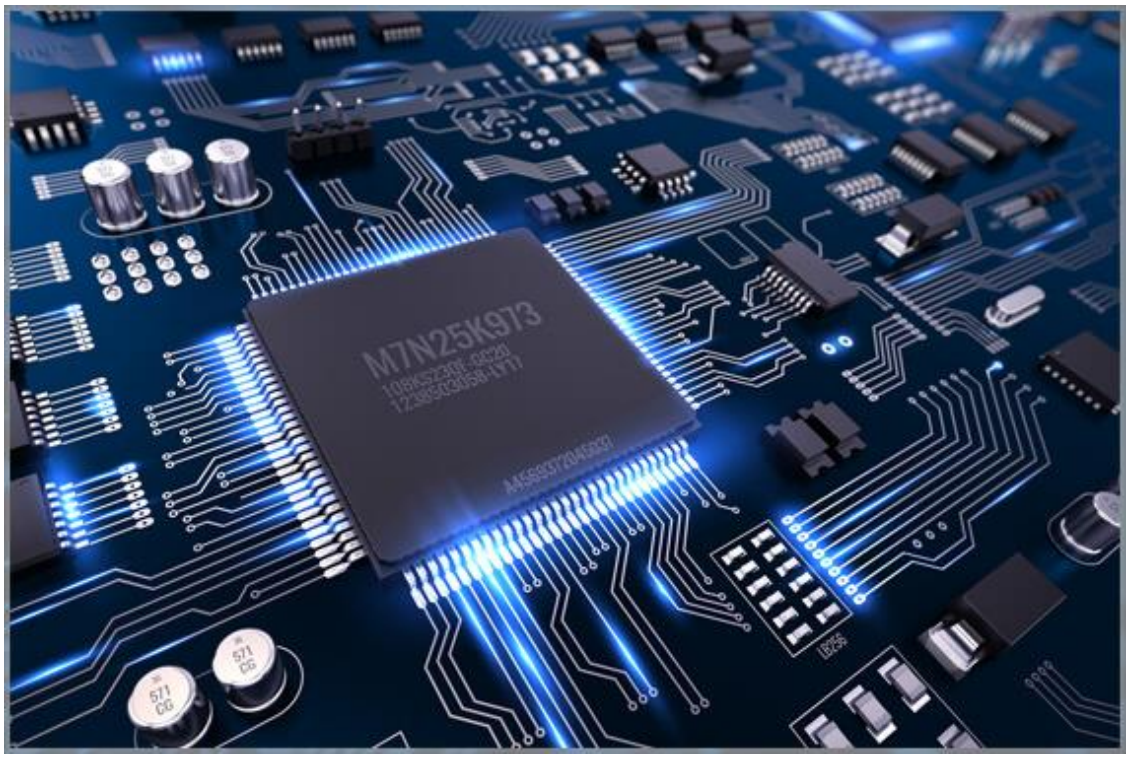
ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

