EE 3171 Lab 1 Submission

Code Entry and Debugging

1 Submission Requirements

Answer every question. Make sure that if you use any resource not provided to you by the TA, instructor or in the textbook you properly cite it. All demonstration requirements have a matrix to be filled out by the TA. This will allow partial credit to be given and for it to be clear how things are graded. Any demonstration required, you need to ask a TA to grade it. Only authorized individuals can grade your demonstrations. You **must** submit your code for all programs. If code is not submitted then you will lose the demonstration points.

2 Code Entry (50 points)

1. After the initial build of the unedited code, indicate how you fixed:
    - (10) The first easy error

      add a space before the instruction ADD

    - (10) The warning on/around Line 28

      add # in front of the #

    - (15) The "terrifying error" (include all the lines that reference k)

      6 fix: add = in front of k

      30

      43

      44

2. (5) If you encountered any other errors or warnings, list them here.

   n/a

3. (5) The initial values for `j` and `k` are both assigned in the section of code labeled `SystemInit`. What are those initial values?

   j = 5
   k = 0x3

4. (5) The three variables are all assigned using different bases and specifications. Did this have any effect on the way the code worked? Why or why not?

   no, the compiler sees them all as the same value anyway.

3 Simulation (100 points)

1. (50) Demonstrate to the TA that your program successfully builds and runs.
   TA Initials: JDR
   Date: 01/29

2. (5) What is the value of the `PC` when the debug session is started?

   0x0000026C

3. (5) Which of the two files does your project start in?

   startup

4. (5) What is the value of the `PC` and what is the instruction that causes the value of `theAnswer` to change in the Watch Window?

   0x 3A0
   STR R5, [R6]

5. (5) What is the value of R4 that finally causes the loop to exit in the initial version of the code?

   0x3

6. (5) In the initial version of the code, how many times is the code on line 14 executed (the `ADD R4, R4, #1` line)?

   3 times

7. (5) What was the final value for `theAnswer` in the initial version of the code?

   0xF

8. (20) Based on your reading of the code (and other values you used to test your hypothesis), what does this program do?

   in short: theAnswer = j × K

   technically j is being added K times, with the value being temporarily stored in R5. In the end the value of R5 is stored into R6 (theAnswer). K is used as a loop counter, set to the number of times addition occurs. Every loop there is a comparison to check if the proper number of loops happened.

   After all, the program spends its time in an infinite loop on line 21; branching to itself forever.