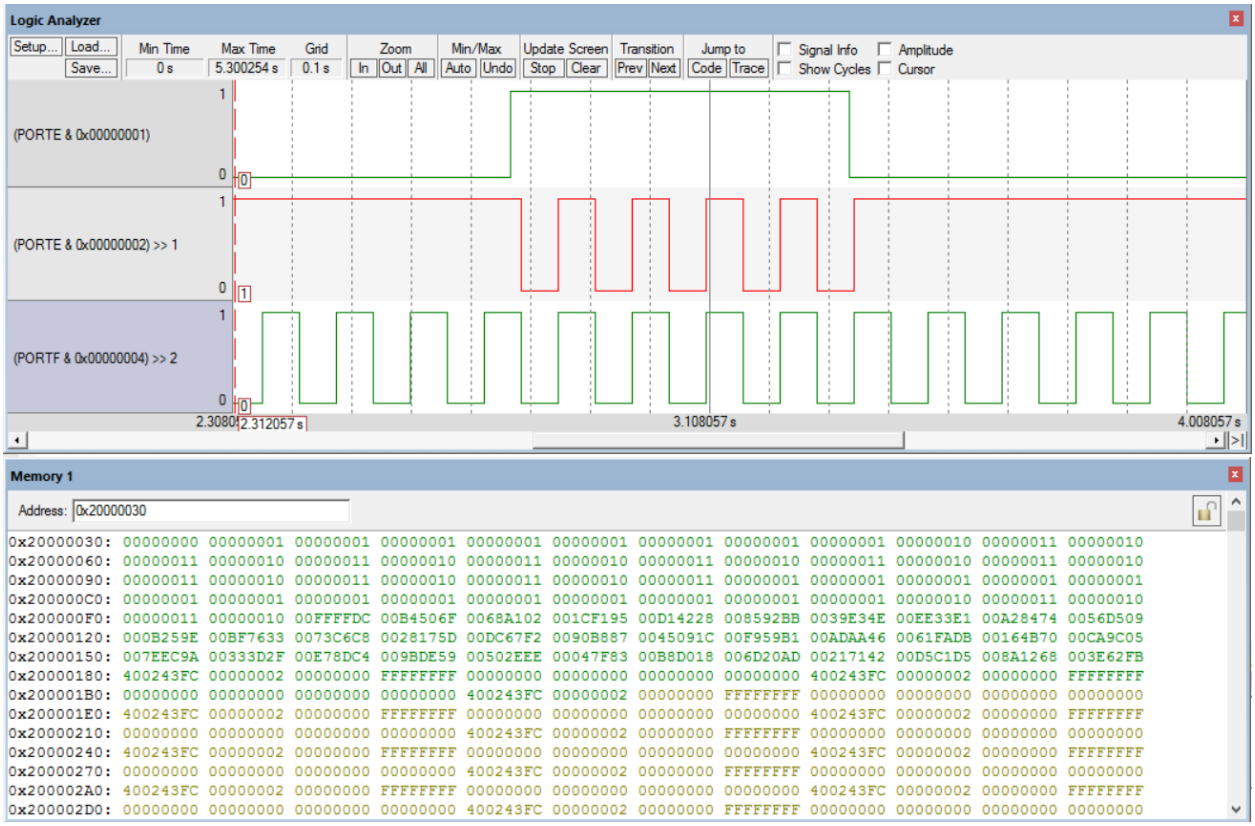
Eduardo Zueck Garces and Mai Phan

Logic and Memory Windows



Eduardo Zueck Garces and Mai Phan

Program

```
;****************** main.s ***************
; Program written by: Eduardo Zueck Garces and Mai Phan
; Date Created: 3/1/2015
; Last Modified: 3/2/2015
; Section 11am-12pm     TA: Wooseok Lee
; Lab number: 4
; Brief description of the program
;   If the switch is presses, the LED toggles at 8 Hz
; Hardware connections
;  PE0 is switch input  (1 means pressed, 0 means not pressed)
;  PE1 is LED output (1 activates external LED on protoboard)
;Overall functionality of this system is the similar to Lab 3, with four changes:
;1-  activate the PLL to run at 80 MHz (12.5ns bus cycle time)
;2-  initialize SysTick with RELOAD 0x00FFFFFF
;3-  add a heartbeat to PF2 that toggles every time through loop
;4-  add debugging dump of input, output, and time
; Operation
;           1) Make PE1 an output and make PE0 an input.
;           2) The system starts with the LED on (make PE1 =1).
;   3) Wait about 62 ms
;   4) If the switch is pressed (PE0 is 1), then toggle the LED once, else turn the LED on.
