

На правах рукописи

Краснобаев Антон Александрович

**МЕТОД ДЕКОМПОЗИЦИИ АЛГОРИТМОВ СИСТЕМ
ТЕХНИЧЕСКОГО ЗРЕНИЯ НА ПАРАЛЛЕЛЬНО-
КОНВЕЙЕРНОЕ ПРОГРАММНО-АППАРАТНОЕ
ИСПОЛНЕНИЕ В АРХИТЕКТУРЕ ПЛИС-ЦСП**

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

А в т о р е ф е р а т
диссертация на соискание учёной степени
кандидата физико-математических наук

Москва 2008

Работа выполнена в Институте прикладной математики им. М.В.Келдыша РАН

Научный руководитель – доктор физико-математических наук, профессор
Платонов Александр Константинович

Официальные оппоненты:

- доктор физико-математических наук

Лацис Алексей Оттович

-кандидат физико-математических наук

Андреев Виктор Павлович

Ведущая организация:

Государственный научно-исследовательский институт авиационных систем
(ГосНИИАС)

Защита состоится 10 июня 2008 г. в 11 часов

на заседании Диссертационного совета Д 002.024.01 в Институте прикладной
математики им. М.В.Келдыша РАН по адресу: 125047, Москва, Миусская пл. 4.

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

Автореферат разослан 8 мая 2008 г.

Учёный секретарь совета

доктор физико-математических наук



Т. А. Полилова

Общая характеристика работы

Актуальность работы

Потребность в общедоступных *системах технического зрения (СТЗ)* диктует необходимость исследований новых способов программно-аппаратных решений для обработки и анализа изображений. Очевидным фактом является необходимость использования комбинации из параллельных и последовательных реализаций различных частей алгоритмов анализа изображений для достижения заданного быстродействия при минимальной себестоимости изделия. Использование для этой цели персональных компьютеров недостаточно эффективно и дорого, ввиду их избыточной универсальности и особенностей их архитектуры, мало приспособленной для параллельной и конвейерной обработки двумерных данных в масштабе реального времени. Поэтому для получения конкурентоспособных видеосистем с хорошими показателями по скорости обработки данных и себестоимости, необходима и актуальна разработка специализированных программно-аппаратных средств, допускающих программирование параллельно-конвейерного исполнения алгоритмов обработки видеоданных.

Для решения этой задачи желательно найти формальные методы оценки используемых алгоритмов анализа изображений, с точки зрения возможностей их эффективной параллельно-конвейерной программно-аппаратной реализации на существующей микропроцессорной элементной базе. К таким алгоритмам, прежде всего, относятся широко используемые алгоритмы первичной обработки видеоданных - алгоритмы нижнего и среднего уровней работы с изображениями.

Хорошо известно, что на рынке интегральных микросхем существуют выпускаемые массово по сравнительно низкой цене цифровые процессоры, приспособленные к использованию всего богатства возможностей программных алгоритмов, и логические интегральные схемы, позволяющие достигнуть высокой степени параллельности преобразования данных. Среди этих микросхем для обработки видеоданных наилучшими качествами универсальности и возможностей специализации обладают *цифровые сигнальные процессоры (ЦСП)* и *программируемые логические интегральные схемы (ПЛИС)*. Предполагается, что в архитектуре их совместного использования плохо параллелизуемые, логически ветвящиеся программные части алгоритмов должны выполняться на ЦСП, а ПЛИС должна использоваться для выполнения хорошо параллелизуемых и логически жёстко фиксированных частей алгоритма. В этом случае актуальна задача наилучшего разделения алгоритмов обработки изображений на части, выполняемые ЦСП, и на части, выполняемые ПЛИС. С другой стороны, опыт разработки видеосистем показал, что в таких системах возникает противоречие между алгоритмическими требованиями процессов доступа к большим объёмам памяти видеоданных, ограничениями объёмов внутренней памяти микросхем и быстродействием процессов обмена информацией с модулями внешней памяти.

Это приводит в процессе анализа способа программирования СТЗ к необходимости учёта особенностей памяти аппаратных модулей, влияющих на процесс программной реализации алгоритмов работы с видеоданными.

В связи с этим актуально решение поставленной в диссертации задачи поиска способов параллельно-последовательного программирования алгоритмов обработки изображений с одновременным формированием архитектуры микропроцессорных вычислительных средств и элементов памяти.

Цель работы

Целью работы является разработка метода параллельно-конвейерной программно-аппаратной реализации требуемого алгоритма системы технического зрения в минимальной по цене и эффективной по производительности архитектуре СТЗ.

Научная новизна

- Предложен метод описания алгоритмов преобразования цифрового изображения, позволяющий выявить особенности адресации обращений к памяти, содержащей изображение, и определить возможность параллельно-конвейерного исполнения алгоритма.
- Выделены структура и набор формальных параметров в *параметризованной схеме потоков данных (ПСПД)* алгоритмов для определения возможности их отображения на множество параметров микросхем.
- Предложен алгоритм использования внутренней памяти микросхем для реализации алгоритмов, использующих “оконные” операции.
- Построен формальный метод отображения алгоритмов обработки видеоданных на архитектуру вычислительных средств.

