SANTA CLARA UNIVERSITY
Electrical and Computer Engineering Department

ELEN 120 – Embedded Computing Systems

Lab 0 – The Keil IDE

Andrew Wolfe

Assignment: In this assignment, you will:

1. Experiment with the Keil IDE to observe its features and capabilities
2. Trace the execution of ARM assembly programs and modify assembly programs according to directions.

You are going to use the Keil IDE along with the ARM Cortex-M4 software simulator to step through and analyze some assembly programs.

Step 1: Preparing to use the Keil Simulator

1. Log into the lab machines using one student's ECC login (your SCU login)
2. Create an ELEN-120-Labs directory to keep your shared files. It is best to do this on your Z: drive. You may want to also create a shared Google Drive directory so that you can copy the completed labs somewhere more useful and accessible to both students. Inside of the directory, create subdirectories for Lab0, Lab1, … Lab8. Within those subdirectories, you will want to create another subdirectory for each program you write., e.g. Lab0p1 for problem 1.

Step 2: Using the Keil Simulator

Instructions for using the Keil simulator are in the handout *Software Simulation - Keil*. You should try them. They have been demoed in class.

Next – do these problems:

Problem 1

Create a new Keil project in a new directory and enter the following code into main.s. Note that the assembly directives must be indented.

```
                AREA    main, CODE, READONLY
                EXPORT    __main
                ENTRY

__main    PROC

          LDR  r0, =0x13
          LDR  r1, =0x07
          ADD  r2, r0, r1
          EOR  r3, r0, r1
endless   B    endless
               ENDP

               END
```

Answer the following questions using the *Disassembly* and *Registers* windows in the simulator:

1. What is the result in R0 after the following instruction is executed?
   ```
   LDR r0, =0x13
   ```

2. What is the result in R1 after the following instruction is executed?
   ```
   LDR  r1, =0x07
   ```

3. What is the result in R2 after the following instruction is executed?
   ```
   ADD  r2, r0, r1
   ```

4. What is the result in R2 after the following instruction is executed?
   ```
   EOR  r3, r0, r1
   ```

5. How many times will the following instruction execute?

   ```
   EOR  r3, r0, r1
   ```

6. How many times will the following instruction execute?

   ```
   B    endless
   ```

Problem 2

Create a new Keil project in a new directory and enter the following code into main.s

```
                AREA    main, CODE, READONLY
                EXPORT    __main
                ENTRY

__main  PROC

        EOR  r0, r0, r0
        MOV  r2, r0
        ADD  r1, r0, #1
        MOV  r3, r1
count   ADD  r2, r2, r1
        ADD  r3, r3, r3
        B    count
endless B    endless

                ENDP
                END
```

1. Step through this program to reach __main. (should be 2 steps from reset)
2. Step through your code another 100 steps.
3. What are the values in r0, r1, r2, and r3 after 100 steps?

Problem 3

Modify the code in Problem 2 to match this example:

```
                AREA    main, CODE, READONLY
                EXPORT    __main
                ENTRY

__main  PROC

        EOR  r0, r0, r0
        MOV  r2, r0
        ADD  r1, r0, #1
        MOV  r3, r1
count   ADD  r2, r2, r1
        ROR  R3, #31
        B    count
endless B    endless

                ENDP
                END
```

1. Step through this program to reach __main. (should be 2 steps from reset)
2. Step through your code another 100 steps.
3. What are the values in r0, r1, r2, and r3 after 100 steps?
4. What happens differently in this program than in Problem 2?