

На правах рукописи

Анисимов Игорь Юрьевич

**ИССЛЕДОВАНИЕ И РАЗРАБОТКА МЕТОДОВ ПОВЕДЕНЧЕСКОГО
СИНТЕЗА КОНВЕЙЕРНЫХ СХЕМ ДЛЯ ЦИФРОВОЙ ОБРАБОТКИ
ВИДЕОИЗОБРАЖЕНИЙ**

Специальность 05.13.05 – Элементы и устройства вычислительной техники и систем управления

АВТОРЕФЕРАТ

диссертации на соискание ученой степени
кандидата технических наук

Санкт-Петербург – 2008

Работа выполнена на кафедре «Вычислительной техники»
государственного образовательного учреждения высшего профессионального
образования «Санкт-Петербургский государственный электротехнический уни-
верситет «ЛЭТИ» им. В.И.Ульянова (Ленина)»

Научный руководитель: доктор технических наук, профессор
Мурсаев Александр Хафизович

Официальные оппоненты: доктор технических наук, профессор
Лыпарь Юрий Иванович

кандидат технических наук,
Суворова Елена Александровна

Ведущая организация: **ФГУП «ЦНИИ Электроприбор»,**
г. Санкт-Петербург

Защита состоится _____ 2008 г. в _____ часов на заседании диссертаци-
онного совета Д 212.229.18 ГОУ ВПО «Санкт-Петербургский государственный
политехнический университет» по адресу: 195251, Санкт-Петербург, ул. Поли-
техническая, 21, IX учебный корпус, ауд. 325.

С диссертацией можно ознакомиться в фундаментальной библиотеке ГОУ
ВПО «Санкт-Петербургский государственный политехнический университет»
Автореферат разослан « ____ » _____ 2008 г.

Ученый секретарь
Диссертационного совета Д212.229.18
кандидат технических наук доцент

Васильев А. Е.

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность проблемы

Тенденцией развития систем визуализации, включая всевозможные оптикоэлектронные приборы, телекоммуникационные и телевещательные системы, тренажеры, компьютерные игры и т.п., является непрерывное расширение предоставляемых услуг и усложнение алгоритмов обработки. Средством для этого является цифровая техника. В современных видеосистемах вся обработка электрического сигнала, получаемого с выхода приемника излучения или из эфира (часто аналогового), осуществляется цифровыми методами и, наоборот, результат цифровой обработки может подаваться на аналоговые системы отображения. Поэтому каждая система содержит блок цифровой обработки. В этом блоке среди множества алгоритмов обработки изображений выделяется особый класс алгоритмов, реализуемых отдельно от остальных, называемый алгоритмами предварительной обработки изображений. К предварительной обработке обычно относят алгоритмы, связанные с:

- компенсацией погрешностей фотоприемного или отображающего устройства,
- улучшением качества визуального восприятия изображения,
- сжатием видеоданных,
- оценкой некоторых свойств изображения (например, диапазон значений, спектральные свойства) и подготовкой изображения к анализу автоматической системой.

В блоке цифровой обработки алгоритмы должны выполняться в реальном масштабе времени. Это требование строго задает необходимую пропускную способность используемых вычислительных схем. Например, поток изображений высокой четкости (HDTV), имеет интенсивность 110 Мбайт/с, такую же пропускную способность должен иметь вычислительный блок. Для современных универсальных процессоров это практически невыполнимые требования. Современный сигнальный процессор с производительностью 3 Млрд.оп./с способен выполнять около 30 операций на такт. Это эквивалентно всего лишь реализации фильтра 5 на 5, а для перечисленных выше задач требуется гораздо больше операций. На современных вычислительных средствах для большинства алгоритмов предварительной обработки изображений достичь такой высокой пропускной способности можно только с помощью параллельной реализации алгоритмов, причем параллельной на уровне элементарных операций. Именно такой реализацией являются конвейерные схемы. В силу сказанного предварительную обработку обычно выносят в отдельный специализированный блок, имеющий конвейерную архитектуру.

Современной перспективной элементной базой для специализированных конвейерных схем являются ПЛИС (FPGA) и полузаказные СБИС (ASIC). Благодаря универсальности ПЛИС, они выпускаются большой серийностью, поэтому современной тенденцией является быстрое снижение цен на них при од-

новременном росте степени интеграции. ПЛИС являются матрицей конфигурируемых элементарных однобитовых процессоров объединенных между собой конфигурируемой системой связей. ПЛИС в основном ориентированы на изделия малой серийности, тогда как полужаказные СБИС - на изделия средней серийности.

