

Fall 2006
CS/ECE 333
Lab 1 – Programming in SRC Assembly

Lab Objectives:

1. How to assemble an assembly language program for the SRC using the SRC assembler.
2. How to simulate and debug programs using the SRC ISA level simulator.
3. How to write simple SRC programs including programs which make procedure calls.

Lab Procedure

This lab will be comprised of an in-lab exercise to be completed in lab and a post-lab exercise to be completed and electronically submitted by the start time of your lab section the following week. For instance, if you have lab on Wednesday, then your postlab is due the following Wednesday.

You will need your textbook (Appendix B1) and the slides from class on implementing procedure calls in SRC to complete this lab.

In-lab Deliverable:

- Demonstrate your working `findMax()` function to the TA.
- **2 points extra in-lab credit** for spotting a minor bug/typo in the lab document.

Post-lab Deliverables:

Electronically submit the following to Toolkit (One code submission per pair):

1. `findMax.asm`. The procedure you generated to calculate the `findMax` function during the in-lab exercise.
2. `call_findMax.asm` program. The header of your assembly file must contain a comment header that contains your name and email ID and the name and email ID of any partner you may have worked with. If you collaborated significantly with others/other groups, please also credit them in the comments and describe the nature of the help you received.

Each code submission should contain comments at the top which include the following information:

- Name and email ID of both partners (if working with a partner)
- Typed Honor Pledge
- Comments crediting any help received

Collaboration Policy

You may work with one partner on the in-lab and post-lab exercise. If you choose to work with a partner, this partner must be the same partner for both in-lab and post-lab. You must put the names and email IDs of both partners in the top of your submitted assembly code files. **If you discuss with another person/group to solve the problem, you must also credit them in the comments of your program describing the nature of the help you received.**

You may talk with others about aspects of the coding exercise, but you may not copy code directly from others.

In-lab Exercise

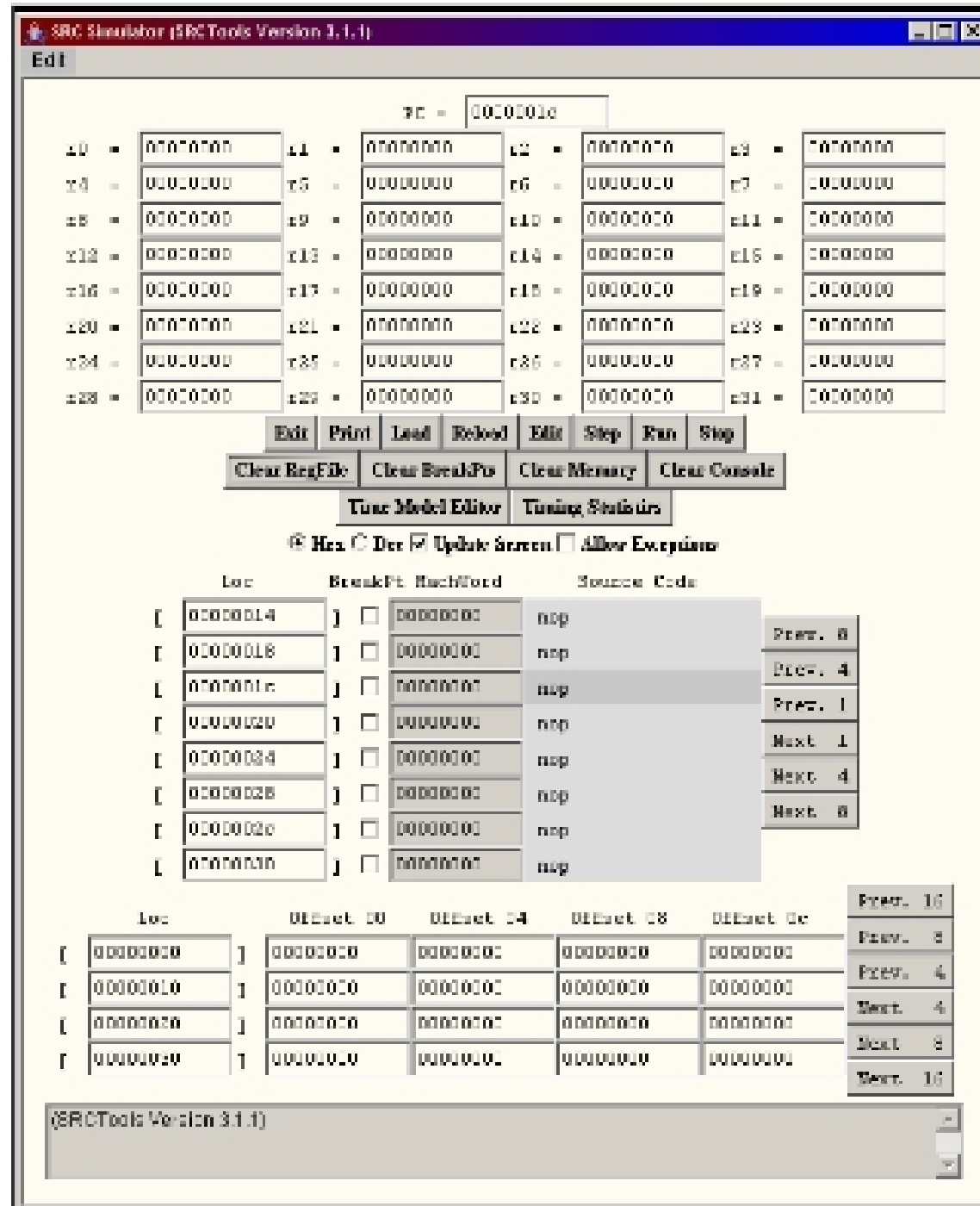
Getting started

This lab exercise is designed to show you how to use the SRC assembler and the SRC ISA level simulator. The assembler translates SRC assembly language, written according to the conventions shown in Table B.1 of Appendix B in the Computer Systems Design and Architecture text, into valid machine code for the SRC microprocessor. The SRC simulator simulates the SRC microprocessor at the ISA level. That means that it executes the SRC instruction set and shows the effects that each instruction has on the content of the programmer visible registers and the memory of an SRC-based system.

The SRC assembler and simulator have been built as a Java application which requires a minimum of Java 1.1 Runtime Environment to run.

1. If SRCTools is not already on the machine you are working on, download [SRCToolsv3.1.1.jar](http://www.cs.virginia.edu/~cs333/f06_simulators/SRCToolsv3.1.1/) (http://www.cs.virginia.edu/~cs333/f06_simulators/SRCToolsv3.1.1/) from the class website.

The simulator should be runnable in Windows by double-clicking on the application. You should see an interface that looks like this:



2. The display shows an editable text field showing the contents of the program counter (PC) register, an editable text fields displaying the 32 general-purpose registers numbered from r0 to r31, a row of control buttons, a display of 8 machine words with their addresses, complete with checkboxes for setting breakpoints, disassembled source code, and navigation buttons for displaying the program code in other memory locations (these fields are not editable) and an editable display of memory addresses and contents. [NOTE: All numbers are represented in hexadecimal.]

The functions of the control buttons are as follows:

- The **Exit** button exits the simulator by terminating the application.
- The **Load** button allows the loading of a binary SRC file by bringing up the standard file opening dialog that your machine supports.
- The **Reload** button allows the reloading of a binary SRC file.
- The **Edit** button opens the SRC editor and assembler. It allows you to open an existing file or create new files.
- The **Step** button executes the single machine instruction pointed to by the PC. Note that this instruction is displayed in the center of the screen and is highlighted.