ELEC 5260/6260 Homework #4
Functions and Subroutines

PROGRAM 1:

To practice writing and using subroutines, write a subroutine that implements the function \( Z = (a_0 \times x_0) - (a_1 \times x_1) \), and a main routine that calls this subroutine.

The values of the four variables \( a_0, a_1, x_0, x_1 \) are to be passed to the subroutine in registers. The address of variable \( Z \) is to be passed to the subroutine in a register.

The “main” program is to call the subroutine two times, once for each of the following sets of data. You may not use “immediate” data – all values are to be read from memory and results are to be stored in memory. The data formats are to be as follows.

- \( x_0 \) and \( x_1 \) are to be 32-bit integers.
- \( a_0, a_1, \) and \( z \) are to be stored using Q24.8 format, i.e. 24 integer bits and 8 fraction bits, stored in a 32-bit word as follows: \( N_{23}N_{22}N_{21}... N_1N_0 . N_{-1}N_{-2} ... N_{-7}N_{-8} \)

Qn.m format is described on the ARM web page:

For the first call: \( x_0=200, x_1=100, a_0=5.25, a_1=6.75. \)
Store the answer at variable \( z_1. \)

For the second call: \( x_0=300, x_1=200, a_0=3.5, a_1=4.125. \)
Store the answer at variable \( z_2. \)

Define all ten variables in a DATA section, and use a debug initialization file to set the eight “arguments” to the test values indicated above.

You may test this with the simulator, but use of the Discovery board is recommended. Submit a printout of your source program, and a screen image of the Debug window, with the results displayed in a Watch window and/or highlighted (circled) in a Memory window.

(Program 2 on next page.)
PROGRAM 2:

Implement the two functions of Program 1 in the C language. Note that the startup code must be used in this project, even if you did not use it for Program 1.

- Define all data as global variables of type int (integer). However, as in the first program, values of a0, a1, z1, and z2 should all be in Q24.8 format, thereby allowing integer operations to be used.

- The function to compute \( Z = (a0 \times x0) - (a1 \times x1) \) should be passed values of the four variables a0, a1, x0, x1. However, instead of passing the address of Z, have the function return the result, to be stored/used by the calling program.

- The main program should call the function at least twice, using the data from Program 1.

Execute the program, and then submit a printout of the program and a screen capture of a Watch window from the debugger, showing all 10 data values.

PART 3:

Answer the following questions.

1. How did the assembly language generated by the C compiler differ from your implementation in Program 1?

2. In the assembly language generated by the C compiler, how did the compiler choose to pass parameters to and from the function that computes Z. (Did it use registers and/or stack?)

3. Describe how the use of the DSP discussed in class might affect your solution to Program 1.