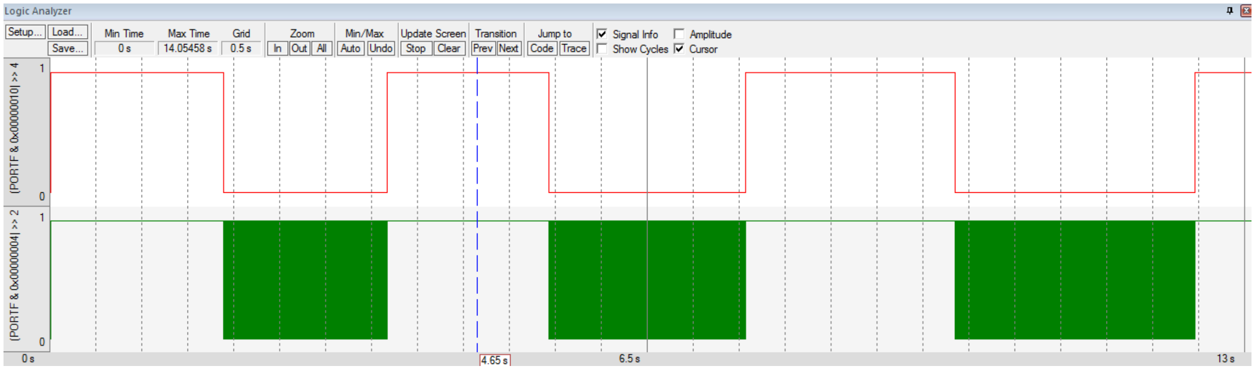
Kassandra Smith and Madhumitha Venkataraman
Professor R.Y.

Lab 2 Deliverables

1. Two screenshots

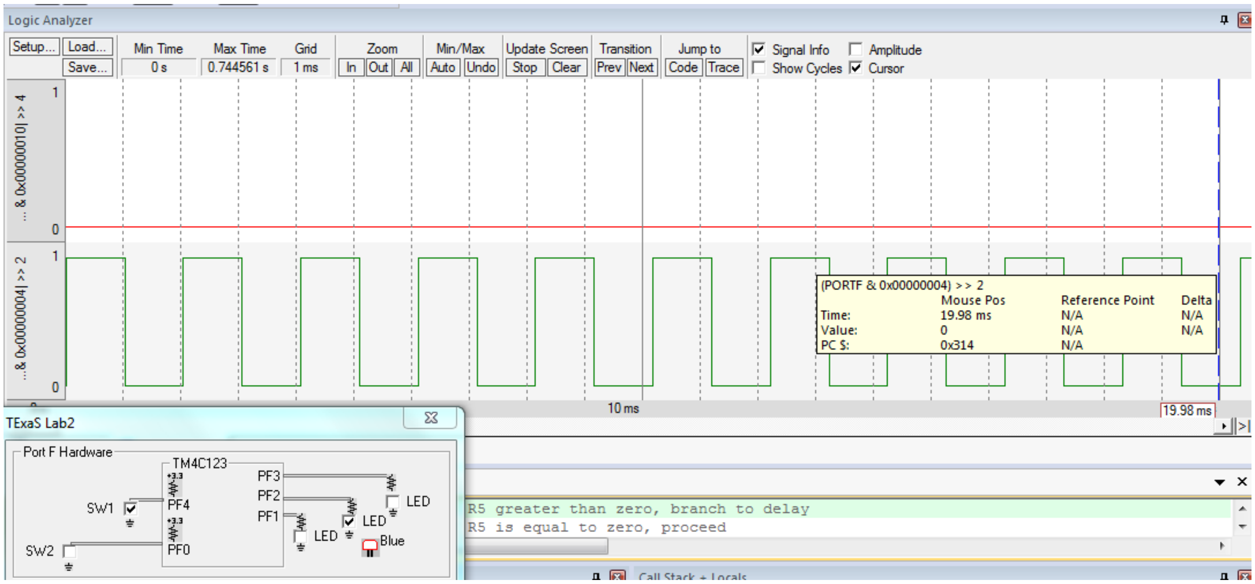
Touch and Release:



Min Time 0 s | Max Time 14.05458 s | Grid 0.5 s

☑ Signal Info □ Amplitude
□ Show Cycles ☑ Cursor

Pressed:



