

Lab 6: Graphics Processing Unit

CME433-01

Addi Amaya

Caa746

11255790

Dec 6th, 2021

Contents

[Single-Line Cache 2](#_Toc86866874)

[Microprocessor Layout 2](#_Toc86866875)

[My Program Sequencer 2](#_Toc86866876)

[Questions 4](#_Toc86866877)

[Question 1 4](#_Toc86866878)

[Question 2 4](#_Toc86866879)

[Question 3 4](#_Toc86866880)

[Question 4 4](#_Toc86866881)

[Cache Memory Benefits 6](#_Toc86866882)

[My Program Sequencer 6](#_Toc86866883)

[Questions 11](#_Toc86866884)

[Question 1 11](#_Toc86866885)

[Question 2 11](#_Toc86866886)

[Question 3 12](#_Toc86866887)

# Step 1: matrixMul.c

This will outline the proof that confirms step one is working as intended

## Expected Output

The below photo shows in the input to the matrix calculator

A picture containing black, calculator

Description automatically generated

The below image shows the output to the matrix multiplication between matrix A and matrix B.

Table

Description automatically generated

## Actual Output

After finishing matrixMul.c and compiling the code. The below picture shows the outcome with the same inputs.

Text

Description automatically generated

## Table of Matrixes

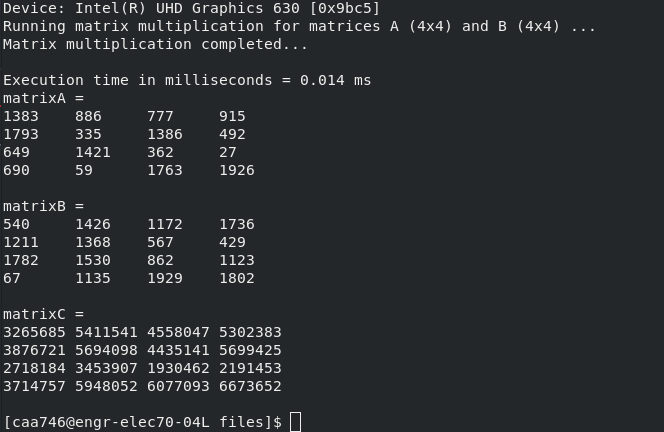
|  |  |
| --- | --- |
| Matrix Size | Time |
| 3x3 | 1ms |
| 10x10 | 5ms |
| 123x123 | 8077ms |
| 210x210 | 40031ms |
| 421x421 | 323071ms |
| 512x512 | 579135ms |

## Conclusion

The matrixMul.c does work and its quite slow when it hits 512x512

# Step 2: matrixMul\_host.c and matrixMul\_kernel.cl

The below photo shows the outcome of a 4x4 matrix to confirm that the outcome is the same as in step 1



## Table of Matrixes

|  |  |
| --- | --- |
| Matrix Size | Time |
| 3x3 | 0.3ms |
| 10x10 | 0.018ms |
| 123x123 | 0.755ms |
| 210x210 | 2.895ms |
| 421x421 | 84.242ms |
| 512x512 | 42.656ms |

# Conclusion

The parallel processing is a magnitude faster than the single thread processing. Using the GPU implementation shows the value of using processing on the GPU and the speed that it can save you. OpenCL implementation on the GPU offers faster processing than the just a single core CPU. There were some strange instances in which bigger matrices results in a faster time. Additionally, Theses calculations were only implemented with square by square matrix multiplication