Assignment #4 Fall 2011
SYSC-3006 Computer Organization
Department of Systems and Computer Engineering

Due Date: Assignment 4 Due: November 4 @ 7:00 PM

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 I : An exercise in indirect addressing (assign41.asm, .exe)

We will build a simple light controller for a night club. There are five lights you will control (each of which connected to the same circuit in the LED box in our lab). The lab's switch box will be eventually changed by the actual lights (i.e., when a LED turns on, the corresponding light will be turned on).

In this assignment, every time you send a command to the light controller, you must wait APPROXIMATELY 1 second (write a wait loop that takes approximately 1s; don't worry about precision in timing).

The lights are programmed using 2 arrays:
LightConfiguration = { 11h, 0Fh, 1Fh, 10h }
Delay = { 20, 30, 20, 20 }

The first array contains the current light configuration. The second array, the delay between one light configuration and the next one.
Write a program to traverse the 2 arrays and print the contents of the LightsConfiguration array on screen, and following the array with timing information.

To be sure your program works, change the contents of the arrays, and check that it still works with a different schedule.

Part II : An exercise in Based-Indexed indirect addressing (assign42.asm, .exe)

Now, we will use a more complex data structure: instead of two arrays, you will use one, as follows:

DancingLights = { (11h, 20); (0Fh, 30); (1Fh, 20); (00h, 20) }

Repeat Part I using this new structure. You have to declare and initialize the matrix. Give fixed initial values to each of the elements.

(your program still needs to display the light configuration in Hexadecimal, like in Part 1)

Part III: an exercise in stacks and subroutines (assign43.asm, .exe).

Modify Part II and reorganize your code using subroutines. You must define the following routines:

void printLedInfo(byte); // prints the current LED configuration on screen
void waitOneSecond(); // waits one second

Use the stack to pass any parameters needed.

Part IV [BONUS]: an exercise about parallel I/O (assign44.asm, .exe)

i) Modify assign43.asm, and include code to turn on the LED lights. Use the data showed in Part II for testing (for partial marks, if you did not solve Part III, do it with Part II. If you did not solve Part II, do it with Part I. Put a clear note indicating what you have done at the beginning of your program).

ii) **(assign45.asm, .exe)**

Modify assign44.asm to receive inputs from the switches. If, during the execution of your program, a switch is set manually, the corresponding light is not set. When the switch is turned off, the schedule for the corresponding LED resumes.

Assignment 4 Marking Criteria (12 marks total + 6 marks bonus = 4%+2%):

3 marks per exercise.

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.