Experiment 1: Keil uVision, ARM Assembly, Debugging

Objectives:

- To gain experience with the ARM Assembly and the TI TIVA C Launchpad Development Board.
- To gain experience the Keil ARM development environment and debugging features.

Preparation:

Install Keil uVision 4 or 5 on your laptop or computer.

Introduction:

In this experiment you will be compiling and running a simple LED blinking program on the TIVA C Launchpad development board. You will learn to manage projects in Keil uVision, how to build programs, debug running applications and use various features of the Keil IDE.

Experiment:

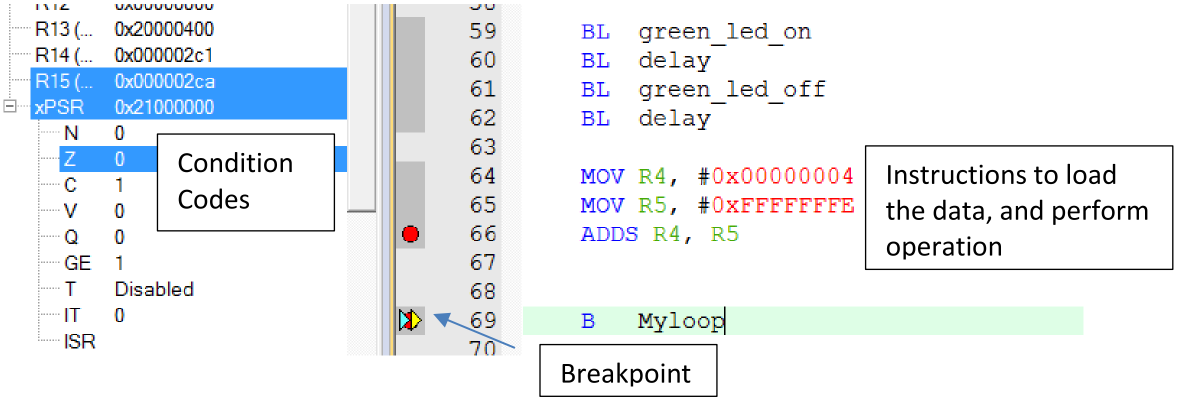
1. First, download the project in Lab1.zip and unzip it. Next, using Keil, Build and load this program on the Launchpad evaluation board and verify its correct operation. Confirm that blinky is running and experiment with various blink patterns.

2. Rewrite the program to blink the Blue LED once, the Red LED twice, and the Green LED three times. Demonstrate this for the lab instructor. Use looping structures to achieve this (do not just repeat the code segments). Note, R0 and R1 are used extensively in the LED blinking functions, use the other CPU registers in your code segments to make sure we do not interfere with those functions. Demonstrate your new blinking pattern as a video recording.

3. Next, write additional code segments and use breakpoints to analyze their effect on the CPU registers and condition codes. Let X, Y, and Z, be three numbers as shown below. In your project, write assembly instructions to perform the operations listed in the table, record the Result of the listed addition or subtraction operations, and the states of the condition flags N, Z, V, and C in the PSR register**.** The figure below shows how this can be achieved in Keil.

X = 0x0000 0010, Y = 0xFFFF FFF4, Z = 0x7FFF FFFD

Operation	Result (R)	Condition Codes N Z V C
R = X + Y		
R = X – X		
R = Z + X		
R = Y - Z		
R = Y + Z		



```
R13 (...   0x20000400
R14 (...   0x000002c1
R15 (...   0x000002ca
xPSR       0x21000000
  N        0
  Z        0
  C        1
  V        0
  Q        0
  GE       1
  T        Disabled
  IT       0
  ISR
```

