

На правах рукописи

Штейнберг Роман Борисович

Автоматическое отображение программ
на конвейерные и многоконвейерные архитектуры

05.13.11 – математическое и программное обеспечение вычислительных машин,
комплексов и компьютерных сетей

АВТОРЕФЕРАТ

диссертации на соискание ученой степени кандидата
физико-математических наук

Москва – 2012 год

Работа выполнена на факультете математики, механики и компьютерных наук Южного федерального университета.

Научный руководитель:

заслуженный работник высшей школы РФ, кандидат физико-математических наук,
профессор Ерусалимский Яков Михайлович.

Официальные оппоненты:

доктор физико-математических наук,
Лацис Алексей Оттович.
доктор технических наук,
Лебедев Валентин Григорьевич.

Ведущая организация:

Санкт-Петербургский государственный университет.

Защита состоится 13.03.2012 в 11.00 на заседании диссертационного совета Д 002.024.01 Института прикладной математики имени М.В. Келдыша РАН по адресу: 125047, Москва, Миусская пл., 4.

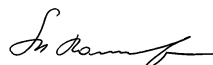
С диссертацией можно ознакомиться в библиотеке Института прикладной математики имени М.В. Келдыша РАН.

Автореферат разослан 09.02.2012

Ученый секретарь

диссертационного совета

доктор физико-математических наук



Полилова Т.А.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Сегодня с использованием высокопроизводительных вычислительных систем решаются задачи прогноза погоды, моделирования межмолекулярных взаимодействий, биоинформатики, нефтегазовой разведки, наноинженерии, механики, гидродинамики и т.д. Разработка параллельного программного обеспечения (ПО), так же как и разработка инструментов для создания параллельного ПО являются приоритетными направлениями научных исследований в России и других странах.

Суперкомпьютерные архитектуры, допускающие конвейерные вычисления, широко используются во всем мире. Программные конвейеры применяются в работе большинства современных процессоров производства Intel, AMD, Apple. Для построения специализированных вычислителей, таких как бортовые компьютеры, устройства фильтрации сигналов и др., активно используется конвейерный подход.

Сегодня в России, как и в других странах мира, ведутся исследования в области и аппаратного, и программного обеспечения суперкомпьютеров, в том числе связанные с перестраиваемыми архитектурами (reconfigurable architecture). Компьютеры с такой архитектурой позволяют создавать перестраиваемые (программируемые) конвейеры, показывающие большую эффективность при обработке информационно сильно связанных задач. В этой архитектуре основной идеей является конфигурирование системы для организации вычислений по заданной программе. Как правило, для этого настраиваются связи между элементарными вычислителями внутри компьютера. При создании компиляторов с языков высокого уровня для таких архитектур, желательно организовать автоматическое формирование связей между вычислителями в соответствии с потоком данных в компилируемой программе. В противном случае перевод программы в машинный код будет осуществляться вручную и может занимать время

порядка месяца вместо нескольких минут. Примером проекта по созданию суперкомпьютера с архитектурой перестраиваемого конвейера является проект Merrimac.

Настройку связей между вычислителями в компьютерах с программируемой архитектурой удобно реализовывать на базе ПЛИС (программируемые логические интегральные схемы, FPGA). Конфигурирование (программирование) ПЛИС осуществляется с помощью специальных программных пакетов, поставляемых производителем ПЛИС. Эти пакеты получают на вход описание схемы и выполняют настройку. Этот процесс называется синтезом схемы по ее описанию. Наиболее распространенными языками для описания схем являются VHDL и VeriLog.

Одним из главных преимуществ ПЛИС по сравнению с универсальными ЦПУ (CPU) и графическими процессорами (GPGPU) является более эффективное использование ресурсов (площадь на кристалле, количество транзисторов и т.п.). Это преимущество позволяет получать большую скорость выполнения операций и энергоэффективность. Единственным, известным сегодня способом такого эффективного использования является построение сверхдлинных конвейеров, структура связей между которыми специализируется под конкретный алгоритм. В связи с вышесказанным, становится ясно, что развитие архитектур на базе ПЛИС может оказаться качественным шагом по сравнению с многократным копированием ядер CPU и размещением их на одном кристалле. Это понимают и в России, и во всем мире.

Нужно отметить, что создавать программы на языках описания схем видится неудобным, так как тестирование и верификацию алгоритмов удобнее делать на языках высокого уровня, таких как, например, C. Кроме того, культура программирования на этих языках уже сложилась и сообщество программистов пишущих на C-подобных языках очень велико. Таким образом, возникает необходимость отображать программы с языков высокого уровня на конвейерные вычислители на базе ПЛИС.

В последнее время в области ПО для рынка программируемых суперкомпьютеров происходят существенные изменения. Например, 1 сентября 2010 г. журнал HPCWire опубликовал пресс-релиз программного продукта C-to-FPGA, который разработан компанией Stone Ridge Technology в рамках проекта ImpulseC. Этот продукт предназначен для генерации HDL-описаний в RTL-форме на основе алгоритма, написанного на языке высокого уровня. По оценкам экспертов, это позволит сэкономить до 2/3 времени разработки проектов на ПЛИС. Есть и другие проекты, предназначенные для автоматизации программирования ПЛИС, но пока еще во многих случаях качество генерируемого этими продуктами кода неудовлетворительно.