Практическая значимость и реализация

Полученные методы и правила позволяют формализовать процесс построения СТЗ, минимальной по цене при заданной производительности, с эффективной реализацией требуемых алгоритмов первичной обработки видеоизображений. В работе показано, что большое многообразие алгоритмов первичной обработки изображений может быть эффективно реализовано на однотипной аппаратной части в виде “рефлексного” видеодатчика, позволяющего совместить процессы получения видеоданных и их предобработки. Результаты диссертационного исследования были использованы при разработке камеры с её программируемыми откликами, на основе которой реализован сканер линейных и двухмерных штриховых кодов с разрешением 1280x1024 при частоте 30 кадров в секунду. По функциональным характеристикам сканер существенно превосходит известные аналоги при вдвое меньшей себестоимости. Имеются 2 патента на особенности разработанных алгоритмов и архитектуру сканера.

Апробация работы

Основные результаты работы докладывались и обсуждались на:

- Конференция "Мобильные роботы 2003", Россия, Москва, ИМЭХ МГУ (XI/03)
- Конференция "Мобильные роботы 2005", Россия, Москва, ИМЭХ МГУ (III/05)
- Объединенном семинаре по робототехническим системам ИПМ им. М.В. Келдыша РАН, МГУ им. М.В.Ломоносова, МГТУ им. Н.Э. Баумана и ИНОТиИ РГГУ; Москва, ИПМ им. М.В.Келдыша РАН (II/05 и X/06)
- XVII Международной конференции "Экстремальная робототехника", Санкт-Петербург, ЦНИИ РТК (IV/06)
- Семинаре по компьютерной графике и машинному зрению Ю.М. Баяковского Россия, Москва, МГУ, факультет ВМиК (X/07).

Структура и объём работы

Диссертация состоит из Введения, пяти глав, Заключения, Списка литературы и Приложения. Содержание работы изложено на 160 страницах (включая 18 страниц приложения). Список литературы включает 59 наименований. В работе содержится 75 рисунков и 15 таблиц.

Содержание работы

В первой главе даётся определение терминов, использованных в работе, описаны этапы и уровни обработки изображения в задачах машинного зрения. Показано, что алгоритмы обработки изображений нижнего и среднего уровней имеют большие потоки входных данных, что предъявляет высокие требования к каналу передачи данных между датчиком изображения и частями, выполняющими первичные алгоритмы. Вместе с этим приведён перечень задач с необходимостью использования зрительных данных в вычислительных модулях, расположенных на удалённом расстоянии от видеокамеры. Делается заключение, что с целью уменьшения уровня требований к пропускной способности канала передачи данных желательно создавать малогабаритные и малопотребляющие "рефлексные" видеокамеры со способностью осуществлять первичную обработку получаемого изображения. При этом целесообразно в конструкции таких камер предусмотреть аппаратную реализацию "рефлексных" механизмов предварительной обработки получаемых видеоданных.

Для этого необходим анализ существующих алгоритмов первичной обработки и существующих аппаратных средств, с точки зрения взаимной приспособленности их свойств в ограничениях быстродействия, памяти и цены. В главе приведен обзор существующих вычислительных средств и определены проблемы аппаратной реализации алгоритмов обработки изображений.

Во второй главе, с целью выделения общих и вариативных частей процедур обработки видеоданных, производится анализ известных из литературы

и широко используемых алгоритмов первичной обработки изображения. Показано, что характерной чертой большинства этих алгоритмов является наличие “оконных” операций обработки яркостей видеоданных, когда результат операции для каждого выходного пикселя изображения зависит от значений яркостей группы близлежащих входных пикселей. “Оконные” операции позволяют использовать внутреннюю память микросхем для хранения данных, многократно повторяющихся в соседних окнах. Такая возможность позволяет уменьшить обращения к внешней памяти, хранящей кадр видеозображения целиком. Свойство “оконности” операций может присутствовать не только для входных, но и для выходных данных (рис. 1). Наиболее распространённой “оконной” операцией является двумерный фильтр с конечной импульсной характеристикой (**КИХ-фильтр**):

$$y[n_1, n_2] = \sum_{i=0}^p \sum_{j=0}^q a_{ij} x[n_1 - i, n_2 - j], \quad (1)$$

где $x[n_1, n_2]$ - входное изображение, $y[n_1, n_2]$ - выходное изображение, a_{ij} - весовые коэффициенты фильтра. Из этой формулы следует важность аппаратной реализации операций умножения с накоплением (см. ниже).

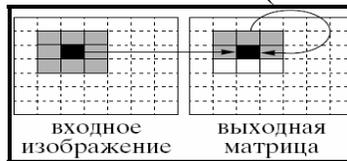


Рис. 1. “Оконность” адресации обращений к входным и выходным данным. Чёрным цветом показаны центральные пиксели окон, серым – остальные пиксели, входящие в окна.



Рис. 2. Группы анализируемых детекторов простых элементов изображения с числом детекторов в группе (в скобках).

