Assignment #2 Fall 2011
SYSC-3006 Computer Organization
Department of Systems and Computer Engineering

Due Date: Monday, October 5, at 7:00 PM (electronic submission)

You must submit the files identified below <u>using the electronic Submit application</u>. The submission process will be cancelled at the deadline. No assignments will be accepted via email or on disk.

Part A: General questions (assign2A.txt)

1. Answer the following questions BRIEFLY (4-5 lines maximum):

a. [1 marks] On Slide 6, Part 1, we discussed various levels for the abstract models used to represent a computer. Why do programmers not use the MicroProgramControl (or the Hardware) to write Applications?
b. [1 marks] Is the Assembler for the Intel family the same as the Assembler used for programming the iPhone? (which uses an ARM 1176JZ(F)-S processor). Why?
c. [1 marks] High level applications like Firefox, Facebook or Tetris run both on Intel-based computers and iPhones. Considering your answer in part b., how is it possible to have such high level applications running in so many different platforms?
d. [1 marks] Why is Assembly Language not used to write large applications like the ones discussed in c.?
e. [1 marks] What is the CPU? What does it do?
f. [1 marks] The CPU is connected to other components using what kind of interconnect?

2. [2 mark] The goal of this exercise is to help you to get used to the Instruction Set in the 80x86. You should find in the course text (any text on the 80x86 can be used) the following instructions that we have not explained in the course, and answer:

. The name of the instruction
. The Assembly Language mnemonic for the instruction
. The operands used
. The flags modified by the instruction
. The type of instruction (Arithmetic, Boolean, Control Flow or Data Transfer).
. The instruction encoding (OPTIONAL)

```
DEC
XCHG
```

Part B: A First Programs (use the provided template.asm file)

A series of exercises are listed below. Complete the small program provided to perform the assembly-language equivalent of these exercises. Your code must be a general solution that will work for a variety of input values (the variables will be changed by the TAs, so test your solution with a variety of values).

Commenting your code is mandatory; the TAs should understand everything you have done in detail. Comment thoroughly.

1. [2 mark] (assign21.asm; assign21.exe) Copy the following program in the template. Assemble the program, execute it and complete the comments (*Var* is an 16-bit variable)

```
MOV BX, 7     ; _____________________
DEC BX        ; _____________________
MOV BX, Var   ; _____________________
; This short program_____________________ (explain what it is doing)
```

2. [2 mark] (assign22.asm; assign22.exe) Write a program that adds two 8-bit numbers (suggestion for testing: the total of the addition should be between 30h and 39h; submit a solution with the following two numbers: 30h and 7h). Store the result in register AL, and display the result on screen. The numbers must be in memory locations. Write comments in your source code explaining what of the options you have used.
3. [2 mark] (assign23.asm; assign23.exe) Write a program that computes the following formula: $Z = X^*Y + 2$. Here, X and Y are 8-bit unsigned integer numbers. Z should be stored in register AX. The numbers must be stored in memory. You must submit a solution using an initial value of X = 5, Y=10; display the value of AL on screen using the routine provided.
4. [2 mark] (assign24.asm; assign24.exe) The Arithmetic Series is the sum of a sequence $\{a_i\}$ with $i = 1, 2, 3, ...$ in which each term is computed from the previous one by adding a constant . Hence, for $i>=2$, $a_i = a_{i-1} + k$. Write a program to compute the first 5 terms of the Arithmetic Series (i.e., $i=1,...,5$) with constant $k=3$. The numbers should be computed and then stored in the AL register (on every cycle you should print the number on screen; the first numbers in the series will be displayed properly; numbers of 2 digits and up will not be displayed properly. To verify the rest of the results, you will need to use the debugger set a breakpoint and examine the variables in memory directly).
5. [2 mark] (assign25.asm; assign25.exe) [Bonus] Modify exercise 4 and store the results on an array

Part C: Debugging (assign2C.txt)

1. [1 mark] Explain what is a breakpoint and how to define a breakpoint in your program.

2. [1 mark] Explain three different ways to run a program using the debugger.

3. [2 mark] For exercise assign21.exe, load the executable into the debugger and run it step by step. At every step, write:

- Value of the flags register
- Meaning of these flag values (i.e., what is the result of the previous instruction?)
- Contents of AX, BX, CX, DX
- Contents of CS, IP

You will obtain this information by stopping the execution of the program after each of the line, and getting the information using the Debugger.

(Practice exercise for the Midterm: repeat with the rest of the programs. Think about the meaning of the results of the execution and the registers contents).

(Practice exercise for the Midterm: modify the memory variables, register contents, IP, etc. during the program execution to learn how to do online modifications to your running programs)

Assignment 2 Marking Criteria:

No marks will be given if the assignment is late or does not assemble. You may submit incomplete assignments if there are some working portions, but the submission must assemble and run for those portions AND you must clearly identify which parts of complete with comments.