Другой актуальной областью применения конвейерных схем являются реконфигурируемые сопроцессоры. Идея состоит в том, что в компьютере, помимо центрального универсального процессора, можно иметь специализированный сопроцессор на базе ПЛИС, структура которого перестраивается пользователем под конкретную задачу. Любой алгоритм, выполняемый на компьютере, может иметь какую-то часть, реализуемую аппаратно, что существенно повышает скорость исполнения алгоритма. К примеру, при работе с видеoinформацией в сопроцессор может быть загружен кодек, сжимающий или разжимающий изображения и звуковые сигналы в реальном масштабе времени. Использование сопроцессора актуально при выполнении любых вычислительно сложных задач: научных расчетов, имитационного моделирования, архивирования файлов, играх, работе с трехмерной графикой и др. Необходимо только во время запуска очередного приложения менять конфигурацию сопроцессора, для ПЛИС время изменения конфигурации занимает несколько секунд.

И в случае реконфигурируемых сопроцессоров, и в случае специализированных процессоров обработки видеоизображений остро стоит проблема проектирования. Это связано в основном с тем, что большинство изделий в этой предметной области выпускается малой серийностью, и вклад проектирования в стоимость одного изделия оказывается существенным. Проектирование конвейерных схем непростая задача, поскольку непосредственно связана с проблемой параллельной реализации алгоритмов. Современные САПР используют в качестве исходных данных только уже готовую конвейерную реализацию алгоритма, описанную на специальном языке (VHDL, Verilog, System C, Handle C и т.п.), и доведенную до уровня абстракции регистровых передач. То есть задача конвейерной реализации алгоритма возлагается на пользователя. Вместо этого хотелось бы задавать исходный алгоритм в последовательной форме (например, используя чистое поведенческое описание VHDL, System C, Handle C, или последовательные языки C++, MatLab, и др.), а по ней получать конвейерную схему автоматически. Это позволило бы значительно сократить срок проектирования и значительно снизить стоимость проектирования.

Использование последовательной (программной) формы алгоритма (ПФА) в качестве исходных данных при проектировании схем имеет множество достоинств, ускоряющих процесс проектирования. Во-первых, в силу исторических причин получили сильное развитие методы и средства разработки именно последовательных программ, поэтому разумно использовать уже накопленный опыт. Во-вторых, существует проблема повторного использования уже накопленного багажа готовых последовательных программ. В-третьих, последовательные программы более удобны для спецификации алгоритмов, поскольку обладают лучшей читаемостью. Алгоритм при этом следует доводить

именно до программной реализации, а не брать в качестве исходных данных, например, математическую запись алгоритма. Это позволит учесть в спецификации алгоритма ошибки округления, используемую арифметику, типы данных, порядок действий.

Проблема синтеза схем по последовательной (программной) форме алгоритма называется проблемой поведенческого синтеза. Данная проблема является актуальной, что подтверждают усилия многих фирм в этом направлении: в частности:

- Impulse Accelerated Technologies – программный продукт ImpulseC
- Mentor Graphics – программный продукт Catapult C
- Mitronics Inc. – программный продукт Mitrion Software Development Kit
- Celoxica – программный продукт Agility Compiler

Кампания Mentor Graphics для своего продукта Catapult C утверждает, что размер автоматически сгенерированных схем оказывается на 20-50% больше, чем в случае ручного проектирования. Таким образом, можно утверждать, что, несмотря на работы в данном направлении, поведенческий синтез требует дальнейшего исследования и развития. Данная работа ограничена исследованием методов поведенческого синтеза только конвейерных схем, которые наиболее актуальны в предварительной обработке изображений.

Цель работы

Исследование и разработка методики синтеза конвейерных схем, ориентированных на предварительную обработку изображений, по последовательной форме алгоритма.

Основные задачи:

1. Исследование моделей программ, выбор модели отражающей структуру зависимостей между операциями программы, исследование методов построения такой модели по последовательной форме алгоритма.
2. Исследование и разработка методов нахождения таких операций алгоритма, которые могут выполняться одновременно.
3. Исследование и разработка методов представления алгоритма в конвейерной форме.
4. Исследование и разработка моделей для конвейерных вычислительных устройств, описывающих обмен данными между вычислительными узлами.
5. Исследование и разработка методов синтеза конвейерных вычислительных устройств на основе задания алгоритма в последовательной форме.

Методы исследований

При решении поставленных задач были использованы теория графов, теория информационной структуры программ, математическое моделирование и имитационное моделирование.

Научная новизна работы

1. Развита теория информационной структуры программ применительно к поведенческому синтезу конвейерных вычислительных схем предварительной обработки видеоизображений.
2. Предложена методика проектирования конвейерных вычислительных схем предварительной обработки видеоизображений по последовательной программной реализации алгоритма на основе использования графовых моделей.
3. Предложен метод нахождения конвейерной развертки графа зависимостей программы.
4. Предложен метод синтеза структурной графовой модели вычислительной схемы по графу зависимостей программы.

