Lab Project 1:

Introduction:

In lecture you will be learning the theory and principles of how to interface different types of circuitry with the Intel x86 family of processors. In lab, you will apply those principles by creating software and hardware that runs and utilizes an 80186XL processor. The culmination of the labs will be designing and implementing the interface between external memory and this processor.

To achieve this objective, the labs will be divided into four projects. The first and second projects will deal with creating software to run on the 80186XL processor. By creating this software and through the use of Chameleon, you will learn how the CPU executes code and how the program uses memory for code and data. The final two projects will be the actual design and implementation of the memory interface. This will require hand wiring the circuitry and programming two EPROMs. Ultimately, the design process and implementation of the software and hardware is the ultimate goal of the labs. The first two projects build upon each other as do the second two projects build upon each other. When doing the first and third projects this should be kept in mind. The program developed in the second project will be used in the final project.

Project 1 Goals:

- To write an x86 assembly program to test memory locations
- To gain familiarity with the USP-186 In-Circuit Emulator and assembly development environment.

Requirements:

- Write an x86 assembly program to test the first 128k of memory including the start and stop addresses.
- The memory test should be non-destructive. Meaning that the contents of a particular memory location should be the same before and after the test.
- The following test values should be used: A0h, 0Ah, 50h, 05h.
- The test values should be stored in variables located in the data segment of your program.
- Your program should consist of one data segment and one code segment
- The logic flow of the program is provided in figure 1.
- Use the code from the tutorial program as your skeleton.
- Use ES: [BX] to address the memory locations.
- Use the .cfg file from the tutorial program for the locate software.
Figure 1: Logic Flow of Program

Deliverables:
- Demonstrate your program to your lab instructor
- Turn in a professional quality report. Formatting should coincide with the document on report formats. Correct spelling, grammar, and coherence as well as level of professionalism should be used in your report as these will be factors in grading.
- Include your properly documented source code along with the .CFG file.
- Include a full flow chart of the program operation in your lab report.

Questions:
1. How did you design the program such that the memory test was non-destructive?
2. In what ways could you modify the program to be more efficient? Explain quantitatively how the program will be more efficient? (i.e. less processing required, less memory required, etc).
3. If this memory test program were to test every address of the memory, what issues would have to be dealt with that are not dealt with in this program? In what ways would you over come these issues (hypothetically)?