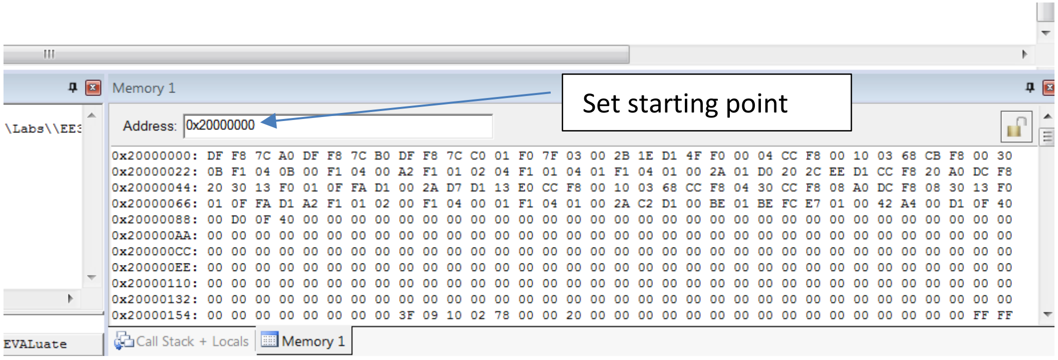
Condition Codes

```
        BL    green_led_on
        BL    delay
        BL    green_led_off
        BL    delay

        MOV R4, #0x00000004
        MOV R5, #0xFFFFFFFE
        ADDS R4, R5


        B    Myloop
```

Instructions to load the data, and perform operation

Breakpoint

4. Next, use the Memory Window to study the values in the Port F Data Register under various conditions. In the memory window, put in the address 0x400253FC (Port F Data Register).



Set starting point

Address: 0x20000000

```
0x20000000: DF F8 7C A0 DF F8 7C B0 DF F8 7C C0 01 F0 7F 03 00 2B 1E D1 4F F0 00 04 CC F8 00 10 03 68 CB F8 00 30
0x20000022: 0B F1 04 0B 00 F1 04 00 A2 F1 01 02 04 F1 01 04 01 F1 04 01 00 2A 01 D0 20 2C EE D1 CC F8 20 A0 DC F8
0x20000044: 20 30 13 F0 01 0F FA D1 00 2A D7 D1 13 E0 CC F8 00 10 03 68 CC F8 04 30 CC F8 08 A0 DC F8 08 30 13 F0
0x20000066: 01 0F FA D1 A2 F1 01 02 00 F1 04 00 01 F1 04 01 00 2A C2 D1 00 BE 01 BE FC E7 01 00 42 A4 00 D1 0F 40
0x20000088: 00 D0 0F 40 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200000AA: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200000CC: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x200000EE: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000110: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000132: 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00
0x20000154: 00 00 00 00 00 00 00 00 3F 09 10 02 78 00 00 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 FF FF
```

What is the value at the Port F Data Register under the following conditions?

- Only blue LED is on, no buttons pressed.
- Only red LED is on, no buttons pressed.
- All LEDs off, left button is pressed.
- All LEDs off, right button is pressed.

- All LEDS off, no buttons pressed.

5. Select and instance where a subroutine is called, such as “BL green_led_on” and study the effects of calling and returning from this subroutine on the Program Counter (PC) and Link Register (LR). You can do this by setting a break point at the subroutine call, note the values for the PC and LR, then “step into” the subroutine, and then note those two registers again. Explain the change in values. Next, in the subroutine step down to the “BX LR” instruction, and note the changes to the PC and LR instructions again as this instruction executes.

In your submission address the following:

- Show a video demonstration for the LED blinking pattern.
- In general, when are each of the N, V, C, Z flags set?
- Show the results table for the condition codes exercise. Describe how you retrieved the results. Show the software you have written for this part. For a few sample cases, explain why the flags were set the way they appeared in the debugger.
- Explain your observations from part 4. How is this happening?
- For part 5, describe the purpose of and how the PC and LR registers are affected when calling and returning from a function.
- Write a conclusion statement summarizing what you have done, problems you have faced, and what you have learned. Instructors will look for a meaningful conclusion, put some thought into it. It should represent the effort the experiment has taken, show your level of understanding, highlight what you have learned, challenges faced, and what lingering questions you have.