;   5) Steps 3 and 4 are repeated over and over


SWITCH                  EQU 0x40024004  ;PE0
LED                     EQU 0x40024008  ;PE1
SYSCTL_RCGCGPIO_R       EQU 0x400FE608
SYSCTL_RCGC2_GPIOE      EQU 0x00000010   ; port E Clock Gating Control
SYSCTL_RCGC2_GPIOF      EQU 0x00000020   ; port F Clock Gating Control
GPIO_PORTE_DATA_R       EQU 0x400243FC
GPIO_PORTE_DIR_R        EQU 0x40024400
GPIO_PORTE_AFSEL_R      EQU 0x40024420
GPIO_PORTE_PUR_R        EQU 0x40024510
GPIO_PORTE_DEN_R        EQU 0x4002451C
GPIO_PORTF_DATA_R       EQU 0x400253FC
GPIO_PORTF_DIR_R        EQU 0x40025400
GPIO_PORTF_AFSEL_R      EQU 0x40025420
GPIO_PORTF_DEN_R        EQU 0x4002551C
NVIC_ST_CTRL_R          EQU 0xE000E010
NVIC_ST_RELOAD_R        EQU 0xE000E014
NVIC_ST_CURRENT_R       EQU 0xE000E018
```

```
        THUMB
        AREA    DATA, ALIGN=4
SIZE      EQU    50
;You MUST use these two buffers and two variables
;You MUST not change their names
;These names MUST be exported
        EXPORT DataBuffer
        EXPORT TimeBuffer
        EXPORT DataPt [DATA,SIZE=4]
        EXPORT TimePt [DATA,SIZE=4]
DataBuffer SPACE  SIZE*4
TimeBuffer SPACE  SIZE*4
DataPt     SPACE  4
TimePt     SPACE  4

    ALIGN
    AREA    |.text|, CODE, READONLY, ALIGN=2
    THUMB
    EXPORT  Start
    IMPORT  TExaS_Init


Start BL   TExaS_Init  ; running at 80 MHz, scope voltmeter on PD3
           BL PortInitE
           BL PortInitF
           BL Debug_Init
    CPSIE  I    ; TExaS voltmeter, scope runs on interrupts

loop      BL Debug_Capture
                  BL Heartbeat
                  BL Delay

          ;We check the state of the switch PF4
                  LDR R0, =GPIO_PORTE_DATA_R
                  LDR R1, [R0]
                  ANDS    R2, R1, #0x01

          ;SwitchToggle if PF4 is 0
                  BEQ SwitchOn
                  BL Toggle
                  B loop
SwitchOn
```

```
                BL TurnOn

                B    loop


;------------Debug_Init------------
; Initializes the debugging instrument
; Input: none
; Output: none
; Modifies: none
; Note: push/pop an even number of registers so C compiler is happy
Debug_Init
                PUSH{R0-R3}
        ;Array Init
                LDR R0, =DataBuffer
                LDR R1, =TimeBuffer
                LDR R2, =DataPt
                LDR R3, =TimePt
                STR R0, [R2] ; We initialize our pointers
                STR R1, [R3]
                LDR R2, =0xFFFFFFFF
                MOV R3, #50
Erase   STR R2, [R0] ;We write xFFF... in both of our arrays
                STR R2, [R1]
                ADD R1, #4
                ADD R0, #4
                SUBS R3, #1
                BNE Erase
                LDR R0, =DataBuffer
                LDR R1, =TimeBuffer
                LDR R2, =DataPt
                LDR R3, =TimePt
                STR R0, [R2]
                STR R1, [R3]

        ;Systick Init
                LDR R0, =NVIC_ST_CTRL_R
                MOV R1, #0
                STR R1, [R0]

                LDR R0, =NVIC_ST_RELOAD_R
                LDR R1, =0x00FFFFFF
```

```
                STR R1, [R0]

                LDR R0, =NVIC_ST_CURRENT_R
                STR R1, [R0]

                LDR R0, =NVIC_ST_CTRL_R
                MOV R1, #0x05
                STR R1, [R0]

                POP{R0-R3}
                BX LR

;------------Debug_Capture------------
; Dump Port E and time into buffers
; Input: none
; Output: none
; Modifies: none
; Note: push/pop an even number of registers so C compiler is happy
Debug_Capture
                PUSH{R0-R7}
                ;Check if array is full
                LDR R0, =DataPt
                LDR R1, [R0]
                LDR R2, [R1]
                LDR R3, =0xFFFFFFFF
                SUBS R2, R3
                BNE Leave

                ;Dump - Get Systick Data
                LDR R0, =NVIC_ST_CURRENT_R
                LDR R1, [R0]

                ;Get PortE Data and Mask
                LDR R0, =GPIO_PORTE_DATA_R
                LDR R2, [R0]
                ADD R3, R2, #0
                AND R2, #0x01
                LSL R2, #4
                AND R3, #0x02
                LSR R3, #1
                ADD R2, R2, R3
```

```
                ;Store Data
                LDR R0, =DataPt
                LDR R7, =TimePt
