PORTB, T

rjmp main
```

C version

```c
#include <avr/io.h>
unsigned char A, B, C, D, notC, temp, F0, F1, ans;

int main(void) {
    DDRA = 0x00;            // input
    DDRB = 0xFF;            // output

    while (1) {
        A = PINA & 0x01;    // keep lsb only
        B = PINA & 0x02;    // keep 2nd lsb
        B = B >> 1;         // and shift it to lsb
        C = PINA & 0x04;    // keep 3rd lsb
        C = C >> 2;         // and shift it to lsb
        D = PINA & 0x08;    // keep 4th lsb
        D = D >> 3;         // and shift it to lsb

        notC = C ^ 0x01;    // C complement
        temp = ((A & B & notC) | (C & D));

        F0 = temp ^ 0x01;
        F1 = (A | B) & (C | D);

        F1 = F1 << 1;       // shift F1 to 2nd lsb
        ans = F0 + F1;      // add F1 and F0 for output
        PORTB = ans;        // send it to PORTB
    }
}
```

3η Άσκηση:

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

unsigned char x = 0x01;              // default value for output

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

    PORTA = x;                       // send lsb 1 to output
    while (1) {
        if(PINC == 0x01) {           // if SW0 is pressed
            while(PINC == 0x01) {}   // loop until it turns off
            if(x == 0x80) {          // if the led is on the leftmost
                x = 0x01;            // position, send it to lsb
            }
            else {
                x = x << 1;          // else shift output left once
            }
        }
        else if(PINC == 0x02) {      // if SW1 is pressed
            while(PINC == 0x02) {}   // loop until it turns off
            if(x == 0x01) {          // if the led is on the rightmost
                x = 0x80;            // position, send it to msb
            }
            else {
                x = x >> 1;          // else shift output right once
            }
        }
        else if(PINC == 0x04) {      // if SW2 is pressed
            while(PINC == 0x04) {}   // loop until it turns off
            x = 0x80;                // send led to msb
        }
        else if(PINC == 0x08) {      // if SW3 is pressed
            while(PINC == 0x08) {}   // loop until it turns off
            x = 0x01;                // send led to lsb
        }
        PORTA = x;                   // send led to output
    }
}
```