Значительная часть главы посвящена анализу основного класса алгоритмов обработки изображения в СТЗ - алгоритмов детекторов простых элементов

изображения. Простыми элементами изображения являются контрастные перепады, контурные линии, угловые точки, круговые элементы и т.п. В диссертации подробно описан состав показанных на рис.2 групп анализируемых детекторов и типов детектируемых ими простых элементов изображения.

Для детекторов типа ДГ характерно использование градиента яркости для поиска элемента изображения. Оценка значений компонент градиента осуществляется при помощи масок, дифференцирующих изображение (являющихся КИХ-фильтрами). Сравнение модуля вектора градиента с порогом и анализ его изменения позволяют выделять простые элементы контуров или ориентации объектов изображения. Детекторы типа ДСШ осуществляют фильтрацию изображения набором “масок-шаблонов” для выявления фрагментов изображения, наиболее совпадающих с одной из них. Особенность детекторов типа ДПП проявляется в использовании преобразований функции яркости двумерной связной области (окна) в элементы некоторого абстрактного пространства. Искомый элемент изображения детектируется, если значения компонент получаемого вектора в абстрактном пространстве попали в некий допустимый диапазон. Примером служит детектор Хюккеля, в котором яркости кругового окна отображаются в параметры отрезка. Детекторы типа ДС обнаруживают искомые свойства элементов изображения, привлекая гипотезу о статистических свойствах функции распределения яркостей в окрестности анализируемого пиксела.

В связи с тем, что общим свойством этих детекторов является упомянутая "оконность" их алгоритмов, оказался возможным синтез универсальных архитектур СТЗ, позволяющих с помощью программной настройки реализовывать рассмотренное множество алгоритмов детектирования простых элементов изображения на широком множестве задач обработки видеоданных. Встроенная функция детектирования добавляет СТЗ свойства детерминированного отклика на получаемое изображение. Поэтому СТЗ с подобной архитектурой аппаратных средств ниже называются “рефлексными”.

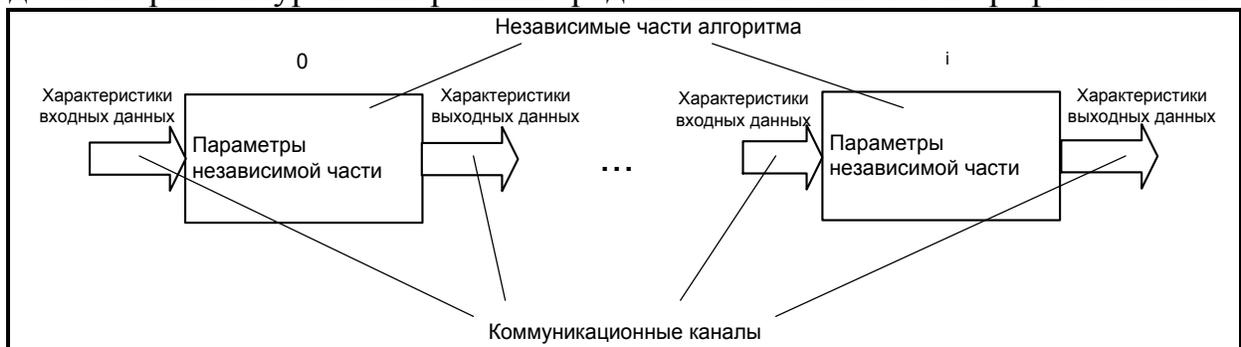


Рис. 3. Элементы ПСПД.

В главе формируются критерии оценки минимального по цене при заданной производительности разбиения алгоритмов детектирования на части,

выполняемые ПЛИС и ЦСП. Каждый из алгоритмов представлен в виде ПСПД¹ (см. рис. 3), учитывающей в себе коммуникационные расходы на передачу данных и параметры, характеризующие эффективность использования различных аппаратных архитектур для реализации каждой независимой части алгоритма. При построении ПСПД в ней не должно быть циклов, и притом независимые части должны содержать минимально возможный набор операций над изображением. Чем меньше будет гранулярность независимых частей, тем точнее можно выбрать место разбиения алгоритма между вычислителями (ПЛИС и ЦСП). Так как ПСПД не содержит циклов, то всегда возможно конвейерное исполнение независимых частей. Для возможности параллельного исполнения части алгоритмов не должны быть зависимыми друг от друга по данным, что с точки зрения ПСПД означает, что ни одна из этих частей не является предшествующей для любой другой.

Алгоритм, каждого из рассмотренных детекторов элементов изображения, можно разложить на операторы преобразования параметров цветовой яркости (V_1, V_2, V_3) и положения (i, j) пикселей изображения в параметры выходных данных детектора. С точки зрения задачи отображения алгоритма на архитектуру аппаратных средств в его составе следует выделить:

- операторы преобразования адресов обращений к памяти,
- сами операторы обращения к памяти и
- операторы преобразований получаемых значений яркости пикселей изображения.