В данной работе рассматриваются конвейерные вычисления и автоматизация разработки программ, которые могли бы быть выполнены конвейерно. Рассматривается отображение гнезд циклов на модель многоконвейерной вычислительной системы. Тем самым описывается класс программ, которые могут быть эффективно выполнены на суперкомпьютерах, допускающих многоконвейерные вычисления, и основы разработки распараллеливающих компиляторов на такие архитектуры.

Предметом исследования являются отображения программ, написанных на языках высокого уровня, на конвейерные и многоконвейерные вычислители.

Целью диссертации является разработка методов автоматизации отображения высокоуровневых программ на конвейерные и многоконвейерные вычислительные архитектуры.

К достижению этой цели приводит решение следующих **задач**:

- 1) автоматическое определение характеристик конвейера, таких как интервал инициализации итераций, буферные задержки, обратные дуги;

- 2) автоматическое составление расписания работы каждого конвейера в рамках многоконвейерной модели вычислений;
- 3) разработка алгоритмов синхронизации конвейеров в рамках многоконвейерной модели вычислений;
- 4) разработка алгоритмов эффективного автоматического отображения программ на многоконвейерную модель вычислений;
- 5) разработка программных средств автоматически выполняющих расчеты характеристик конвейера и расчет задержек в стартах конвейеров, позволяющих синхронизировать эти конвейеры.

Методы исследований

В процессе решения рассматриваемых задач использовались методы теории преобразований программ, теории графов, линейной алгебры. При разработке программного обеспечения использовался объектно-ориентированный подход к программированию.

Основные положения, выносимые на защиту:

1. Алгоритмы автоматической синхронизации взаимодействия конвейеров в многоконвейерной вычислительной системе для случая многомерного гнезда циклов.
2. Алгоритмы и программная реализация автоматической синхронизации взаимодействия конвейеров для случая тесного двумерного гнезда циклов.
3. Модификация и обоснование формулы расчета интервала инициализации итераций.
4. Алгоритмы и программная реализация автоматического расчета интервала инициализации итераций, буферов задержек, выявления обратных дуг, составления расписания в рамках рассматриваемой модели вычислений.

5. Алгоритмы и программная реализация линейаризации выражений для определения информационных зависимостей и оптимизации программ, в том числе для конвейеризации циклов.

Научная новизна

1. Впервые рассмотрена задача отображения многомерных гнезд циклов как тесных, так и нетесных на многоконвейерную архитектуру с использованием информации о зависимостях, описанной с помощью решетчатых графов. Кроме того, этот подход видится наиболее общим, так как решетчатые графы, по сравнению с другими графовыми моделями, более полно описывают зависимости в многомерных циклах.

2. Предложен упрощенный алгоритм отображения на многоконвейерную архитектуру двумерных гнезд циклов. Указанный алгоритм применим к меньшему классу программ, чем общий, но при этом решает рассматриваемую задачу эффективнее. Этот алгоритм отображения, в частности, применим к программам, решающим задачи цифровой фильтрации сигналов, рассмотренные в работах М.С. Яджака, и позволяет вычислять характеристики конвейера автоматически.

3. Предложен новый алгоритм линейаризации выражений на этапе компиляции.

Практическая значимость

Алгоритмы и методы, содержащиеся в диссертации, могут быть использованы при разработке распараллеливающих компиляторов или диалоговых инструментов разработки программ.

Описанные алгоритмы уже используются в реализации проекта C2HDL [13, 15], разрабатываемого группой ОРС (www.ops.rsu.ru). Этот конвертер предполагает по программе на языке С генерацию семейства алгоритмически эквивалентных VHDL-описаний микросхем с разными параметрами синтеза. Итоговое решение по выбору наиболее качественного

описания должен осуществлять специалист-схемотехник, при этом конвертер будет выдавать семейство описаний, в котором будут исключены заведомо неэффективные варианты.

В процессе программной реализации алгоритмов, описанных в диссертации, в рамках проектов ОРС (Открытая распараллеливающая система) и ДВОР (Диалоговый высокоуровневый оптимизирующий распараллеливатель), автору потребовалось выполнить несколько вспомогательных подпроектов. Так, например, была разработана библиотека классов «линейные выражения», включающая алгоритм линеаризации. Эта библиотека активно используется другими разработчиками при построении графа информационных связей, генерации кода MPI и других подпроектов ДВОР.

Система тестирования, описываемая в диссертации, использовалась для тестирования преобразований в ОРС, а также при тестировании высокопроизводительного программного комплекса HPC-NASIS.

Внедрение результатов работы

Результаты работы реализованы в виде программных модулей в рамках проектов ОРС и ДВОР.

Система автоматизации тестирования программ по принципу «черного ящика» использовалась при тестировании высокопроизводительного программного комплекса HPC-NASIS.

Визуализация программы построения графа вычислений и расчета характеристик конвейера используется в составе «Тренажера параллельного программиста» в учебном процессе на мехмате ЮФУ в спецкурсе «Параллельные вычисления и преобразования программ».