Практическая значимость

1. Предложенный метод поведенческого синтеза конвейерных реализаций алгоритма позволяет автоматизировать процесс проектирования специализированных процессоров обработки изображений, сократить временные затраты и трудоемкость проектирования.
2. Исследование алгоритмов с помощью теории информационной структуры программ позволяет находить эффективные аппаратные реализации этих алгоритмов, оптимизированные под выбранный элементный базис и заданные требования производительности, что позволяет повысить качественные характеристики проектируемых устройств.
3. Исследованы алгоритмы, и получены конвейерные схемы для алгоритмов предварительной обработки изображений: двумерной свертки, вычисления градиентного поля, корреляционной функции, внедренные в сетевые камеры высокого разрешения ЗАО «Ареконт Вижн» (Акт внедрения прилагается к диссертации).

На защиту выносятся:

1. Методика проектирования конвейерных вычислительных схем предварительной обработки изображений по последовательной программной реализации алгоритма на основе использования графовых моделей.
2. Метод нахождения конвейерного режима выполнения программы на основе нахождения специального вида разверток графа зависимостей.
3. Метод синтеза структуры конвейерной вычислительной схемы для заданного алгоритма по графу зависимостей этого алгоритма.
4. Параллельные схемы для операций: двумерной свертки, вычисления градиентного поля изображения, двумерной корреляционной функции.

Реализация результатов работы

Результаты диссертационной работы были использованы в разработке H264 кодера для камер высокого разрешения проводимой ЗАО «Ареконт Вижн», в ОАО «НПО «Карат» при разработке блока предварительной обработ-

ки изображения, во ФГУП «НПК «ГОИ им. С.И. Вавилова» при разработке блока улучшения изображения. Акты внедрения приложены к диссертации.

Апробация работы и публикации

По основным результатам работы были опубликованы 3 научные работы, и сделано 2 доклада:

1. Анисимов И.Ю. Методика отображения алгоритмов на параллельные вычислительные структуры. //VII Международная научно-техническая конференция «Распознавание-2005», 4-7 октября 2005 г., г.Курск.

2. Анисимов И.Ю., Мурсаев А.Х. Методика поведенческого синтеза конвейерных схем предварительной обработки изображений. // 61-я научно-техническая конференция профессорско-преподавательского состава СПбГЭТУ «ЛЭТИ», 29 января - 8 февраля 2008 г., г. Санкт-Петербург.

Структура и объем диссертации

Диссертация состоит из введения, четырех глав, заключения, списка литературы, приложений. Она содержит 143 страницы машинописного текста (без приложений), 49 рисунков. Список литературы содержит 127 наименований. Нумерация формул и рисунков сквозная по всей диссертации.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность темы диссертации, формулируются цели, задачи, научная новизна и практическая значимость работы, приводятся выносимые на защиту положения.

В первой главе диссертации обосновывается выбор конвейерных схем как средства реализации алгоритмов предварительной обработки, а также выполняется анализ современного состояния проблем проектирования специализированных параллельных архитектур.

Поскольку конвейерные схемы состоят из множества параллельно работающих устройств, проблема их проектирования фактически сводится к проблеме параллельной реализации алгоритмов. На основе исследований методов проектирования параллельных процессоров, в диссертационной работе в качестве общего направления проектирования выбран процесс отображения алгоритмов на матричные процессоры, состоящий в следующем:

1. Разработка программы с однократным присваиванием и построение графа зависимостей.
2. Построение графа потока сигнала (граф-машины) на основе графа зависимостей.
3. Разработка матричного процессора по графу потока сигнала.

Граф зависимостей это графовая модель алгоритма. Граф потока сигнала или граф-машина это графовая модель матричного процессора. Однако ранее известные методы, использовавшиеся в процессе отображения, были ориенти-

рованы только на матричные процессоры, структура которых регулярна. Кроме этого принятые там графовые модели ориентированы только на исследование отдельно взятых тесно вложенных циклов программы. Анализ графов выполняется визуально, без достаточной математической формализации, что позволяет работать только с простыми алгоритмами. На основе анализа литературы по параллельным вычислениям в качестве перспективного средства преодоления указанных недостатков, в работе выбрана теория информационной структуры программ.

Теория информационной структуры программ предлагает математический аппарат для анализа структуры зависимостей между операциями алгоритма. Каждый оператор присваивания в программе обозначается как итерация. Итерации в программе могут быть связаны отношениями зависимости. Граф зависимостей может строиться для широкого, но ограниченного класса программ, содержащих произвольные комбинации вложенных циклов, ветвлений и линейных участков. Данный класс программ называется линейным классом. Граф зависимостей может строиться по программе из линейного класса формальным способом.

Для выявления параллелизма используются специальные функции на графах - развертки. Развертка определяется как линейный функционал, определенный на пространстве итераций и возрастающий в направлении дуг графа зависимостей. На любой поверхности уровня развертки никакие точки пространства итераций не могут быть связаны путями графа. Поэтому точки, принадлежащие одной и той же поверхности уровня развертки, соответствуют параллельным операциям алгоритма. Линия уровня развертки совпадает с понятием вычислительный фронт.

