

## Lab for ITSC 3181, Introduction to Computer Architecture, Spring 2023

### Lab #06: Encoding and decoding RISC-V instructions. **Due on 02/23 and count for 2% of the final grade percentage.**

In this lab, you will perform two tasks, one as an assembler and the other as disassembler. An assembler is a computer program that translates assembly language to machine language, by encoding assembly instructions with their binary representation. A disassembler is a computer program that translates binary machine language into assembly language—the inverse operation to that of an assembler. The two tasks are:

- 1) given RISC-V assembly instructions, identify the class and encoding format of the instructions and encode them into 32-bit instruction words, and
- 2) given instruction words in hexadecimal format, decode the instructions and write down their assembly instructions.

Encoding format and decoding of RISC-V instructions are studied in book chapter 2.4 (2's complement binary representation), 2.5 (R-, I- and S-format instruction encoding for arithmetic, logic, load and store instructions) and 2.10 (SB- and J-Format encoding for branch and jump instructions, and decoding instructions). You should review the related lecture slides or the text book sections before working on the lab. For R-, I- and S-Format encoding, please check lectures from

[https://passlab.github.io/ITSC3181/notes/Chapter02\\_Instructions\\_OperandsNumberEncoding.pdf#page=40](https://passlab.github.io/ITSC3181/notes/Chapter02_Instructions_OperandsNumberEncoding.pdf#page=40), and for SB-format (branch instruction), please check [https://passlab.github.io/ITSC3181/notes/Chapter02\\_Instructions\\_ProcedureStringAddressing.pdf#page=25](https://passlab.github.io/ITSC3181/notes/Chapter02_Instructions_ProcedureStringAddressing.pdf#page=25).

The RISC-V standard for encoding instructions can be found from

the RISC-V instruction manual at <http://content.riscv.org/wp-content/uploads/2017/05/riscv-spec-v2.2.pdf#page=116>. Part of the encoding/decoding table are given in this document for your reference. Your answer should be completed in the following answer table. Please note that immediate numbers used in instructions are encoded using 2's-complement format. The usages of immediate include the offset of load and store instructions, relative and absolute address for branch and jump instructions, and immediate operands of I-Format arithmetic and logic instructions.

#### **Answer sheet, submission and grading:**

Using the answer sheet from [https://passlab.github.io/ITSC3181/notes/Lab\\_06/InstructionEncoding.xlsx](https://passlab.github.io/ITSC3181/notes/Lab_06/InstructionEncoding.xlsx) to work on your solution. After completing answers with the sheet, submit the sheet on canvas. There is total 16 instructions for you to finish. For each instruction, there are total 9 table cells that are needed to fill in for the answers, for the maximum 7 points. Starting from 7 points, each incorrect answer causes deduction of 1 point until all the 7 points are deducted for that instruction.