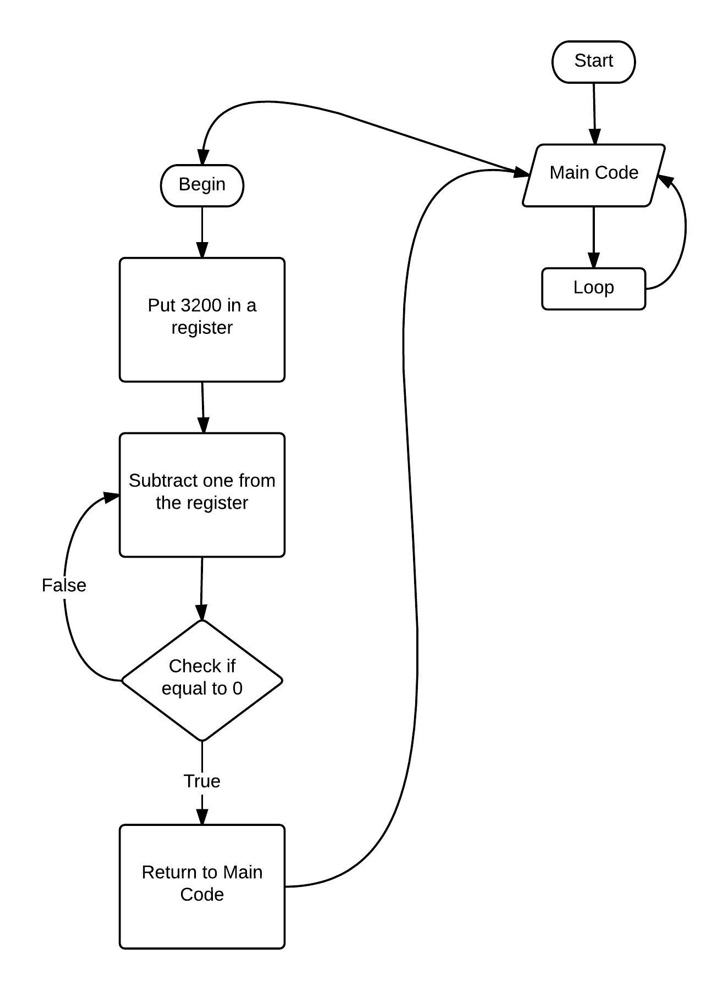
**Lab 2 Deliverables**

Delay

Set 3200 as count

Decrement

Count-1

Check if count = 0

If Count=0 go to End

If Count≠0 go to Decrement

End

Return to caller program

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

; Program written by: Zain Modi and Ali Ziyaan Momin

; Date Created: 9/20/2015

; Last Modified: 9/22/2015

; Section WED 4-5pm TA: Jenny Chen

; 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\_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

BL PortF\_Init ;subroutine that initializes port F

LDR R1, =GPIO\_PORTF\_DATA\_R ;turn on led via PF2

LDR R0, [R1]

ORR R0, R0, #0x04

STR R0, [R1]

loop

BL Delay ;go to delay subroutine

LDR R1, =GPIO\_PORTF\_DATA\_R ;the following steps compare PF4 to zero

LDR R0, [R1]

AND R2, R0, #0x10 ;mask data register to obtain value of PF4

CMP R2, #0x00 ;compare PF4 to zero

BNE TON ;if not equal to zero then go to TON (turn on subroutine), this means switch is not pressed

LDR R1, =GPIO\_PORTF\_DATA\_R ;the following steps will turn off the LED via PF2

LDR R0, [R1]

EOR R0, R0, #0x04 ;the exclusive or will make the LED turn off since 1EOR1 = 0

STR R0, [R1] ;store the PF2 modification back to the data register

B DONE

TON ;the section toggles led via PF2 bit being flipped

LDR R1, =GPIO\_PORTF\_DATA\_R

LDR R0, [R1]

ORR R0, #0X04

STR R0, [R1]

DONE

B loop

PortF\_Init

LDR R1, =SYSCTL\_RCGCGPIO\_R ;activate clock

LDR R0, [R1]

ORR R0, R0, #0x0020

STR R0, [R1]

NOP

NOP

NOP

NOP ;delay to allow time for clock to activate

LDR R1, =GPIO\_PORTF\_DIR\_R ;the subroutine that turns on the led via PF2

LDR R0, [R1]

ORR R0, R0, #0x04

BIC R0, #0x10

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_AFSEL\_R ;turn off alternate function for bits PF4 and PF2

LDR R0, [R1]

BIC R0, R0, #0x14

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_DEN\_R ;digital enable on PF4 and PF2 bits

LDR R0, [R1]

ORR R0, R0, #0x14

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_PUR\_R ;pull up resistor is enabled on bit PF4

LDR R0, [R1]

ORR R0, R0, #0x10

STR R0, [R1]