Таким образом, для решения поставленной в диссертационной работе цели, принято решение ориентироваться в целом на методы, использовавшиеся при проектировании матричных процессоров, но при этом использовать формальный аппарат графов зависимостей и разверток из теории информационной структуры программ. Методы данной теории покрывают только первый этап процесса проектирования конвейерных схем, поэтому в работе ставятся новые задачи нахождения математической интерпретации конвейерного режима выполнения алгоритма, а также формализации процесса перехода от графа зависимостей к граф-машине.

Во второй главе описывается методика проектирования конвейерных вычислительных схем на ПЛИС и полужаказных СБИС, основанная на чистом поведенческом описании алгоритма (или последовательной программной форме алгоритма - ПФА), и ориентированная на алгоритмы предварительной обработки. А также описываются методы положенные в основу методики.

Для схем последовательного действия существуют формальные методы синтеза схем по программному представлению алгоритма, описанные, в частности, в работах Баранова С.И., Майорова С.А. и Новикова Г.И. Эти известные

методы эффективно порождают реализации с временным распределением исполнения операций, такие как автоматную и микропрограммную.

Предлагаемая в работе методика расширяет возможности использования поведенческого синтеза также и на конвейерные схемы. Процесс проектирования конвейерных схем содержит следующие этапы:

1. Описание алгоритма в последовательной форме (другими словами чистое поведенческое описание).
2. Построение графа зависимостей алгоритма, то есть модели, отражающей информационные связи между операциями без учета содержания операций.
3. Нахождение параллельной формы графа зависимостей путем поиска допустимого множества разверток графа зависимостей (развертка есть линейная функция на графе, монотонно возрастающая в направлении дуг графа).
4. Нахождение конвейерной развертки графа зависимостей.
5. Синтез граф-машины - структурной модели вычислительного устройства.
6. Описание конвейерной схемы на предназначенном для этого языке (например, VHDL).

На уровне последовательной формы алгоритма должны быть определены:

- Типы и разрядность данных, а также тип компьютерной арифметики и правила округления.
- Содержание операций. Все сложные операции должны быть разбиты на элементарные, принятые за таковые в целевой элементной базе.
- Порядок элементарных операций.
- Программа должна быть приведена к линейному классу.

В данной главе все этапы процесса синтеза конвейерной формы иллюстрируются примером алгоритма свертки двух последовательностей, используемой в задачах фильтрации. Программная реализация алгоритма выглядит следующим образом

```
for u = 1..n1
  p = 0; % V1
  for v = 1..n2
    p = p + x(u-v+1)*h(v); % V2
  end;
  y(u) = p;
end;
```

Процесс построения графа зависимостей состоит из:

- Разбиения программы на абстрактные элементарные программы, содержащие только два вхождения одной и той же переменной, слева и справа от оператора присваивания.
- Построения элементарного графа зависимостей для каждой элементарной программы. Множеству вершин ставится в соответствие множество итераций программы, итерацией при этом считается срабатывание опе-

ратора присваивания. Дугам ставятся в соответствие зависимости между итерациями. Дуги находятся путем поиска для каждой вершины лексикографически старшей вершины, от которой зависит текущая.

- Объединения элементарных графов зависимостей с отбрасыванием при этом лишних дуг исходя из условия лексикографического максимума при выборе дуг.

Элементарные (абстрактные) программы выглядят следующим образом:

```
for u = 1..n1
  p =          % V1
  for v = 1..n2
    = p      % V2
  end;
end;
```

```
for u = 1..n1
  for v = 1..n2
    p = p      % V2
  end;
end;
```

Графы для этих программ изображены на рисунке 1:

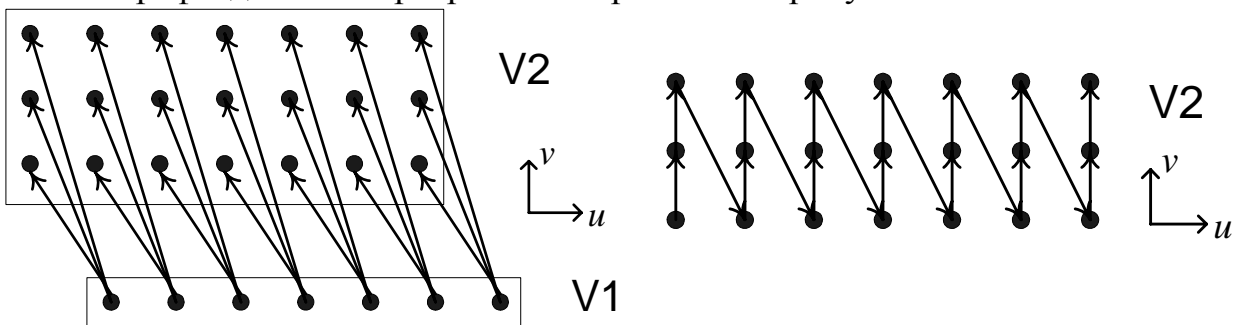


Рисунок 1. Графы зависимостей элементарных программ

Результат объединения этих графов, и окончательный вариант графа с добавлением зависимостей от входных данных изображены на рисунке 2.