С целью описания свойств независимых частей алгоритмов в главе введено понятие *графа зависимости алгоритмических параметров* (называемого ниже – "ЗАП-граф"). На рис. 4 представлено графическое изображение операторов ЗАП-графа.

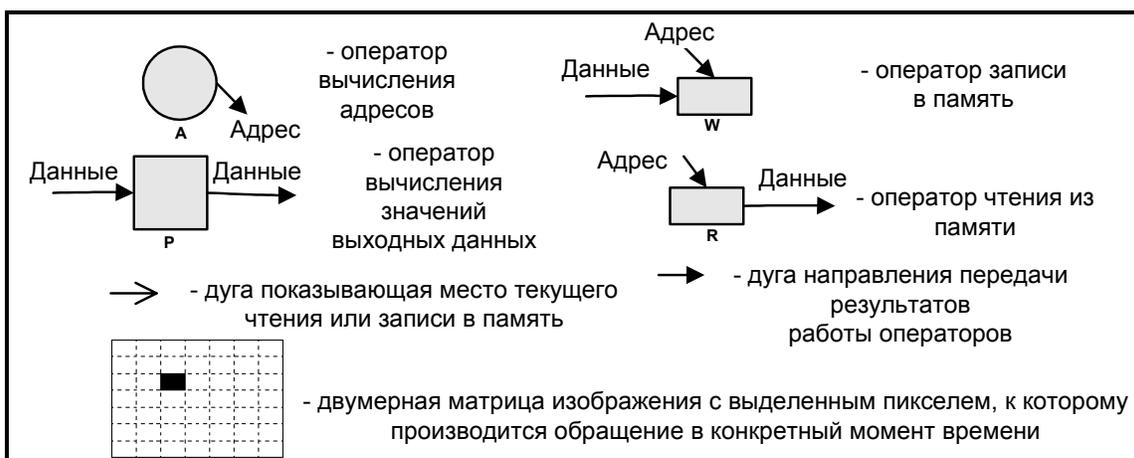


Рис. 4. Элементы ЗАП-графа.

В диссертации построены ЗАП-графы для всех независимых частей ПСПД алгоритмов детекторов простых элементов изображения приведённых на рис. 2.

¹ В диссертации приводятся ПСПД для алгоритмов детекторов.

ЗАП-граф для наиболее распространённых “**оконных**” операций показан на рис. 5. Оператор $A_{вх}$ определяет адреса пикселей, входящих в последовательно перемещаемое по всему изображению окно, $R_{вх}$ получает значения их яркостей, P выполняет операцию над значениями яркостей нескольких пикселей в окне, и оператор $W_{вых}$ записывает результат по адресу, определяемому оператором $A_{вых}$ и зависящему только от адреса центрального пикселя окна.

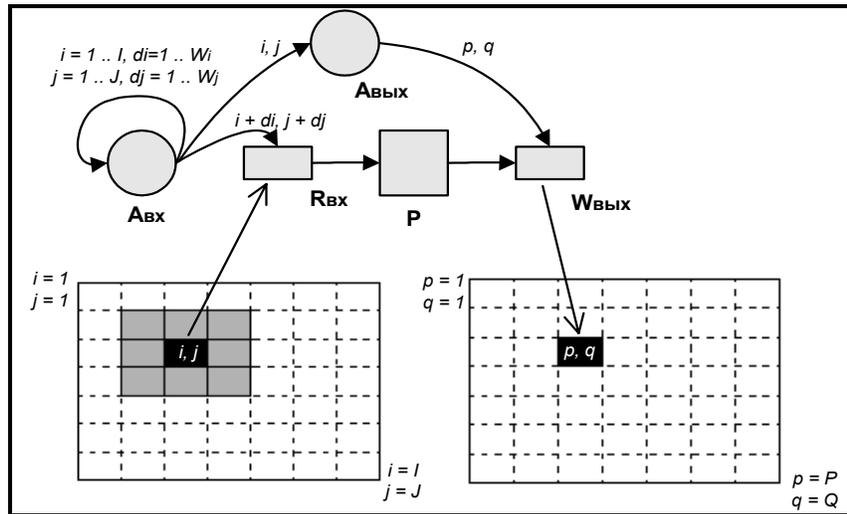


Рис. 5. ЗАП-граф для “оконных” операций.

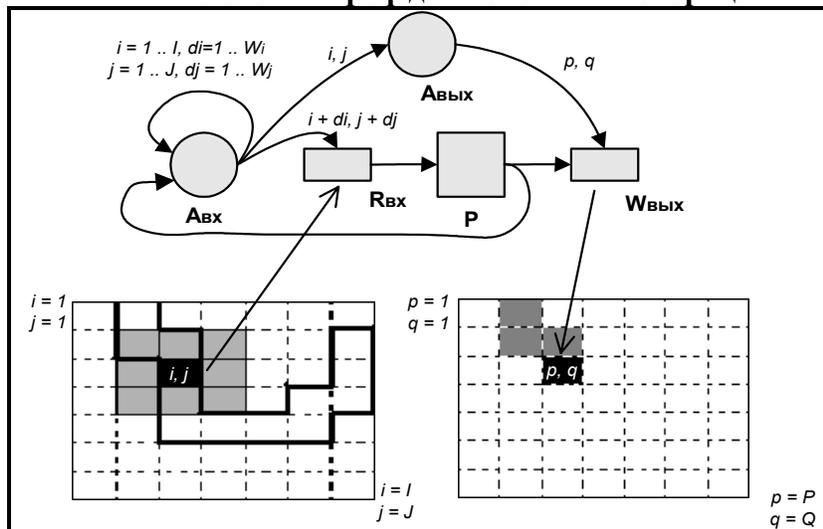


Рис. 6. ЗАП-граф алгоритма прослеживания контуров в детекторе Канны.

