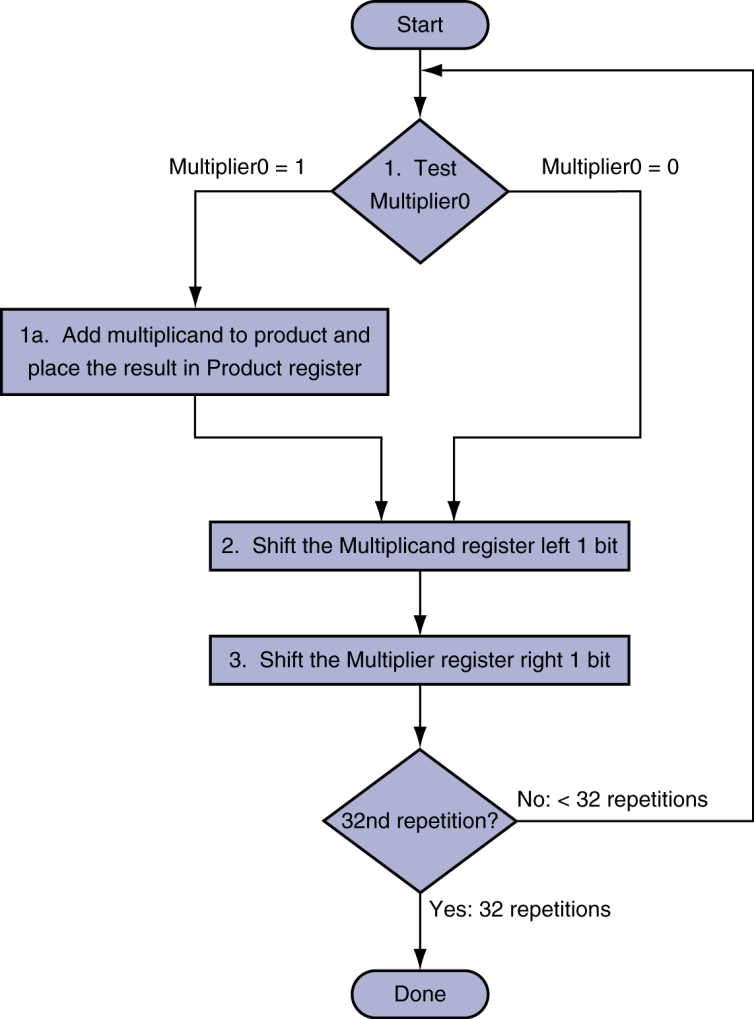
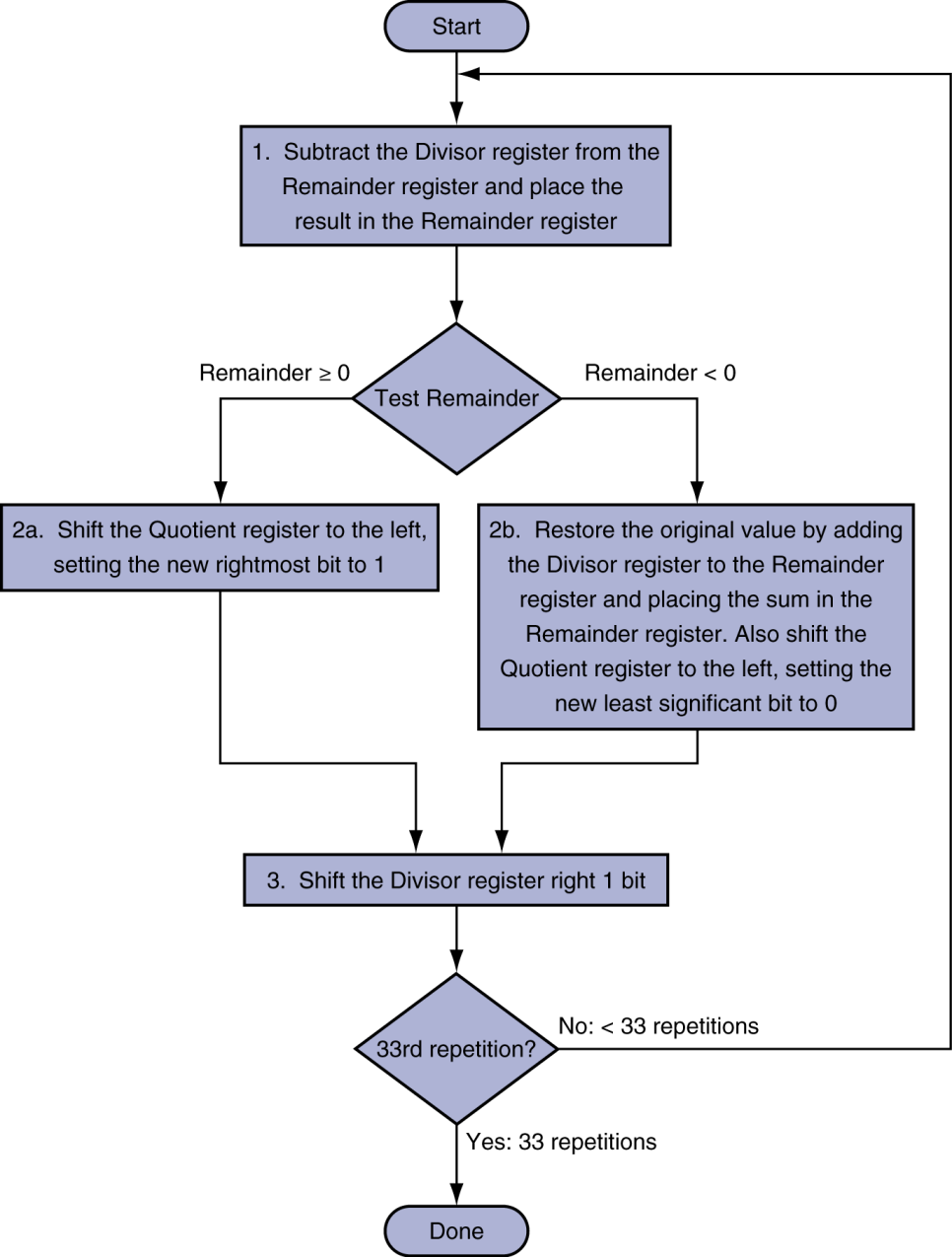
**Lab #2: Multiplication and Division in RISC-V Assembly**

