

Exploring Arm Architecture and Keil uVision

Objective

The objective of this lab is to explore Arm assembly language and gain hands-on experience with Keil uVision, an Integrated Development Environment (IDE) for Arm microcontrollers. You will write an Arm assembly language program and observe how different instructions affect the flags in the CPSR (Current Program Status Register).

Prerequisites

- A computer with Keil uVision installed.
- Basic knowledge of Arm architecture and assembly language.
- What is the purpose of **S** suffix to the mnemonic?

Lab Assignment

In this lab, you will write an Arm assembly program named **FlagEffects.s** and execute various instructions to observe the effects on the flags in the CPSR. Follow the instructions below and answer the questions as you proceed.

Writing the Assembly Program

1. Create a new file named **FlagEffects.s** within your Keil uVision project.
2. Write the following Arm assembly code in **FlagEffects.s**:

```
1. ; Lab_1: Arithmetic and Logical Operations in ARM Assembly
2.
3. ; Description:
4. ; This lab focuses on performing arithmetic and logical operations in
   ARM assembly using various instructions.
5. ; Students are required to load their last 5 digits of student IDs
   into registers, perform different operations, and observe the results
   and status flags.
6.
7. ; Export the __main label as the entry point
8.     EXPORT __main
9.     AREA flagEffect, CODE, READONLY
10.        ; Start of the main program
11.        __main
12.        ; Your assembly code goes here
```

```

13.
14.          ; Task 1: Load last 5 digits of your IDs into R0 and R1
15.          ; Hint: Use MOV or LDR instructions to load the values into
    registers.
16.
17.          ; Task 2: R2 = R0 + R1
18.          ; Observation 2: Display the values of R0, R1, R2, and xPSR
    (CPSR).
19.
20.          ; Task 3: R3 = R1 - R2 (use SUB)
21.          ; Observation 3: Display the values of R0, R1, R3, and xPSR
    (CPSR).
22.
23.          ; Task 4: R4 = R1 - R2 (use SUBS)
24.          ; Observation 4: Display the values of R0, R1, R4, and xPSR
    (CPSR).
25.
26.          ; Task 5: Compare Observation 3 and Observation 4.
27.
28.          ; Task 6: R6 equals to a bitwise OR operation between R0
    and R1
29.          ; Observation 6: Display the values of R0, R1, R6, and xPSR
    (CPSR).
30.
31.          ; Task 7: R8 equals to arithmetic shift left R0 by 2 bits
32.          ; Observation 7: Display the values of R0, R8, and xPSR
    (CPSR).
33.
34.
35.          ; End of the main program
36.
37.          ; End of program
38.      END

```

Running the Program and Observations

1. Open your project in Keil uVision5.
2. Include *startup_bare_minimum.s* file (you can download it from github).
3. Build the project to ensure there are no syntax errors.
4. Choose simulator.
5. Run the program step by step, observing the register values and flags in the CPSR after each instruction.

Submission

1. Take a screenshot of the register and flag values for each observation.
2. Save your FlagEffects.s file with the answers included. Ensure that your code file includes the answers of the comments.
3. Submit both the screenshot and the FlagEffects.s file as part of your lab assignment.