Апробация работы

Изложенные в диссертации результаты обсуждались на международных и российских научно-технических конференциях:

международная научно-техническая конференция «Интеллектуальные и многопроцессорные системы ИМС-2003» (2003 г., Дивноморское); научно-методическая конференция «Современные информационные технологии в образовании: Южный федеральный округ» (2004 г., Ростов н/Д); всероссийская научно-техническая конференция «Параллельные вычисления в задачах математической физики» (2004 г., Ростов н/Д); международная конференция «Математика. Экономика. Образование» (2005 г., Ростов н/Д); всероссийская научная конференция «Научный сервис в сети Интернет: технологии распределенных вычислений» (2005 г., Новороссийск); VI международная конференция памяти академика А.П. Ершова «Перспективы систем информатики», рабочий семинар «Наукоемкое программное обеспечение» (2006 г., Новосибирск); международная конференция PACO-2006 (2006 г., Москва); международная конференция IEEE East & West Design and Test (2009 г., Москва); международная суперкомпьютерная конференция «Научный сервис в сети Интернет: суперкомпьютерные центры и задачи» (2010 г., Новороссийск); международная конференция «Параллельные вычисления и задачи управления» PACO'2010. (2010 г., Москва).

Достоверность полученных результатов подтверждается строгими математическими формулировками и доказательствами, программно реализованными моделями и вычислительными экспериментами, тестированием разработанных программных модулей.

Личный вклад

Постановка задачи создания многоконвейерной модели вычислений, расчета параметров конвейеризации, а также отображения многомерных гнезд циклов на многоконвейерную архитектуру с помощью анализа информационных зависимостей, описанных решетчатыми графами, принадлежит Б.Я. Штейнбергу. Все теоретические результаты получены

автором лично. Программная реализация графа вычислений и алгоритмов, описанных в диссертации, выполнена лично, велась в рамках проектов OPC и ДВОР и использовала смежные проекты, выполненные другими разработчиками группы OPC, в особенности проекты В.В. Петренко и Р.И. Морылева.

Все рисунки в данной диссертации выполнены с помощью программных средств, в разработке которых автор принимал участие. Часть из них являются экранными формами, созданными с помощью OPSDemo 3, OPSDemo 5 — программных систем, сделанных в рамках проекта Открытая распараллеливающая система (OPC). Другие рисунки — с помощью макросов, написанных на VBA под MS Word лично автором.

В совместных работах [7, 8, 11, 15–18] личным вкладом автора является разработка алгоритмов отображения программ на многоконвейерную модель вычислений, алгоритмов построения и анализа графа вычислений. В совместных работах [16, 18] также описывается и используется линеаризация выражений, выполненная лично автором. В совместных работах [13, 14] описывается тестирующая система, в разработке которой автор принимал участие вместе с соавторами. В частности, формат представления файлов с данными разработан лично автором.

Гранты, поддерживавшие исследования диссертации.

- НИОКР «Разработка системной поддержки высокопроизводительного программного комплекса для квантово-механических расчетов и моделирования наноразмерных атомно-молекулярных систем и комплексов», госконтракт 02.524.11.4005, шифр: 2008-04-2.4-15-02-003.
- ФЦП «Научные и научно-педагогические кадры инновационной России» по лоту «Проведение научных исследований коллективами научно-образовательных центров в области информатики» по теме: «Диалоговый высокоуровневый оптимизирующий распараллеливатель программ и его

приложения», госконтракт 02.740.11.0208 от 07 июля 2009 г., шифр: 2009-1.1-113-050-043.

- ФЦП «Научные и научно-педагогические кадры инновационной России» по лоту «Проведение научных исследований коллективами научно-образовательных центров в области: геномных и постгеномных технологий создания лекарственных средств; клеточных технологий; биоинженерии; биоинформационных технологий» по теме: «Создание биоинформационной технологии поиска взаимосвязанных сценариев организации в геномах животных и человека не кодирующей ДНК и кодирующей белок ДНК», госконтракт 14.740.11.0006 от 01 сентября 2010 г.
- Грант Российско-Белорусского сотрудничества «ТРИАДА», НИЦЭВТ, 2005 г.
- Внутренний грант Южного федерального университета «Учебно-научная лаборатория оптимизации и распараллеливания программ», 2007г.
- ФЦП «Научные и научно-педагогические кадры инновационной России» на 2009–2013 гг., в рамках реализации мероприятия № 1.4 «Развитие внутрироссийской мобильности научных и научно-педагогических кадров путем выполнения научных исследований молодыми учеными и преподавателями в научно-образовательных центрах», госконтракт 14.740.11.0442 от 30 сентября 2010 г.

Публикации

Результаты диссертации опубликованы в 18 печатных работах, из которых 5 статей, в т.ч. 4 в журналах перечня ВАК и 1 в украинском журнале, и 13 публикаций в сборниках трудов и тезисов конференций.

Объем и структура диссертации

Диссертация состоит из введения, четырёх глав, заключения, приложения и списка литературы (93 наименования). Объем диссертации 152 страницы. Работа содержит 6 лемм, 6 теорем, 54 рисунка, 6 таблиц.

СОДЕРЖАНИЕ РАБОТЫ

В первой главе диссертации приведены общие сведения из теории графов, необходимые для формулирования теорем и результатов работы. Также сообщаются сведения из теории преобразования программ. Приводятся понятия информационной зависимости, графа информационных связей, решетчатого графа и его подвидов (элементарного решетчатого графа, минимального решетчатого графа и т.д.). Описываются способы представления решетчатых графов в памяти компьютера.

