

علیرضا اسلامی خواه

99521064

تمرین دوم ریزپردازنده و زبان اسمبلی

(سوال 1)

$$L3: 1 + 255*(1+1+2) - 1 = 1020$$

$$L2: 1 + 255*(L3+ 1 +2) - 1 = 260865$$

$$L1 : 1 + 16*(1 + L2 + 1 + 2) - 1 + 4 = 4173908$$

$$TOTAL DELAY = 4173908 \text{ Cycle} = 4.17s$$

(سوال 2)

در این کد هنگامی که ret فراخوانی میشود و به طبع برنامه به خطی که delay فراخوانی شده بود برمیگردد چون استک ما خالی است دچار خطا میشویم.

```
AssemblerApplication1  main.asm  ✎ ✕
#include "m32def.inc"
start:
    ldi r16, HIGH(Ramend)
    out SPH, r16
    ldi r16, LOW(Ramend)
    out SPL, r16
    call delay
    rjmp exit
delay:
    ldi r20, 0x10
L1:
    ldi r21, 0xff
L2 :
    ldi r22, 0xff
L3:
    NOP
    dec r22
    brne L3
    dec r21
    brne L2
    dec r20
    brne L1
    ret
exit:
    rjmp exit
```

(سوال 3)

مانند سوال اول اگر اینگونه تغییرات را انجام دهیم به کلاک یک چهارم (یعنی 1 ثانیه) میرسیم.

طبق این فرمول باید یک حلقه چهارمی هم اضافه شود که مقدار آن 0x04 است.

```
L3 = 1 + 255*(1+1+2) -1  
L2 = 1 + 255*(L3+ 1 +2) -1  
L1 = 1 + 4*(1 + L2 + 1 + 2) -1 + 4
```

مقدار کلاک نهایی در اینجا 8347820 است که اگر تبدیل به ثانیه شود حدودا 1.04 طول میکشد.

کد نهایی :

```
.include "m32def.inc"  
start:  
    call delay  
delay:  
    ldi r20, 0x04  
L1:  
    ldi r21, 0xff  
L2 :  
    ldi r22, 0xff  
L3:  
    NOP  
    dec r22  
    brne L3  
    dec r21  
    brne L2  
    dec r20  
    brne L1  
    ret
```

البته میتوان مانند سوال 2 مشتقات بیشتر هم به آن اضافه کرد که دچار ارور نشویم.

سوال 4)

در این برنامه ما یک حلقه کلی داریم که هنگامی که عدد دلخواهش را دریافت میکند یکی به آن اضافه کرده و دوباره اجرا میشود.

```
main.asm*
; Load immediate value 0 into register R16
ldi R16, 0
; Load immediate value 0xFF into register R17
ldi R17, 0xFF
; Load immediate binary value 11001010 into register R18
ldi R18, 0b11001010
; Set PORTA as output (all bits to 0)
out DDRA, R16
; Set PORTB as output (all bits to 0)
out DDRB, R16
; Set PORTD as input (all bits to 1)
out DDRD, R17
loop:
; Read input from PORTA and store in R19
in R19, PINA
; Compare R18 with R19
cp R18, R19
; If not equal, go back to the loop
brne loop
; Read input from PORTB and add to R19
in R20, PINB
add R19, R20

; Write the result to PORTD
out PORTD, R19

; Increment R18
inc R18

; Jump back to the loop
rjmp loop
```

تصویر پروتئوس :