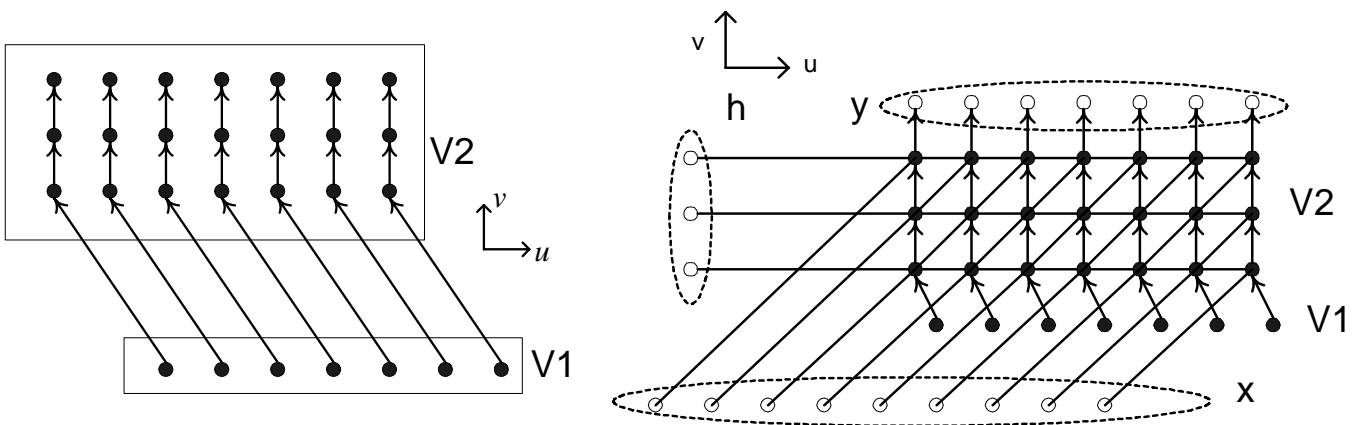


Рисунок 2. Объединение элементарных графов и граф зависимостей программы

Данные графы описываются с помощью системы линейных функций заданных на множестве вершин, называемых покрывающими функциями. Таким образом, на первом этапе синтеза исходная программа описывается системой линейных покрывающих функций. Для рассматриваемого алгоритма система функций имеет вид :

$$\Phi_{V_2V_1} : V_2 \rightarrow V_1; \quad \Phi_{V_2V_1}(u, v) = u, \quad v = 1 \quad (1)$$

$$\Phi_{V_2V_2} : V_2 \rightarrow V_2; \quad \Phi_{V_2V_2}(u, v) = (u, v - 1), \quad v = 2..n_2; \quad (2)$$

$$\Phi_{V2X} : V2 \rightarrow X; \quad \Phi_{V2X}(u, v) = u - v + 1; \quad (3)$$

$$\Phi_{V2H} : V2 \rightarrow H; \quad \Phi_{V2H}(u, v) = v; \quad (4)$$

$$\Phi_{YV2} : Y \rightarrow V2; \quad \Phi_{YV2}(u) = (u, n2). \quad (5)$$

Здесь, например выражение (1) означает, что имеется дуга из вершины с координатой u принадлежащей множеству $V1$ в вершину с координатами (u, v) из множества $V2$.

Следующий этап это нахождение допустимого множества разверток графа и максимальной параллельной формы графа. Так как развертка по определению представляет собой линейный функционал, то ее будем представлять в форме:

$$f(x) = (a, x) + \alpha, \quad (6)$$

то есть как скалярное произведение вектора координат x и вектора коэффициентов a , плюс свободный член α . По определению развертка должна возрастать в направлении дуг, следовательно, должна выполняться следующая система неравенств:

$$f(\Phi_{V2V1}(x)) < f(x), \quad (7)$$

$$f(\Phi_{V2V2}(x)) < f(x). \quad (8)$$

Решениями данных неравенств является множество наборов коэффициентов развертки, задающих допустимое множество разверток. Чтобы из этого множества выбрать какую-нибудь конкретную развертку, необходимо задать дополнительные ограничения. Например, критерий минимума линий уровня развертки, которыми можно покрыть граф зависимостей, дает развертку соответствующую максимальной параллельной форме графа зависимостей. Для рассматриваемого примера эта развертка будет иметь вид:

$$f(u) = 0 \text{ на } V1, \quad (9)$$

$$f(u, v) = v \text{ на } V2. \quad (10)$$

Линии уровня данной развертки изображены на рисунке 3 в виде горизонтальных линий, отмеченных переменной t .

