Автоматическое отображение программ на процессор с ПЛИС-ускорителем

Д.В. Дубров[1], С.А. Рошаль[1], Р.Б. Штейнберг[1], Б.Я. Штейнберг[1]

Южный федеральный университет[1]

В работе рассматривается задача автоматического отображения высокоуровневых программ на процессор с ПЛИС-ускорителем. Для такого отображения разрабатывается и используется генератор HDL-кода из внутреннего представления распараллеливающей системы.

1. Введение

Конвейерные вычислители используются во многих программно-аппаратных комплексах и показывают на некоторых задачах производительность значительно более высокую, чем универсальные процессоры. Существенный прогресс в развитии конвейерных вычислителей внесли технологии программируемых логических интегральных схем (ПЛИС). Традиционно, конвейерные вычислители используются в программно-аппаратных комплексах. Для применения к широкому классу программ разрабатываются вычислители с архитектурой программируемого конвейера или реконфигурируемые конвейеры. Известны работы по реконфигурируемым конвейерам [1, 3, 4]. Узкое место таких вычислителей – долгое время перепрограммирования ПЛИС. В данной работе описывается проект, разрабатываемый на мехмате ЮФУ на основе распараллеливающей системы ДВОР (Диалоговый высокоуровневый оптимизирующий распараллеливатель программ). Этот проект ориентирован на расширение множества разработчиков и пользователей прикладных программ для параллельно-конвейерных суперкомпьютеров. Основная идея проекта связана с созданием генератора VHDL-кода из внутреннего представления распараллеливающей системы.

2. Конвертор программ языка Си в язык VHDL

На входе конвертора текст на языке Си, описывающий алгоритм вычислений создаваемого генератора. Вводятся также некоторые характеристики создаваемой схемы: диапазоны значений входных и выходных переменных, допустимая площадь на кристалле. На выходе HDL-описание электронной схемы. Генератор позволяет создавать встроенные системы, конвейерные вычислители, параллельно-конвейерные вычислители, автоматы, системы с программируемой архитектурой и др. По диапазонам значений входных и выходных данных, определяются диапазоны промежуточных переменных и определяется минимальная разрядность аргументов вычислительных устройств проектируемой схемы. Генератор имеет свою библиотеку VHDL-описаний базовых операций и функций. В этой библиотеке представлены базовые операции и функции для аргументов различной разрядности, которые сопровождаются оценками количества тактов и занимаемой площади кристалла [5].

В настоящее время на рынке представлен целый ряд программных продуктов, обладающих возможностью преобразования высокоуровневых языков в языки описания электронных схем. В следующей таблице приведена информация об основных из них. Как можно видеть, к данному направлению проявляют интерес как небольшие исследовательские коллективы (разработки C-to-Verilog, CoolKit), так и ведущие разработчики средств проектирования электронных устройств (Xilinx, Mentor Graphics и др.) В последнем случае наблюдается общая для ИТ-индустрии тенденция, выраженная в приобретении крупными компаниями интеллектуальной собственности независимых разработчиков. Так например, язык высокого уровня Handel-C был изначально разработан в вычислительной лаборатории Оксфордского университета [6], после чего в 1996 г. был коммерциализирован созданной компанией Embedded Solution Limited (переименованной позднее в Celoxica), которая в 2008 г. была приобретена компанией Agility, чьи

активы год спустя были в свою очередь приобретены Mentor Graphics. В настоящее время Handel-C поддерживается их продуктом DK Design Suite. Другим подобным примером является язык Streams-C, разработанный в Лос-Аламосской национальной лаборатории и лицензированный компании Impulse Accelerated Technologies под именем Impulse C.

Таблица 1. Обзор основных средств, преобразующих высокоуровневые программы в HDL.

Название	Разработчик	Web-страница	Время разработки	Тип лицензии	Входные языки	Выходные языки
Catapult C	Mentor Graphics	http://www.mentor.com/esl/catapult/overview	с 2004 г.	коммерческая	C, C++	RTL
Vivado Design Suite	Xilinx	http://www.xilinx.com/products/design-tools/vivado/	с 2012 г.	коммерческая	C, C++, MATLAB, Simulink	VHDL, Verilog
Impulse CoDeveloper	Impulse Accelerated Technologies	http://www.impulseaccelerated.com/products.htm	с 2003 г.	коммерческая	C, Impulse C	VHDL, Verilog
Altium Designer	Altium	http://www.altium.com/en/products/altium-designer	с 2008 г.	коммерческая	C	VHDL, Verilog
Mitrion SDK	Mitrionics	http://www.mitrionics.com/?page=mitrion-sdk	с 2005 г.	коммерческая	Mitrion-C	VHDL
MATLAB + Simulink + HDL Coder	MathWorks	http://www.mathworks.com/products/simulink/	с 2007 г.	коммерческая	MATLAB, Simulink, Stateflow	VHDL, Verilog
C-to-Verilog	Nadav Rotem (Haifa University)	http://www.c-to-verilog.com/	с 2009 г.	GPL	C	Verilog
CoolKit	СПбГУ	http://oops.math.spbu.ru/projects/coolkit	с 2008 г.	“No license”	HaSCoL	VHDL

