Εργαστήριο Μικροϋπολογιστών
Θοδωρής Παπαρρηγόπουλος
el18040
Ομάδα 21

2ο Εργαστήριο στον AVR

Άσκηση 1

Σε C

```c
/*
 * ask1_c.c
 *
 * Created: 11/2/2021 6:23:46 PM
 * Author : thodpap
 */
/*
 * ask1_c.c
 *
 * Created: 11/1/2021 8:20:10 PM
 * Author : thodpap
 */

#include <avr/io.h>

char A,B,C,D;
char F0;
char F1;

int main(void)
{
       DDRC = 0x00; // C is input
       DDRB = 0xff; // B is output


    /* Replace with your application code */
    while (1)
    {
              char input = PINC;
              A = (input & 0x01);
              B = (input & 0x02) >> 1;
              C = (input & 0x04) >> 2;
              D = (input & 0x08) >> 3;

              F0 = !( ((!A) & B) | ((!B) & C & D) );
              F1 = (A & C) | (B | D);
              F1 = F1 << 1;
              PORTB = F1 | F0;

    }
}
```

Σε assembly

```
;
; ask1_s.asm
;
; Created: 11/1/2021 8:31:08 PM
; Author : thodpap
;

;.include "m16def.inc"
; Replace with your application code
ser r24
out DDRB, r24 ; B is output
clr r24
out DDRC, r24 ; C is input

start:
	in r25, PINC ; xxxxDCBA, input
	mov r26, r25
	andi r26, 0x01 ; input & 0x01 -> A

	lsr r25 ; input is xxxxxDCB

	mov r24, r25 ; B
	andi r24, 0x01 ;

	lsr r25 ; input is xxxx xxDC

	mov r23, r25 ; C
	andi r23, 0x01;

	lsr r25 ; xxxx xxxD

	mov r22, r25 ; D
	andi r22, 0x01

F0:
	mov r16, r26 ; A
	mov r17, r24 ; B
	mov r18, r23 ; C
	mov r19, r22 ; D

	com r16; A'
	andi r16, 0x01

	and r16, r17 ; A' B

	com r17 ; B'
	andi r17, 0x01
	and r17, r18 ; B'C
	and r17, r19 ; B'CD

	or r16, r17 ; F0 = A'B | B'CD
	com r16  ; F0 = F0'
	andi r16, 0x01


F1:
	and r26, r23 ; AC
	or r24, r22 ; B | D
```

```
        or r26, r24 ; Result of F1

        lsl r26

        or r16, r26 ; results of F1F0

out PORTB, r16 ; result port B
rjmp start
        ;;lsr
```

Άσκηση 2

```
.org 0x0
		rjmp reset
.org 0x4
		rjmp ISR1

reset:
	ldi r24, (1<<ISC11)|(1<<ISC10)
	out MCUCR, r24
	ldi r24, (1<<INT1)
	out GICR, r24
	sei
	clr r31		; initialize interrupt

	ldi r24 , low(RAMEND) ; initialize stack
	out spl , r24
	ldi r24 , high(RAMEND)
	out sph , r24


	ser r26
	out DDRC, r26 ; count up to 255
	out DDRB, r26 ; count interrupt
	clr r26
	out DDRA, r26 ; Read from PA
	out DDRD, r26 ; read pind

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


ISR1:
	inc r31		; push stack
	push r26		; INT1 service routine
	in r26, SREG
	push r26

	in r30, PINA  ; get input
	andi r30, 0xc0		; is PA7A6  11 ?
	cpi r30, 0xc0 ;
	brne skip		;
	mov r26, r31  ;
	rjmp getout
skip:
	clr r26
getout:
	out    PORTB, r26

	pop r26 ;; restore stack
	out SREG, r26
	pop r26
	reti
```

Άσκηση 3

```
/*
 * ask3.c
 *
 * Created: 11/2/2021 10:18:14 PM
 * Author : thodpap
 */

#define F_CPU 8000000UL // set clock speed -- not needed here but added it nevertheless
#include <avr/io.h>
#include <avr/interrupt.h>

ISR(INT0_vect)
{
	// if input 1000 0001
	// PA2 ON ->  PORTC : 0000 0010 (2 light bulbs)
	// PA2 OFF -> PORTC : 0000 0011 (2 is 11 in binary)
	// Hence we just need to count ones

	/*  i = 1, 2, 4, 8, 16, 32, 64, 128
	    j = 0, 1, 2, 3, 4,  5,  6,  7

		input & i save the j-th digit 0..010...0 >> j -> 1
                                                                0..000...0 >> j -> 0

		if 0 then count += 0
		if 1 then count += 1
	*/
	char A = PINA;
	char input = PINB;

	int count = 0;
	for(int i = 1, j = 0; i < (1 << 8); i *= 2, ++j) {
		count += (input & i) >> j;
	}
	if (A & 0x04) { // if PA2 is ON transform to binary
		int t = 0;
		for(int i = 0; i < count; ++i) {
			t *= 2; // 000000 -> 000001 -> 0000010 + 1-> 0000011 -> 0000110 + 1 0000111
			t += 1;
		}
		count = t;
	} // else if PA2 is OFF just keep the counter as it is
	PORTC = count;
}
int main(void)
{
	DDRA = 0x00; // A input
	DDRB = 0x00; // B input
	DDRC = 0xff; //     C output

	GICR = 1<<INT0;            /* Enable INT0*/
	MCUCR = (1<<ISC01) | (1<<ISC00);  /* Trigger INT0 on rising edge */
	asm("sei");                /* Enable Global Interrupt */

    while (1) {
		asm("nop");
	}
}
```