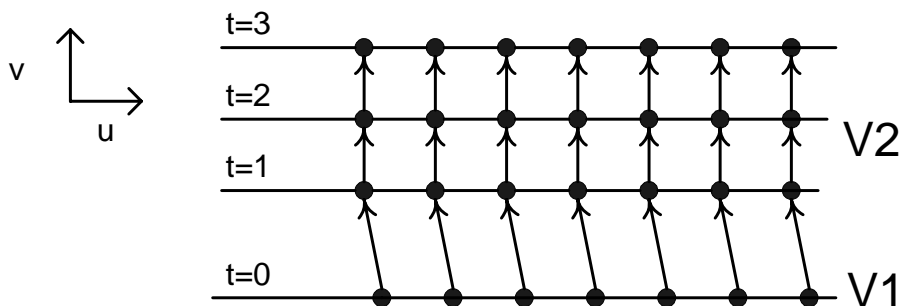


Рисунок 3. Линии уровня развертки

Вершины лежащие на линии уровня развертки параллельны по графу зависимостей, и соответствуют тем операциям алгоритма, которые могут выполняться одновременно. Линия уровня развертки может быть интерпретирована как такт срабатывания абстрактной параллельной вычислительной машины.

Конвейерному режиму выполнения алгоритма свойственна согласованность обработки с поступлением данных. Порядок поступления данных можно описать некоторым линейным функционалом. Для того чтобы развертка была конвейерной коэффициенты этого линейного функционала должны совпадать с соответствующими коэффициентами развертки. Остальные коэффициенты развертки находятся также как и для максимальной параллельной формы, описанной выше. Для примера свертки порядок поступления данных (обозначим как p) описывается выражением, содержащим только одну переменную

$$p = u, \quad (11)$$

где u – номер итерации внешнего цикла. То есть в данном случае порядок поступления отсчетов совпадает с номером итерации внешнего цикла. Конвейерная развертка будет иметь вид:

$$f(u) = u \text{ на } V1, \quad (12)$$

$$f(u, v) = u + v \text{ на } V2. \quad (13)$$

Линии уровня полученной развертки изображены на рисунке 4.

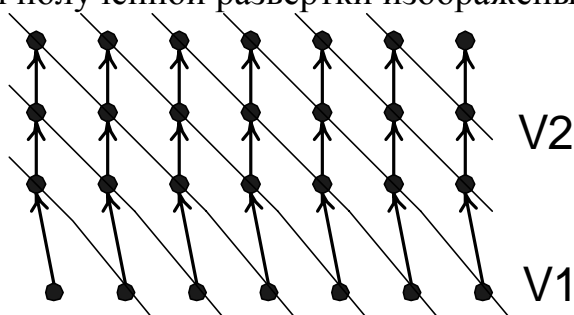


Рисунок 4. Линии уровня конвейерной развертки

Следующий этап – переход от графа зависимостей к граф-машине. Граф-машина это графовая модель параллельной вычислительной схемы. Для перехода необходимо выполнить следующие действия:

1. Лексикографически упорядочить итерации по возрастанию переменной p , описывающей порядок поступления данных.
2. Перейти к новому координатному базису, связанному с конвейерной разверткой.
3. Построить проекцию графа зависимостей.

Для примера свертки покрывающие функции в новом координатном базисе будут иметь вид:

$$\Phi_{v_2v_1}(t, v) = t - 1, \quad v = 1, \quad (14)$$

$$\Phi_{v_2v_2}(t, v) = (t - 1, v - 1), \quad v = 2..n_2, \quad (15)$$

$$\Phi_{v_2x}(t, v) = t - 2v + 1. \quad (16)$$

$$\Phi_{v_2H}(t, v) = v. \quad (17)$$

Переменная t здесь обозначает номер линии уровня развертки, и соответствует тактам срабатывания граф-машины. Например, покрывающая функция (15) обозначает, что операция в узле v на текущем такте t зависит от результата операции в узле $v-1$ на такте $t-1$. Структура граф-машины, восстанавливаемая по этим функциям, представлена на рисунке 5.

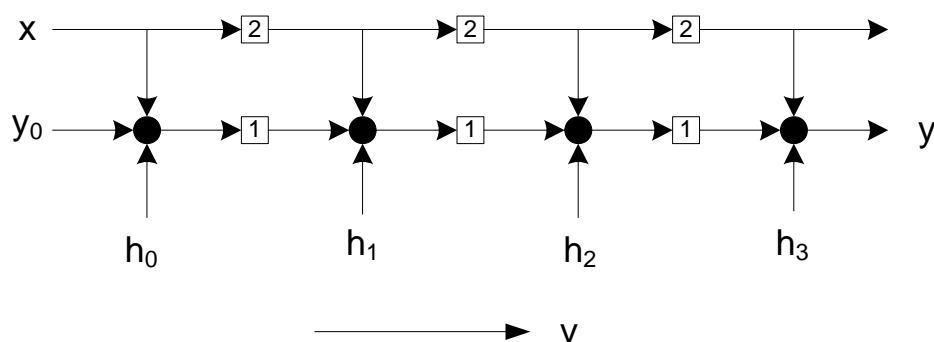


Рисунок 5. Граф-машина свертки

На рисунке вершины графа, обозначенные кружками, соответствуют операционным устройствам (при переходе к VHDL описанию им ставится в соответствие умножения с накоплением). Дуги указывают направление передачи данных (при переходе к VHDL – сигналы), веса на дугах – задержки на такты (при переходе к VHDL – регистры). В работе продемонстрировано также, что для получения последовательно-параллельных вычислительных структур может быть использована гомоморфная свертка граф-машины.

