ELL 782: Computer Architecture
Assignment-1

Submission Deadline: September 3, 2023
Released: 2 August, 2023

1 Introduction

The Gaussian elimination method, alternatively referred to as row reduction, is a computational procedure employed to solve systems of linear equations. The process involves a series of operations conducted on the matrix of coefficients.

In this assignment, you need to solve a set of linear equations using the Gaussian elimination method. You have to create an assembly code using RISC-V ISA. This assignment needs to be done **individually**.

2 Implementation

You are given a set of five linear equations represented as $AX = B$, where A is a 5×5 matrix, X is a 5×1 matrix.

$$
\begin{aligned}
a_{11}x_1 + a_{12}x_2 + a_{13}x_3 + a_{14}x_4 + a_{15}x_5 &= b_1 \\
a_{21}x_1 + a_{22}x_2 + a_{23}x_3 + a_{24}x_4 + a_{25}x_5 &= b_2 \\
a_{31}x_1 + a_{32}x_2 + a_{33}x_3 + a_{34}x_4 + a_{35}x_5 &= b_3 \\
a_{41}x_1 + a_{42}x_2 + a_{43}x_3 + a_{44}x_4 + a_{45}x_5 &= b_4 \\
a_{51}x_1 + a_{52}x_2 + a_{53}x_3 + a_{54}x_4 + a_{55}x_5 &= b_5
\end{aligned}
$$

You need to find the value of $X = x_1$, x_2, x_3, x_4, x_5 that satisfy this system.

2.1 Inputs

Two matrices A and B are given as inputs. These matrices need to be stored in the row-major format in the memory. Since, RISC-V does not support floating-point operations, use the fixed-point notation for storing the values (format: **12.16**) where 12-bits are used for integer part, 16 bits for the decimal part, and rest 4 MSB bits for the sign (refer to Figure 1).

2.2 Outputs

The elements of the output matrix X should also follow the fixed-point notation (12.16). There are various possible solutions for a given system of linear equations and the format of the output for each of those cases is as follows:

- **No solution:** Print "No solution exists"
- **Unique solution:** Print the value of X.
- **Infinitely many solutions:** Print "Infinitely many solution exist".

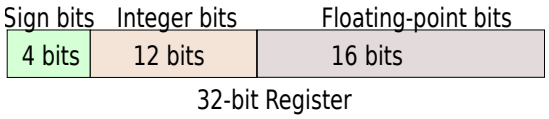

Figure 1: Fixed-point representation of a floating point number

2.3 Setup

- A quick start to RISC V assembly language programming is available here.
- You have to use the Microsoft VS-CODE-based locally installed RISC-V simulator for assembly language code development, execution, and debugging.
- First, you need to install the VS code. You can install the RISC-V Venus Simulator using the Link

3 Code Verification

In this section, you will implement a simple Python script to test the code you created. Your program should generate random test cases and then solve the Gaussian elimination problem. Python will generate the solutions in a floating-point format. You need to convert it to the fixed point format. Compare the results generated from your RISC-V code and your Python script. Report the error percentage.

- Be prepared to explain the fine details of your sanity test.
- Make sure your script covers all possible cases.
- Make sure to report the average error percentage.

You are free to use any in-built libraries for this part.

4 Report

The report should clearly mention the implementation methodology for all the parts of the assignment. Small code snippets are alright, additionally, the pseudo-code should also suffice.

- Details that are relevant to the implementation.
- Say what you have done that is extra (this should be the last section in the document).
- Limit of 10 pages (A4 size) and must be in PDF format (name: report.pdf).

5 Submission Guidelines

- We will run MOSS on the submissions. Any cheating will result in a zero in the assignment, a penalty as per the course policy and possibly much stricter penalties (including a fail grade and/or a DISCO).
- There will be demos for assignment 1.
- We will be evaluating your submission with a different set of input numbers. So, make sure your implementation works for all the possible cases.
- You need to submit your RISC-V code, report, and the Python script.

How to submit:
Create a tar ball using:

```
tar czvf assignment1_<entryNumber1>.tar.gz *
```

This will create a tar ball with name, assignment1_<entryNumber1>.tar.gz. Submit this tar ball on Moodle. **Entry number format: 2019EEZ8358**

The submission deadline is **23:59, 03 September 2023**.