Lab 2 – Assembly Programming 2 (Debouncing)

Short Description: Turnstile Counter
Difficulty Level: 3/10
Grading: 20 points

This assignment is more difficult than the last, a pattern we will only vary from once this semester. In this assignment you are asked to model a turnstile counter. Each time the button is pressed (a person walks through), you are to increment the count (number of people that have gone through).

Below is a picture of what happens when a mechanical switch is pressed. As you can see there is some noise. Imagine what would happen if you were trying to count the number of times that a sensor (button) was triggered and this happened. What would happen? The count would be incorrect (especially with fast hardware).

So, how do we fix this problem? There are two ways, only one of which you will really try in this course. We will use a software delay method to remove the bouncing of the switch. To do this we will need to come up with a way to create a delay within our code. Below is pseudo code example that describes one method of debouncing.

Loop:
Read I/O Port()
If SW == 0 goto Loop
Delay()
Read I/O Port()
If SW == 0 goto Loop
Else proceed now that SW==1
Well this is great, but how do we implement the delay? We need to write a delay function! So how do we do that? Let us use index registers X and Y as our delay variables (that gives us the ability to use up to 16 bits instead of only 8 bits) since these things are very fast. X is the upper byte and Y is the lower byte. Below is a pseudo code example of how to implement a delay.

Delay:
Load X with ##
Load Y with ##

If Y>0
    Decrement Y
Else
    Decrement X
    Y = 0xFF
End if

If X==0 and Y==0
    Return

For this assignment you will need to implement a counter that increments the count when SWX is pressed and clears the count when SWY is pressed. The count is to be displayed via the four LEDs that were used in the last Lab. You get to choose the two switches that increment and clear (these should be chosen from the switches used in the previous lab). Be sure to use the debouncing method covered so that you do not add to your count when you do not need to.

QUESTIONS TO BE ANSWERED IN COMMENTS AT THE END OF YOUR CODE:

1) Why is debouncing important?
2) How can I debounce in hardware?
3) What other electro-mechanical device might “bounce”? 
4) List the number of cycles it takes for the opcodes used in your delay function.