

Assignment 1 Instructions

Wednesday, April 2, 2025 1:20 PM

Objective:

This lab aims to implement a matrix-matrix multiplication algorithm in assembly language for the RISC-V architecture. This exercise will help you understand the concepts of matrix multiplication and assembly language programming for RISC-V.

Requirements:

- Basic understanding of RISC-V assembly language.
- Familiarity with matrix multiplication algorithms.
- RISC-V assembly development environment

Background:

Matrix multiplication is a fundamental operation in linear algebra and finds applications in various fields, such as computer graphics, scientific computing, and machine learning. The algorithm for matrix-matrix multiplication involves multiplying each element of a row from the first matrix with each element of a column from the second matrix and summing up the results to obtain the corresponding component of the resulting matrix.

Consider two matrices A and B:

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix} \quad B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

The resulting matrix C, obtained by multiplying A and B, will be:

$$C = \begin{bmatrix} (a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31}) & (a_{11}b_{12} + a_{12}b_{22} + a_{13}b_{32}) & (a_{11}b_{13} + a_{12}b_{23} + a_{13}b_{33}) \\ (a_{21}b_{11} + a_{22}b_{21} + a_{23}b_{31}) & (a_{21}b_{12} + a_{22}b_{22} + a_{23}b_{32}) & (a_{21}b_{13} + a_{22}b_{23} + a_{23}b_{33}) \\ (a_{31}b_{11} + a_{32}b_{21} + a_{33}b_{31}) & (a_{31}b_{12} + a_{32}b_{22} + a_{33}b_{32}) & (a_{31}b_{13} + a_{32}b_{23} + a_{33}b_{33}) \end{bmatrix}$$

This code in C maybe useful to understand the implementation:

```
for (int i = 0; i < R1; i++) {
    for (int j = 0; j < C2; j++) {
        result[i][j] = 0;

        for (int k = 0; k < R2; k++) {
            result[i][j] += m1[i][k] * m2[k][j];
        }

        printf("%d\t", result[i][j]);
    }

    printf("\n");
}
```

Procedure:

1. **Initialize Matrices:** Define A, B, and C with appropriate dimensions and initial values.
2. **Create a function to perform Matrix-Matrix Multiplication Algorithm using the stack correctly and the RISC-V calling convention as explained in class**
3. **Call the function you just created with the initialized Matrices; you can find initialization for 3x3,...,50x50 arrays in [datasets.zip](#)**
4. **Check that the implementation is correct by comparing your results with the values generated by a matrix multiplication calculator like <https://matrix.reshish.com/multiplication.php>**

UPLOAD

1. pdf with names of 2-4 project members and describe each person's contribution
2. source code
3. show that your implementation works by comparing the results you are producing with a diff file (use step 4 above for the truth result your implementation should create)