В последнем параграфе первой главы приведено понятие графа вычислений, который в других источниках называется редуцированным графом алгоритма. Этот граф является промежуточным представлением программы для многоконвейерной модели вычислений. Дадим упрощенное определение этого графа.

Определение: Графом вычислений называется такой граф, в котором каждой операции (арифметической, логической, чтения, записи и т.п.) исходной программы соответствует вершина графа, а дуга на графе соединяет вершину x с вершиной y , если результат выполнения операции, соответствующей вершине x , является аргументом для операции, соответствующей вершине y .

Отображение графа вычислений на конвейерную архитектуру может быть построено следующим образом: каждой вершине графа вычислений ставится в соответствие процессорный элемент, выполняющий соответствующую операцию, а каждой дуге – канал связи.

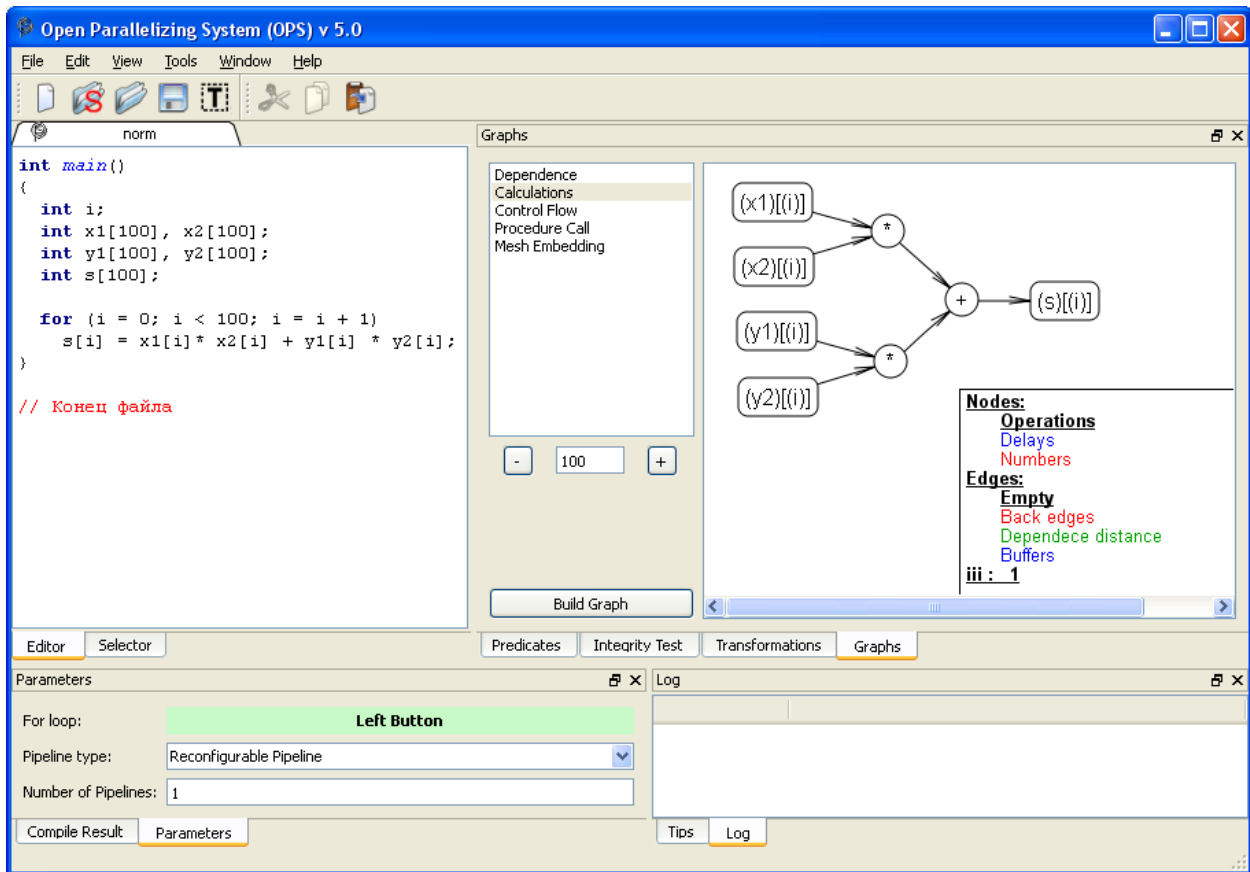


Рис. 1. Граф вычислений в OPS 5 (ДВОР)

Во второй главе рассматриваются конвейерные вычисления и отображения программ на конвейерные вычислительные устройства. Приводятся алгоритмы отображения одномерных циклов на конвейер, алгоритмы расчета параметров конвейера (интервала инициализации итераций, буферов задержек) и вспомогательные алгоритмы.

