

**СРЕДСТВА ОТЛАДКИ ИСПОЛНЯЕМОГО КОДА  
ДЛЯ ГРАФИЧЕСКИХ ПРОЦЕССОРОВ**  
Ларочкин Г.И. (Университет ИТМО, г. Санкт-Петербург)  
Научный руководитель – к. т. н, доцент Перминов И.В.  
(Университет ИТМО, г. Санкт-Петербург)

Графические процессоры (ГП) получили широкое распространение как для работы с графикой, так для вычислений общего назначения. Целью данной работы является описание подхода к отладке ГП на уровне ассемблерных инструкций. В результате был получен готовый инструментарий для отладки исполняемого кода, применяющийся на практике группой разработчиков.

**Введение.** Современные языки программирования (ЯП) высокого уровня хорошо подходят для решения большинства задач разработки алгоритмов для GPGPU вычислений. Такие ЯП поставляются совместно с набором инструментов для разработчиков. Однако возможно значительно увеличить производительность за счет использования специализированных инструкций ассемблера и жесткого контроля над выполнением программы, но многие разработчики предпочитают не использовать ассемблер ввиду отсутствия инструментария.

Вследствие наличия в графических процессорах большого количества вычислительных блоков, каждый из которых выполняет векторные операции, исполняемая программа (шейдер) обладает высокой степенью параллельности и асинхронности выполнения, которые в свою очередь накладывают свои требования и ограничения для отладчиков.

**Основная часть.** Традиционный процесс отладки состоит из трех основных этапов – приостановка программы в точке останова, сохранение контекста исполнения и возобновление программы. Такой отладчик требует дополнительной поддержки оборудования, драйверов и генерации отладочной информации, чего нет на уровне ассемблера.

В данной работе предлагается альтернативный подход, в котором точка останова реализуется как фрагмент кода, попадая в который сохраняются необходимые регистры и состояния потоков исполнения в предварительно выделенную область памяти, а после продолжает исполнение основной программы. Поскольку ассемблирование и отправка шейдеров для исполнения на графическом процессоре обычно занимают незначительное количество времени, на практике такой подход может быть неотличим от традиционной отладки.

Он прозрачен для оборудования и не требует специальной поддержки драйверов. Еще одним преимуществом является то, что символические имена разрешаются на уровне ассемблера и для такого отладчика не нужна таблица символов, вследствие чего в именах наблюдаемых регистров могут быть использованы препроцессорные конструкции, а точка останова находится в макросах.

**Выводы.** В результате был разработан отладчик исполняемых программ на уровне ассемблера, а также дополнительные инструменты автоматизации процесса сборки, вставки точки останова и выделения дополнительной памяти для наблюдаемых регистров. Применение данных инструментов позволило значительно ускорить процесс разработки шейдеров, а также обнаружить большое количество программных ошибок.

Ларочкин Г.И. (автор)

Подпись

Перминов И.В. (научный руководитель)

Подпись