

ΑΝΑΦΟΡΑ ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ
7-6-2020 ΣΗΜΜΥ
ΑΪΒΑΛΗΣ ΘΕΟΔΩΡΟΣ 03117099
ΚΟΡΟΥΛΗΣ ΧΡΗΣΤΟΣ 03117159
ΣΑΛΙΑΡΑΚΗΣ ΠΑΥΛΟΣ 03117135

ΑΣΚΗΣΗ 1

Θέλουμε να προγραμματίσετε σε assembly πρόγραμμα που να απεικονίζει ένα αναμμένο led που να αντιστοιχεί στα bit της θύρας PB0-PB7. Το αναμμένο led να κινείται συνεχώς ξεκινώντας από τα LSB προς τα MSB και αντίστροφα όταν φτάνει στο άλλο άκρο. Η κίνηση του αναμμένου led να ελέγχεται από το push button PC2. Όταν αυτό είναι πατημένο η κίνηση να σταματάει, ενώ διαφορετικά να συνεχίζεται.

```
.INCLUDE "m16def.inc"

.def counter=r20 ; ΜΕΤΡΗΤΗΣ ΓΙΑ ΝΑ ΕΛΕΓΧΩ ΣΕ ΠΟΙΟ ΨΗΦΙΟ
BΡΙΣΚΟΜΑΙ
.def temp=r21
.def lightBit=r22
.def inputC=r23
.org 0

reset:
    ldi r24,low(RAMEND) ; ΑΡΧΙΚΟΠΟΙΗΣΗ RAMEND=ΤΕΛΕΥΤΑΙΑ ΘΕΣΗ SRAM
    out SPL,r24 ; ΔΕΙΚΤΗ ΣΤΟΙΒΑΣ
    ldi r24,high(RAMEND)
    out SPH,r24
    clr lightBit

main:
    clr temp ; ΚΑΝΩ reset TO temp(=00000000)
    out DDRC,temp ; ΟΡΙΖΩ ΤΗΝ portC ΓΙΑ ΕΙΣΟΔΟ
    ser temp ; ΚΑΝΩ set TO temp(=11111111)
    out DDRB,temp ; ΟΡΙΖΩ ΤΗΝ portB ΓΙΑ ΕΞΟΔΟ

    ldi lightBit,0x01 ; ΟΡΙΖΩ TO lightbit=1
    out PORTB,lightBit ; ΕΜΦΑΝΙΖΩ TO 01 ΣΤΟΝ Β(flash the LSB)
    ldi counter,7 ; ΟΡΙΖΩ ΤΟΝ ΜΕΤΡΗΤΗ ΣΤΟ 7

left:
    in inputC,PINC ; ΔΙΑΒΑΖΩ ΤΗΝ ΕΙΣΟΔΟ ΑΠΟ ΤΗΝ ΠΥΛΗ C
    subi inputC,4 ; ΑΦΑΙΡΩ 4 ΑΠΟ ΤΗΝ ΕΙΣΟΔΟ
    breq left ; ΕΛΕΓΧΩ ΑΝ Η ΕΙΣΟΔΟΣ ΕΙΝΑΙ 4(ΑΝ ΕΙΝΑΙ
    ; ΠΑΤΗΜΕΝΟ ΔΕΝ ΚΑΝΩ ΤΙΠΟΤΑ
    ; ΚΑΙ ΠΕΡΙΜΕΝΩ ΝΑ ΣΒΗΣΕΙ)
    ; ΑΝ ΔΕΝ ΕΙΝΑΙ ΠΡΟΧΩΡΑΩ ΣΤΑ ΕΠΟΜΕΝΑ
    clr ; ΚΑΝΩ clear ΤΟΝ ΚΑΤΑΧΩΡΗΤΗ ΚΡΑΤΟΥΜΕΝΟΥ
    rol lightBit ; ΟΛΙΣΘΗΣΗ ΤΩΝ LEDS ΜΙΑ ΘΕΣΗ ΑΡΙΣΤΕΡΑ
    out PORTB,lightBit ; ΑΝΑΒΩ ΤΟ ΕΠΟΜΕΝΟ LED(ΑΜΕΣΩΣ ΑΡΙΣΤΕΡΑ
    ; ΚΑΘΕ ΦΟΡΑ)
    dec counter ; ΜΕΙΩΝΩ ΤΟΝ ΜΕΤΡΗΤΗ
    cpi counter,0 ; ΑΝ 0 counter == 0 ΕΙΜΑΙ ΣΤΟ MSB ΚΑΙ ΤΟΤΕ
```

```

    breq right          ; ΞΕΚΙΝΑΩ ΚΑΙ ΠΑΩ ΠΡΟΣ ΤΑ ΔΕΞΙΑ
    rjmp left           ; ΑΛΛΙΩΣ ΣΥΝΕΧΙΖΩ ΚΑΙ ΠΑΩ ΠΑΛΙ ΠΡΟΣ ΤΑ ΑΡΙΣΤΕΡΑ

right:
    in inputC,PINC      ; ΔΙΑΒΑΖΩ ΤΗΝ ΕΙΣΟΔΟ ΑΠΟ ΤΗΝ ΠΥΛΗ A
    subi inputC,4       ; ΑΦΑΙΡΩ 4 ΑΠΟ ΤΗΝ ΕΙΣΟΔΟ
    breq right          ; ΕΛΕΓΧΩ ΑΝ Η ΕΙΣΟΔΟΣ ΕΙΝΑΙ 4(ΑΝ ΕΙΝΑΙ
                        ; ΠΑΤΗΜΕΝΟ ΔΕΝ ΚΑΝΩ ΤΙΠΟΤΑ ΚΑΙ ΠΕΡΙΜΕΝΩ ΝΑ ΣΒΗΣΕΙ)
                        ; ΑΝ ΔΕΝ ΕΙΝΑΙ ΠΡΟΧΩΡΑΩ ΣΤΑ ΕΠΟΜΕΝΑ
    clc                 ; ΚΑΝΩ clear ΤΟΝ ΚΑΤΑΧΩΡΗΤΗ ΚΡΑΤΟΥΜΕΝΟΥ
    ror lightBit        ; ΟΛΙΣΘΗΣΗ ΤΩΝ LEDΣ ΜΙΑ ΘΕΣΗ ΔΕΞΙΑ
    out PORTB,lightBit  ; ΑΝΑΒΩ ΤΟ ΕΠΟΜΕΝΟ LED(ΑΜΕΣΩΣ ΔΕΞΙΑ
                        ; ΚΑΘΕ ΦΟΡΑ)
    inc counter         ; ΑΥΞΑΝΩ ΤΟΝ ΜΕΤΡΗΤΗcounter--
    cpi counter,7       ; ΑΝ 0 counter == 7 ΕΙΜΑΙ ΣΤΟ LSB ΚΑΙ ΤΟΤΕ
    breq left           ; ΞΕΚΙΝΑΩ ΚΑΙ ΠΑΩ ΠΡΟΣ ΤΑ ΑΡΙΣΤΕΡΑ
    rjmp right          ; ΑΛΛΙΩΣ ΣΥΝΕΧΙΖΩ ΚΑΙ ΠΑΩ ΠΑΛΙ ΠΡΟΣ ΤΑ ΔΕΞΙΑ

end:

```

ΑΣΚΗΣΗ 2

Θέλουμε να υλοποιηθούν σε ένα σύστημα Μικροελεγκτή AVR οι λογικές συναρτήσεις:

$$F0 = (AB' + BC'D)'$$

$$F1 = (A+C) \cdot (B+D)$$

Οι τιμές των F0- F1 να εμφανιστούν αντίστοιχα στα δύο LSB της θύρας εξόδου PORTA (0-1). Οι μεταβλητές εισόδου (A, B, C, D) υποθέτουμε ότι αντιστοιχούν στα 4 bit της θύρας εισόδου PORTB (0-3), με το A στο LSB.

```

.INCLUDE "m16def.inc"
.def second=r17
.def input=r18
.def temp=r19
.def A=r20
.def B=r21
.def C=r22
.def D=r23
.def F0=r24
.def F1=r25

main:
clr temp
out DDRB,temp          ; ΘΕΤΩ ΤΟ B ΩΣ ΕΙΣΟΔΟ
ldi temp,0b00000011
out DDRA,temp          ; ΚΡΑΤΑΩ ΤΑ 2 LSBS ΤΟΥ A ΓΙΑ ΕΞΟΔΟ

in input,PINB
mov A,input
andi A,0x01            ; ΑΠΟΜΟΝΩΝΩ ΤΟ ΠΡΩΤΟ BIT ΕΞΟΔΟΥ

mov B,input
andi B,0x02            ; ΑΠΟΜΟΝΩΝΩ ΤΟ ΔΕΥΤΕΡΟ BIT ΕΞΟΔΟΥ
clc
ror B                  ; ΚΑΝΩ ΜΙΑ ΔΕΞΙΑ ΟΛΙΣΘΗΣΗ ΓΙΑ