3. Диалоговый высокоуровневый оптимизирующий распараллеливатель программ (ДВОР)

Одна из особенностей распараллеливающей системы ДВОР – высокоуровневое внутреннее представление, именуемое Reprise. Генерация описания конвейерных электронных схем из регистровых внутренних представлений (GCC, LLVM) представляется очень сложной. Система ДВОР содержит автоматическое построение графа вычислений программы [4], который является промежуточным представлением между Reprise и HDL-описанием конвейерной схемы.

Привязка конвертора языка программирования Си в язык описания электронных схем к распараллеливающей системе дает возможность использования библиотеки преобразований высокоуровневых программ. Благодаря этому, исходный код программы Си может быть преобразован в другие эквивалентные программы. Это позволяет по исходному коду сгенерировать несколько эквивалентных электронных схем, а не одну единственную. Это важно, поскольку к электронным схемам предъявляются различные требования, иногда противоречащие одно другому: минимизация площади кристалла, быстродействие, энергопотребление,…

4. Отображение гнезд циклов на программируемый вычислитель

Параллельные вычисления, в том числе и конвейерные, следует применять к долго считаемым фрагментам программ. В данной работе будем рассматривать ускорение фрагментов кода, содержащих гнезда программных циклов. Конвейеризуется самый глубоко вложенный цикл

гнезда. Счетчики более высоко вложенных циклов могут рассматриваться как параметры, определяющие узел кластера, на ускорителе которого должен выполняться самый глубоко вложенный цикл (этот же цикл для разных значений счетчиков внешних циклов выполняется на разных узлах).

Основная идея проекта следующая. Разрабатываемая система получает на входе программу языка Си, содержащую гнездо циклов. Гнездо циклов преобразуется к виду, удобному для отображения распараллеливающей системой на конвейерную архитектуру. Самый вложенный цикл гнезда преобразуется в VHDL-описание конвейера конвертором C2HDL. Программируемая часть вычислительного узла прожигается полученной схемой. В исходном гнезде циклов самый вложенный цикл заменяется обращением к конвейерному ускорителю, на котором этот цикл будет вычисляться.

Условие эффективности использования ПЛИС-ускорителя – время передачи данных на ускоритель должно быть существенно меньше времени обработки этих данных. Учитывая, что при современных технологиях время пересылки одного четырехбайтного числа между чипами на порядок дольше умножения чисел, можно заключить, что в ускоряемом фрагменте кода количество операций над данными должно в несколько десятков раз превосходить количество данных. Значит, конвейеризуемый самый вложенный цикл должен содержать в своем теле несколько десятков операций. Необходимое для эффективности отношение количества операций к количеству данных существенно снижается при использовании коротких чисел (например, двух- или однобайтных).

5. Заключение

В данной работе частично реализованный проект создания компилятора с языка высокого уровня на процессор с ПЛИС-ускорителем. Этот проект ведет к решению проблемы создания высокоуровневых языков для вычислительных устройств с архитектурой программируемого конвейера.

Литература

1. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений //М., «Янус-К», 2003, 380 с.

2. Штейнберг Р. Б. Отображение гнезд циклов на многоконвейерную архитектуру, Программирование, 2010, № 3. Steinberg R. Mapping loop nests to multipipelined architecture. // MAIK Nauka/Interperiodica distributed exclusively by Springer Science+Business Media LLC., May 2010, Vol 36, №3, pp 177-185.

3. Яджак М.С. Высокопараллельные алгоритмы и методы для решения задач массовых арифметических и логических вычислений // Диссертация на соискание ученой степени д.ф.-м.н. Институт прикладных проблем механики и математики. Львов. 2009г – 298с. (на украинском языке).

4. Bondalapati K. Modeling and Mapping for Dynamically Reconfigurable Hybrid Architecture. Ph.D. Thesis, University of Southern California, August, 2001.

5. Dubrov D.V., Roshal A.V., Generating Pipeline Integrated Circuits Using C2HDL Converter // Proceedings of IEEE East-West Design & Test Symposium (EWDTS'2013), Rostov-on-Don, Russia, September 27-30, 2013, p. 215-219.

6. Self R. P., Fleury M., Downton A. C. A Design Methodology for Construction of Asynchronous Pipelines with Handel-C, IEEE Software, 1988, Vol. 150, pp. 39-47.