RISC-V Assembler

Prof. Naga Kandasamy
ECE Department, Drexel University

Develop an assembler that translates programs written in RISC-V assembly language into the binary code understood by the processor. This problem is due August 2, 2024, by 11:59 pm. You may work on this problem in a group of up to two people. Please submit original work.

You have been provided with starter code in C for the assembler which take the RISC-V assembly-language file as a command-line argument. If the input file name is `trace_1`, usage is as follows:

```
$./assembler trace_1
```

The trace file consists of text lines, each representing a RISC-V assembly instruction. Your assembler must translate each of these instructions into machine code and store them in instruction memory. Use the skeleton code provided to you as a starting point for your solution. In particular, an example of how to assemble an R-type instruction is provided within the *parse_R_type()* function in *parser.c*.

Testing Your Assembler

Test the correctness of the machine code generated by your assembler using the provided trace file which contains the following instruction classes: R-type, I-type, and SB-type.

```
add x10, x10, x25
ld x9, 0(x10)
addi x22, x22, 1
slli x11, x22, 3
bne x8, x24, -4
```

First, translate these instructions into their corresponding binary representation by hand. Then, run the file through your assembler implementation to compare the results and verify correctness.

Submission Instructions

Submit via BBLearn, the source code for the assembler along with a README file that details how to execute it.

The code must compile and run on the `xunil-05.coe.drexel.edu` machine. Run `make clean` in your source directory prior to submitting the files. We must be able to build your code from source and do not want pre-compiled executables or intermediate object files.