Иначе обстоит дело с **алгоритмом прослеживания контуров** (см. рис. 6), являющегося частью **детектора Канны**. Здесь происходит выбор пикселей вдоль контура, определяемого яркостями соседних пикселей. Следовательно, в операторе $A_{вх}$ определяется адрес очередного входного пикселя по результату работы оператора P и адресу предыдущего входного пикселя. Затем оператор $R_{вх}$ осуществляет доступ к элементу памяти по этому адресу и считывает значение яркости очередного пикселя входного изображения. Алгоритм оператора P отвечает на вопрос, является ли данный пиксель частью контура, и, если ответ

положительный, результат записывается оператором $W_{\text{вых}}$ по адресу, определяемому оператором $A_{\text{вых}}$, с учётом адреса входного пикселя.

ЗАП-граф преобразования Хака на рис.7 отличается от описанных детекторов свойством более сложной адресации к входным и выходным данным. В наиболее принятой реализации этого алгоритма выполняется суммирование значений яркостей пикселей, попавших на последовательно генерируемые линии, заданной формы. Алгоритмом оператора $A_{\text{вых}}$ перебираются все возможные значения параметров линий, т.е. осуществляется последовательная адресация ко всем элементам выходной матрицы, описывающей пространство параметров линий. Для очередного вектора параметров оператор $A_{\text{вх}}$ осуществляет вычисление адресов пикселей, по которым проходит линия внутри кадра изображения. При помощи оператора $R_{\text{вх}}$ реализуется чтение в этих пикселях значений яркостей, и выполняется их суммирование оператором P . Используя $W_{\text{вых}}$ осуществляется запись полученного значения суммарной яркости линии по адресу выходной памяти, определяемому оператором $A_{\text{вых}}$. Очевидно, что, несмотря на сложность вычислений, этот алгоритм допускает конвейерное исполнение его частей.

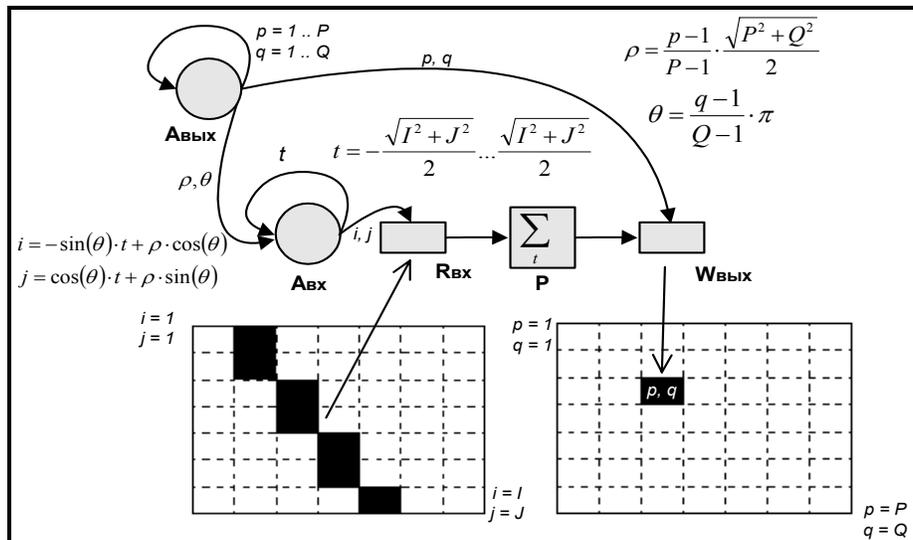


Рис. 7. ЗАП-граф алгоритма преобразования Хака (Hough) для прямых линий. На вход алгоритма поступает бинарное изображение.

Анализ содержания ЗАП-графов показал, что важными параметрами алгоритмов для их отображения на архитектуру ПЛИС-ЦСП с учётом условий быстродействия и эффективности использования АЛУ являются:

1. Количество элементарных операций на один пиксель входного изображения.
2. Содержательная зависимость входных-выходных адресов (СЗА) алгоритма.
3. “Оконность” алгоритмических операций.
4. Отсутствие-присутствие циклов ЗАП - графа.

