Microprocessor and Assembly Lab: Lab 3

Dr. Upama Kabir, Professor
Computer Science and Engineering, University of Dhaka

Jan 30, 2023

Contents

1 Objectives

The objectives of this lab is to understand and have familiarize with register based assembly programming for Cortex M4 processor for arithmetic operation.

2 Assessment Policy

The student must complete and demonstrate the lab in the laboratory during the lab hour. After completing the lab student must submit a report and upload the code to the Microprocessor and Microcontroller google classroom.

3 Example

```
AREA  Example, CODE, READONLY
ENTRY ; starting point of the code execution
EXPORT main ; declaration of identifier main

main; address of the main function
; user code starts from the next line
ADD r0,r1,r2;      ;add Q to R and put in P
ADD r0,r0,r3      ;add S to P and put the result in P

Stop B Stop
END
```

3.1 Notes

- The semicolon indicates a user-supplied comment. Anything following a semicolon on the same line is ignored by the assembler.
- The first line is AREA Example1, CODE, READONLY is an assembler directive and is required to set up the program. It is a feature of the development system and not the ARM assembly language. An assembler from a different company may have a different way of defining the start of a program. In this case, AREA refers to the segment of code, Example1 is the name we've given it, CODE indicates executable code rather than data, and READONLY state that it cannot be modified at run time.
- Anything starting in column 1 (in this case Stop) is a label that can be used to refer to that line.
- The instruction Stop B Stop means 'Branch to the line labelled Stop' and is used to create an infinite loop. This is a convenient way of ending programs in simple examples like these.
- The last line END is an assemble directive that tells the assembler there is not more code to follow. It ends the program.

4 Creating a Project in keil-IDE

- Click on the Project/New µVision Project.
- Create a new folder and provide the desired project name and click the save button in the window .
- As soon as you click on Save, the next window opens. There you need to select your device which is STM32F446RE
- Then the Manage Run-Time Environment window will open.
  1. Expand CMSIS and select Core.
  2. Expand Device and select Startup
  3. Select 32F469IDISCOVERY as the Board Support

- You now have a new project
- You have created a blank µVision project using MDK 5 Software Packs. All we need to do now is add our own source files.
- Right-click on Source Group 1 in the Project window and select Add New Item to Group 'Source Group 1'
- Highlight the Asm file (.s). And enter your source file's name.
- Click on Add to close this window.

5 Debugging

Use the KEIL debugger for the debugging. It is excellent and helps you to see the inside. KEIL using GDB.

- Click on the Build Target under Project
- Run Start/Stop Debug Session. You will get an error message "No ULINK2/ME Device found" if you don't connect the STM32 board with your computer and also if you don't select the correct debugger for this device.
- Check Options for Debug " Target 1" and select ST-Link Debugger and click OK.
- You can explore the other options to become accustomed with this.
- As everything is configured correctly, now you can Run Start/Stop Debug Session and press F11 for step by step debugging

6 What to do?

The lab has the following tasks:

1. Write a simple program to calculate: P = Q + R + S. Let Q = 2, R = 4, S = 5. Assume that r1 = Q, r2 = R, r3 = S. The result P will go in r0.
2. Write a simple program to calculate: P = Q - R . Assume that r1 = Q, r2 = R, and Q¿R. The result P will go in r0.
3. Write a simple program to calculate: P = Q - R- S. Let Q = 12, R = 4, S = 5. Assume that r1 = Q, r2 = R, r3 = S. The result P will go in r0.
4. Write a simple program to calculate: P = Q x R . The result P will go in r0.
5. This problem is same as the problem 1. W = X + Y + Z . Once again, let X = 9, Y = 8, Z = 5 and we assume that r4 = X, r3 = Y, r2 = Z. In this case, you will put the data in memory in the form of constants before the program runs.

7 Useful Hints

Use the following directives and instructions

- Directives
  - EQU: The EQU directive gives a symbolic name to a numeric constant, a register-relative value or a PC-relative value.

8 Submission

You must demonstrate your solution, submit the code, and report (Latex source files and code) to the course website. The submission date is by midnight January 31, 2023 .