Min Time 0 s | Max Time 0.744561 s | Grid 1 ms

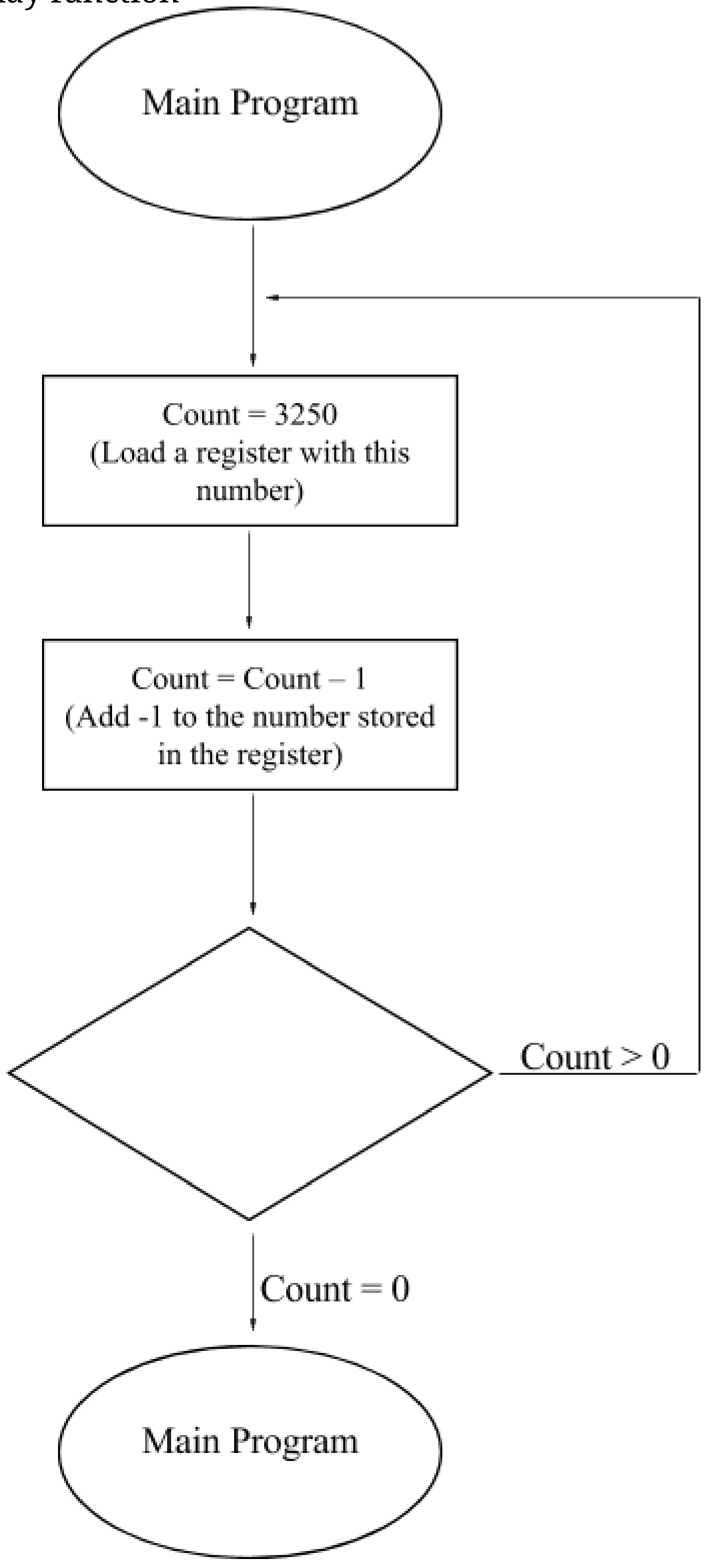
☑ Signal Info □ Amplitude
□ Show Cycles ☑ Cursor

(PORTF & 0x00000004) >> 2

	Mouse Pos	Reference Point	Delta
Time:	19.98 ms	N/A	N/A
Value:	0	N/A	N/A
PC $:	0x314	N/A	

```
R5 greater than zero, branch to delay
R5 is equal to zero, proceed
```

2. Flowchart of the delay function


Main Program
Count = 3250
(Load a register with this number)
Count = Count – 1
(Add -1 to the number stored in the register)
Count > 0
Count = 0
Main Program

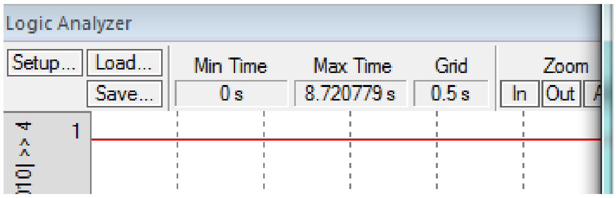
3. Pseudocode of the delay function

```
start
{
Main code
ADD 3250 to R5
While (R5>0) {
Subtract 1 from R5          }
Main code
          }
```