Основным результатом этой главы является алгоритм автоматической синхронизации конвейера, включающий расчет интервала инициализации итераций и расчет буферов задержек. Задача вычисления минимального интервала инициализации итераций в общем случае является NP-полной, и точный алгоритм её решения состоит в переборе всех циклов на графе вычислений. Известная формула расчета интервала инициализации итераций выглядит следующим образом:

$$iii = \max_c \lceil d(c) / p(c) \rceil, \quad (1)$$

где максимум берется по всем c — циклам графа вычислений, $d(c)$ — сумма задержек операций по вершинам цикла c , $p(c)$ — сумма расстояний зависимости по дугам зависимости входящим в цикл c , а символом $\lceil a \rceil$ обозначено округление с избытком числа a .

В диссертации доказана следующая теорема.

Теорема 1: Всегда существует элементарный цикл, на котором достигается максимум выражения (1).

Таким образом, предложенный в диссертации алгоритм уменьшает объем вычислений для определения минимального iii , так как для вычисления значения выражения (1) требуется перебрать не все циклы на графе вычислений, а только элементарные. Также приводятся обоснования корректности этих алгоритмов.

В третьей главе рассматривается модель многоконвейерных вычислений. Вводится понятие задержки между стартами конвейеров. Рассматривается возможность запуска набора независимых конвейеров и набора конвейеров с передачей информации между ними.

В качестве примера, можно привести следующий фрагмент программы:

```
for (i=0; i<N; ++i)
{
    y[i] = b[i];
    for (j=0; j<N; ++j)
        y[i] = y[i]+A[i][j]*x[j];
}
```

Этот фрагмент программы может быть отображен на многоконвейерную систему как набор независимых конвейеров.

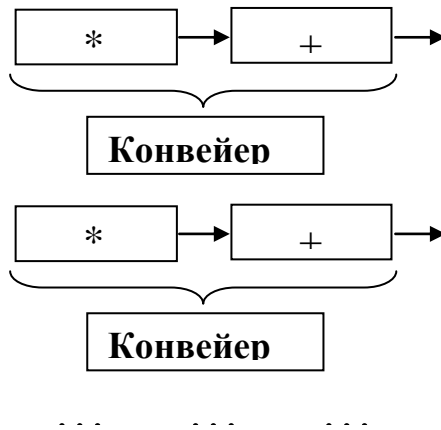


Рис. 2. Организация вычислений без передачи информации между конвейерами

Рассмотрим другой пример:

```
for (i=1; i<N; ++i)
  for (j=1; j<M; ++j)
    x[i][j] = a[i][j]*x[i-1][j]+b[i][j]*x[i][j-1]+c[i][j];
```

Для выполнения этого фрагмента программы можно организовать набор конвейеров, причем они будут обмениваться информацией.

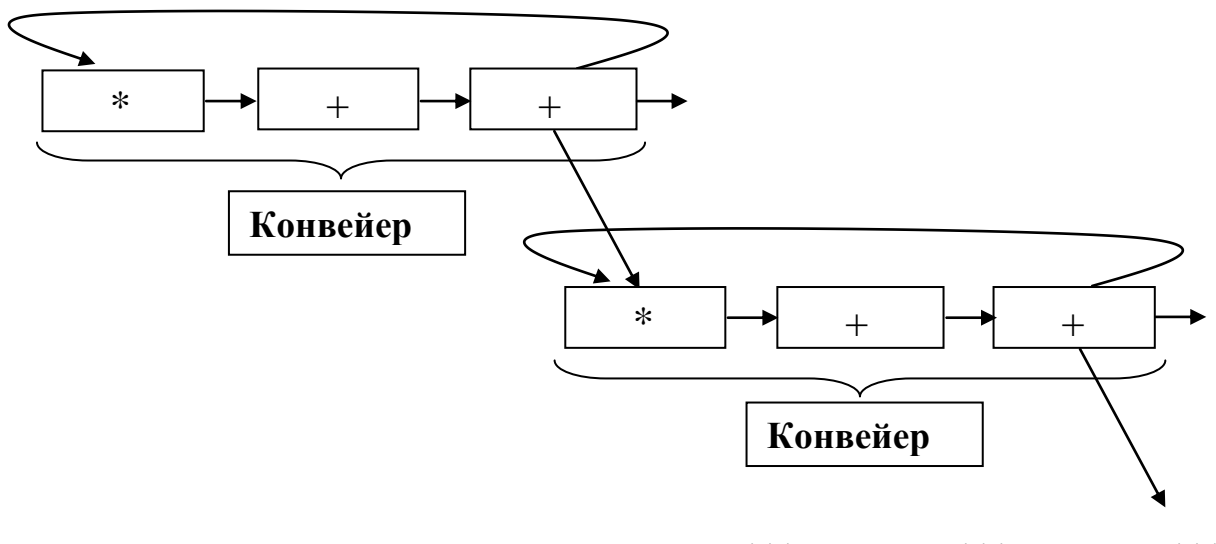


Рис. 3. Организация вычислений с передачей информации между конвейерами

Исследуется влияние информационных зависимостей на синхронизацию и составление расписания. Для анализа зависимостей используются решетчатые графы.

Вводятся два понятия: цепочки вхождений и функции зависимости цепочки. С помощью этих понятий формулируется и доказывается критерий отображения гнезда циклов на многоконвейерную модель вычислений. Этот критерий формулируется в виде следующей теоремы.