Этот анализ показал также, что в большом числе случаев “оконных” алгоритмов нижнего и среднего уровней обработки изображений используются

операции без СЗА преобразования видеоданных с последовательным перемещением центра окна по адресам кадра (т.е. - с графом зависимости, показанным на рис. 5). Поэтому алгоритмам с этими свойствами далее в главе было уделено наибольшее внимание. Была исследована зависимость минимально необходимого объёма внутренней памяти от параметров размера окна и его шага. Далее для “оконных” типов обращения к памяти (в предположении использования прогрессивной, а не "чересстрочной" развёртки кадра видеоданных) исследованы два способа использования внутренней памяти микросхем. Первый способ заключается в **буферизации входных данных**, поступающих последовательно по элементам строк с внешнего модуля памяти или - с датчика изображения с его прогрессивной развёрткой. Второй – в **накапливании промежуточных результатов вычислений** для аналогичных входных данных.

Преимущество буферизирующего способа - в независимости требуемого объёма буфера памяти от количества используемых фильтров. Преимущество накапливающего способа – в экономии требуемого объёма памяти при использовании прореженных результатов “оконной” операции. Ограничением накапливающего способа является требование к коммутативности операций, использующих на входе данные разных пикселей изображения.

Значения объёмов внутренней памяти для буферизирующего метода “оконных” преобразований вычисляются по формуле (2), для накапливающего - по формуле (3).

$$M_{BUF} = (C + 1) \cdot (W - 1) \cdot B_{pix}, \quad (2)$$

$$M_{acc} = N_{acc} \cdot ceil(B_{acc}) = ceil\left(\frac{C}{d}\right) \cdot ceil\left(\frac{W}{d}\right) \cdot ceil(2 \log_2 W + B_{pix} + B_{filt}). \quad (3)$$

Здесь C – длина строки изображения, d – шаг окна, W – размер окна, $ceil()$ – функция округления до большего среднего, B_{pix} – количество бит для представления яркости пикселей изображения, B_{filt} – количество бит для представления весовых коэффициентов фильтра.

Для более полной параметризации ПСПД необходимо было охарактеризовать и свойства коммуникационных каналов. Здесь было желательным найти характеристику, независящую от пространственного разрешения кадра и частоты смены кадров. Было принято, что каждая независимая часть алгоритма характеризуется её коэффициентом КД изменения объёмов данных в процессе их преобразования. Например, для алгоритма бинаризации монохромного 256-градационного (8-ми битного) изображения $KD=1/8$.

Полученный набор схемных параметров (см. рис. 8), характеризующий независимые части алгоритмов с точки зрения их аппаратной реализации, включает:

1) свойства, выделенные на ЗАП–графах

- a) “оконность” алгоритмических операций,
 - b) СЗА входных-выходных адресов алгоритма,
 - c) отсутствие-присутствие циклов,
- 2) количество элементарных операций, приходящихся на один пиксель входного изображения,
 - 3) необходимый объём внутренней памяти,
 - 4) коэффициент изменения объёма выходных данных по отношению к входным.

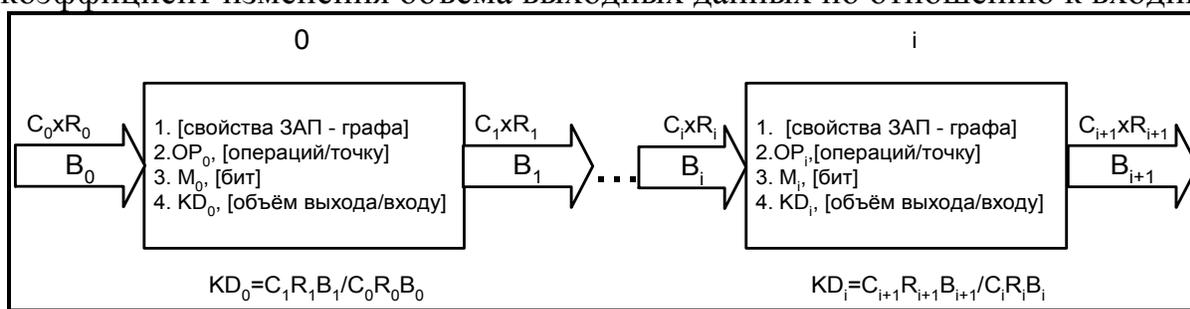


Рис. 8. Параметры ПСПД.

Здесь OP – количество операций на один элемент входных данных, M – требуемый объём внутренней памяти, C, R - количество столбцов и строк входных данных, B – разрядность данных.

Третья глава содержит описание архитектур микросхем, применяемых для построения СТЗ. Анализируются и оцениваются параметры приспособленности ПЛИС, ЦСП и различных видов памяти к алгоритмическим требованиям реализации первичных алгоритмов обработки зрительных данных. Определена наиболее эффективная (в вычислительном плане) роль каждой микросхемы в обработке потока данных изображения.