4Η ΣΕΙΡΑ ΑΣΚΗΣΕΩΝ



ΜΑΥ 22, 2021

ΘΟΔΩΡΗΣ ΑΡΑΠΗΣ – EL18028
ΚΡΙΣ ΚΟΥΤΣΗ – EL18905
ΑΡΙΑΔΝΗ ΚΑΖΔΑΓΛΗ – EL18838

ΣΥΣΤΗΜΑΤΑ ΜΙΚΡΟΥΠΟΛΟΓΙΣΤΩΝ

Άσκηση 1

```
.include "m16def.inc"
.DEF temp = r16
.DEF leds = r17

reset:
	ldi r24, low(RAMEND) ;αρχικοποίηση Stack Pointer
	out SPL, r24
	ldi r24, high(RAMEND)
	out SPH, r24
start:
	ldi leds,0x01
	clr temp
	out DDRB,temp		;θύρα B ως είσοδος
	ser temp
	out PORTB,temp		;pull-up θύρας B
	out DDRA,temp		;θύρα A ως έξοδος
left:
	out PORTA,leds		;άναψε το LED
	in temp,PINB		;έλεγξε αν πατήθηκε το PB0
	sbrc temp,0		;αν όχι τότε κάνε skip
	rcall wait		;αν ναι τότε περίμενε
	lsl leds		;μετακίνησε το LED μία θέση αριστερά
	sbrs leds,7		;αν έφτασε το LED στο MSB κάνε skip
	rjmp left
right:
	out PORTA,leds		;άναψε το LED
	in temp,PINB		;έλεγξε αν πατήθηκε το PB0
	sbrc temp,0		;αν όχι τότε κάνε skip
	rcall wait		;αν ναι τότε περίμενε
	lsr leds		;μετακίνησε το LED μία θέση δεξιά
	sbrs leds,0		;αν έφτασε το LED στο LSB κάνε skip και πήγαινε στο
left
	rjmp right
	rjmp left
wait:
	in temp,PINB		;περίμενε μέχρι να γίνει το PB0 = 0
	andi temp,1
	cpi temp,0x01
	breq wait

	ret
```

Άσκηση 2

```
.include "m16def.inc"
.DEF temp = r16
.DEF A = r17
.DEF B = r18
.DEF C = r19
.DEF D = r20
.DEF F0= r21
.DEF F1= r22

reset:
       ldi r24,low(RAMEND)  ;αρχικοποίηση Stack Pointer
       out SPL,r24
       ldi r24,high(RAMEND)
       out SPH,r24
start:
       clr temp
       out DDRA,temp        ;θύρα A ως είσοδος
       ser temp
       out PORTA,temp       ;pull-up θύρας A
       out DDRB,temp        ;θύρα B ως έξοδος
main:
       clr F0               ;clear F0
       in temp,PINA         ;ανάγνωση ακροδεκτών PORTA
       mov A,temp           ;το A στο LSB του καταχωρητή A
       lsr temp
       mov B,temp           ;το B στο LSB του καταχωρητή B
       lsr temp
       mov C,temp           ;το C στο LSB του καταχωρητή C
       lsr temp
       mov D,temp           ;το D στο LSB του καταχωρητή D
       mov temp,C           ;προσωρινή αποθήκευση του C
       mov F0,C
       com F0               ;συμπλήρωμα C
       and F0,B             ;LSB(F0) = BC'
       and F0,A             ;LSB(F0) = ABC'
       and C,D              ;LSB(C) = CD
       or  F0,C             ;LSB(F0) = (ABC' + CD)
       com F0               ;LSB(F0) = (ABC' + CD)'

       or A,B               ;LSB(A) = A + B
       or temp,D            ;LSB(temp) = C + D
       and A,temp           ;LSB(A) = (A+B)·(C+D)
       lsl A                ;2o LSB(A) = (A+B)·(C+D)
       mov F1,A             ;2o LSB(F1) = (A+B)·(C+D)

       andi F0, 0x01        ;απομόνωση του LSB
       andi F1, 0x02        ;απομόνωση του 2ου LSB
       or   F0,F1
       out PORTB,F0


       rjmp main
```

Άσκηση 3

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

char x;

int main()
{

	DDRA=0xFF;		// Αρχικοποίηση PORTA ως output
	DDRC=0x00;		// Αρχικοποίηση PORTC ως input

	x = 1;			// Αρχικοποίηση μεταβλητής για αρχικά αναμμένο led

	while(1){
		if ((PINC & 0x01) == 0x01) { // έλεγχος για SW0
			if (x == 0x80)		// Έλεγχος υπεχείλισης
				x = 0x01;
			else
				x = x << 1;	// Ολίσθηση αριστερά
		}
		if ((PINC & 0x02) == 0x02) { // έλεγχος για SW1
			if (x == 0x01)		// Έλεγχος υπεχείλισης
				x = 0x80;
			else
				x = x >> 1;	// Ολίσθηση δεξιά
		}
		if ((PINC & 0x04) == 0x04) { // έλεγχος για SW2
			x = 0x80;
		}
		if ((PINC & 0x08) == 0x08) { // έλεγχος για SW3
			x = 0x01;
		}
		PORTA = x;
	}

	return 0;
}
```