Lab 5: External Interrupts
Key wakeup –interrupt mode -flag polling mode
ECE3731

INTRODUCTION

Often an external input may trigger a certain event. Your Dragon12 board supports external interrupts on PTH (Port H), which conveniently has switches connected to it. There are two types of switches connected to Port H (as you already know from the previous lab).

DIP (dual-inline pin) switches – 8 switches that can set corresponding bits high (up) or low (down).

Pushbutton switches – 4 switches on the 4 lower bit positions. These switches are termed "active low". (The DIP switches on the lower four-bit positions are set high (up) to avoid interfering with the pushbutton switches.)

Pressing a pushbutton switch causes a high to low logic transition, which causes an interrupt flag bit to be set since the switches are on Port H, which supports a type of external interrupt called "key-wake-up". A flag bit can cause an interrupt, if enabled to do so.

Note:
In this lab you are required to create to different projects, one for part(a) and the other for part (b)

PRE-LAB
1) Write an assembly program which flashes Blue LED 10 times with on time of 0.25 seconds and off time of 0.25 seconds
2) Write a program that decrements 4-bit counter on PORTB. When a value of zero is reached and the counter is decremented, it becomes 1111.

ASSIGNMENT

To be done in Assembly

PART A: -interrupt driven interface
This program is to have a main program loop and one interrupt service routine.

(i) Main Program Loop:

The main program also has code or calls code to initialize an interrupt on PORT H.

The main Program constantly calls a Subroutine that flashes the GREEN LED <mark>10 times</mark> on the RGB display at a 2Hz rate as a type of "heartbeat" display.
(This means on for 250ms and off for 250 ms.)
(You do not need to use the timer subsystem to create the 250 ms delay.)
This is done in an endless loop.

(ii) Interrupt Service Routine:
The interrupt service routine responds to an external interrupt on PORT H (PTH).
Specifically, your program is written so that any one of the pushbutton switches PH0, PH1 and PH2 can cause an interrupt. (There is one interrupt service routine for PTH which handles all of these switches.)
The interrupt service routine determines which switch caused the interrupt, *__by testing IF flag bits, not the switches__* and performs the corresponding action for the switch.
The Actions are as follows:

PH0:
- The value of all bits on Port H are read.
- The four leftmost bits as set by the DIP switch bank, represent a 4-bit counter value for this program.
- PH0 causes the four leftmost bits on PTH to be written to the four rightmost bits of PORTB (LED's). (PH7-4 are written to PORTB3-0 resp.) The upper 4-bits of PORTB will always be cleared.
- A message is displayed on LCD that SW0 is pressed.

PH1:
- The value of the 4-bit counter displayed on PORTB (LED's) is incremented by one.
- When a value of four ones is reached (1111), the counter rolls over to zero (0000).
- A message is displayed on LCD that SW1 is pressed.
- Blue LED flashes 2 times

PH2:

- The value of the 4-bit counter displayed on PORTB (LED's) is decremented by one.
- When a value of zero is reached and the counter is decremented, it becomes 1111.
- A message is displayed on LCD that SW1 is pressed.
- Red LED flashes 2 times

HINT:

To set up the Port H interrupt vector add the following lines in your Interrupt Vectors section in your project.

```
;***********************************************************
;*              Interrupt Vectors               *
;***********************************************************
        ORG   Vreset
        DC.W  Entry       ; Reset Vector

        ORG    Vporth     ; setup  Port H interrupt Vector
        DC.W   PTHISR
; (This assumes your interrupt service routine is name PTHISR)
```

PARTB: -flag polling interface –no interrupts

You have to implement the same functionality as in Part (a) but you are not allowed to use interrupts, but instead *polls the interrupt flag bits*.

This program is a polling flag driven interface between the DIP switch bank and pushbuttons on Port H and the 9S12 CPU. Port H bits 7-4 (upper 4-bits) represent the data inputs. Port H bit 0 (PH0) going high to low sets an interrupt flag. This flag is what is referred to as a Strobe signal indicating a command to input the 4-bit data value from PTH 7-4 as currently set by the DIP switches.

Questions for Discussion:

1. In PARTA, to what extent does pressing a pushbutton affect the blinking GREEN light? Is it possible to stop the light from blinking by holding down (not just pressing) a pushbutton switch? Why or why not?

2. What type of logic transition is created by pressing a pushbutton on Port H? How can you configure the interrupt flag bit for the pushbutton to be set by this transition? (Give example lines of code).

3. What happens if pushbutton interrupt flag bits are not cleared before exiting the Interrupt service routine?

4. Is it possible to **set** the IF (interrupt flag bits) for Port H under software control?
(If you are claiming yes, give example code.)

5. Give flowcharts or pseudo-code for your programs