Теорема 2: Для организации параллельно-конвейерного вычисления гнезда циклов на многоконвейерной вычислительной системе эквивалентно последовательному выполнению, необходимо и достаточно, чтобы все индексные выражения были регулярными, гнездо циклов не содержало антизависимостей и зависимостей по выходу, а также, чтобы для задержки между стартами конвейеров, обозначаемой z , выполнялись следующие два неравенства:

$$z \geq \max_{d \in G} \max_{1 \leq \alpha \leq m} \max_{J \in D_{d,\alpha}} T_1(d, F_{d,\alpha}(J), J),$$

$$z \geq \max_{1 \leq \alpha \leq r} \max_{J \in D_\alpha} T_2(\omega_{\alpha,1}, \omega_{\alpha,2q(\alpha)+2}, F_\alpha(J), J),$$

где G – граф информационных связей тела самого вложенного цикла в гнезде циклов, d – дуга графа G , система функций $\{F_{d,\alpha}\}$, $1 \leq \alpha \leq m$ описывает элементарный решетчатый граф, построенный по дуге d , а набор выпуклых множеств $\{D_{d,\alpha}\}$, $1 \leq \alpha \leq m$, каждое из которых является областью определения соответствующей функции $F_{d,\alpha}$, семейство $\{W_\alpha\}$, $1 \leq \alpha \leq r$ представляет собой семейство цепочек зависимостей такое, что цепочка W_α имеет вид $\{\omega_{\alpha,\beta}\}$, $1 \leq \beta \leq 2q(\alpha)+2$, при этом $\{F_\alpha\}$, $1 \leq \alpha \leq r$ семейство соответствующих функций зависимости с областями определения D_α , функции T_1 и T_2 определяются следующими равенствами:

$$T_1(d, I, J) = \frac{T_{\text{calc}}(u, I) + T_{\text{send}} - iii \times (j_s - i_s)}{\sum_{\alpha=1}^{s-2} \left((j_\alpha - i_\alpha) \prod_{\beta=\alpha}^{s-2} N(\beta+1) \right) + j_{s-1} - i_{s-1}},$$

$$T_2(u, v, I, J) = \frac{T_{\text{calc}}(u, I) + q \times T_{\text{send}} + \sum_{\alpha=1}^q T_{\text{calc}}(w_{2\alpha+1}) - iii \times (j_s - i_s)}{\sum_{\alpha=1}^{s-2} \left((j_\alpha - i_\alpha) \prod_{\beta=\alpha}^{s-2} N(\beta+1) \right) + j_{s-1} - i_{s-1}},$$

$I = (i_1, i_2, \dots, i_s), J = (j_1, j_2, \dots, j_s), s$ – количество циклов в гнезде, $N(x)$ – количество итераций в цикле с номером x из рассматриваемого гнезда циклов, T_{send} – время пересылки данного между вычислительными элементами системы, $T_{\text{calc}}(u, I)$ – время, которое пройдет с момента старта

итерации I на конвейере с номером $N_{\text{pipe}}(I) = \left[\sum_{\alpha=1}^{s-2} \left(i_\alpha \prod_{\beta=\alpha}^{s-2} N[\beta+1] \right) + i_{s-1} + 1 \right]$ до

момента, когда будет получен результат вычисления значения генератора u , принадлежащего самому вложенному циклу, $T_{\text{calc}}(u)$ – время, которое потребуется для вычисления значения генератора u , не принадлежащего самому вложенному циклу.

Далее в третьей главе рассматривается практически важный случай двумерного гнезда циклов и выводятся более удобные с точки зрения вычислений формулы для вычисления задержки между стартами конвейеров. Обсуждаются вопросы составления расписания для организации вычислений гнезда циклов.

В четвертой главе формулируются вспомогательные и дополнительные результаты, которые были получены в процессе работы над диссертацией: анализ и линеаризация выражений, система тестирования, конвертера с языка С в VHDL.

Линеаризацией выражения относительно набора переменных a_i называется преобразование произвольного арифметического выражения к виду: $e_1 * a_1 + e_2 * a_2 + \dots + e_n * a_n + e_n + 1$, где e_i – выражения, a_i – переменные, входящие в исходное выражение, и при этом ни одна из них не входит ни в одно из выражений e_i . Таким образом, программная реализация должна определить по входному выражению и набору переменных a_i коэффициенты его линейного представления e_i . Также в первом параграфе рассматривается

применение линеаризации к различным подпроектам ДВОР. Описывается процесс развития библиотеки, реализующей линеаризацию в ДВОР, с целью соответствия как можно большему покрытию языка С.

Вторым вспомогательным результатом является система тестирования. Во втором параграфе, посвященном системе тестирования, разъясняются принципы тестирования, определяются компоненты системы, в частности интерфейс, а также специально разработанный формат описания файлов с входными и выходными данными. Разработанная автором библиотека позволяет по описаниям файлов входных и выходных данных генерировать случайный файл с данными, проверять, подходит ли конкретный файл с данными под указанное описание и сравнивать два файла на эквивалентность данных, в нем содержащихся. Используя эти возможности, тестирующая система может осуществлять следующие методики тестирования:

- тестирование на случайном наборе входных данных,
- тестирование на predetermined наборе входных данных с эталонными результатами,
- тестирование параллельной версии программы на эквивалентность последовательному эталону,
- тестирование масштабируемости параллельной программы,
- тестирование преобразований программ, в частности, входящих в проект ДВОР.

