

ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ ΣΧΟΛΗ ΗΛΕΚΤΡΟΛΟΓΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ VLSI

Εργαστηριακή Αναφορά

στο μάθημα "Συστήματα Μικροϋπολογιστών" 6ου Εξαμήνου

Ομάδα φοιτητών:

Νικόλαος Παγώνας	el18175
Αναστάσιος Παπαζαφειρόπουλος	el18079
Νικήτας Τσίννας	el18187

Ομάδα εξέτασης: ΜΥ_ 17

Ημερομηνία εξέτασης: 20/5/2021

Ακολουθεί ο κώδικας για κάθε ζήτημα αντίστοιχα μαζί με τα απαραίτητα σχόλια:

Ζήτημα 4.1

```
.include "m16def.inc"
start:
      ldi r24, low(RAMEND) ; initialize stack pointer
      out SPL, r24
      ldi r24, high(RAMEND)
      out SPH, r24
      ser r24
                                    ; r24 = 1111 1111
      out DDRA, r24
                                    ; Initialize PORTA for output
      clr r27
                                     ; r27 = 0000 0000
      out DDRB, r27
                                     ; Initialize PORTB for input
      ldi r28, 0b00000001
                                    ; r28 holds the output
                                     ; Print initial r28 = 0000 0001
      out PORTA, r28
left:
      in r26, PORTB
                                     ; Read input from PORTB
                                     ; Extract LSB
      andi r26, 0b00000001
      cpi r26, 0b00000001
      brne left
                                     ; Wait in this loop until PBO becomes 1
      1s1 r28
                                    ; Shift output one bit to the left
      out PORTA, r28
cpi r28, 0b10000000
                                     ; Output to leds
                                    ; If you have reached the end,
                                    ; start going to the right,
      breq right
      jmp left
                                     ; else continue going to the left
right:
                                     ; Read input from PORTB
      in r26, PORTB
      andi r26, 0b00000001
                                     ; Extract LSB
      cpi r26, 0b00000001
      brne right
                                     ; Wait in this loop until PB0 becomes 1
      1sr r28
                                    ; Shift output one bit to the right
      out PORTA, r28
                                    ; Output to leds
      cpi r28, 0b00000001
                                    ; If you have reached the beginning,
      breq left
                                     ; start going to the left,
      jmp right
                                     ; else continue going to the right
```

Ζήτημα 4.2

Έγινε σύνταξη του προγράμματος στην γλώσσα C:

```
#include <avr/io.h>
char A, B, C, D, F0, F1, OUT;
int main (void)
{
      DDRA = 0x00;
                        // initialize PINA as input
      DDRB = 0xFF; // initialize PORTB as output
      while (1)
      {
            A = PINA & 0x01;
                              // LSB (1st bit)
            B = PINA & 0x02; // 2nd bit
            C = PINA & 0x04; // 3rd bit
            D = PINA & 0x08; // 4th bit
            B = B \gg 1;
                               // shifting right
            C = C >> 2;
                               // shifting right
            D = D \gg 3;
                                // shifting right, all bits are in the 1st place
                                // (LSB)
            F0 = \sim ((A \& B \& (\sim C)) | (C \& D)); // expression 1
            F0 = F0 & 0x01;
            F1 = ((A|B) & (C|D)); // expression 2
            F1 = F1 << 1;
                                      // shift left, F1 is in the 2nd bit place,
                                      // where 1st place = LSB place
            OUT = F0 | F1;
            PORTB = OUT;
      }
      return 0;
}
```

Ζήτημα 4.3

```
#include <avr/io.h>
char x;
int main(void)
      DDRA = 0b11111111; // Αρχικοποίηση του PORTA ως output
      DDRC = 0b00000000; // Apxικοποίηση του PORTC ως input
      x = 0b00000001;
                         // Αρχικοποίηση μεταβλητής για αρχικά αναμμένο LED
      PORTA = x;
      while(1)
      {
             if ((PINC & 0b0001) == 0b0001) { // Έλεγχος πατήματος push-button SW0
                    while ((PINC & 0b0001) == 0b0001); // Έλεγχος επαναφοράς
                                                        // push-button SW0
                    // SW0: Ολίσθηση-περιστροφή του 1ed μία θέση αριστερά (κυκλικά)
                    if (x == 0b10000000)
                          x = 0b00000001;
                    else
                          x = x << 1;
             }
             if ((PINC & 0b0010) == 0b0010) { // Έλεγχος πατήματος push-button SW1
                    while ((PINC & 0b0010) == 0b0010); // Έλεγχος επαναφοράς
                                                       // push-button SW1
                    // SW1: Ολίσθηση-περιστροφή του led μία θέση δεξιά (κυκλικά)
                    if (x == 0b00000001)
                          x = 0b10000000;
                    else
                          x = x \gg 1;
             }
             if ((PINC & 0b0100) == 0b0100) { // Έλεγχος πατήματος push-button SW2
```