

# **Lab 08: Arm Programming using Keil Software**

## **[Exercise]**

COSC2440: Computer Organization and Architecture

**Submit in Blackboard**

Dr. Kevin Long

# What to Submit?

## Minimum Requirements:

- 1) You need to submit ZIP file of your KEIL ARM Program.
- 2) You must use assembly file [**NO C/C++ files**]
- 3) Your code will be tested with different values!
- 4) Make sure you name your submission "**COSC2440\_Lab08\_YourFirstName\_YourLastName**"
- 5) Make sure you name your AREA in your **Program** "**Lab\_08\_YourFirstName\_YourLastName**"

## How are we going to grade?

- 1) If you **don't submit**, you will receive **0**.
- 2) If you **use C/C++ files**, you will get **0**.
- 3) If you submit **Empty Project**, you will get **0**.
- 4) If you **don't use one loop**, you will get **50** assuming that everything else is correct.
- 5) If You **use instructions other than the allowed instructions**  
[**ADD, BEQ, BNE, CLZ, CMP, LDR, LSL, MOV**], **you will lose 50**.
- 6) If you program works with only the sample number, you will receive **0**.
- 7) If your **project doesn't compile** and we see that you have your correct code, you will get **50**.
- 8) If you submit **only** the Code file [**Assembly.s**], you will get **50**, assuming you have the correct Code.

## Exercise

**Step 1:** Create a new **ASSEMBLY** project using Keil Software

(1) Make sure you use the **M3** option

(2) Make sure you name the folder "**COSC2440\_Lab08\_YourFirstName\_YourLastName**"

### Step 2:

"Write an ARM assembly language program that counts the number of 1's for any value in R0. The program must assemble/compile in KEIL and must be able to run in the KEIL simulator. Generally, R0 may contain any value, but for purpose of this exercise, you may move 0x2345ABCD into R0. The number in R0 does not need be preserved. You may use any other registers as you need. The result, total count of 1's in R0, should be in R1 when the program ends. "

UHCL: Unwala CENG 3371Microcontroller Programming Spring 2017

### Step 3: Copy the template below to your assembly file

```
AREA Lab_08_YourFirstName_YourLastName, CODE, READONLY
```

```
EXPORT __main
```

```
__main
```

```
LDR R0, =0x2345ABCD; This is how you assign large value to R0. MOV will NOT work!
```

```
stop B stop
```

```
END
```

### Step 3:

**Change YourFirstName to your first name.**

**Change YourLastName to your last name.**

**Step 4:** You are allowed to use ONLY the instructions below:

1. LDR
2. MOV
3. CMP
4. BEQ
5. CLZ
6. ADD
7. LSL
8. BNE

**You may use Keil ARM documentation to learn about any of the instructions listed above.**

[http://www.keil.com/support/man/docs/armasm/armasm\\_dom1361289850039.htm](http://www.keil.com/support/man/docs/armasm/armasm_dom1361289850039.htm)

### Step 5:

#### Results:

Hexadecimal: 0x2345ABCD

Decimal: 591,768,525

Binary: 10 0011 0100 0101 1010 1011 1100 1101

**The expected result will be 16 [Count # of 1 in the Binary Value above]**

**R1 should be 0x00000010**

### **Step 6:**

#### **Hints.**

1. You will need to use one loop
2. Use Windows Calculator to convert between hexadecimal to binary to save yourself time as shown in previous labs.

### **Step 7:** After testing your program, submit your project in Blackboard.

- (1) Make sure you delete the "Objects" folder before you submit
- (2) Make sure you zip & submit the correct folder "COSC2440\_Lab08\_YourFirstName\_YourLastName"