В последнем параграфе второй главы приводится информация об экспериментальном конвертере с языка С в VHDL. Описывается поэтапное преобразование исходной программы на языке С в описание схемы на VHDL. Основным преимуществом перед всеми существующими на сегодняшний день аналогичными программными средствами является то, что по программе на языке С будет сгенерировано целое семейство описаний схемы на VHDL. Причем эти описания будут получены с помощью оптимизирующих высокоуровневых преобразований исходной программы, ориентированных на распараллеливание. А далее будут выбираться

различные, подходящие для конкретной ПЛИС реализации стандартных функций. Таким образом, инженер-схемотехник получит возможность выбора между различными вариантами автоматически сгенерированных описаний.

В заключении приведены основные результаты диссертации.

Основные результаты работы

1. Описана математическая модель многоконвейерных вычислений. Проведены исследования по отображению программ на языке С на указанную модель. Разработаны алгоритмы расчета параметров конвейеров и параметров синхронизации конвейеров. Доказаны теоремы, показывающие корректность алгоритмов, приведенных в работе.
2. Получены формулы для вычисления характеристик конвейеров и их синхронизации, в случае двумерного тесного гнезда циклов. Этот частный случай видится практически значимым, так как подобные циклы встречаются довольно часто в реальных программах.
3. Получена модифицированная формула вычислений интервала инициализации итераций (*iii*) и доказана ее эквивалентность общеизвестной формуле. В работе сформулирован алгоритм вычисления интервала инициализации итераций, использующий модифицированную формулу вычисления *iii*. Предлагается анализировать только элементарные циклы и доказывается, что этого достаточно для получения точной оценки интервала инициализации итераций. В рамках проекта ДВОР, была сделана программная реализация вычисления интервала инициализации итераций по алгоритмам описанным в данной работе.
4. В работе сформулированы и обоснованы алгоритмы вычисления различных характеристик конвейера: алгоритм определения обратных дуг на графе вычислений, алгоритм составления расписания, алгоритм вычисления буферных задержек. Приводятся

обоснования корректности этих алгоритмов. В рамках проекта ДВОР, была сделана программная реализация вычисления параметров конвейера по алгоритмам описанным в данной работе.

5. В работе сформулированы алгоритмы анализа линейности выражений, а также приведения выражения к линейному виду (линеаризация). Эти алгоритмы могут быть использованы для определения информационных зависимостей и оптимизации программ, в том числе для конвейеризации циклов. В проекте ДВОР для построения графа информационных зависимостей используется анализ индексных выражений массивов, реализованный в соответствии с указанными алгоритмами.

Список публикаций

1. Штейнберг Р. Б. Отображение гнезд циклов на многоконвейерную архитектуру // Программирование, 2010, № 3. С. 69–80. (**Перечень ВАК**)
Steinberg R. Mapping loop nests to multipipelined architecture // Programming and Computer Software. МАИК Наука/Interperiodica distributed exclusively by Springer Science+Business Media LLC. 2010. Vol 36, №3 May, P 177–185.
2. Штейнберг Р.Б. Использование решетчатых графов для исследования многоконвейерной модели вычислений // Известия вузов. Северо-кавказский регион. Естественные науки. 2009. №2. С. 16–18. (**Перечень ВАК**)
3. Бухановский А. В., Марьин С. В., Князьков К. В., Сиднев А. А., Жабин С. Н., Баглий А. П., Штейнберг Р. Б., Шамакина А. В., Воеводин В. В., Головченко Е. Н., Фалалеев Р. Т., Духанов А. В., Тарасов А. А., Шамардин Л. В., Моисеенко А. И. Результаты реализации проекта «Мобильность молодых ученых» в 2010 году: развитие функциональных элементов технологии iPSE и расширение состава прикладных сервисов // Известия вузов. Приборостроение. 2011. № 10. С. 80–87. (**Перечень ВАК**)
4. Штейнберг Б. Я., Кравченко Е. Н., Морылев Р. И., Нис З. Я., Петренко В. В., Скиба И. С., Шаповалов В. Н., Штейнберг О. Б., Штейнберг Р. Б. Особенности реализации распараллеливающих преобразований программ в системе ДВОР // Известия вузов. Приборостроение. 2011. № 10. С. 87–89. (**Перечень ВАК**)

