ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

4η Εργαστηριακή Άσκηση

Ακαδημαϊκό έτος 2020-2021

Χριστίνα Προεστάκη | ΑΜ : 03118877

Νικόλαος Μπέλλος | ΑΜ : 03118183

Αναστάσης Αγγλογάλλος | ΑΜ : 03118641

1η Άσκηση

Κώδικας ASSEMBLY για AVR

```
	.include "m16def.inc"

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

	clr   r24
	out DDRB, r24		;PORT B - input
	ser r24
	out DDRA, r24		;PORTA - output

	clr r24			;count LEDs
	ldi   r25, 0x01

GO_LEFT:
	in r26, PINB	;input in r26
	andi r26, 0x01	;keep PB0
	cpi   r26, 0x01	;check if PB0 is 1
	breq GO_LEFT	;if it is, stop
	out   PORTA, r25	;turn on LSB
	inc   r24		;r24++
	lsl r25		;left rotation (r25 == 0x02 - first rotation)
	cpi r24, 7		;check if r24 == 7
	breq GO_RIGHT	;if so, MSB is turned on - rotate and go right
	rjmp GO_LEFT	;if not, keep going left

GO_RIGHT:
	in r26, PINB
	andi r26, 0x01
	cpi r26, 0x01
```

```
        breq GO_RIGHT
        out PORTA, r25
        dec r24             ;r24 --
        lsr r25             ;right rotation
        cpi r24, 0          ;check if r24 == 0
        breq GO_LEFT        ;if so, LSB is turned on - rotate and go left
        rjmp GO_RIGHT       ;if not, keep going right
```

2η Άσκηση

Κώδικας ASSEMBLY για AVR

```
.include "m16def.inc"
.DEF A = r16        ;input A - PORTA LSB(0)
.DEF B = r17        ;input B - PORTA LSB(1)
.DEF C = r18        ;input C - PORTA LSB(2)
.DEF D = r19        ;input D - PORTA LSB(3)
.DEF I = r20        ;register to store input
.DEF E = r21        ;register for temporary calculations
.DEF F = r22        ;register for temporary calculations

;========== INITIALIZE STACK POINTER , I/O ==========
reset:
            ldi r24 , low(RAMEND)         ;initialize stack pointer (LOW)
            out SPL , r24
            ldi r24 , high(RAMEND)        ;initialize stack pointer (HIGH)
            out SPH , r24
            ser r26
            out DDRB, r26                 ;initialize PORTB
            clr r26
            out DDRA, r26                 ;initialize PORTA

;========== DATA INPUT ==========
main:
            clr E               ; CLEAR E
            clr F               ; CLEAR F
            in I, PORTA         ; I <- INPUT

            mov A, I            ; LSB(0) = A
            lsr I               ; rotate right INPUT
            mov B, I            ; LSB(1) = B
            lsr I               ; rotate right INPUT

            mov C, I            ; LSB(2) = C
            lsr I               ; rotate right INPUT
```

```
            mov D, I          ; LSB(3) = D

;========== ROUTINE FOR F0 =============
            mov F, C          ; F = C
            com F             ; F = C'
            and F, A          ; F = AC'
            and F, B          ; F = ABC'
            mov E, C          ; E = C
            and E, D          ; E = CD
            or  F, E          ; F = (ABC' + CD)
            com F             ; F = (ABC' + CD)' -> F0

;========== ROUTINE FOR F1 =============
            or A, B                 ; A = A + B
            or C, D                 ; C = C + D
            and A, C          ; A = (A + B)(C + D) -> F1
            lsl A             ; rotate left A

            andi F, 1         ; F = F0 (0000 0001 MASK)
            andi A, 2         ; A = F1 (0000 0010 MASK)
            or F, A                 ; COMBINE F0, F1
            out PORTB, F      ; OUTPUT F0-F1 IN PORTB

            rjmp reset
```

3η Άσκηση

Κώδικας C για AVR

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

char x = 1;

int main(void)
{
      DDRC = 0x00;  // Αρχικοποίηση PORTC ως input
      DDRA = 0xFF; // Αρχικοποίηση PORTA ως output
      PORTA = x;  //  Θέτουμε output το x

    while (1)
    {

      if(PINC == 1){              // Push Button SW0
```

```
            if  (x==128) x=1;
            else x=x<<1;        // Ολίσθηση-περιστροφή του led αριστερά
      }
      else if(PINC == 2){       //  Push Button SW1
            if (x==1) x=128;
            else x=x>>1;      // Ολίσθηση-περιστροφή του led δεξιά
      }
      else if(PINC == 4)     // Push Button SW2
            x=128;                                  // output -> MSB

      else if(PINC == 8)    // Push Button SW3
            x=1;                            // output -> 1st LSB

      while(PINC!=0);      // Waiting to apply changes after release of
button
      PORTA = x;          // Show output

    }
}
```