Как известно, микросхемы внешней памяти (**МВП**), в зависимости от интерфейса, бывают синхронными и асинхронными, а в зависимости от их физического устройства - статическими и динамическими. Заметными недостатками динамической памяти являются значения длительности времени доступа к ней, наличие циклов регенерации содержимого (1 раз в 64 ms, занимает около 0.3% от общего времени доступа) и переключение страниц памяти T_{NSA} (на что требуются затраты времени не менее 7-ми тактов работы шины этой памяти). Опыт использования динамической памяти показывает, что циклы регенерации могут вызвать сбой при сохранении в памяти изображения, поступающего в систему в реальном времени. А переключение её страниц существенно тормозит процесс обращения к данным, расположенным на разных строках изображения. В то же время память статического типа, несмотря на отсутствие в ней всех описанных недостатков, является более дорогостоящей и более энергопотребляющей. Общим недостатком для всех типов памяти (в отличие, например, от регистрового файла процессора) является наличие задержки T_{RL} момента времени чтения данных (времени между выставлением адреса и получением результата). Это свойство существенно для оценки реализуемости

алгоритмов с СЗА входных данных, т.к. невозможно определить адрес до получения и завершения обработки предыдущей порции данных.

В главе приведены рекомендации, уменьшающие влияние перечисленных недостатков на цену и энергопотребление при работе с МВП. В соответствии с этими рекомендациями, выбирается одна из предложенных в четвёртой главе структур соединения ПЛИС, ЦСП и МВП.

Далее в главе рассмотрены свойства микросхем ЦСП. Основное достоинство ЦСП связано с его универсальностью и логическими возможностями смены программ. Это делает удобным его использование на верхних уровнях распознавания изображений (позволяя строить гибкую модель содержания наблюдаемой сцены). Вместе с тем наличие достаточного количества внутренней памяти делает удобным использование ЦСП и на нижнем уровне обработки видеоданных (позволяя быстро реализовывать “оконные” операции). Наличие в системе команд ЦСП операции умножения с накоплением позволяет заметно оптимизировать производительность вычислений при реализации КИХ-фильтров. К преимуществам ЦСП следует отнести и наличие средств аппаратной поддержки работы с памятью (от реализации интерфейсов с внешней памятью до механизмов кэширования). Недостатком ЦСП является низкий уровень возможного параллелизма реализации алгоритмов преобразования видеоданных (не более 8-ми одновременных 16 x 16 битных умножений за один такт ЦСП)¹.

Затем в главе рассмотрены свойства микросхем ПЛИС. Так как частота работы ПЛИС не является высокой, в сравнении с частотой сигнальных процессоров, то основное преимущество использования ПЛИС заключается в возможности программирования структуры соединения их внутренних элементов. Учитывая это, наиболее эффективно использовать ресурсы ПЛИС для обработки изображений, реализуя на них **ZIMD** архитектуры (zero instructions multiple data/ ноль инструкций множество данных). ПЛИС является идеальной базой для мелко-гранулярных параллельных вычислений, благодаря возможностям одновременного использования большого количества конфигурируемых и независимых вычислительных элементов. Кроме того, эти микросхемы позволяют организовывать конвейерные вычисления с большой длиной конвейера. Архитектура ПЛИС даёт возможность распределить требуемые процедуры вычислений между независимыми частями АЛУ ПЛИС и блоками внутренней памяти (каждый блок памяти ПЛИС фирмы Altera составляет 4608 бит), позволяя избежать простоев АЛУ в ожидании данных. Возникает возможность работать с видеоканальными данными, обрабатывая их ещё до момента времени сохранения изображения в МВП, что позволяет уменьшить трафик через шину памяти. Через ПЛИС, образно говоря, “прокачивается” поток видеоданных с прогрессивной развёрткой. Очевидными преимуществами такого подхода является малая задержка между моментами

¹ Данные на январь 2008 г.

получения видеоданных и получением результатов обработки (данные одного кадра изображения обрабатываются по мере их поступления с датчика изображения). Преимущество ПЛИС, связанное с наличием большого количества конфигурируемых портов ввода/вывода, позволяет использовать ПЛИС как коммутатор потоков данных шин нескольких МВП, способствуя этим разделению трафика через различные шины памяти (архитектура с распределённой памятью). С помощью ПЛИС можно организовать подключение различных потребителей к различным МВП (архитектура с обобщённой памятью). Накладными расходами при этом является удлинение конвейера доступа к внешней памяти, что увеличивает задержку обращения к ней T_{RL} . Невысокая тактовая частота ПЛИС (~250 МГц) накладывает большие ограничения (по сравнению с ЦСП) при выполнении алгоритмов с СЗА или с циклами на ЗАП-графе. Это связано с отсутствием возможности одновременного выполнения операторов графа, входящих в цикл.

В главе показывается, что с учётом описанных свойств элементной базы СТЗ, для выбора архитектуры имеют важные значения следующие свойства ЗАП-графа:

1. Количество элементарных операций, приходящихся на один пиксель входного изображения, определяет необходимую суммарную производительность АЛУ. Эти значения для детекторов показаны внутри операторов их ЗАП-графов.
2. СЗА входных-выходных адресов алгоритма. При СЗА снижается скорость выполнения оператора R как минимум на время T_{RL} по причине большой длины конвейера доступа к памяти. Здесь T_{RL} время работы конвейера обращения к памяти (время на выставление адреса и получение данных). Алгоритмы, ЗАП-графы которых имеют СЗА, эффективнее исполняются при использовании памяти с меньшей задержкой чтения.

