

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

; Program written by: \*\*\*Elvin Galarza and Ankith Kandikonda\*\*\*

; Date Created: 1/22/2016

; Last Modified: 1/22/2016

; Section \*\*\*Tuesday 3-4\*\*\*

; Instructor: \*\*\*Vijay Janapa Reddi\*\*\*

; Lab number: 4

; Brief description of the program

; If the switch is presses, the LED toggles at 8 Hz

; Hardware connections

; PE1 is switch input (1 means pressed, 0 means not pressed)

; PE0 is LED output (1 activates external LED on protoboard)

;Overall functionality of this system is the similar to Lab 3, with three changes:

;1- initialize SysTick with RELOAD 0x00FFFFFF

;2- add a heartbeat to PF2 that toggles every time through loop

;3- add debugging dump of input, output, and time

; Operation

; 1) Make PE0 an output and make PE1 an input.

; 2) The system starts with the LED on (make PE0 =1).

; 3) Wait about 62 ms

; 4) If the switch is pressed (PE1 is 1), then toggle the LED once, else turn the LED on.

; 5) Steps 3 and 4 are repeated over and over

datacmp EQU 0x200000E8

TIMER EQU 1653333

SWITCH EQU 0x40024008 ;PE0

LED EQU 0x40024004 ;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

; initialize Port E

LDR R1, =SYSCTL\_RCGCGPIO\_R

LDR R0, [R1]

ORR R0, R0, #0x10

STR R0, [R1]

NOP

NOP

LDR R1, =GPIO\_PORTE\_DEN\_R

LDR R0, [R1]

ORR R0, R0, #0x03

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DIR\_R ;LED is high

LDR R0, [R1]

ORR R0, R0, #0x01

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DIR\_R ;SW1 is low

LDR R0, [R1]

AND R0, R0, #0xFD

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_AFSEL\_R

LDR R0, [R1]

AND R0, R0, #0xFC

STR R0, [R1]

LDR R1, =GPIO\_PORTE\_DATA\_R ;makes PE0 on, originally

LDR R0, [R1]

ORR R0, R0, #0x01

STR R0, [R1]

; initialize Port F

LDR R1, =SYSCTL\_RCGCGPIO\_R

LDR R0, [R1]

ORR R0, R0, #0x20

STR R0, [R1]

NOP

NOP

LDR R1, =GPIO\_PORTF\_DEN\_R

LDR R0, [R1]

ORR R0, R0, #0x04

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_DIR\_R

LDR R0, [R1]

ORR R0, R0, #0x04

STR R0, [R1]

LDR R1, =GPIO\_PORTF\_AFSEL\_R

LDR R0, [R1]

AND R0, R0, #0xFB

STR R0, [R1]

; initialize debugging dump, including SysTick

CPSIE I ; TExaS voltmeter, scope runs on interrupts

BL Debug\_Init

loop

BL Debug\_Capture

;// HEARTBEAT = toggles onboard LED on/off to signify that the code is running

LDR R1, =GPIO\_PORTF\_DATA\_R

LDR R0, [R1]

EOR R0, R0, #0x04

STR R0, [R1]

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;|||||||| |||||||||

;|||||||| Lab 3 code |||||||||

;|||||||| |||||||||

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;|||||||||||||||||||||||||||||||||||||||||||||||||||||

;delay

;input PE1 test output PE0

BL delay

LDR R1, =GPIO\_PORTE\_DATA\_R

LDR R0, [R1]

LSLS R2, R0, #30 ;puts PE1 in most sig bit

BMI toggle ;if switch is pressed, go to toggle

LDR R1, =GPIO\_PORTE\_DATA\_R ;if not then fall through, keep LED on

LDR R0, [R1]

ORR R0, R0, #0x01

STR R0, [R1]

B loop

delay LDR R3, =TIMER

wait SUBS R3, R3, #0x01

BNE wait

BX LR

toggle EOR R0, R0, #0x01

STR R0, [R1]

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

; // R0-R3 (parameters)

; // THIS IS JUST CODE TO INITIALIZE THE DEBUGGING SEQUENCE

; // THE SAME CONCEPT APPLIES TO PUTTING THE REAL DATA IN

NoDataFF EQU 0xFFFFFFFF

; // Data Buffer Stack Initialization

LDR R0, =DataBuffer

LDR R1, =DataPt

LDR R2, =SIZE

LDR R3, =NoDataFF

STR R0, [R1]

DLoop

SUBS R2, #1 ;counts down, where SIZE = size of array (number of elements) = counter (8bytes)

BMI TArray ;go to 2nd Array (time array) when done with I/O array (data buffer)

STR R3, [R0] ;stores NoData into element space

ADD R0, R0, #4 ;goes to next valid space to store elements

B DLoop

;//Time Buffer Stack Initialization

TArray

LDR R0, =TimeBuffer

LDR R1, =TimePt

LDR R2, =SIZE

LDR R3, =NoDataFF

STR R0, [R1]

TLoop

SUBS R2, #1

BMI SysTick

STR R3, [R0]

ADD R0, R0, #4

B TLoop

;// SysTick Initialization

SysTick

IMPORT SysTick\_Init

MOV R2, R14 ;saves return address

BL SysTick\_Init ;changes Link Register value... need to save Link Register

;and load it back after calling SysTick

MOV R14, R2

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

LDR R2, =DataPt

LDR R1, =datacmp

LDR R0, [R2]

CMP R1, R0

BMI done

LDR R3, [R2]

LDR R1, =GPIO\_PORTE\_DATA\_R

LDR R0, [R1]

AND R0, R0, #0x03

LSLS R1, R0, #30 ; checking if shifting is needed

BPL next ; checks to see if switch is not pressed, if not pressed br next

ADD R0, R0, #0x02 ; shifting bit 1 to bit 5

ADD R0, R0, #0x04 ; 0000,00XY becomes..

ADD R0, R0, #0x08 ; 000X,000Y

next STR R0, [R3]

ADD R0, R3, #4 ; increment pointer \*\*\*

STR R0, [R2]

LDR R3, =TimePt

LDR R1, =NVIC\_ST\_CURRENT\_R

LDR R0, [R1]

LDR R2, [R3]

STR R0, [R2]

ADD R2, R2, #4

STR R2, [R3]

done BX LR

ALIGN ; make sure the end of this section is aligned

END ; end of file