**Introduction**

This project is intended to get you familiar with some of the basic operations and instructions involved in writing an assembly program in RISC-V. You will be implementing shift-and-add multiplication and shift-and-subtract division of \***unsigned\*** numbers without using the explicit multiplication and division instructions. For more information regarding multiplication and division algorithms please review lecture viewgraphs. The algorithm flowcharts are also provided below for convenience.

**Figure 1.** Flowcharts for (left) shift-and-add multiplication and (right) shift-and-subtract division algorithms.

**Overview**

The goal of this project is to write a function that performs integer multiplication and a function that performs integer division in RISC-V using base integer set of instructions (RV32I), e.g. any instruction from reference sheet page (Figure 2). Avoid using pseudoinstruction (that are not real instructions and translated to one or several real insructions upon assembly) except for “j” jump instruction that was discussed in class and can be used instead of “jalr” which will be discussed later. Just for a reference, see a list of pseudo-instruction (that should be avoided) in “Assembler Pseudo-instructions” sections at <https://michaeljclark.github.io/asm.html> .

Note that you cannot implement multiplication by repeatedly (multiplier times) adding multiplicand. The same applies to division.

We will use RARS RISC-V emulator <https://github.com/TheThirdOne/rars> . RARS is distributed as an executable jar. You will need at least Java 8 to run it. For convenience the latest RARS executable file is included in the lab package.

Also provided are a skeletons of code for each function. You are to write the body of each function.

**Multiplication**

For the multiplication function, start with skeleton code mult.s. You are provided the two operands in a0 and a1 which can be up to 8‐bits in length. Place the result of the multiplication in a2. The contents of a0 and a1 must be preserved.

a2 = a0 x a1

**Division**

For the division function, start with skeleton code div.s. You are provided the divisor in a0 and the dividend in a1. Each operand can be up to 8‐bits in length. The quotient of the division should be placed in a2 and the remainder in a3. The contents of a0 and a1 must be preserved.

a2 + reminder a3 = a1/a0

**Hints**

The code for each function can be greatly simplified through looping. For testing, you can edit op1 and op2 located at the top of the files. op1 is loaded into a0 and op2 is loaded into a1. Make sure that your code covers all relevant corner cases, e.g., division by zero, etc.

**Lab Report**

Your lab report should be a single PDF file, which has all the following items in the following order and clearly labeled:

1. **Please indicate how many hours you spent on this lab.** This will be helpful for calibrating the workload for the next time the course is taught.
2. Your “mult.s” file \*
3. Your “div.s” file. \*
4. If you have any feedback on how we might make the lab even better for next quarter, that’s always welcome. Please submit it in writing at the end of your lab

\* For your code files, please copy and paste your code clearly in a monospace font (ex., Courier New). Provide in the comment the equivalent C code for each added line of the code in the skeleton.

**Autograder**

You will need to submit your “mult.s” and “div.s” files to the Gradescope assignment, “test lab2”. When you make a submission, an autograder will run several tests to ensure your design is correct. You must pass all the tests to get a full score on the lab. You can resubmit as many times as you like.

**What to Turn In**

You will need to make two separate Gradescope submissions: “lab2 Code” and “lab2 Report”.

* For “lab2 Code”, please submit your “mult.s” and “div.s” files.
* For “lab2 Report”, please submit your lab report PDF file.

Only one submission per group is needed; be sure to add all your group members.

****

**Figure 2.** Set of instructions that can be used in this lab.