Lab Time_____   Name_____

Lab 2. Conditional Branches and Creating Loops in Arm Assembly

Preparation

You will need a LaunchPad and a laptop/computer with Keil uVision5 installed. Download the Lab2 starter project from Canvas.

Starter project   Lab2

Purpose

The purpose of this lab is for you to become familiar with using conditional branches in ARM assembly by creating looping structures. You will learn about how to use loops to create software delays.

Useful References

Section 3.3.1 of the *ARM Cortex-M4 Process Technical Reference Manual* (available online) will help with Part 1 as it includes instruction cycle counts.

Procedure

1. The provided code simply blinks the Red LED at an indeterminate rate by making calls to the function "delay_dim" whose code can be found in the file "delay.s". Knowing that the clock speed is 16 MHz, calculate how long the delay function is. To do this determine how many times each instruction is executed within "delay_dim" and how many clock cycles each instruction takes to execute. Use the oscilloscope to verify that your calculations are correct. You can assume that the Red LED will be on or off for the length of the delay. Find and correct your mistake if your calculations are not correct, because the next part of the lab will require you to design your own delay function. Demonstrate to the TA your correct calculations and oscilloscope output.

Signature_____ Date_____

2. Next you will create your own delay function "delay" to replace calls to **"delay_dim"** in Part 1. A placeholder for this function can be found in "delay.s" just after the "delay_dim" function. Be sure to choose a delay large enough that the LED will toggle at a viewable rate. Demonstrate to the TA how you calculated the value used in your delay function and use the oscilloscope to prove its timing.

Signature_____ Date_____

3. At the start of your main loop, uncomment the instruction "B parts34". This will allow the program to skip the code from the previous two parts. This new program will now indefinitely repeat a blinking routine that cycles through each of the LEDs with a blink rate determined by the delay function you wrote for Part 2. Rewrite the provided program to blink the Red LED four times, the Blue LED three times, the Green LED two times. **Use looping structures to achieve this (do not repeat code segments)**. Note that R0 and R1 are used extensively in the LED blinking functions. Be sure to use other CPU registers to track your loop iterations. Demonstrate your new blinking pattern to the TA.

Signature___Cole Farrell_____ Date 2/11/20

4. Now expand your code from the previous part to blink the Red LED four times, the Blue LED three times, the Green LED two times, the Blue LED three time, the Green LED two times, the Blue LED three times, and the Green LED two times (i.e. the blue then green routines will run three times after the red routine). Use nested loops instead of merely copy and pasting the Blue and Green LED code segments from the previous part. Demonstrate your new blinking pattern to the TA.

Signature___Matthew Russell_____ Date 2/15/2020