EE2016 - Microprocessor Theory + Lab

Experiment 7 - Introduction to ARM Assembly Language Programming

1 Introduction

In this experiment, we will observe various aspects relating to the ARM assembly language. We will also write a few programs in ARM assembly. We will use Keil μVision software to run the ARM assembly language programs.

As for AVR programs, you need to open a new project and add the assembly language program file (named with a .s extension) to the project. Choose "Legacy Device Database (no RTE)" and LPC2378 (or LPC2148). RTE stands for Run Time Environment. There is no need to add the LPC*.s file to the project.

2 Tasks to be performed

Task 1: Type in the program given below. Put a breakpoint at the statement " B Stop" and observe the content of register R0 after executing the program via "Start/Stop Debug Session". Using the F11 key on your machine to single step through the code. Note down the content of register R0 at the end.

```
   AREA Program, CODE, READONLY
   ENTRY
     MOV r0,#11
Stop
     B Stop


   END
```

Task 2: Replace 11 in the code in Task 1 by &FFFFFFFF and run the program again. Note down the results (including any error messages).

Task 3: Type in the following program and run the same in Keil *mu*Vision. Record the values of r1 by single stepping through the code.

```
AREA Reset, CODE, READONLY

    ENTRY

          LDR r0, =7
          MUL r1, r0, r0
          LDR r2, =4
          MUL r1, r2, r1

          LDR r3, =3
          MUL r3, r0, r3
          ADD r1, r1, r3
stop
          B stop

    END
```

Task 4: Write an ARM assembly language program to obtain the tenth number in a Fibonacci sequence. By definition, the first two numbers in the Fibonacci sequence are 0 and 1; and each subsequent number is the sum of the previous two.

Task 5: Complete the following ARM assembly language program to divide a 32-bit binary number by a 16-bit binary number and store the quotient as well as the remainder. Note that there is no division instruction as such in the *basic* ARM instruction set (although recent CORTEX-M series datasheets indicate availability of an instruction called UDIV for unsigned division).

```
   AREA Program, CODE, READONLY
   ENTRY

     LDR ... ..... ;load the numbers into R0 and R1
     LDR ... .....
     MOV .......;  assign and clear the quotient register

Loop
     CMP ....... ;test division by 0
     BEQ Error1
     CMP ...... ;is the dividend less than the divisor ?
       ...... Result ;if yes, then we are done
     ADD ....... ;add one to quotient
     SUB ......
```

```
     B Loop
Error1
   MOV R3, #0xFFFFFFFF ;error flag (-1)
Result
   LDR R4, = Remainder ;store the remainder and quotient
   STR .......
   LDR R5, = Quotient
     STR .........
     SWI &11


Num1 DCD &.............
Num2 DCW &..............
   ALIGN

   AREA Data2, DATA, Readwrite
Quotient  DCD 0
Remainder DCD 0


   END
```

Task 6

Submit a report on the experiment on Moodle (within a week of this experiment). One report per group (with the names of the group members) is sufficient. The report should contain details of the solution (including the code) and your observations and experience (in programming, debugging etc.). Please note that reports that closely match those of other groups will be penalized.