Εργαστήριο Μικροϋπολογιστών

Πρώτη Εργαστηριακή Άσκηση

Γεράσιμος Μαρκαντωνάτος el19149
Οδυσσεύς Δρυς-Πεντζάκης el19192

Ζήτημα 1.1

Η εντολή sbiw αφαιρεί από τον διπλό καταχωρητή όταν εκτελείται. Αφαιρούμε μια φορά από τον διπλό καταχωρητή έξω από το loop ώστε στην περίπτωση που είμαστε στην τελευταία επανάληψη να εκτελέσουμε μια διαφορετική ακολουθία καθώς η τελευταία εκτέλεση θα διαρκέσει περισσότερο από τις προηγούμενες. Επίσης όταν θέλουμε να διαρκέσει 1 ms η εντολή εκτελούμε επίσης την διαφορετική ακολουθία που αναφέραμε πάνω.

Κώδικας assembly:

```
.include "m328PBdef.inc"

main:
   rcall wait_x_msec
   jmp main

wait_x_msec:
   ldi r24,0xFF
   ldi r25, 0x01
   sbiw r24,1
   breq jump1
loop1:
   rcall wait1m
   sbiw r24,1
   brne loop1
   rcall waitm
   ret

wait4:
   ret

wait1m:
   ldi r26, 98

loop:
   rcall wait4
   dec r26
   brne loop
   nop
   nop
```

```
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    ret

waitm:
    ldi r26, 98

loop2:
    rcall wait4
    dec r26
    brne loop2
    nop
    nop
    ret
jump1:
    ldi r26, 98

loop3:
    rcall wait4
    dec r26
    brne loop3
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    ret
```

Ζήτημα 1.2

Κώδικας assembly:

```
.include "m328PBdef.inc"
.def A=r16
.def B=r17
.def C=r18
.def D=r19
.def temp=r20
.def f0=r21
.def f1=r22
.def count=r23
```

```
ldi A,0x55
ldi B,0x43
ldi C,0x22
ldi D,0x02
ldi count,0x06

start:
    mov f0,A
    com f0
    mov temp,B
    com temp
    and f0,temp
    and temp,D
    or f0,temp
    com f0
    mov f1,A
    or f1,C
    mov temp,D
    com temp
    or temp,B
    and f1,temp
    ldi temp,0x02
    add A,temp
    inc temp
    add B,temp
    inc temp
    add C,temp
    inc temp
    add D,temp
    dec count
    brne start
```

Πίνακας τιμών:

A	B	C	D	F0	F1
0x55	0x43	0x22	0x02	0x57	0x77
0x57	0x46	0x26	0x07	0x56	0x76
0x59	0x49	0x2A	0x0C	0x59	0x7B
0x5B	0x4C	0x2E	0x11	0x4E	0x6E
0x5D	0x4F	0x32	0x16	0x4F	0x6F
0x5F	0x52	0x36	0x1B	0x56	0x76

Ζήτημα 1.3

Όταν θέλουμε το βαγονέτο να μετακινηθεί προς τα αριστερά θέτουμε το T flag ως 1 μέσω της εντολής set και όταν θέλουμε το βαγονέτο να μετακινηθεί προς τα δεξιά θέτουμε το T flag ως 0 μέσω της εντολής clt. Κάθε φορά που μετακινείται προς τα αριστερά ή δεξιά ελέγχουμε μέσω των εντολών brts και brtc την τιμή του T flag και ανάλογα την τιμή τους πραγματοποιούμε την αντίστοιχη κίνηση του βαγονέτου.

Κώδικας assembly:

```
.include "m328PBdef.inc"
.def temp=r16
.def led=r17
.def data=r18
.def position=r19

    clr temp
    out DDRD, temp
    ldi led,0x01
    set
    clr position
    out PORTD , led
    jmp left
left1:
    set
    rcall wait_x_msec
    rcall wait_x_msec
left:
    inc position
    rcall wait_x_msec
    lsl led
    out PORTD , led
    cpi position , 7
    breq right1
    brts left

right1:
    clt
    rcall wait_x_msec
    rcall wait_x_msec

right:
    dec position
    rcall wait_x_msec
    lsr led
    out PORTD , led
    cpi position , 0
    breq left1
```

```
    brtc right

wait_x_msec:
    ldi r24,0xD0
    ldi r25, 0x07
    sbiw r24,1
    breq jump1
loop1:
    rcall wait1m
    sbiw r24,1
    brne loop1
    rcall waitm
    ret

wait4:
    ret

wait1m:
    ldi r26, 98

loop:
    rcall wait4
    dec r26
    brne loop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    ret

waitm:
    ldi r26, 98

loop2:
    rcall wait4
    dec r26
    brne loop2
    nop
    nop
    ret
jump1:
    ldi r26, 98
```

```
loop3:
    rcall wait4
    dec r26
    brne loop3
    nop
    nop
    nop
    nop
    nop
    nop
    nop
    ret
```