

16.317 Microprocessor I, Spring 2007
Lab 3: Assembly Language Programming
Due on 03/28/2007, 12:30pm EST

Reference

[1] “compare numbers” example and “stepper motor” example coming with Emu8086.

Objective

Learn the development of an assembly language program.

Content

In this lab, you need to develop an assembly program that computes the Greatest Common Divisor (GCD) of two positive integers and use the result to control the rotation of a stepper motor. Specifically, your program should function as follows.

- (1) Get the first integer from the keyboard. The input will be a hexadecimal number ranging from 0 to F. **Bonus points will be given if you can handle two digit numbers.**
- (2) Obtain the second **decimal** number (0 through F) from the keyboard.
- (3) Calculate the GCD of the two numbers.
- (4) Rotate the stepper motor for $GCD * 11.25$ degrees in a clock-wise direction.

You will use Emu8086 for this lab.

The following C code segment describes how to calculate the GCD of two numbers (a and b).

```
unsigned int gcd (unsigned int a, unsigned int b)
{
    if (a == 0 && b == 0)
        b = 1;
    else if (b == 0)
        b = a;
    else if (a != 0)
        while (a != b)
            if (a < b)
                b = b - a;
            else
                a = a - b;

    return b;
}
```

Steps

- (1) Make sure you understand the problem clearly.
- (2) Download the skeleton code, run the skeleton code in Emu8086 to get an idea of what the program does. The skeleton code can be obtained from http://faculty.uml.edu/ylo/Teaching/MicroprocessorI/lab3_skeleton.asm
- (3) Once the two numbers are obtained from the keyboard, you need to convert them from ASCII code to its corresponding value. For example, “1” entered from keyboard is coded as 31H in ASCII. Your program should interpret it as 1 in hexadecimal. Think of using a subroutine to do this conversion. You can assume only numbers and upper case letters are possible inputs.
- (4) Make sure you understand the algorithm of calculating the GCD of two numbers. Draw a flowchart to guide your implementation of this algorithm in assembly language. **You need to put the flow chart in your report.**
- (5) Code your design with assembly language. **Debug and run the program in Emu8086.**

Check-off and Turn-in

You need to check off with the TA by demonstrating your program. You need to assemble the source code and run your program.

You are expected to turn in a report with your source code in hardcopies. In addition, you need to email your source code to Professor Yan Luo (Yan_Luo@uml.edu) as an attachment with email subject as “16.317 Lab 3 Source Code Turn-in”.

In the report, you need to describe your development procedure in detail, including your solution in flowchart and subroutines you design.

Please put your name and your partner’s name in the beginning lines of your source code. Detailed comments are required in the source code.

Report format

Your report needs to follow the format below.

Lab # and title:
Student Name:
Partner’s Name:

Lab Purpose:
<It is usually the objective of the lab.>

Lab Content:

< Answer the questions in lab specification. Describe what you do in the lab, e.g. what commands you practiced. It has to be at least one page with 11pt font size. Try to organize and summarize the lab in itemized lists.>

Difficulties:

< state what difficulties you encountered in the lab and how you managed to solve it. If not, what have you tried? >

Conclusion and Suggestions:

Date of Grading _____

Student Name _____

Student ID _____

LAB THREE RUBRICS

Component	Grade	Actual points
Convert ASCII to hexadecimal value (check-off needed)	20	
Calculation of GCD (check-off needed)	30	
Rotate stepper motor (check-off needed)	30	
Flowchart in report	10	
Detailed comments in the source code	10	
Total	100	

You will receive ZERO in any of the following situations:

- (1) Copy other people's code. We have advanced software tools to test the similarity of source codes. Please do not risk. **You will be subject to university administrative penalty.**
- (2) Your source code produces errors when assembling. You need to at least get your program assembled and obtain an executable file correctly.
- (3) You do not submit your source code electronically. The source code you submitted will be checked.