Σχολή Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών ΕΜΠ

ΤΟΜΕΑΣ ΤΕΧΝΟΛΟΓΙΑΣ ΠΛΗΡΟΦΟΡΙΚΗΣ ΚΑΙ ΥΠΟΛΟΓΙΣΤΩΝ
ΕΡΓΑΣΤΗΡΙΟ ΜΙΚΡΟΫΠΟΛΟΓΙΣΤΩΝ ΚΑΙ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ ΑΚΑΔ. ΕΤΟΣ 2024-2025

1ή εργαστηριακή άσκηση.

Ομάδα : 58
Ονοματεπώνυμο : Φέζος Κωνσταντίνος / Α.Μ : 03118076
Ονοματεπώνυμο : Αρθούρος Ρήγας Τσουκνίδας Οδυσσέας / Α.Μ : 03120043

Ζήτημα 1.1

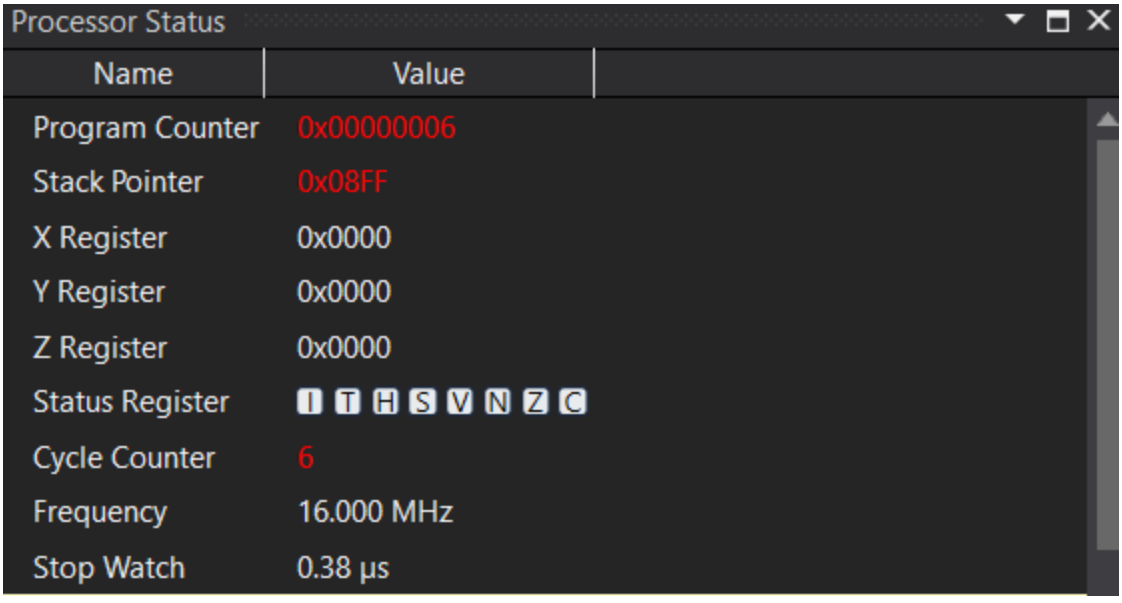
Παρακάτω ακολουθεί ο κώδικας που χρησιμοποιήσαμε για να δημιουργήσουμε πρόγραμμα assembly που παράγει καθυστέρηση ίση με X ms. Ο ακέραιος X ορίζεται στην αρχή του προγράμματος και ανήκει στο διάστημα [1,65535]. Χρησιμοποιούμε 3 loops έτσι ώστε να καταλήξουμε στο τελικό επιθυμητό αριθμών κύκλων. Ο αριθμός κύκλων που παράγεται σε κάθε βρόχο φαίνεται στα σχόλια του παρακάτω κώδικα. Παρακάτω φαίνεται και η ένδειξη του χρονομέτρου στο περιβάλλον του Microchip σε συνθήκες προσομοίωσης.(Χρησιμοποιούμε X = 10ms = 10,000μs καθώς υπάρχει αρκετά μεγάλη καθυστέρηση στην προσομοίωση για μεγάλες τιμές του X).