                LDR R5, [R0]
                LDR R6, [R7]
                STR R2, [R5]
                STR R1, [R6]

                ;Increment pointers
                ADD R5, #4
                ADD R6, #4
                STR R5, [R0]
                STR R6, [R7]

                POP{R0-R7}
Leave   BX LR

;**********Initialization Sequence**********
PortInitE
;Initialize the Clock for Port E
        LDR R1, =SYSCTL_RCGCGPIO_R  ;Load the clock location
        LDR R0, [R1]                ;Load the clock setting
        ORR R0, R0, #0x10           ;Activate Port E
        STR R0, [R1]

;Wait two cycles
        NOP
        NOP

;Set inputs and outpurs in the direction register
        LDR R1, =GPIO_PORTE_DIR_R   ;Load the DIR location
        LDR R0, [R1]                        ;Load the DIR setting for PortE                 ;
        MOV R0, #0x02                       ;Set PE1 as output
        STR R0, [R1]                        ;Store the DIR setting


;Disable alternate functions
        LDR R1, =GPIO_PORTE_AFSEL_R;Load AFSEL location
        LDR R0, [R1]
        BIC R0, R0, #0xFF                   ;Disable the functions
        STR R0, [R1]
```

Eduardo Zueck Garces and Mai Phan

```
;Enable the digital port
        LDR R1, =GPIO_PORTE_DEN_R
        LDR R0, [R1]
        MOV R0, #0xFF
        STR R0, [R1]

;Go back
        BX LR

;**********Initialization Sequence PortF***********
PortInitF
;Initialize the Clock for Port F
        LDR R1, =SYSCTL_RCGCGPIO_R  ;Load the clock location
        LDR R0, [R1]                ;Load the clock setting
        ORR R0, R0, #0x20           ;Activate Port E
        STR R0, [R1]

;Wait two cycles
        NOP
        NOP

;Set inputs and outpurs in the direction register
        LDR R1, =GPIO_PORTF_DIR_R   ;Load the DIR location
        LDR R0, [R1]                    ;Load the DIR setting for PortE                 ;
        MOV R0, #0x04                   ;Set PF2as output
        STR R0, [R1]                    ;Store the DIR setting


;Disable alternate functions
        LDR R1, =GPIO_PORTF_AFSEL_R;Load AFSEL location
        LDR R0, [R1]
        BIC R0, R0, #0xFF               ;Disable the functions
        STR R0, [R1]

;Enable the digital port
        LDR R1, =GPIO_PORTF_DEN_R
        LDR R0, [R1]
        MOV R0, #0xFF
        STR R0, [R1]

;Go back
        BX LR
```

```
;************Delay Subroutine for 8HZ ***************
Delay
        MOV R0, #20000 ;This gives a 1 ms delay
        MOV R1, #62 ;Number of ms we want
        MUL R0, R0, R1
Repeat
        SUBS R0, R0, #1
        BNE Repeat
        BX LR

;************Turn on the switch*************
TurnOn
        LDR R0, =GPIO_PORTE_DATA_R
        LDR R1, [R0]
        ORR R1, R1, #0x02
        STR R1, [R0]
        BX LR

;************Toggle the switch************
Toggle
        LDR R0, =GPIO_PORTE_DATA_R
        LDR R1, [R0]
        EOR R1, R1, #0x02
        STR R1, [R0]
        BX LR

;************Heartbeat************
Heartbeat
        LDR R0, =GPIO_PORTF_DATA_R
        LDR R1, [R0]
        EOR R1, R1, #0x04
        STR R1, [R0]
        BX LR


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

Eduardo Zueck Garces and Mai Phan

Estimation of Execution Time

Time in Debug = 0.00002455 – 0.00002374 = 0.00000081
Time in Loop = 0.06202510 – 0.00002455 = 0.06200055

Intrusiveness: 0.0013%

This means this debugging method is barely intrusive in our program.

Debugging Data

00000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
01000000010000001000000011000000
10000000110000001000000011000000
10000000110000001000000011000000
10000000110000001000000011000000
10000000110000000100000001000000
01000000010000000100000001000000
01000000010000000100000001000000
0100000001000000DCFFFF006F50B400
02A1680095F11C002842D100BB928500
4EE33900E133EE007484A20007D55600
9A250B002D76BF00C0C6730053172800
E667DC0079B890000C0945009F59F900
32AAAD00C5FA6100584B1600EB9BCA00
80EC7E00153D3300AA8DE7003FDE9B00
D42E5000697F0400FECFB80093206D00
28712100BDC1D50052128A00E7623E00
7CB3F2001104A700A6545B003BA50F00
CEF5C30061467800F4962C0087E7E000
1A389500AD88490040D9FD00D329B200
667A6600F9CA1A008C1BCF001F6C8300

00BF7631→ 0073C6C6
12547633→7587526
4960107 *12.5 nanoseconds = 6.20013375 * E7 nanoseconds = 62.0013375 milliseconds