Лабораторная работа № 6
Определение параметров видеокарты с поддержкой технологии CUDA в среде Microsoft Visual Studio

Цель работы: познакомиться с особенностями интеграции инструментария CUDA в среде разработки Microsoft Visual Studio, научиться определять основные параметры видеокарт, влияющие на скорость вычислений.

Настройка среды Microsoft Visual Studio

Инструментарий CUDA предоставляет возможность интеграции в состав популярной среды разработки Microsoft Visual Studio, что избавляет разработчика от необходимости напрямую обращаться к компилятору `nvcc` с использованием командной строки или разрабатывать `make`-файлы. Для создания проекта с поддержкой CUDA необходимо (подразумевается, что инструментарий CUDA установлен на машине):

1. Создать новый проект.
2. Добавить к нему необходимые `.cu`-файлы.
3. Для каждого из них указать инструмент, с помощью которого производится компиляция (правый клик по файлу в дереве проекта → Properties → General → Tool → в выпадающем списке выбрать «CUDA Runtime API»):

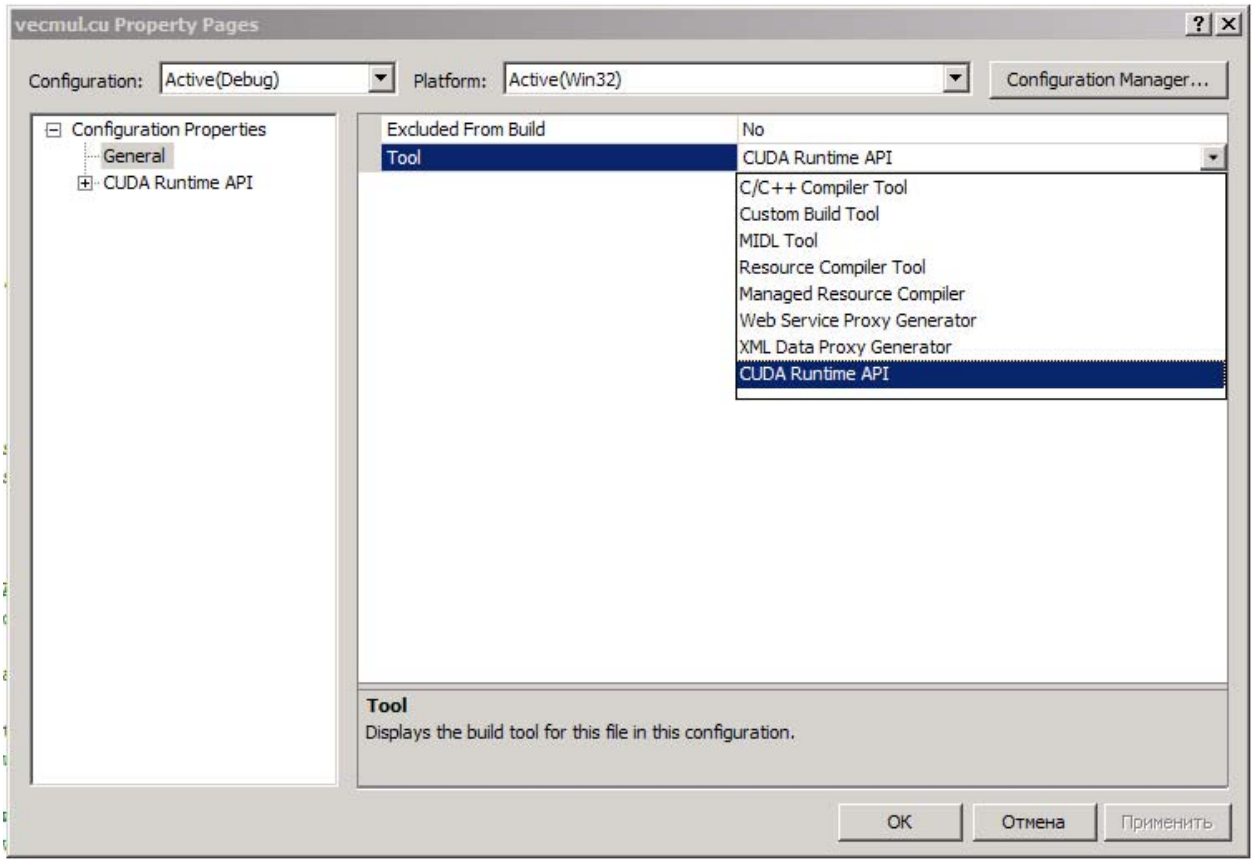

Рис. 1. Выбор типа компиляции для `.cu`-файлов

4. Указать путь к необходимым статически подключаемым библиотекам (выбрать Главное меню → Project → Properties → Configuration properties → Linker → Input → Additional dependencies, указать значение

```
"C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.0\lib\Win32\cudart.lib"
```

(в кавычках, путь показан на примере Windows XP x86!)

Другой вариант – указать в коде программы ссылку на библиотеку:

```
#pragma comment(lib,"cudart.lib")
```

5. Указать путь к расположению инструментария CUDA (выбрать Главное меню → Project → Properties → Configuration properties → Linker → Additional library directories, указать значение `$(CUDA_LIB_PATH)`).
6. Указать путь к расположению заголовочных файлов CUDA (выбрать Главное меню → Project → Properties → Configuration properties → C/C++ → Additional include directories, указать значение

```
"C:\Program Files\NVIDIA GPU Computing Toolkit\CUDA\v5.0\include\"
```

(в кавычках, путь показан на примере Windows XP x86!)