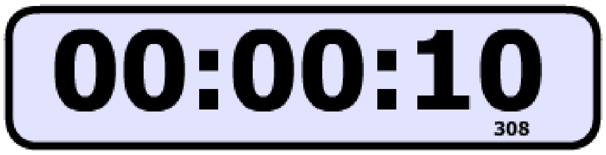
4. Assembly source code of your final program

```
;****************** main.s ***************
; Program written by: Kassandra Smith and Madhumitha Venkataraman
; Date Created: 1/24/2015
; Last Modified: 2/4/2015
; Section 1-2pm TA: Saugata Bhattacharyya
; Lab number: 2
; Brief description of the program
; The overall objective of this system is a digital lock
; Hardware connections
; PF4 is switch input  (1 means SW1 is not pressed, 0 means SW1 is pressed)
; PF2 is LED output (1 activates blue LED)
; The specific operation of this system
; 1) Make PF2 an output and make PF4 an input (enable PUR for PF4).
; 2) The system starts with the LED ON (make PF2 =1).
; 3) Delay for about 1 ms
; 4) If the switch is pressed (PF4 is 0), then toggle the LED once, else turn the LED ON.
; 5) Repeat steps 3 and 4 over and over

GPIO_PORTF_DATA_R   EQU   0x400253FC
GPIO_PORTF_DIR_REQU   0x40025400
GPIO_PORTF_AFSEL_R  EQU   0x40025420
GPIO_PORTF_PUR_REQU   0x40025510
GPIO_PORTF_DEN_REQU   0x4002551C
GPIO_PORTF_AMSEL_R  EQU   0x40025528
GPIO_PORTF_PCTL_R   EQU   0x4002552C
SYSCTL_RCGCGPIO_R   EQU   0x400FE608
GPIO_PORTF_LOCK_R                   EQU   0x40025520
GPIO_PORTF_CR_R                     EQU   0x40025524


  AREA|.text|, CODE, READONLY, ALIGN=2
  THUMB
  EXPORT  Start
Start
                                                                          ;Enable the clock for the port,
                                                                          ;Initialize clock,
        LDR R1, =SYSCTL_RCGCGPIO_R                    ;activate clock
        LDR R0, [R1]
        ORR R0, R0, #0x20                                    ;set bit 5 to turn on clock
        STR R0, [R1]                                         ;put it back
        NOP                                                  ;wait for stabilization,
        NOP


        LDR R1, =GPIO_PORTF_LOCK_R  ;unlock the lock register
        LDR R0, =0x4C4F434B ;unlock GPIO Port F Commit Register
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_CR_R;enable commit for Port F
        MOV R0, #0xFF   ;1 means allow access
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_AMSEL_R ;disable analog functionality
        LDR R0, [R1]
        BIC R0, #0x14                   ;Clear bits 2 and 4
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_PCTL_R  ;configure as GPIO
        LDR R0, [R1]
        BIC R0, #0x14                   ;0 means configure Port F as GPIO
        STR R0, [R1]
        LDR R1, =GPIO_PORTF_DIR_R               ;set direction register
        LDR R0, [R1]
        ORR R0,#0x04                                    ;PortF bit 2 is set to 1
        BIC R0,#0x10                                    ;clear bit 4
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_AFSEL_R     ;disable alternate function select
        LDR R0, [R1]
        BIC R0, #0x14                           ;We don't need the pins' special functions so we set it to 0
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_DEN_R               ;Set DEN so that the bits are useable, Port F digital port
        LDR R0, [R1]
        ORR R0, #0xFF                                   ;1 enables digital I/O
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_PUR_R               ;enable pull up resistor
        LDR R0, [R1]
        ORR R0, #0x10                                   ;enable PUR for bit 4
        STR R0, [R1]


        LDR R1, =GPIO_PORTF_DATA_R
        LDR R0, [R1]
        ORR R0, #0x04                                   ;starting the program with the LED on
        STR R0, [R1]


        AND R5, R5, #0                                  ;clearing register 5, to be used as counter for delay


loop
        ADD R5, #3250                                   ;set R5 to 3250
delay                                                           ;delay function
                ADD R5, #-1                                     ;subtract one from R5
                CMP R5, #0                                      ;if R5 greater than zero, branch to delay
                BGT delay                                       ;if R5 is equal to zero, proceed

                LDR R2, =GPIO_PORTF_DATA_R
                LDR R6, =GPIO_PORTF_DATA_R
                LDR R6, [R6]                            ;load data from Port F
                AND R6, #0x10                           ;masking for bit 4
                CMP R6, #0x10                           ;check and see is bit 4 is "1" (switch not pressed)
                BEQ turnon                                      ;if switch is not pressed, take the branch
                LDR R6, =GPIO_PORTF_DATA_R
                LDR R6, [R6]                            ;load data from Port F
                AND R6, #0x04                           ;masking for bit 2
                EOR R6, R6, #0x4                ;NOT bit 2
                STR R6, [R2]                            ;store result back to Port F
        B loop                                                  ;Branch back to beginning of loop

turnon                                                          ;turning or keeping the LED on
                LDR R1, =GPIO_PORTF_DATA_R
                LDR R2, =GPIO_PORTF_DATA_R
                LDR R1, [R1]                            ;load data from Port F
                ORR R1, #0x04                           ;set bit 2 to "1"
                STR R1, [R2]                            ;store result back to Port F
                                                                        ;end subroutine
B loop

  ALIGN                                         ;make sure the end of this section is aligned
  END                                   ;end of file
```

5. Measurement of how much microcontroller time is simulated in 10 seconds of actual time.



Min Time 0 s
Max Time 8.720779 s
Grid 0.5 s



00:00:10
308

Other times we measured were:
10.52652 seconds for every real second
6.58 seconds for every real second