BX LR

Delay

MOV R0, #0xC80 ;move the number 3200 into R0

Decrement

ADD R0, R0, #-1 ;decrement by 1

CMP R0, #0x00 ;compare to 0

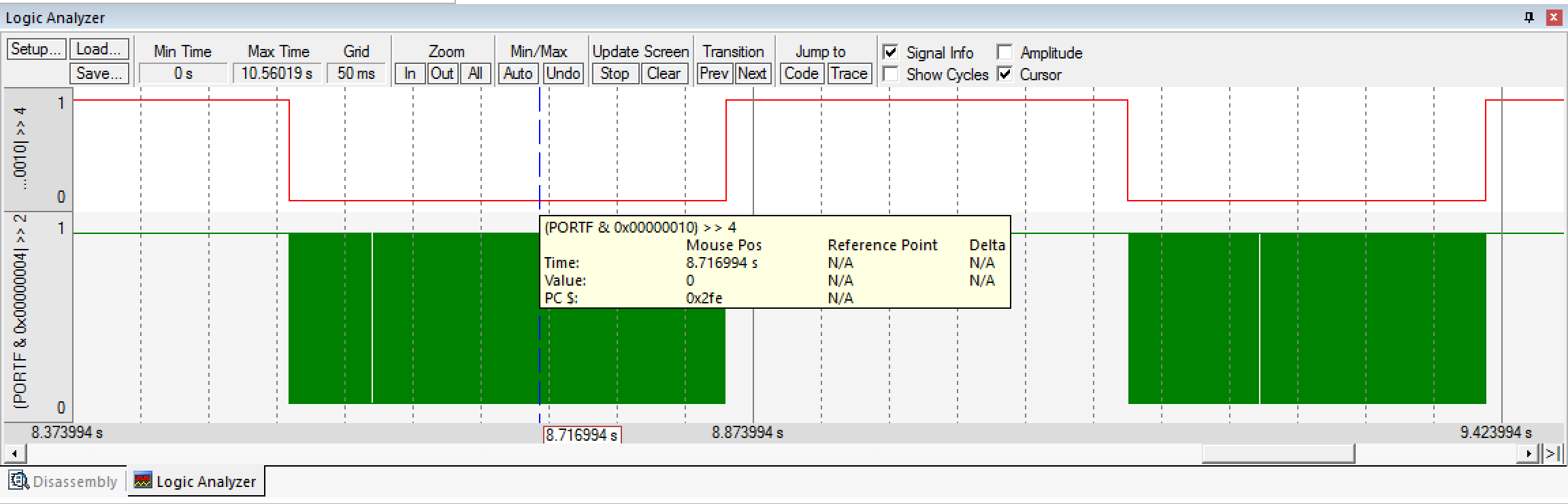
BNE Decrement ;if not equal to zero keep decrementing

BX LR ;equal to zero so return to loop

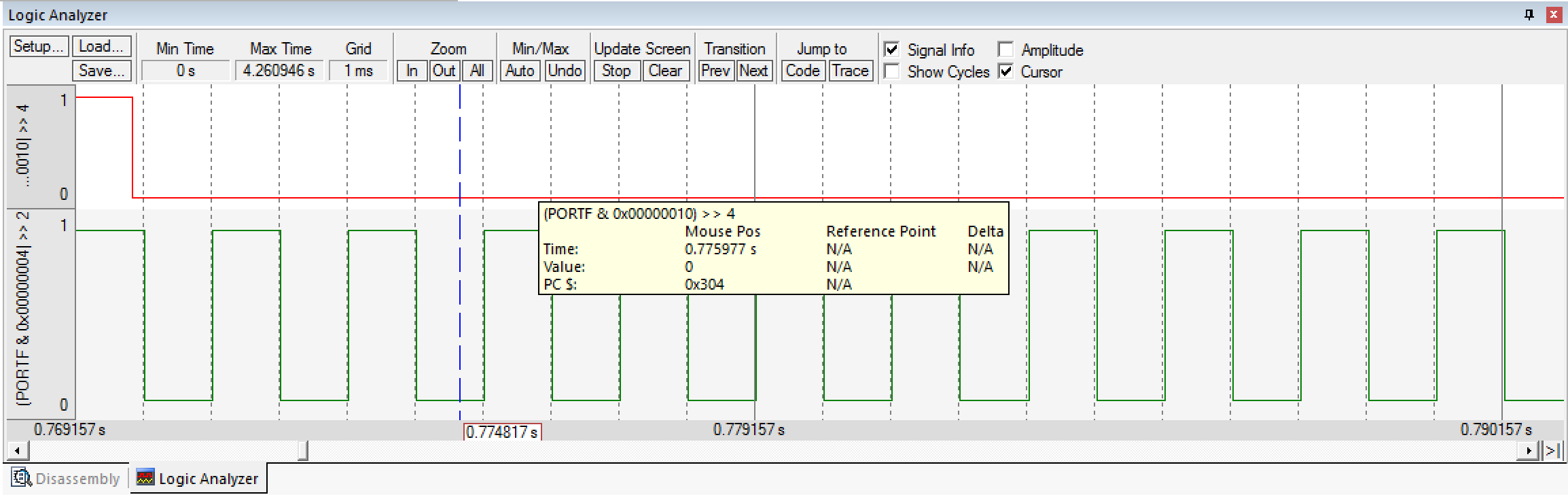
ALIGN ; make sure the end of this section is aligned

END ; end of file

Button Pressed:



LED Toggling:



Microcontroller time simulated in 10s of actual time:

