## Задача 4.

Исходный код

Репозиторий: https://github.com/Grand-OT/MatMultcuBLASS\_MatMul

Система

GPU: NVIDIA GeForce RTX 3050m.

CUDA: 12.6.

Результаты

Выполнялось сравнение разных алгоритмов умножения матриц для разных типов данных. Так же был выполнен анализ производительности средствами инструмента Nsight Compute. Сравнение выполнялось для матриц размером 1024х1024.

Первый алгоритм: каждый поток вычисляет значение одного элемента результирующей матрицы. Матрицы-множители хранятся в глобальной памяти GPU.

Второй алгоритм: каждый блок выполняет умножение подматриц. Сначала подматрицы копируются в разделяемую память блока, после выполняется умножение.

Третий алгоритм: алгоритм умножения матриц средствами библиотеки cuBLAS.

Таблица 1. Суммарная ошибка для значений типа float

|  |  |
| --- | --- |
| Функция GPU | Длительность вычислений, мс |
| Прямое умножение |  |
| Умножение с разделяемой памятью |  |
| cuBLAS |  |

Таблица 2. Суммарная ошибка для значений типа double

|  |  |
| --- | --- |
| Функция GPU | Длительность вычислений, мс |
| Прямое умножение |  |
| Умножение с разделяемой памятью |  |
| cuBLAS |  |

Использование инструментов профилирования показало проблему низкого использования FMA-инструкций при выполнении умножения cuBLAS для чисел типа double.