CS 2321
Final Laboratory Problem Set
Putting It All Together

This Project Replaces
The Final Examination

Due Monday, May 8, 2006 at 2:00 p.m.

This project is to be done on an individual (with the exception of a group project as indicated below) basis following the Trinity University Academic Integrity Policy. or Honor Code.

1 Academic Integrity and Honor Code

All students are covered by a policy that prohibits dishonesty in academic work. The Academic Integrity Policy (AIP) covers all students who entered Trinity before the Fall of 2004. The Academic Honor Code covers all those who entered the Fall of 2004 or later.

The Integrity Policy and the Code share many features: each asserts that the academic community is based on honesty and trust; each contains the same violations; each provides for a procedure to determine if a violation has occurred and what the punishment will be; each provides for an appeal process. The main difference is that the faculty implements the AIP while the Honor Code is implemented by the Academic Honor Council. Under the Academic Integrity Policy, the faculty member determines whether a violation has occurred as well as the punishment for the violation (if any) within certain guidelines. Under the Honor Code, a faculty member will (or a student may) report an alleged violation to the Academic Honor Council. It is the task of the Council to investigate, adjudicate, and assign a punishment within certain guidelines if a violation has been verified.

Students who are under the Honor Code are required to pledge all written work that is submitted for a grade: On my honor, I have neither given nor received any unauthorized assistance on this work and their signature. The pledge may be abbreviated pledged with a signature.

Laboratory problems should be submitted electronically (e-mail to cs2321@ariel.cs.trinity.edu) on or before the due date and should contain a problem write-up, source code to any programs and data sets used in solving the problem. The submitted files should be ASCII text files having Unix end-of-line characters (please convert all Windows and Mac text files to Unix format–I have found that Emacs seems to do a reasonable job of such conversions). If several files need to be submitted, put them in a directory having name *your-last-name-problem-set-number* and create a tar archive of this file system and attach it to your e-mail problem submission.

A brief (about 10 minute) oral presentation and demonstration of your project will be done during the Final examination period for this course. The final course project and course project presentation will be used as a substitute for the course final examination.

Group projects

Students may form groups of size two to complete this final project, however, written materials and final presentation must indicate the contribution of each group member.

The Project Problem

The final project will involve the design of a simple computer and the implementation of a model of same. The machine may have any register structure and address structure desired. It is assumed that the machine will have at least one general purpose register and implement, at least, the 2's complement binary operations of *add, subtract, multiply, divide, and, or* and *not*.

The machine should also implement some sort of subroutine linkage machinism. The *alu* for the machine must be modeled at the circuit level while other parts of the architecture may (or may not) be implemented at the circuit level.

In building the various models that are part of your machine design, you may use any of the J based models developed in class or in solutions to homework problems. You are not constrained to use these tools, but may use other tools or programming languages to accomplish these tasks.

It is not necessary to build elaborate user interfaces to your machine model, in fact, it is preferable to spend time on machine design issues rather than user interface programming. The most rudimentary uses of standard input and output will suffice for this project.