EC605: Computer Engineering Fundamentals Fall 2021
Lab 4: ARM Assembly

NOTE: **This lab is to be completed individually, and not with your lab partner.**

Goals

- Introduction to the ARM Development studio 5 (DS-5) design and test environment.
- Simple programming in LEGv8, a subset of ARMv8 assembly.

Overview

In this lab you will compile and debug a working C & assembly project, while implementing simple functions in LEGv8 assembly.

Deliverables

Place your modified assembly files in a zipped folder with your last name in the format Lab4_(BUEmailID).zip, and submit on Blackboard. (If you submit the (.s) files directly, TAs will not be able to download the file and therefore will not be able to give you a grade.)

There is no demo for this lab and submissions are individual.

Tasks

Task 0: DS-5 Design Environment Tutorial

Before you start to work on this lab, follow the provided tutorial to running DS-5 in a virtual machine in the computer lab.

Provided C – Code for Tasks 1 - 3: Lab4_main.c

```
#include <stdio.h>

/* Declare the assembly function */
extern void Fib_array(long *array);
extern void array_swap(long *A, long *Y);
extern void array_even_odd(long *A);

int main()
{
  long array[32];              /* Part 1 - array will contain the specified initial sequence */
  long swap[32];    /* Part 2 - array will have swapped every other pair of indices */
  long odd[32];     /* Part 3 - array will swap the content of odd indices */

  int i;

  /*Part 1 fib array*/
  Fib_array(array);
```

```
    for (i = 0; i < 32; i++)
    {
       printf("%ld, ", array[i]);
    }
    printf("\n\n");


    /* Uncomment the following line to call Part 2 assembly function*/
    /*array_swap(array, swap);*/

    /* Uncomment the following lines to print out Part 2 array */
     /*for (i = 0; i < 32; i++)
     {
       printf("%ld, ", swap[i]);
     }
     printf("\n\n");*/

    /* Uncomment the following lines to call Part 3 assembly function*/
    /*array_swap(array, odd);
    array_even_odd(odd);*/

    /* Uncomment the following lines to print out Part 3 array */
    /*for (i = 0; i < 32; i++)
    {
       printf("%ld, ", odd[i]);
    }
    printf("\n");*/

     return (0);
   }
```

Task 1: Array Initialization

1. The provided *Lab4_main.c,* listed above, calls the *Fib_array* function to initialize an array, and then prints the content of the array.

   Extend the provided Fib_array.s file to initialize the array to contain the first 32 elements of the Fibonacci sequence. The Fibonacci sequence is defined as follows:
   Fib[0] = 0; Fib[1] = 1; Fib [i] = Fib[i-1] + Fib[i-2] for i>1. The function receives its input, array, in register x0. You do not have to do anything to initialize the array to register x0, just assume it's there in your code.

   Your code should utilize a loop, and consist of instructions from the green card, along with immediate instructions. Note that ARMv8 does not actually support unique immediate instructions. Instead of: *addi x1, x2, 0x100*, use *add x1, x2, 0x100*. And remember: an integer in ARM-legv8 is 64-bits or 8 bytes, like in the example gone over in class.

2. Follow the *DS-5 Tutorial* to test your code on the DS-5 in debug mode.

Task 2: Array Swap

1. The provided *Lab4_main.c*, includes a commented out *array_swap* function. This function receives two arrays of size 32 as inputs: one source array and one destination array. It copies the data from the source array to the destination but reverses the order of elements, i.e., array[31] will contain the previous content of array[0]. Uncomment this function, and implement it in array_swap.s.

2. Test your code on the DS-5.

Task 3: Even-Odd Manipulation

1. The provided *Lab4_main.c*, includes a commented out *array_even_odd_flip* function. This function receives the reversed order array as input and performs in-place operations on it. It performs a single bit left shift of the content of all even indices, and a single bit right shift of the content of all odd indices. Uncomment this function, and implement it in array_even_odd.s.

2. Test your code on the DS-5.