В третьей главе исследуются алгоритмы и строятся конвейерные вычислительные структуры для распространенных в задачах обработки изображений вычислительных операций. Данные примеры в силу их простоты позволяют изобразить все графы и развертки на плоскости и визуально проконтролировать процесс синтеза. На примере данных алгоритмов показывается возможность синтеза структур с различной степенью конвейеризации. Рассмотрены схемы параллельные на битовом уровне, на уровне слов, а также бит-последовательные структуры.

Сначала для иллюстрации методики рассмотрен обычный алгоритм умножения, реализуемый через операции сложения и сдвига. Данный алгоритм был описан программно с детализацией на уровне битовых операций. Исследование графа зависимостей программы с помощью предлагаемой методики показало, что в случае перемножения последовательностей чисел для данного алгоритма существует конвейерная реализация, причем конвейерная на уровне битовых операций. Для алгоритма умножения, описанного на уровне многоразрядных операций, с помощью методики получена схема, конвейерная уже на уровне многоразрядных операций. Если входные данные поступают последовательно по битам, то результатом синтеза схемы будет бит-последовательный умножитель, состоящий из однобитовых сумматоров последовательного действия. Таким образом, проиллюстрирована возможность синтеза с помощью предлагаемой методики схем с различной степенью конвейеризации, в зависимости от задаваемых требований.

Следующий рассматриваемый пример – алгоритм двумерной свертки, широко используемый в задачах фильтрации изображений. Данный пример интересен тем, что граф зависимости для него четырехмерный и не может быть изображен на плоскости. Этот пример иллюстрирует преимущество математи-

ческой формализации процесса синтеза, поскольку анализировать такой граф визуально по проекциям трудно. Для данного алгоритма была получена конвейерная вычислительная структура, которая содержит по два элемента задержки (регистра) на каждый узел для входных данных. Анализ конвейерной развертки графа показал, что, изменив порядок действий в алгоритме свертки на обратный, можно получить более эффективную схему вычисления свертки, не требующую вообще элементов задержки для входных данных (за исключением задержек на строки) .

В предварительной обработке изображений часто встречаются линейные преобразования и свертки, обладающие свойством делимости. Данные преобразования выполняются как два (в двумерном случае) преобразования меньшей размерности, при этом требуется операция транспонирования. Предложена эффективная схема выполнения транспонирования, требующая использования только одной однопортовой памяти (но с отдельными шинами записи и считывания).

В четвертой главе предлагаемая в диссертации методика применяется к реальному примеру из предметной области, алгоритму вычисления градиентного поля изображения. Данный пример показателен в том смысле, что содержит все типичные конструкции чистого поведенческого описания алгоритма: вложенные циклы, ветвления, линейные участки.

Вычисление поля градиентов изображения является одним из распространенных способов подчеркивания границ на изображении для выделения контура объекта. Алгоритм состоит в сворачивании изображения с восьмью масками размером 3 на 3, и выбором максимального по модулю среди всех таких сверток. Данный алгоритм был описан на языке системы MatLab. В соответствии с предложенной методикой были выполнены этапы построения графа зависимостей, нахождения допустимого множества разверток, нахождения конвейерной развертки, синтеза граф-машины. Для записи покрывающих функций, областей их определения, разверток, систем неравенств, а также всех преобразований была использована матричная форма, удобная для реализации методики на компьютере. В результате формальным путем была получена конвейерная вычислительная структура для алгоритма вычисления градиентного поля изображения. Каждая ступень конвейера по величине задержки срабатывания не превышает задержку срабатывания сумматора, более глубокая конвейеризация приводит к разрастанию аппаратных затрат. Для семейств ПЛИС большой серийности Spartan3E или Cyclone2 скорость работы полученной конвейерной схемы составит около 150 МГц, при затратах сравнимых с неконвейерными реализациями, то есть порядка 500 функциональных ячеек. Частота работы конвейера 150 МГц вполне способна обеспечить обработку видеоизображений формата HDTV, либо монохромных размером 6 мегапикселей, идущих с частотой 25 кадров в секунду. Данная структура внедрена в изделия НИИ ВЦ «Кара-рат» при проведении ОКР «Сфера», а также ФГУП ВНИЦ ГОИ «Государственный оптический институт им. С.И. Вавилова» при проведении ОКР «Сатрап».

