

***** main.s *****

; Program written by: Yerraballi

; Date Created: 1/15/2018

; Last Modified: 1/15/2018

; Brief description of the program: Solution to Lab1

; The objective of this system is to implement a Car door signal system

; Hardware connections: Inputs are negative logic; output is positive logic

; PF0 is right-door input sensor (1 means door is open, 0 means door is closed)

; PF4 is left-door input sensor (1 means door is open, 0 means door is closed)

; PF2 is Safe (Blue) LED signal - ON when both doors are closed, otherwise OFF

; PF1 is Unsafe (Red) LED signal - ON when either (or both) doors are open, otherwise OFF

; The specific operation of this system

; Turn Unsafe LED signal ON if any or both doors are open, otherwise turn the Safe LED signal ON

; Only one of the two LEDs must be ON at any time.

; NOTE: Do not use any conditional branches in your solution.

; We want you to think of the solution in terms of logical and shift operations

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_LOCK_R EQU 0x40025520

GPIO_PORTF_CR_R EQU 0x40025524

GPIO_PORTF_AMSEL_R EQU 0x40025528

GPIO_PORTF_PCTL_R EQU 0x4002552C

GPIO_LOCK_KEY EQU 0x4C4F434B ; Unlocks the GPIO_CR register

SYSCCTL_RCGCGPIO_R EQU 0x400FE608

```

    Red    EQU 0x02
    Blue   EQU 0x04

    SW1     EQU 0x10
    SW2     EQU 0x01

    THUMB

    AREA    DATA, ALIGN=2

;global variables go here

    ALIGN

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

    EXPORT  Start

Start

    ; initialization code goes here

        BL    PortF_Init ;Function from InputOutputs example

        MOV R3, #2

        MOV R4, #0x01 ; Set R4 to x01

loop

    ; the body of the code goes here

        AND R1, R0, #0x01 ; if switch 1 is on R1= 0x01
        AND R2, R0, #0x10 ; if switch 2 is on R2= 0x10
        LSR R2, R2, #4 ; if switch 2 is on R2= 0x01
        ORR R1, R1, R2 ; if either switch is on R1=0x01
        EOR R1, R1, R4 ; Exclusive OR, if both have 1 in LSB R1=0x00
        MUL R1, R1, R3 ; if switch was off R1=2, otherwise 0
        ADD R1, R1, #2 ;add 2 to R2
        AND R0, R0, #0x00 ;clear contents of R0
        ORR R0, R0, R1

```

BL PortF_Output

B loop

; put your main engine here

;-----PortF_Init-----

; Initialize GPIO Port F for negative logic switches on PF0 and

; PF4 as the Launchpad is wired. Weak internal pull-up

; resistors are enabled, and the NMI functionality on PF0 is

; disabled. Make the RGB LED's pins outputs.

; Input: none

; Output: none

; Modifies: R0, R1, R2

PortF_Init

LDR R1, =SYSCTL_RCGCGPIO_R ; 1) activate clock for Port F

LDR R0, [R1]

ORR R0, R0, #0x20 ; set bit 5 to turn on clock

STR R0, [R1]

NOP

NOP ; allow time for clock to finish

LDR R1, =GPIO_PORTF_LOCK_R ; 2) 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 ; 3) disable analog functionality

```

MOV R0, #0          ; 0 means analog is off
STR R0, [R1]
LDR R1, =GPIO_PORTF_PCTL_R    ; 4) configure as GPIO
MOV R0, #0x00000000          ; 0 means configure Port F as GPIO
STR R0, [R1]
LDR R1, =GPIO_PORTF_DIR_R      ; 5) set direction register
MOV R0, #0x0E                 ; PF0 and PF7-4 input, PF3-1 output
STR R0, [R1]
LDR R1, =GPIO_PORTF_AFSEL_R    ; 6) regular port function
MOV R0, #0                    ; 0 means disable alternate function
STR R0, [R1]
LDR R1, =GPIO_PORTF_PUR_R      ; pull-up resistors for PF4,PF0
MOV R0, #0x11                 ; enable weak pull-up on PF0 and PF4
STR R0, [R1]
LDR R1, =GPIO_PORTF_DEN_R      ; 7) enable Port F digital port
MOV R0, #0xFF                 ; 1 means enable digital I/O
STR R0, [R1]
BX LR

```

;-----PortF_Input-----

; Read and return the status of the switches.

; Input: none

; Output: R0 0x01 if only Switch 1 is pressed

; R0 0x10 if only Switch 2 is pressed

; R0 0x00 if both switches are pressed

; R0 0x11 if no switches are pressed

; Modifies: R1

PortF_Input

LDR R1, =GPIO_PORTF_DATA_R ; pointer to Port F data

LDR R0, [R1] ; read all of Port F

AND R0,R0,#0x11 ; just the input pins PF0 and PF4

BX LR ; return R0 with inputs

;-----PortF_Output-----

; Set the output state of PF3-1.

; Input: R0 new state of PF

; Output: none

; Modifies: R1

PortF_Output

LDR R1, =GPIO_PORTF_DATA_R ; pointer to Port F data

STR R0, [R1] ; write to PF3-1

BX LR

ALIGN ; make sure the end of this section is aligned

END ; end of file

Flowchart Lab 1



10000
0010
16
2

1001
0001

1110

Start

Port F initialization

Mov R4 0x01
Mov R3 #12

Port F input

AND R1, R0, #0x01
AND R2, R0, #0x00
LSR R2, R2, #4
ORR R1, R1, R2
EOR R1, R1, R4
MUL R1, R1, R3
ADD R1, R1, #1
AND R0, R0, #0x00
ORR R0, R0, R1

loop

B loop

Port F Output

