

МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МО ЭВМ

ОТЧЕТ
по лабораторной работе №5
по дисциплине «Параллельные алгоритмы»
ТЕМА: ЗНАКОМСТВО С ПРОГРАММИРОВАНИЕМ ГЕТЕРОГЕННЫХ СИСТЕМ
В СТАНДАРТЕ OPENCL

Студент гр. 0303

Мыратгелдиев А. М.

Преподаватель

Сергеева Е. И.

Санкт-Петербург

2023

Цель работы.

Научиться работать с фреймворком OpenCL и понять принципы вычислений на видеопроцессорах.

Задание.

Реализовать расчет фрактала Мандельброта на OpenCL. Визуализировать результат.

Выполнение работы.

Работа была выполнена в среде Microsoft Visual Studio: подключены библиотеки и настроено окружение.

В первую очередь были написаны функции для вычисления множества Мандельброта (*mandelbrot*, *compute_iterations*, *to_color_gray*) на Host. Вычисление точек производится в 100 итераций. Для вывода вычисленных точек в изображение, использовались оттенки серого цвета.

В kernel данная функция выглядит аналогичной, но вместо цикла, в котором пробегаемся по пикселям изображения, используются work-item-ы.

В Host были написаны вспомогательные функции для работы с OpenCL:

- *create_device* – функция, которая возвращает ID первого попавшегося устройства типа GPU. Если таковые устройства не найдены, то возвращает ID устройства на CPU. Если вообще нет устройств на которых можно вычислять при помощи OpenCL – выбрасывается исключение;
- *build_program* – загружает текст кода, которая будет запускаться на kernel, и собирает программу;
- *align* – для выравнивания размера рабочей области изображения (понадобится при делении задач на рабочие группы, кратные размеру всего рабочего пространства/контекста);
- *invoke_kernel* – для запуска kernel-а. Данная функция устанавливает параметры kernel-а, кладет задачу в очередь. В этой же функции

задается размер рабочей группы (work-group) и рабочей единицы (work-item). И в конце читает результат в переменную, которая была передана в качестве аргумента;

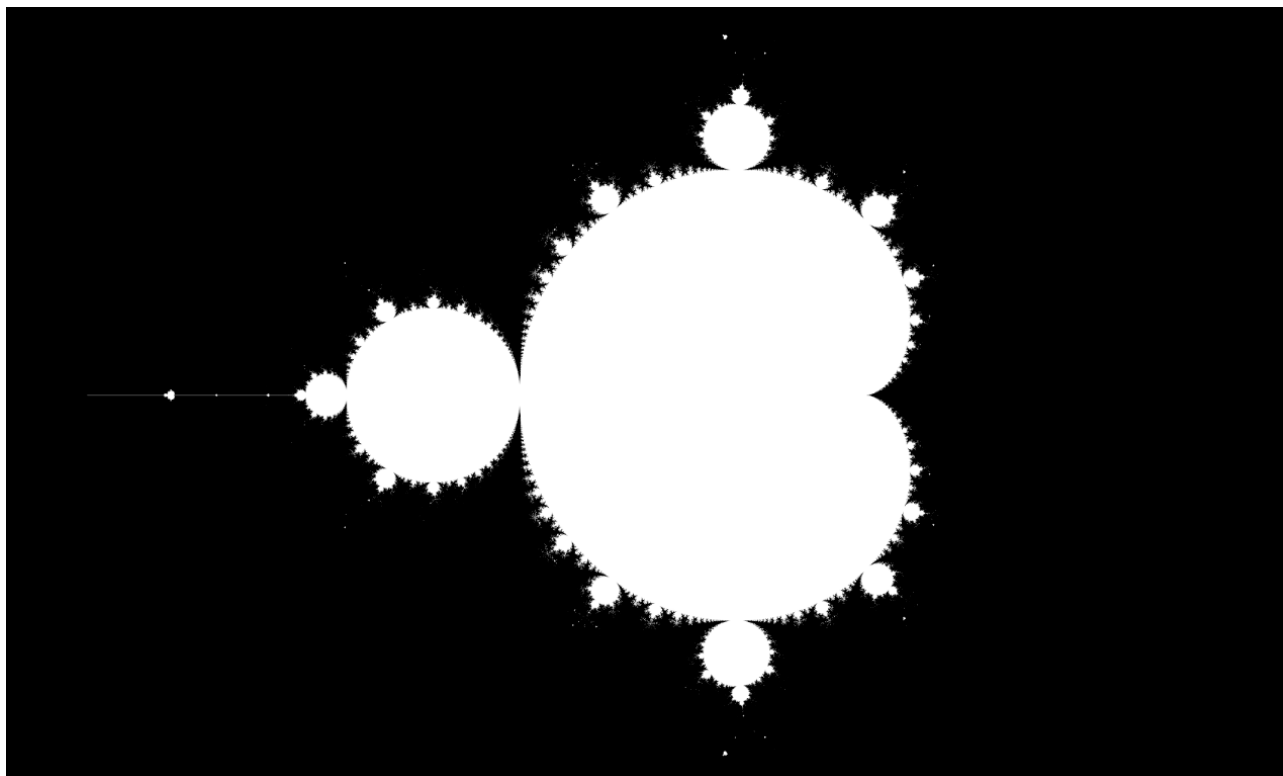
В функции *main* в первую очередь была вызвана функция *create_device*, создан контекст, собрана программа, получен kernel и создана очередь команд. Затем был создан буфер для хранения результатов вычислений, вызвана функция *invoke_kernel*, результат сохранен в изображение и освобождены ресурсы.

Сравним время вычислений на CPU и на GPU меняя размер изображения.

Размер изображения	Время работы на GPU	Время работы на CPU
1200 x 640	1 мс	134 мс
1920 x 1080	2 мс	289 мс
2048 x 2048	4 мс	406 мс
4096 x 2048	8 мс	1101 мс

Как и ожидалось, вычисления на GPU намного быстрее, чем на CPU.

Полученный результат визуализации вычисленных точек в виде изображения в формате .ppm:



Выводы.

В данной лабораторной работе была написана программа, которая вычисляет множество Мандельброта на GPU и на CPU и сравнивает время выполнения. Для вычислений на GPU был использован фреймворк OpenCL - фреймворк для написания компьютерных программ, связанных с параллельными вычислениями на различных графических и центральных процессорах.