ESET 369 LAB 2 REPORT

Student name	Bradley Brewington, Kyle Rex, and Andew Sarabia
Group number	34
Lab Section	501
Lab session date	February 3, 2025
Lab instructor	Xin Zhuang

INTRODUCTION

The purpose of this lab is to utilize the ARM Assembly coding language with the code composer studio to control LED outputs on the BH boards and to perform basic mathematical operations with the MSP430FR5994 microcontroller. With the proper software configuration on the launchpad the mathematical equations should be solved and with the proper physical configuration of the BH board the LED simulated outputs should work as well. Doing this will result in the user becoming more familiar with launchpad GPIO configuration as well as the configuration of the BH board. System A involved performing basic math operations and System B involved creating a two LED blinking sequence controlled by a button.

SYSTEM A

The purpose of System A is to correctly configure the launchpad to perform mathematical operations using word operations in assembly code. The first operation is an adder operation (R11⟵14+32), the second, a subtraction operation (R12⟵32-14), the third, another subtraction operation (R13⟵14-32), and finally, a summation operation (Equation 1) which requires the implementation of a loop in the assembly code.

$$R14 \longleftarrow \left(\sum_{k=2}^{21}(2k+1)\right) \qquad \textbf{Equation 1}$$

The first step in solving this problem is building the assembly code itself to solve these operations. The ARM assembly code that was built to address this problem performs two main tasks: basic arithmetic operations and a summation calculation. First, it defines values in registers and performs addition and subtraction operations. It initializes R11 with 14 and adds 32, storing the result (46). Similarly, it initializes R13 with 14 and subtracts 32, storing -18, and sets R12 to 32, then subtracts 14, storing 18. The second part of the code calculates the summation of the expression 2k+1 for k ranging from 2 to 21. It initializes R4 (k) to 2, R14 (sum) to 0, and R5 to 22 (loop end value). In each iteration, it calculates 2k+1 by doubling R6 (which holds R4's value) and adding 1, then adds this result to R14. R4 increments in each iteration, and the loop continues until R4 reaches 21. The final result stored in R14 is the summation of 2k+1 for values of k from 2 to 21. The result of this code sequence can be seen in Figure 1 where the final values for each register are displayed and correctly correlate with the expected final decimal values of the operations.

Table 1: Final Register Values

Register	Result (Hexadecimal)
R11	0x00002E

R12	0x000012
R13	0x00FFEE
R14	0x0001E0

SYSTEM B

The purpose of System B involves configuring the launchpad and board using assembly code to display a two LED (LED_A and LED_B) simultaneous blinking sequence indefinitely while a button is being pressed. When the button is released the blinking sequence of the LEDs should stop. The duration of the blinking sequence must be between 0.5 to 1 second.

The first step in addressing this problem is making the physical board connections between the BH board and the MSP430FR5994 microcontroller. This can be done by connecting three specific pins together using male-to-female jumper wires, checking that all the DIP switches on the BH board are in the correct configuration, and ensuring the microcontroller itself is properly seated on the BH board. Pin P3.1 controls LED_B, pin P3.0 controls LED_A, and pin P6.0 controls the "Right" button. The next step in solving this problem involves building the assembly code itself to blink the LEDs. The ARM assembly code that was built to address this problem starts by initializing the microcontroller, setting up GPIO pins, and running a loop to control LEDs based on a button press. It first sets the stack pointer and disables the watchdog timer. Then, it configures P3.0 and P3.1 as outputs for LEDs, ensuring they start off, and sets P6.0 as an input with an enabled pull-up resistor for button detection. The main loop continuously checks if the button is pressed. If pressed, a delay loop executes (ensuring the blinking falls between 0.5 and 1 second) before toggling the LEDs. P3OUT is toggled using a "XOR" operation with 0x03. If the button is not pressed an AND operation is used to make sure both the LEDs are off and it returns the sequence back to the first line of the main routine. The program loops indefinitely, monitoring the button and updating the LEDs accordingly. The loop decrements 0xF400 down to 0x0000 and R14 was used to store this initial value of 0xF400. Some of the code used to do this was adapted from the original code found in the book: *Learning Embedded Systems with MSP430 FRAM microcontrollers* by B. Hur. The result of this code sequence achieves the goal of the system which was to create a simultaneous blinking LED sequence.

CONCLUSION

This lab successfully demonstrated the use of ARM assembly language to perform mathematical operations and control LED outputs using the MSP430FR5994 microcontroller. System A focused on arithmetic operations and summation calculations, reinforcing concepts of register manipulation and loop implementation in assembly. The correct final values in the registers confirmed the expected results. System B implemented a two-LED blinking sequence controlled by a button press, showcasing GPIO configuration and conditional branching in assembly. The successful execution of both systems validates the accuracy of the implemented code and hardware setup. This lab provided a better understanding of how to use assembly programming, GPIO control, and debugging techniques, which are essential skills for embedded systems development.

REFERENCES

B. Hur, *Learning Embedded Systems with MSP430 FRAM microcontrollers*, 2nd ed., 2023.