5. Штейнберг Р.Б. Вычисление задержки в стартах конвейеров для суперкомпьютеров со структурно процедурной организацией вычислений // Искусственный интеллект. (научно-теоретический журнал) / Институт проблем искусственного интеллекта НАНУ. Украина, Донецк: ДонДИШИ, «Наука и Освита». 2003. № 4. С. 105–112.
6. Штейнберг Р.Б. Вычисление задержки в стартах конвейеров для суперкомпьютеров со структурно-процедурной организацией вычислений // Интеллектуальные и многопроцессорные системы ИМС-2003: материалы Международной научно-технической конференции. Россия, Дивноморское, 22–27 сентября 2003г. Т. 2.
7. Штейнберг Б.Я., Арутюнян О.Э., Бутов А.Э., Гуфан К.Ю., Морылев Р.И., Науменко С.А., Петренко В.В., Тузаев А., Черданцев Д.Н., Шилов М.В., Штейнберг Р.Б., Шульженко А.М. Обучающая распараллеливанию программа на основе ОРС // Научно-методическая конференция «Современные информационные технологии в образовании: Южный федеральный округ»: тезисы докладов. Ростов-на-Дону, 12–15 мая 2004 г. Ростов н/Д, 2004.
8. Штейнберг Б.Я., Бутов А.Э., Науменко С.А., Петренко В.В., Черданцев Д.Н., Штейнберг Р.Б., Шульженко А.М. Полуавтоматическое распараллеливание на основе ОРС // Всероссийская научно-техническая конференция «Параллельные вычисления в задачах математической физики»: тезисы докладов. Ростов-на-Дону, 21–25 июня 2004 г. Ростов н/Д, 2004.
9. Штейнберг Р.Б. Реализация модели конвейерных вычислений в ОРС // XIII Международная конференция “Математика. Экономика. Образование”: тезисы докладов. Ростов н/Д, 2005. 195 с. - ISBN 5-94153-097-8.
10. Штейнберг Р.Б. Некоторые оптимизирующие преобразования программ для компьютеров, допускающих многоконвейерную обработку данных : труды Всероссийской научной конференции «Научный сервис в сети Интернет: технологии распределенных вычислений», Новороссийск, 19-23 сентября 2005г. М.: Изд-во МГУ, 2005. С. 85-87.
11. Штейнберг Б.Я., Нис З.Я., Петренко В.В., Черданцев Д.Н., Штейнберг Р.Б., Шульженко А.М. Состояние и возможности открытой распараллеливающей системы (лето 2006г.): сборник трудов VI международной конференции памяти академика А.П. Ершова «Перспективы систем информатики», рабочий семинар «Наукоемкое программное обеспечение» (Workshop on Science Intensive Applied Software). 2006г. Новосибирск, 2006. С. 122–125.

12. Штейнберг Р.Б. Вычисление задержки между стартами конвейеров с учётом времени пересылки данных: труды международной конференции РАСО-2006. М.: ИПУ РАН, 2006.
13. Штейнберг Б.Я., Алымова Е.В., Баглий А.П., Морылев Р.И., Нис З.Я., Петренко В.В., Штейнберг Р.Б. Автоматизация тестирования элементов высокопроизводительного программного комплекса // Научный сервис в сети Интернет: масштабируемость, параллельность, эффективность: труды Всероссийской суперкомпьютерной конференции. Новороссийск, 21-26 сентября 2009 г. М.: Изд-во МГУ, 2009. С. 287–291.
14. Steinberg B., Alimova E., Baglij A., Morilev R., Nis Z., Petrenko V., Steinberg R. The System for Automated Program Testing. Proceedings of IEEE East-West Design & Test Symposium (EWDTS'09). Moscow, Russia, September 18-21, 2009. P. 218–220.
15. Steinberg B., Abramov A., Alyмова E., Baglij A., Guda S., Demin S., Dubrov D., Ivchenko A., Kravchenko E., Makoshenko D., Molotnikov Z., Morilev R., Nis Z., Petrenko V., Povazhniy A., Poluyan S., Skiba I., Suhoverkhov S., Shapovalov V., Steinberg O., Steinberg R. Dialogue-based Optimizing Parallelizing Tool and C2HDL Converter. Proceedings of IEEE East-West Design & Test Symposium (EWDTS'09). Moscow, Russia, September 18-21, 2009. P. 216–218.
16. Штейнберг Б.Я., Абрамов А.А., Алымова Е.В., Баглий А.П., Гуда С.А., Дубров Д.В., Кравченко Е.Н., Морылев Р.И., Нис З.Я., Петренко В.В., Полуян С.В., Скиба И.С., Шаповалов В.Н., Штейнберг О.Б., Штейнберг Р.Б., Юрушкин М.В. Диалоговый высокоуровневый оптимизирующий распараллеливатель (ДВОР) // Научный сервис в сети Интернет: суперкомпьютерные центры и задачи: труды международной суперкомпьютерной конференции. Новороссийск, 20-25 сентября 2010г., М.: Изд-во МГУ, 2010. С. 71–75.
17. Дубров Д.В., Штейнберг Р.Б. Экспериментальный конвертер с языка С в HDL на основе диалогового высокоуровневого оптимизирующего распараллеливателя // РАСО'2010. Москва, 26-28 октября 2010 г., М.: ИПУ РАН, 2010. С. 865–878.
18. Штейнберг Б.Я., Абрамов А.А., Баглий А.П., Морылев Р.И., Петренко В.В., Полуян С.В., Штейнберг Р.Б. Уточнение зависимостей программы в ДВОР: труды международной конференции «Параллельные вычисления и задачи управления» РАСО'2010. Москва, 26-28 октября 2010 г., М.: ИПУ РАН, 2010. С. 855-864.

Подписано в печать 25.01.2012 г. Формат 60×84 ¹/₁₆ .
Бумага офсетная. Печать офсетная.
Усл. печ. л. 1,0. Уч.-изд. л. 1,0. Тираж 120 экз. Заказ № 2178.

Отпечатано в типографии ЮФУ.
344090, г. Ростов-на-Дону, пр. Стачки, 200/1.Тел. (863) 247-80-51.