ECE353 In Class Exercise

ARM Assembly – EABI

Problem 4B

In this exercise, you will write a function in ARM assembly that is compliant with ARM Procedure Call Standard. This is also referred to as the EABI (Embedded Application Binary Interface).

You will write a function that accepts five parameters. The first four parameters contain the addresses of four arrays of 8-bit signed numbers. The 5th parameter holds the number of elements in the arrays. You can assume that all of the arrays are of the same size.

The C declaration of the function is shown below.

```
int32_t addMultiply(
      int8_t *a0,                 // Address of array a0
      int8_t *a1,                 // Address of array a1
      int8_t *a2,                 // Address of array a2
      int8_t *a3,                 // Address of array a3
      int32_t array_size          // Array Size
);
```

Validate that none of the array addresses is equal to 0 (NULL) and that array_size is greater than or equal to 0. If all the parameters are valid, return 0. If any of the parameters is invalid, return -1.

For each element in the array, the following calculation is carried out.

```
; a3[i] = (a0[i] + a1[i]) * a2[i]
```

There has been a test bench provided for you in main.c. This test bench is written in C and will call your EABI compliant assembly routine. You do not need to modify anything in main.c.

Write an EABI compliant ARM assembly function for addMultiply. When you are done, load the project on the Launchpad and enter the debugger. Set a break point at line 49. If your program reaches this break point, then you function was correctly implemented. **Remember, the location of the parameters is defined by the EABI.**