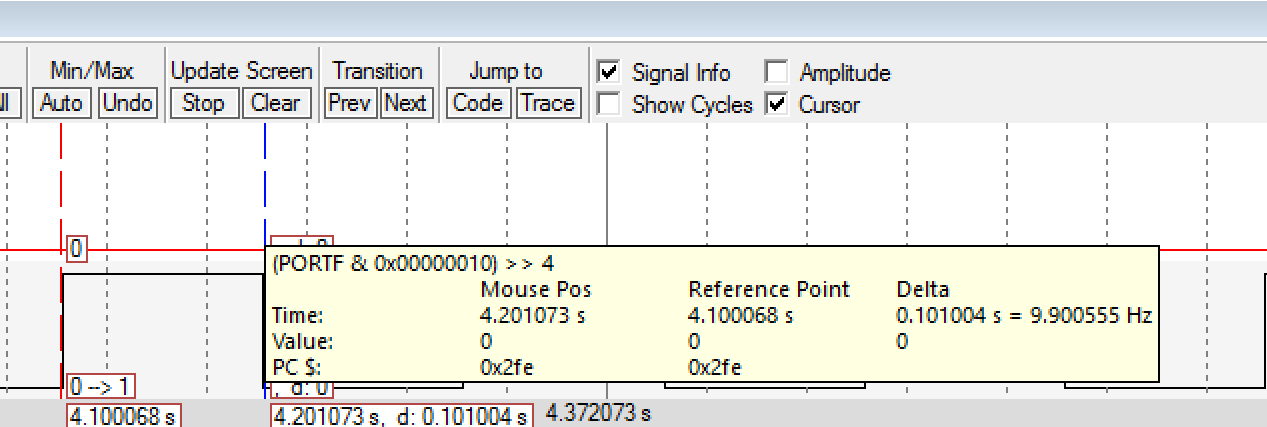
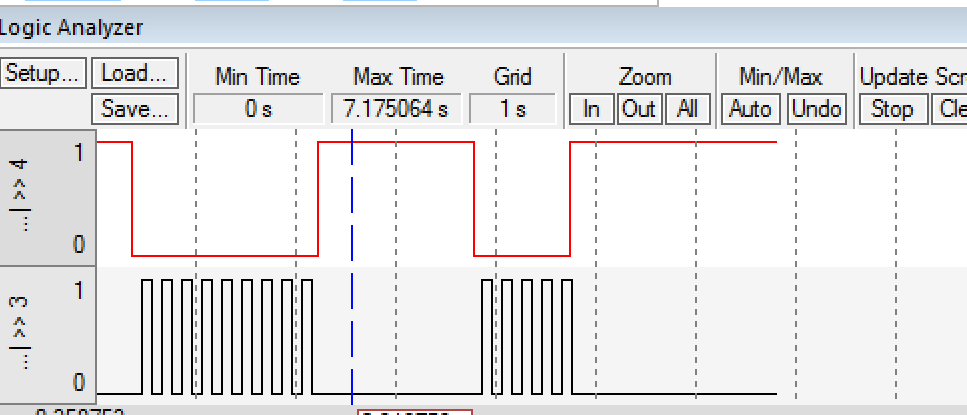
**LAB 2 Deliverables**

Akaash Chikarmane, Milan Feliciello

Section #: 16085

EIDs: avc536, MFF426



**Flowchart for Delay Subroutine**

****

**Pseudo-Code for Delay**

start of delay subroutine \*16000 cycles/ms

;\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* main.s \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

; Program written by: Akaash Chikarmane

; Date Created: 1/22/2016

; Last Modified: 2/12/2016

; Section Tuesday 2-3

; Instructor: Ramesh Yerraballi

; Lab number: 2

; Brief description of the program

; The overall objective of this system an interactive alarm

; Hardware connections

; PF4 is switch input (1 means SW1 is not pressed, 0 means SW1 is pressed)

; PF3 is LED output (1 activates green LED)

; The specific operation of this system

; 1) Make PF3 an output and make PF4 an input (enable PUR for PF4).

; 2) The system starts with the LED OFF (make PF3 =0).

; 3) Delay for about 100 ms

; 4) If the switch is pressed (PF4 is 0), then toggle the LED once, else turn the LED OFF.

; 5) Repeat steps 3 and 4 over and over

GPIO\_PORTF\_DATA\_R EQU 0x400253FC

GPIO\_PORTF\_DIR\_R EQU 0x40025400

GPIO\_PORTF\_AFSEL\_R EQU 0x40025420

GPIO\_PORTF\_PUR\_R EQU 0x40025510

GPIO\_PORTF\_DEN\_R EQU 0x4002551C

GPIO\_PORTF\_AMSEL\_R EQU 0x40025528

GPIO\_PORTF\_PCTL\_R EQU 0x4002552C

SYSCTL\_RCGCGPIO\_R EQU 0x400FE608

AREA |.text|, CODE, READONLY, ALIGN=2

THUMB

EXPORT Start

Start

LDR R0,=SYSCTL\_RCGCGPIO\_R

LDR R1, [R0]

ORR R1, #0x20 ;enable clock for Port F (0010 0000)

STR R1, [R0]

NOP ;stabilize clock

NOP

LDR R0,=GPIO\_PORTF\_DEN\_R

LDR R1, [R0]

ORR R1, #0x18 ;enable digital I/O for PF3, PF4

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_DIR\_R

LDR R1, [R0]

ORR R1, #0x08 ;PF3=output

BIC R1, #0x10 ;PF4=input

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_AFSEL\_R

LDR R1, [R0]

BIC R1, #0x18 ;no alternate function for PF3,4

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_PUR\_R

LDR R1, [R0]

ORR R1, #0x10 ;PUR enable for PF4

STR R1, [R0]

LDR R0,=GPIO\_PORTF\_DATA\_R

LDR R1, [R0]

BIC R1, #0x08 ;LED is initially off

STR R1, [R0]

loop

BL DELAY

LDR R0,=GPIO\_PORTF\_DATA\_R

LDR R1, [R0] ;read the switch (R1 = adjusting PF3)

MOV R2, R1 ;R2 = checking PF4

MOV R3, R1 ;R3 = original data

AND R2, #0x10 ;clear all but PF4

SUBS R2, #0x10

BEQ CLEAR ;if PF4 = 1, clear PF3

AND R1, #0x08 ;clear all but PF3

EOR R1, #0x08 ;toggle PF3

BIC R3, #0x08 ;clear original PF3

ORR R3, R1, R3 ;recombine

STR R3, [R0]

B loop

CLEAR

BIC R3, #0x08

STR R3, [R0]

B loop

DELAY LDR R0,=COUNT ;start of delay subroutine \*16000 cycles/ms

LDR R1, [R0]

REPEAT SUBS R1, R1, #1

BNE REPEAT

BX LR

COUNT DCD 400000

ALIGN ; make sure the end of this section is aligned

END ; end of file

**Simulation Timing**

