

УДК 004.431.4

## ПОДДЕРЖКА СЛОЖНЫХ УПРАВЛЯЮЩИХ КОНСТРУКЦИЙ И АРХИТЕКТУРЫ RDNA В ДЕКОМПИЛЯТОРЕ ДЛЯ AMD GPU

**Громов Д.С.** (Национальный исследовательский университет ИТМО), **Михайленко К.И.**  
(Национальный исследовательский университет ИТМО)  
**Научный руководитель – к.т.н. Лукин М.А.**  
(Национальный исследовательский университет ИТМО)

Цель работы – модификация декомпилятора, транслирующего ассемблерный код в эквивалентный исходный код на языке OpenCL, улучшение архитектуры и расширения функционала.

### Введение.

С 2008 года развивается стандарт OpenCL — открытая среда программирования графических процессоров и других ускорителей вычислений. На данный момент существует только один инструмент для анализа ассемблерного кода, который бы позволял обнаруживать оптимизации, применённые компилятором, и проводить обратную разработку программ – OpenCLDecompiler, разрабатываемый с 2020 года. Остальные известные декомпиляторы не поддерживают видеокарты AMD, поэтому их использование в данной работе невозможно. OpenCLDecompiler позволяет осуществлять поддержку программ, для которых нет исходного кода, а также даёт возможность проверять программы на наличие недокументированных функций. Однако данный инструмент ещё не может декомпилировать произвольное входное ядро, поэтому нуждается в доработке.

### Основная часть.

Для развития существующего декомпилятора ассемблерного кода видеокарт AMD были выделены следующие направления:

1. добавление поддержки новой архитектуры RDNA,
2. доработка архитектуры декомпилятора,
3. добавление нового шаблона для конструкции if-else,
4. добавление шаблона для циклов,
5. добавление поддержки деления.

Поддержка RDNA потребовала реверс инжиниринга и исследования ассемблерного кода. В результате были определены различия в ABI (application binary interface) и список и семантика новых ассемблерных инструкций, поддержка которых необходима для успешной декомпиляции большинства ядер.

Второе направление выделено в отдельную задачу, так как его цель – упростить добавление новых ассемблерных инструкций и повысить качество кода, поэтому она является подготовительной и важна для дальнейшего расширения функционала.

Добавление новой архитектуры видеокарт также повлияло на структуру декомпилятора – разбор ассемблерного кода был выделен в отдельный модуль в результате чего стало возможным проводить разбор ядер для разных архитектур.

Направления три и четыре расширяют функциональные возможности декомпилятора, позволяя улучшить качество результата работы декомпилятора, за счет реализации распознавания комплексных ассемблерных конструкций и их декомпиляции в простые исходные аналоги.

В ходе работы по этим направлениям были решены следующие задачи:

- добавление поддержки новой архитектуры RDNA,
- добавление поддержки новых ассемблерных инструкций,
- реализация декомпиляции деления,
- реализация декомпиляции циклов,

- доработка поддержки конструкции if-else,
- реализация обработки произвольного числа параметров ядра,
- доработка поддержки векторных типов,
- улучшение архитектуры декомпилятора.

**Выводы.** Результаты внедрены по месту работы. Соответственно тестирование результатов происходит не только на синтетических ядрах, но и в реальных условиях. Проверка работоспособности и чистоты кода разрабатываемых доработок системы в проекте осуществляется с помощью тестов и линтера запускаемых средствами Github actions.

Громов Д.С. (автор)

Подпись

Лукин М.А. (научный руководитель)

Подпись