

|  |
| --- |
| Convolutional Operation on the GPU for Mini Project 1 for DSA 2  CMP202: Data Structures & Algorithms 2023/24 Term 2  Mini-Project : Mini-Project 2 GPU  Student Name: Snow White  Student Number: 2302444 |

*Note that Information contained in this document is for educational purposes.*

Contents

[Introduction 3](#_Toc164631465)

[Application Overview 4](#_Toc164631466)

[Parallelisation Strategy and Implementation 5](#_Toc164631467)

[Performance Evaluation 6](#_Toc164631468)

[Conclusion 7](#_Toc164631469)

# Introduction

The purpose of the application is convolutional image filtering, this was given out in the practical between weeks 7-10 and was then put as an idea for the Mini project 2.

This application is going to be compared with the cpu vs the gpu to see what speeds we get and how different they are if there is any. As well as the speed difference between using things like USM and implementing local memory.

# Application Overview

The application is a Convolutional Operation for image filtering.

The rationale behind the algorithm selection was to implement GPU parallelisation and to implement things like USM, buffers/accessors and implement local memory and tiling for the convolutional operation.

Kernel Execution – Convolution kernel executes on the GPU, performing a significant computational task with appropriate task decomposition to efficiently utilize GPU cores.

Resource Management on GPU – Ensures safe sharing of resources between GPU threads using SYCL accessors and barriers to synchronize memory accesses.

Performance Optimisation – Applies optimization strategies specific to GPU architectures, including the use of SYCL buffers and accessors, Unified Shared Memory (USM), and optimizing memory access patterns to enhance the performance of the application and using local memory.

Performance Analysis – There is a section in the main recording how long the program takes from start to finish which will be outputted to screen once program has been completed, which will be inputted into a graph. There will be a comparison between the CPU running the application and the GPU running the application.

# Parallelisation Strategy and Implementation

**Utilise Threads:**

**Task Decomposition:**

**Kernel Function (Convolution Kernel):**

The applyConvolutionalParallel function submits a SYCL kernel for parallel execution, the kernel function then iterates over each pixel in the image. It computes the convolution by traversing the kernel and applying it to the neighbouring pixels. The convolution is performed in parallel, with each work-item handling one pixel. The work-items work independently, allowing the convolution to be applied simultaneously to different pixels. The convolution operation is applied to each pixel by reading from the input buffer, applying the convolution, and writing the result to the output buffer.

**Resource Sharing Strategies:**

**Unified Shared Memory (USM):**

USM is used for memory management, it allows for easy data sharing between the host and the device.

‘malloc\_shared’ allocates shared memory accessible by both the host and the SYCL device, ‘free’ frees the allocated memory and ‘accessor’ allows the kernel to access the shared memory.

**Performance Enhancement Strategies for GPU:**

Parallelism: SYCL provides parallelism by allowing the execution of multiple threads simultaneously, the convolution operation is executed in parallel which greatly speeds up the process, especially on GPU architectures.

**Data Transfer Optimisation**:

**Memory Access Optimisation:**

**Kernel Optimisation:** The kernel is optimised for parallel execution. Each work-item processes one pixel, which ensures maximum utilisation of the GPU’s resources.

# Performance Evaluation

The GPU used for testing was an Intel (R) UHD Graphics 630

The CPU used for testing was an AMD Ryzen & 5800X 8-Core Processor with 3.8GHZ and 16.0 GB of Ram.

The graph below shows the difference in time taken when the program is run on the CPU to when it is running on the GPU.

# Conclusion