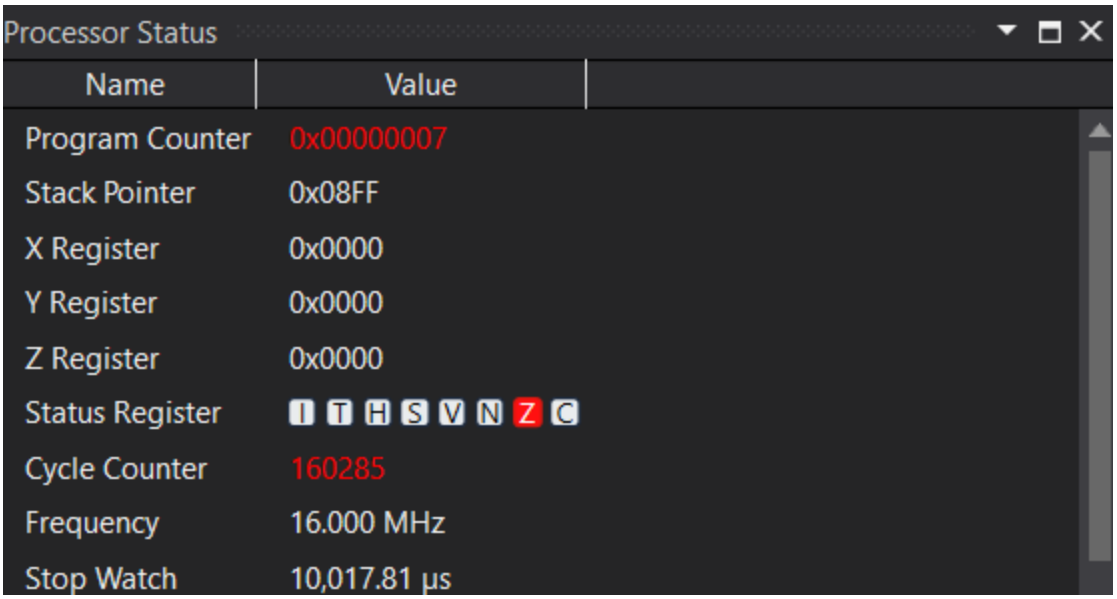
Ο κώδικας για το ζήτημα 1.1 :

```
.include "m328PBdef.inc"
.equ X1 = 10   ; mS

; initialize stack pointer
    ldi r24, LOW(RAMEND)
    out SPL, r24
    ldi r24, HIGH(RAMEND)
    out SPH, r24

    ldi r24, low(X1)
    ldi r25, high(X1)

loop1:
    rcall wait_x_msec      ; mS ; cycle = 16*x*1000
    rjmp loop1
wait_x_msec:
    ldi r16, 16
extra_outer_delay:
    rcall delay_outer
    subi r16,1
    brne extra_outer_delay
    ret
;this routine is used to produce a delay 993 cycles
delay_inner:
    ldi r23, 247          ; 1 cycle
loop3:
    dec r23               ; 1 cycle
    nop                   ; 1 cycle
```

```
    brne loop3              ; 1 or 2 cycles
    nop                     ; 1 cycle
    ret                     ; 4 cycles
;this routine is used to produce a delay of (1000*X1) cycles
delay_outer:
    push r24                ; (2 cycles)
    push r25                ; (2 cycles) Save r24:r25
loop4:
    rcall delay_inner       ; (3+993)=996 cycles
    sbiw r24,1              ; 2 cycles
    brne loop4              ; 1 or 2 cycles
    pop r25                 ; (2 cycles)
    pop r24                 ; (2 cycles) Restore r24:r25
    ret                     ; 4 cycles
```

Βάζοντας δυο breakpoint σε κάθε μία εντολή του loop1 βλέπουμε τα παρακάτω στο περιβάλλον προσομοίωσης του microchip studio.Η πρώτη εικόνα είναι εκτέλεση του προγράμματος έως και το πρώτο breakpoint και η δεύτερη η ολοκλήρωση της πρώτης καθυστέρησης των 10 ms.





Ζήτημα 1.2

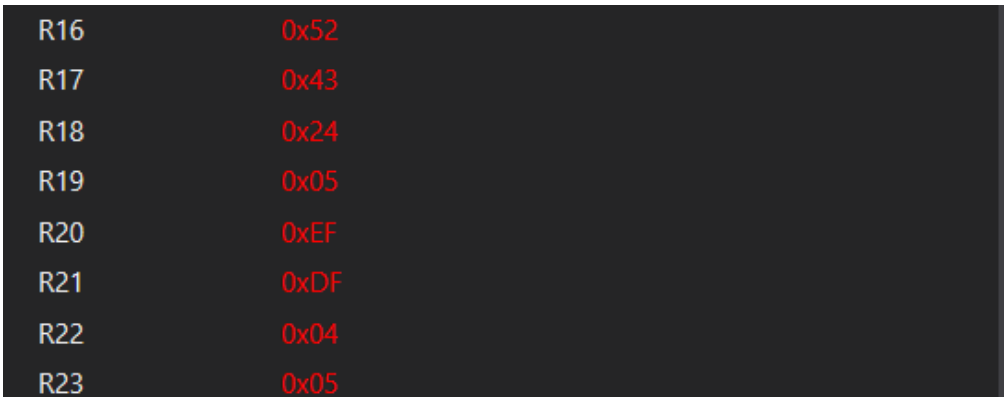
Στο ζήτημα 1.2 χρησιμοποιήσαμε ορισμένους καταχωρητές για να αποθηκεύσουμε τα αποτελέσματα που μας ενδιέφεραν κατά την εκτέλεση των λογικών πράξεων. Παρακάτω ακολουθεί ο κώδικας και ο πίνακας που μας ζητήθηκε συμπληρωμένος. Επίσης ένα screenshot από το microchip studio που μας δείχνει ενδεικτικά τις τιμές των καταχωρητών μετά το πέρας της πρώτης επανάληψης.

