Southern Polytechnic College at Kennesaw State University

Electrical Engineering Department

EE 3501 Laboratory Exercise 2: Programming in Assembly

Name:______________________ Date: _____________

Learning Objectives:

- How to create an assembly program and integrate this in the programming environment.
- How to measure the internal workings of the processor.
- Apply the concepts learned to achieve a desired result.
- Analyze the performance of the ARM processor based on experimental results.

Prelab

1. **Add a new file to the blink_led project from lab 1.**

   Right click on the project name and select new file from the dropdown menu. Enter add_asm.s

   Double click on the add_asm.s file name. Copy and paste in the following lines

```
     AREA   |.text|, CODE, READONLY

add_asm PROC
      EXPORT add_asm
      ADD   R0, R1
      BX   LR
      ENDP

      ALIGN
      END
```

   Make sure there are 5 spaces between add_asm and PROC. Make sure all the other lines begin with 12 spaces and case matters.

Add the following line to the main.cpp file on any line before int main()

```
extern "C" int add_asm(int a, int b);
```

Add the following lines inside the while loop right after the printf statement you added in lab 1:

```
printf("Assembler: %d + %d = %d\n", 2, 3, add_asm(2,3));
asm("AND R0,R0"); // this is a non-op in ARM programming
```

2. **Compile and test**

Compile the program and load the .bin file on the microcontroller board. Verify that the LED blinks. Open the terminal and verify the output is:

```
Hello ARM World.
Assembler: 2 + 3 = 5
```

Laboratory procedure

3. **Start a new project.**

Right click on My projects and add a new Program with the name lab2_asm. Cut and paste (replace) the following into the main.cpp:

```
#include "mbed.h"

DigitalOut myled(LED1);

extern "C" void delay_asm(void);

int main() {
    while(1) {
         delay_asm();
```

```
            myled = !myled; // invert LED state
    }
}
```

4. **Add an assembly file to your project**

   Add file called delay_asm.s to your project. Cut and paste the following into that file (be sure to type 10 spaces to indent the lines):

```
AREA  |.text|, CODE, READONLY

delay_asm   PROC
     EXPORT delay_asm
     LDR R0, =0x00800000
     MOV R1, #1
LOOP SUBS  R0, R1
     BNE  LOOP
     BX   LR
     ENDP

     ALIGN
     END
```

5. **Compile and run**

   Compile the program, download to the board and run.

   Connect oscilloscope to LED1 pin. Measure the time the LED is on or off. Locate the number in the delay_asm.s file that controls the time the processor spends in the LOOP. Change the delay several times and measure. What is the ratio between the number used and the delay time (in microseconds)?

   ____

6. **Use the ratio from 5 to determine the setting to give a delay of exactly 1 ms.**

   Setting   ____________________

7. **Bonus homework: Change the delay_asm code in part 4 to turn on and off the LED and never exit the delay_asm.s subroutine. Measure the delay compared to the results you obtained in part 5.**

Discussion

Explain what every line of code in the assembly program used in this lab does.

What is the longest delay (in seconds) you can achieve with the code in part 4?

Based on measured data, calculate the exact clock rate in MHz of your ARM core processor. To do this you will have to look up the number of clock cycles of each instruction. Refer to
http://infocenter.arm.com/help/index.jsp?topic=/com.arm.doc.100166_0001_04_en/ric1417175924567.html
Note: Assume that *P* = 1 for computing the number of cycles corresponding to the BNE instruction.

If you were to change the name of the asm file (or add a new .s file), where in the code would you have to edit to make this work?

Approved by:_______________ Date:_________ Results due: _________