

УДК 004.415.2.031.43

**Оптимизация целевого алгоритма в системе автоматического проектирования специализированных вычислителей NITTA**

**Прохоров Д.А.** (Национальный исследовательский университет ИТМО)

**Научный руководитель – кандидат технических наук Пенской А.В.**

(Национальный исследовательский университет ИТМО)

В работе рассматриваются оптимизации целевого алгоритма в системе автоматического проектирования специализированных вычислителей NITTA, а именно: Peephole-оптимизация арифметических операций для вычислительного блока аккумулятора и свертка констант.

Использование языков высокого уровня при разработке алгоритмов позволяет значительно сократить время и трудоемкость работы. В тоже время, в отличие от низкоуровневых языков программирования, оно не позволяет оптимизировать алгоритм под особенности конкретной вычислительной системы. Это в первую очередь актуально для специализированных вычислителей, когда их организация может меняться на лету, а количество особенностей превышает возможности рядового разработчика. Далее будет рассмотрен ряд оптимизаций во времени компиляции в САПР NITTA.

**NITTA (Not Instruction Transport Triggered Architecture)** – микроархитектура и метод синтеза для оркестрации низкоуровневых многопроцессорных систем, реализованный в виде САПР с таким же названием.

Этапы работы САПР NITTA:

1. Задается алгоритм на языке программирования высокого уровня Lua, который трансформируется в граф потока данных (Data Flow Graph, DFG), где узлы – функции, ребра – пересылки данных.
2. Определяется микроархитектура, где каждый элемент является вычислительным блоком (Processor Unit, PU), который выполняет набор заданных типов операций для хранения, обработки или передачи данных. Вычислительные блоки объединены общей шиной, что позволяет передавать данные между ними.
3. Процесс синтеза на основе DFG и микроархитектуры позволяет автоматически назначить функции прикладного алгоритма вычислительным блокам и сформировать расписание пересылок данных между ними. Процесс синтеза также включает изменение DFG с целью оптимизации и/или организации вычислительного процесса.
4. Генерация целевой системы включая аппаратную часть (синтезируемый проект на языке Verilog) и программное обеспечение (управляющая программа и начальные конфигурации вычислительных блоков).
5. Верификация целевой системы.

Под оптимизацией в работе понимается процесс сокращения времени выполнения работы программы за счет более эффективного использования вычислительных ресурсов без изменения её функциональности.

**Peephole-оптимизация арифметических операций** (сложение, вычитание, инверсия). Обычно, когда в программе существует несколько подряд арифметических операций, например, когда мы хотим сложить три числа, то парсер разделяет данную операцию на две операции сложения. Однако, аппаратные возможности вычислительного блока аккумулятора позволяют проводить сложение  $N$  чисел за одну операцию без лишних пересылок данных. Нахождение таких цепочек операций и их объединение в единую операцию является задачей данной оптимизации.

**Свертка констант** позволяет уменьшить количество вычислений во время исполнения за счет вычисления операций на этапе компиляции в случае, если данные доступны. Это также

упрощает процесс написания программ, так как нет необходимости высчитывать все константы вручную.

Изначально, эти оптимизации планировалось реализовать на уровне формирования DFG (этап 1), однако от этой идеи пришлось отказаться, так как она не предоставляет нужной степени гибкости. К примеру, слишком раннее объединение арифметических операций не позволит осуществить их параллельное вычисление. Поэтому было принято решение внедрить оптимизации на этап процесса синтеза (этап 3) в виде операций рефакторинга, который также как и оптимизации, изменяет DFG, но не изменяет функциональность работы алгоритма.

Таким образом достигается нужная степень гибкости, потому что за оптимизацию отвечает процесс синтеза, который располагает всеми данными для принятия решения. Процесс синтеза может происходить в автоматическом или ручном режиме, когда пользователь сам выбирает, какую оптимизацию и где он должен сделать, а дальше система завершает процесс синтеза с этими условиями.

Для внедрения новых методов оптимизации (рефакторинга), требуется реализация функций, представляющих список опций возможных изменений для DFG и применяющих определенное изменение к текущему DFG.

Алгоритм reephole-оптимизации арифметических операций представлял собой следующую последовательность действий: нахождение зависимых переменных и объединение их в кластеры, преобразование кластера операций сложения и вычитания в кластер суммирования, замена операций в DFG на новые.

Алгоритм свёртки констант: создание кластеров на основе информации о том, что входные данные функции состоят только из констант, вычисление данных кластеров с помощью реализованного модуля функциональной симуляции, замена в DFG функций с константами из кластеров на вычисленные на этапе процесса синтеза.

Были реализованы, внедрены и протестированы оптимизации, позволяющие производить reephole-оптимизацию арифметических операций, а также производить свёртку констант на этапе процесса синтеза прикладного алгоритма.

Прохоров Д.А.

Подпись

Пенской А.В.

Подпись