Ο κώδικας :

```
.include "m328PBdef.inc"
.org 0x00

.def A= r16
.def B= r17
.def C= r18
.def D= r19
.def F0= r20
.def F1= r21
.def dummy= r22
.def counter= r23

main:
    ldi A, 0x51
```

```
    ldi B, 0x41
    ldi C, 0x21
    ldi D, 0x01
    ldi counter, 0x05
loop:
    mov F0,B ; F0 = B
    com F0 ; F0 = B'
    mov dummy, F0 ; dummy = B'
    and F0, A ; F0 = A*B'
    and dummy, D ; dummy = B'*D
    or F0, dummy ; F0 = A*B' + B'*D
    com F0 ; F0 = (A*B' + B'*D)'
    mov F1, C ; F1 = C
    com F1 ; F1 = C'
    or F1, A ; F1 = A+C'
    mov dummy, D ; dummy = D
    com dummy ; dummy = D'
    or dummy, B ; dummy = B+D'
    and F1, dummy ; F1 = (A+C')*(B+D')
    ldi dummy, 0x01
    add A, dummy ; A += 0x01
    ldi dummy, 0x02
    add B, dummy ; B += 0x02
    ldi dummy, 0x03
    add C, dummy ; C += 0x03
    ldi dummy, 0x04
    add D, dummy ; D += 0x04
    dec counter
    brge loop ; ----
END:
    rjmp END
```

Οι τιμές των καταχωρητών που χρησιμοποιήσαμε για τις μεταβλητές του ζητήματος :



Ο πίνακας των τιμών συμπληρωμένος :

A	B	C	D	F0	F1
0x51	0x41	0x21	0x01	0xEF	0xDF
0x52	0x43	0x24	0x05	0xEB	0xDB
0x53	0x45	0x27	0x09	0xE5	0xD3
0x54	0x47	0x2A	0x0D	0xE7	0xD5
0x55	0x49	0x2D	0x11	0xEB	0xC7
0x56	0x4B	0x30	0x15	0xEB	0xCB

Ζήτημα 1.3

Στο ζήτημα 1.3 έπρεπε να εκτελέσουμε την κίνηση ενός βαγονέτου στην έξοδο PORTD. Για τις καθυστερήσεις ανάμεσα στις θέσεις του βαγονέτου χρησιμοποιήσαμε τον κώδικα του ζητήματος 1.1. Στο T flag αποθηκεύουμε την κατεύθυνση του βαγονέτου ως εξής :

- Από το LSB στο MSB T flag = 0
- Από το MSB στο LSB T flag = 1

Παρακάτω ακολουθεί ο κώδικας και οι ενδείξεις από το περιβάλλον του microchip μετά την μετακίνηση της πρώτης θέσης. Το delay εδώ είναι ένα δευτερόλεπτο σε κάθε μετακίνηση και δύο δευτερόλεπτα στην αλλαγή κατεύθυνσης. Τα παραπάνω έχουν επιβεβαιωθεί μέσω του προσομοιωτή αλλά και με τη χρήση της πλακέτας NTUAboard_G1 στο εργαστήριο.

Ο κώδικας :

```
.include"m328PBdef.inc"
.org 0
.DEF temp=r17
.DEF leds=r18
.equ X1=1000

reset:
    ldi r24 , low(RAMEND)
    out SPL , r24
    ldi r24 , high(RAMEND)
    out SPH , r24

start:
    ldi leds,0x01
    ser temp
    out DDRD,temp
    ldi r24, low(X1)
    ldi r25, high(X1)
    jmp go_left
change_dir_left:
    clt                         ;left(T = 0)
    rcall wait_x_msec
go_left:
    out PORTD,leds
    sbrc leds,07
    jmp change_dir_right
    rcall wait_X_msec
    lsl leds
    rjmp go_left
change_dir_right:
    set                         ;right(T = 1)
    rcall wait_X_msec
go_right:
    out PORTD,leds
    sbrc leds,00
    rjmp change_dir_left
    rcall wait_X_msec
    lsr leds
    rjmp go_right
```

```
wait_x_msec:
    ldi r16, 16
extra_outer_delay:
    rcall delay_outer
    subi r16,1
    brne extra_outer_delay
    ret
delay_inner:
    ldi r23, 247          ; 1 cycle
loop3:
    dec r23               ; 1 cycle
    nop                   ; 1 cycle
    brne loop3            ; 1 or 2 cycles
    nop                   ; 1 cycle
    ret                   ; 4 cycles

;this routine is used to produce a delay of (1000*X1) cycles
delay_outer:
    push r24              ; (2 cycles)
    push r25              ; (2 cycles) Save r24:r25

loop4:
    rcall delay_inner     ; (3+993)=996 cycles
    sbiw r24,1            ; 2 cycles
    brne loop4            ; 1 or 2 cycles

    pop r25               ; (2 cycles)
    pop r24               ; (2 cycles) Restore r24:r25
    ret                   ; 4 cycles
```

Οι ενδείξεις του microchip studio :