Electrical Engineering
& Computer Science
University of Missouri

ECE 4270: Computer Architecture, Spring 2025

LAB 2: RISC-V Assembler

In this lab assignment, you will develop an assembler for RISC-V ISA. It will take a program written in RISC-V assembly and will convert it into RISC-V machine code. In the previous lab assignment, you developed an instruction-level simulator for RISC-V and tested it out with some test programs provided with the assignment. For this assignment, you will use the output of your RISC-V assembler as an input to the RISC-V simulator you have developed during the previous lab assignment.

Your RISC-V assembler should accept an input file containing the program written in the RISC-V assembly and generate an output file containing the corresponding RISC-V machine code. The machine code will be like the ones that you used in the first lab assignment (each line represents an instruction and is encoded as a 32-bit hexadecimal value). As an example, the following C code excerpt:

```
int A[20];
int sum = 0;
for (int i=0; i<20; i++)
     sum +=A[i]
```

can be written in RISC-V ISA as:

address instruction

```
# Assume x8 holds pointer to A
# Assign x10 = sum, x11 = i
add x10, zero, 0        # sum = 0
add x11, zero, 0        # i = 0
addi x12, zero, 20      # x12 = 20
Loop:
bge x11, x12, exit:
sll x13, x11, 2         # i * 4
add x13, x13, x8        # & of A + i
lw x13, 0(x13)          # *(A + i)
add x10, x10, x13       # increment sum
addi x11, x11, 1        # i++
 j Loop                 # Iterate
exit:
```

Of course, you can write a better code for the same C excerpt; however, for illustration purposes bear with this one.

Your RISC-V assembler should convert the above RISC-V instructions to the following machine code (first 3 instructions):
00000533
000005b3
01400613

After generating the above instruction stream and saving it into an output file, you should be able to run the RISC-V simulator that you developed during the first lab assignment with this file.

Once you have a working RISC-V assembler, you should write the RISC-V assembly program for the following problems and convert them into RISC-V machine code using your RISC-V assembler and test them out using your instruction-level RISC-V simulator.

Problem 1: You have an array that contains 10 integers, A = {5,3,6,8,9,1,4,7,2,10}. Use the bubble sort algorithm to sort it in ascending order. In your RISC-V program, you should store these values in the data segment of the memory, before trying to sort them out.

Problem 2: Write a RISC-V program that finds the Fibonacci number of a given value. Calculate the Fibonacci number of 10 to test your RISC-V program.

If you must use an instruction that you haven't implemented in the RISC-V simulator already, you should also implement that instruction and specify it in your lab report.

Grading Rubric

Code: RISC-V assembler, and solutions to the problems (75)
Report: 25 points

Code and RISC-V programs (75 points):

To get full credit for the code, your RISC-V assembler should convert the RISC-V programs you wrote for the given problems correctly, and the RISC-V simulator should run them properly.

Lab report (25 points):
Your report should give details about the work, milestones in your work, and your implementation decisions (why did you choose the way you did it, and/or how did you do that).

If you added new instructions to the RISC-V simulator to support the operations you needed in solving the problems, please specify them in the report, as well.

Submission

You should submit the lab report along with the assembler and simulator code that you worked on. Each member has to submit both report and the code. Please, generate a pdf file for your report and name it lab2_report_name.pdf, then place it into the folder called lab2_name . The folder lab2_name should also contain the src/ and input/ folders that contain your code and given input program, respectively. Then, please compress the lab2_name folder as lab2_name.tar.gz and submit it through the Canvas.

Due Date

Your lab is due on:
03/11/2025 (Tuesday)