```

```

; NA EXΩ TO ΔΕΥΤΕΡΟ BIT ΣΤΟ LSB

mov C,input
andi C,0x04
clc
ror C
ror C
; ΚΑΝΩ 2 ΔΕΞΙΕΣ ΟΛΙΣΘΗΣΕΙΣ ΓΙΑ ΝΑ
; ΕΧΩ ΤΟ ΤΡΙΤΟ BIT ΣΤΟ LSB

mov D,input
andi D,0x08
clc
ror D
ror D
ror D
; ΚΑΝΩ 3 ΔΕΞΙΕΣ ΟΛΙΣΘΗΣΕΙΣ ΓΙΑ ΝΑ
; ΕΧΩ ΤΟ ΤΕΤΑΡΤΟ BIT ΣΤΟ LSB

mov F0,A
com B
and F0,B
com B
mov temp,B
and temp,D
com C
and temp,C
com C
or F0,temp
com F0
andi F0,0x01
; F0 = A
; F0 = A*not(B)
; temp = B
; temp = B*D
; temp = B*D*not(C)
; F0 = (A*not(B)) + (B*D*not(C))
; F0 = not((A*not(B)) + (B*D*not(C)))
; ΕΧΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΣΤΟ LSB

mov F1,A
or F1,C
mov temp,B
or temp,D
and F1,temp
andi F1,0x01
; F1 = A
; F1 = A+C
; temp = B
; temp = B+D
; F1 = (A+C) * (B+D)
; ΕΧΩ ΤΟ ΑΠΟΤΕΛΕΣΜΑ ΣΤΟ LSB

clc
rol F1
add F0,F1
out PORTA,F0
rjmp main
end:
; ΤΟ F0 ΕΧΕΙ ΣΤΑ 2 LSBS ΤΑ ΑΠΟΤΕΛΕΣΜΑΤΑ
;ΑΠΕΙΡΟ LOOP

```

ΑΣΚΗΣΗ3

Θέλουμε να γραφτεί πρόγραμμα σε C για τον προγραμματισμό AVR Atmega16 το οποίο αρχικά να ανάβει το led0 που είναι συνδεδεμένο στο bit0 της θύρας εξόδου PORTB. Στην συνέχεια με το πάτημα των διακοπών (Push-buttons) SW0-3 που υποθέτουμε ότι είναι συνδεδεμένα στα αντίστοιχα bit της θύρας εισόδου PORTA να συμβαίνουν τα εξής:

- SW0 ολίσθηση-περιστροφή του led μια θέση δεξιά (κυκλικά).
- SW1 ολίσθηση-περιστροφή του led μια θέση αριστερά (κυκλικά).
- SW2 μετακίνηση του αναμμένου led στην αρχική του θέση (LSB-led0).
- SW3 μετακίνηση του αναμμένου led στην θέση MSB (led7).

Όλες οι αλλαγές να γίνονται όταν αφήνεται κάθε Push-button SWx (bitx PORTA).

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

char z; /*ορίζω μια μεταβλητή που θα μου δείχνει την έξοδο*/

int main(void){
    DDRB=0xFF; /*αρχικοποιώ το PORTB σαν έξοδο*/
    DDRA=0x00; /*αρχικοποιώ το PORTA σαν είσοδο*/
    z=1; /*ανάβω αρχικά το πρώτο λεντ
    PORTB=z; /*και το απεικονίζω στην έξοδο*/

    while (1) {

        if ((PINA&0x01)==1) { /*αν είναι αναμμένο το πρώτο λεντ της εισόδου*/
            while((PINA&0x01) == 1); /*περιμένω μέχρι να σταματήσει να είναι αναμμένο*/
            if(z==1) /*άμα είναι προηγουμένως αναμμένο το πρώτο λεντάκι*/
                z=128; /*τότε ανάβει το τελευταίο λεντάκι(έλεγχος υπερχειλίσης)*/
            else z=z>>1; /*αλλιώς γίνεται ολίσθηση μια θέση δεξιά*/
        }

        if ((PINA & 0x02)==2) { /*αν είναι αναμμένο το δεύτερο λεντ της εισόδου*/
            while((PINA & 0x02)==2); /*περιμένω μέχρι να σταματήσει να είναι αναμμένο*/
            if (z==128) /*άμα είναι προηγουμένως αναμμένο το έβδομο λεντάκι*/
                z=1; /*τότε ανάβει το πρώτο λεντάκι(έλεγχος υπερχειλίσης)*/
            else z=z<<1; /*αλλιώς γίνεται ολίσθηση μια θέση αριστερά*/
        }

        if ((PINA&0x04)==4) { /*αν είναι αναμμένο το τρίτο λεντ της εισόδου*/
            while((PINA&0x04)==4); /*περιμένω μέχρι να σταματήσει να είναι αναμμένο*/
            z=1; /*ανάβει το πρώτο λεντ της εξόδου*/
        }

        if ((PINA & 0x08)==8) { /*αν είναι αναμμένο το τέταρτο λεντ εισόδου*/
            while((PINA&0x08)==8); /*περιμένω μέχρι να σταματήσει να είναι αναμμένο*/
            z=128; /*ανάβει το έβδομο λεντ της εξόδου*/
        }
        /*αλλιώς αμα δεν είναι αναμμένο κανένα από τα πρώτα τέσσερα λεντ δεν αλλάζει
        τίποτα στην έξοδο */

        PORTB=z; /*απεικονίζω την έξοδο στο PORTB*/
    }
    return 0;
}
```