ELEC 464 : Microcomputer System Design
Fall 1995

Assignment 1
Due September 19, 1995

This assignment is meant to help you review assembly language programming and the 68000 instruction set. We will not deal with program design or coding (e.g. Chapter 3) in this course. An excellent reference for those wanting to learn more about these issues is Code Complete: A Practical Handbook of Software Construction, *by Steve McConnell, Microsoft Press, 1993.*

Assignment

(a) Chapter 2, question 28. You must use a loop rather than repeating similar code 32 times or using look-up tables.

(b) Chapter 2, question 31. *(Hint: A bubble sort would be an appropriate sorting algorithm).*

Put your disk (see below) in the assignment box labeled ELEC 464 on the third floor (across from the elevator) no later than 5:30 pm on the due date (September 19 1995).

Submission Instructions

Since your code will be marked semi-automatically you must follow the following instructions or risk receiving a mark of zero.

- submit the source code on an MS-DOS 3.5 inch floppy disk, in files named `ASG1A.X68` and `ASG1B.X68` respectively
- in both cases your code must be written as a subroutine that returns using `RTS`. You may include test code in the files – it will not be marked.
- the entry points for the subroutines must have labels `REVERSE` and `SORT` respectively
- the code should start at address $400
- the last non-blank line in each file should be an `END` statement
- your files must compile with the `X68K.EXE` cross-assembler without errors
- your files may *not* include labels beginning with the string `MARKER`
- write your name, student number, course, assignment number and date on the disk
- the files on the disk must be readable; make sure your disk is good

Marking

The marker will replace the `END` statement in your file with test code that calls your subroutine and checks the results. Your routine will be tested with typical data as well as with extreme cases (e.g. maximum or minimum values, zero-length arrays, etc). It should produce reasonable results for all possible inputs.

The marker will also read your source code. Marks will be deducted for code that is hard to understand and for incorrect, insufficient or excessive documentation. Look at the examples in chapter 2, 3 and 11 for reasonable examples of assembly-language coding style. Use comments before each routine to describe the inputs and outputs and what the routine *does*. Use comments within the routine to explain *how* the routine does it.

These are individual assignments. All copies of obviously plagiarized assignments will be given a mark of zero.

Software

You should use the MS-DOS cross-assembler for the Motorola 68000 provided by the author of the textbook. The cross-assembler is `X68K.EXE` and the emulator is `E68K.EXE`. These commands should be available on the undergraduate PC Novell network in the 3rd floor labs. EE students can get accounts from Ken Madore (basement power lab). You can also download the software from `http://www.ee.ubc.ca/~edc/464` if you would rather run it on a stand-alone PC (the two files are packaged in a "`.zip`" archive).

Instructions for using the cross-assembler and simulator are given in an accompanying handout.