Был рассмотрен алгоритм быстрого вычисления двумерной корреляционной функции на основе использования преобразования с числами Ферма. Прямое вычисление корреляционной функции, а также на основе БПФ (Быстрого Преобразования Фурье) сложны для аппаратной реализации, поскольку содержат много умножений, а БПФ к тому же оперирует с комплексными числами. Указанных недостатков лишено БПФ в кольце вычетов по модулю числа Ферма, так называемое преобразование с числами Ферма (ПЧФ). Кольцо вычетов по модулю числа Ферма представляет собой конечное множество целых чисел, все элементы которого меньше заданного числа Ферма. На этом множестве определены операции умножения и сложения, отличающиеся от обычных операций с целыми числами тем, что выполняются по модулю числа Ферма. Умножение на матрицу коэффициентов ПЧФ, в отличие от БПФ, содержит умножения только на степень двойки, поэтому ПЧФ удобно для аппаратной реализации. Вычисление корреляционной функции требует выполнения двух прямых и одного обратного ПЧФ. С использованием предложенной методики была построена конвейерная вычислительная структура выполняющая прямое и обратное быстрое ПЧФ. Степень конвейеризации схемы доведена до уровня элементарной операции «бабочка ПЧФ», содержащей два сумматора и одно умножение на степень двойки (сдвиг) по модулю числа Ферма. Данный пример иллюстрирует возможность конвейерной реализации даже тех алгоритмов, которые обладают сложной структурой связей.

В заключении сформулированы основные результаты, полученные в диссертационной работе, состоящие в следующем:

1. Разработана методика поведенческого синтеза конвейерных схем, предназначенных для предварительной обработки изображений. В отличие от традиционных методик проектирования, данная методика ориентирована на исходное представление алгоритма в форме чистого поведенческого описания. Методика обеспечивает формальный переход от последовательной формы, наиболее удобной для большинства разработчиков, к конвейерному представлению, что позволяет сократить срок проектирования схем.

2. Для задач синтеза конвейерных схем на ПЛИС и полузаказных СБИС адаптированы элементы теории информационной структуры программ, которая изначально предназначалась для оптимизации программ под суперкомпьютеры и вычислительные кластеры. Использование новых предложенных преобразований позволяет формализовать проблему поиска различных аппаратных реализаций алгоритма, сведя ее к построению и решению систем линейных неравенств.

3. Предложен новый формализованный метод синтеза конвейерных схем. В качестве исходных данных выступает граф зависимости программы, заданный в виде системы покрывающих функций, а также допустимое множество разверток графа зависимостей. Предложенные в диссертации формальные преобразования графа позволяют построить структурную модель устройства, называемую граф-машиной или графом потока сигнала

4. Продемонстрирована возможность применения гомоморфной свертки граф-машины (а также и графа зависимостей) для получения последовательно-параллельных реализаций алгоритма, обеспечивающих компромисс между используемыми ресурсами и требуемой частотой работы. Исследованы алгоритмы, и получены высокопроизводительные схемы для базовых вычислительных операций, встречаемых в предварительной обработке изображений: двумерной свертки и корреляционной функции, двумерной разделимой свертки и двумерных разделимых линейных преобразований, вычисления градиентного поля изображения.

На основе полученных результатов можно сделать следующие выводы:

1. Предлагаемая в диссертации методика позволяет ускорить процесс разработки специализированных схем предварительной обработки изображений. Кроме рассматриваемой предметной области данная методика может использоваться и в других областях, где осуществляется обработка больших массивов данных в реальном времени.

2. Предлагаемые в диссертации методы делают возможным автоматизацию процесса синтеза конвейерных схем по последовательной программной реализации алгоритма. Большой практический интерес представляет создание на основе данной методики компилятора, способного преобразовывать последовательную поведенческую реализацию алгоритма в структурную конвейерную реализацию.

3. Исследование параллельной структуры алгоритма, являющееся основой предложенной методики, позволяет значительно повысить качество проектируемых схем. Применяемые методы анализа программ и синтеза схем позволяют исследовать все возможные параллельные реализации алгоритма и выбирать из них оптимальные в соответствии с заданным критерием, зависящим от целевой элементной базы.

СПИСОК ПУБЛИКАЦИЙ

1. **Анисимов И.Ю.** Оценка качества выравнивания каналов тепловизионных приемников при использовании метода микросканирования / И.Ю.Анисимов, В.И.Соловьев // Оптический журнал. – 2005. – т.72; № 6. – С. 47-50.
2. **Анисимов И.Ю.** Методика синтеза специализированных параллельных вычислительных структур цифровой обработки изображений / И.Ю.Анисимов, И.А.Малышев // Оптический журнал. – 2006. – т.73; № 3. – С. 35-39.
3. **Анисимов И.Ю.** Использование сигнальных процессоров для автоматической обработки изображений в реальном времени / И.Ю.Анисимов // Оптический журнал. – 2005. – т.72; № 4. – С. 44-47.