

**МИНОБРНАУКИ РОССИИ**  
**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**  
**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**  
**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**  
**Кафедра математического обеспечения и применения ЭВМ**

**ОТЧЕТ**  
**по лабораторной работе №6**  
**по дисциплине «Параллельные алгоритмы»**  
**Тема: Оптимизация доступа к памяти в модели OpenCL**

Студент гр. 0304

\_\_\_\_\_

Максимов Е.А.

Преподаватель

\_\_\_\_\_

Сергеева Е.И.

Санкт-Петербург

2023

### **Цель работы.**

Познакомиться с моделью памяти в OpenCL. Рассмотреть способы оптимизации доступа к памяти в модели OpenCL.

### **Постановка задачи.**

Реализовать умножение матриц на OpenCL. Произвести сравнение производительности с реализацией на CPU из лабораторной работы №4.

### **Выполнение работы.**

Была реализована программа, выполняющая умножение матриц наивным способом. Для реализации была подключена сторонняя библиотека OpenCL.

Для уменьшения обращений к глобальной памяти использовалась локальная память рабочей группы. Каждая рабочая группа отвечает за одну строку ячеек левой матрицы и один столбец ячеек правой матрицы.

Была произведена оценка производительности CPU и GPU. Для оценки производительности умножения матриц на CPU использовалась программа из лабораторной работы №4, реализующая параллельное умножение матриц при помощи алгоритма Штрассена. Для оценки производительности умножения матриц на GPU была использована созданная в рамках данной лабораторной работы программа.

Для измерения времени работы программы была использована библиотека `chrono`. Результаты тестирования представлены в приложении А.

### **Выводы.**

В ходе лабораторной работы была рассмотрена модель памяти в OpenCL. Были рассмотрены способы управления памятью. Было проведено сравнение скорости выполнения программы на графическом и центральном процессоре. Было определено, что на графическом процессоре программа выполняется значительно быстрее, чем на центральном.

Практическим результатом лабораторной работы является программный код на языке C++, реализующий умножение матриц на графическом процессоре при помощи библиотеки OpenCL.

**ПРИЛОЖЕНИЕ А**  
**ТЕСТИРОВАНИЕ**

Таблица А1 — Сравнение эффективности выполнения программ

№	Размерность матрицы	Время работы с алгоритмом Штрассена (CPU), мс	Умножение матриц (GPU), мс
1	64	60	< 1
2	256	160	2
3	512	417	13
4	1024	2302	98
5	2048	18426	705