После выполнения указанных действий среда разработки Microsoft Visual Studio будет самостоятельно вызывать компилятор `nvcc` для `.cu`-файлов, осуществлять компиляцию и сборку проекта.

Определение параметров видеокарты

Для определения числа установленных в системе видеокарт с поддержкой технологии CUDA используется функция `cudaGetDeviceCount()`:

```
int device_count;
cudaGetDeviceCount(&device_count);
```

Для определения свойств видеокарты используется функция `cudaGetDeviceProperties()`, принимающая в качестве параметра номер видеокарты и возвращающая в структуре `cudaDeviceProp` интересующие значения:

```
cudaDeviceProp dp;
cudaGetDeviceProperties(&dp, 0);  // Определение параметров GPU с номером 0
```

Например, для определения наименований установленных в системе видеокарт с поддержкой CUDA и их вычислительных возможностей можно использовать следующий код:

```
int device_count;
cudaDeviceProp dp;

cudaGetDeviceCount(&device_count);
cout << "CUDA device count: " << device_count << "\n";

for (int i=0; i<device_count; i++)
{
        cudaGetDeviceProperties(&dp, i);
```

```
        cout << i << ": " << dp.name << " with CUDA compute compatibility " <<
            dp.major << "." << dp.minor << "\n";
}
```

Задание.

1. Создать в среде Microsoft Visual Studio проект, состоящий из пары файлов (`.cpp` и `.cu`) из предыдущей работы. Установить в среде необходимые настройки. Убедиться в том, что проект успешно собирается и запускается.
2. Определить число видеокарт с поддержкой технологии CUDA и следующие параметры видеокарты:
    - наименование;
    - общий объем графической памяти;
    - объем памяти констант;
    - объем разделяемой памяти в пределах блока;
    - число регистров в пределах блока;
    - размер WARP'а;
    - максимально допустимое число потоков в блоке;
    - версию вычислительных возможностей;
    - число потоковых мультипроцессоров;
    - тактовую частоту ядра;
    - частоту памяти видеокарты;
    - объем кэша второго уровня;
    - ширину шины памяти видеокарты;
    - максимальную размерность при конфигурации потоков в блоке и блоков в сетке.

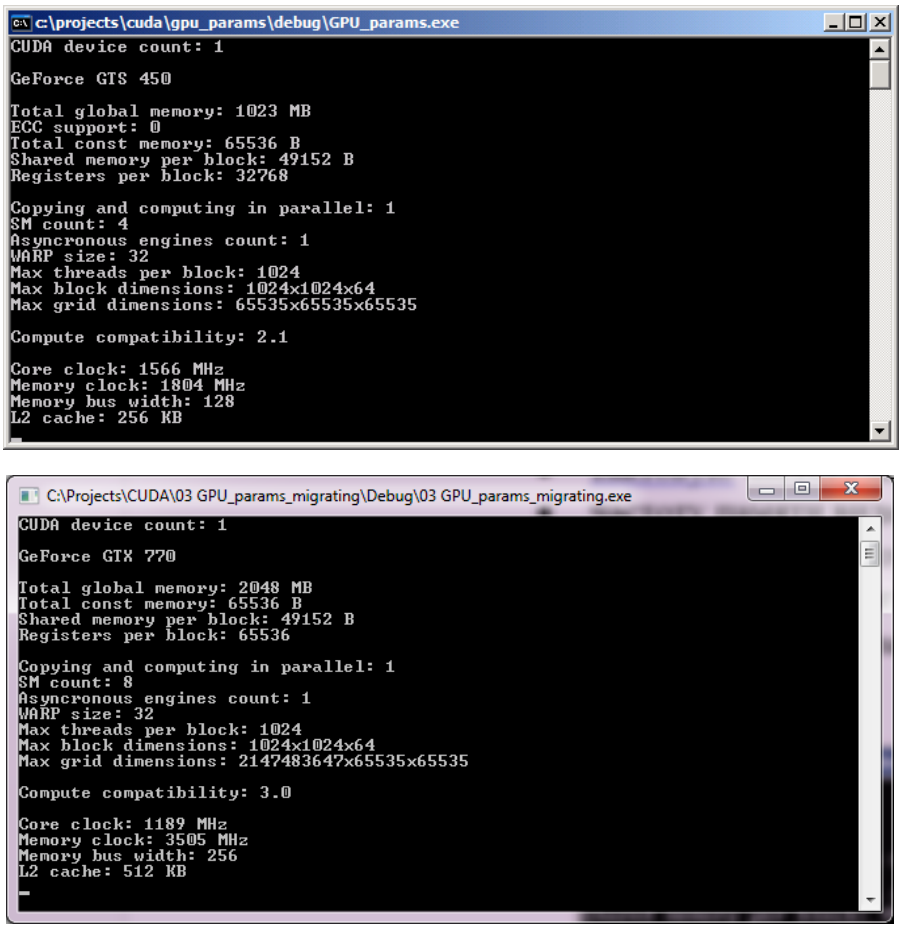

Рис. 2. Основные параметры видеокарты, влияющие на производительность вычислений

Содержание отчета.

1. Титульный лист
2. Цель работы
3. Задание
4. Листинг программы
5. Скриншоты с результатами работы программы
6. Выводы