Особенности свойств генерации адресов видеоданных. Табл. 1.

СЗА		нет СЗА	
входных данных	выходных данных	последовательный доступ к памяти	непоследовательный доступ к памяти
<ul style="list-style-type: none"> • работа только с буферизированными в МВП данными, • накладные расходы на работу с памятью T_{RL} и T_{NSA}. 	<ul style="list-style-type: none"> • возможна работа с видеоканальными данными, следующими с прогрессивной развёрткой. 	<ul style="list-style-type: none"> • возможна работа с видеоканальными данными, следующими с прогрессивной развёрткой. 	<ul style="list-style-type: none"> • работа с буферизированными в МВП данными, • накладные расходы на работу с памятью T_{NSA}.
Примеры:			
прослеживание контуров детектора Канни	Формирование гистограммы яркостей	ДСП	преобразование Хака

С точки зрения быстродействия получения данных возможность организации последовательного доступа к памяти важна при выборе между динамической и статической памятью. Поэтому для операций, не имеющих

свойства СЗА, в главе вводится дополнительная характеристика их ЗАП-графа: *последовательный* (или *непоследовательный*) *доступ к памяти*.

Описанные особенности генерации адресов видеоданных сведены в табл. 1.

3. “Оконность” алгоритмических операций. Наличие описанного во второй главе свойства “оконности” позволяет, используя внутреннюю память, увеличить скорость доступа к видеоданным. Для наиболее распространённых алгоритмов, у которых нет СЗА и при этом реализуемо последовательное перемещение окна, требуемый объём внутренней памяти может быть вычислен по формулам (2) или (3). Если свойства “оконности” нет, то нет и необходимости последовательного использования одних и тех же данных. Это снижает потребность использования внутренней памяти микросхемы.
4. Отсутствие-присутствие циклов ЗАП-графа. Наличие цикла исключает возможность параллельного или конвейерного выполнения его операторов (т.к. над очередной порцией данных необходимо выполнение всех операторов цикла до поступления новой порции данных). Для работы в темпе поступления видеоданных суммарное время работы всех операторов, входящих в цикл, должно быть не больше периода этого темпа. На рис. 9-а показан пример цикла ЗАП-графа, образованного операторами **A**, **R** и **P**. Этот цикл исполняется над одной порцией данных в течение времени $T_A + T_R + T_P$. Ни один из операторов цикла не начинает работу до окончания работы предыдущего, что резко сокращает производительность, в частности, - оператора **R** ($T_R = T_{RL}$). На рис. 9-б операторы **R** и **W** - оба работают с одним и тем же элементом памяти, в результате образуется цикл из операторов **R**, **P** и **W**.

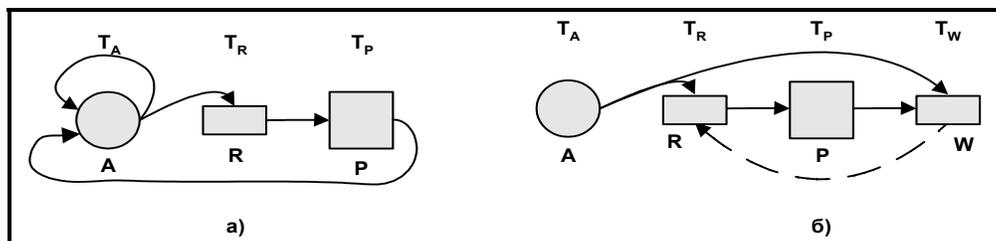


Рис. 9. Циклы на ЗАП-графах.

В четвёртой главе рассматривается организация взаимодействия между микросхемами ПЛИС, ЦСП и МВП, рассчитываются потоки данных между ними. С использованием параметров ПСПД осуществляется поиск разбиений каждого алгоритма i на части, реализуемые в ПЛИС и в ЦСП. Искомое разбиение выполняется путём минимизации значения следующей функции:

$$\min_{k_i} (price_{\text{плис}}(\sum_i OP_{\text{плис}i}, \sum_i M_{\text{плис}i}, \sum_i D_{\text{плис-цсп}i}) + price_{\text{цсп}}(\sum_i OP_{\text{цсп}i}, \sum_i M_{\text{цсп}i}, \sum_i D_{\text{плис-цсп}i})), \quad (4)$$

здесь k_i – переменная, определяющая место разбиения i -ого алгоритма, $price()$ – таблично заданная функция цены микросхемы от параметров, OP – требуемый в алгоритмах параметр производительности (количества операций в секунду), M –

требуемый объём внутренней памяти, $D_{\text{ПЛИС-ЦСП}}$ – поток данных между микросхемами ПЛИС и ЦСП. Параметры производительности, объёма внутренней памяти и потока данных детектора не должны превышать их значений из таблиц микросхем. При этом ограничения, влияющие на разбиение алгоритма между ПЛИС и ЦСП, определяются свойствами, выделенными на ЗАП–графах.

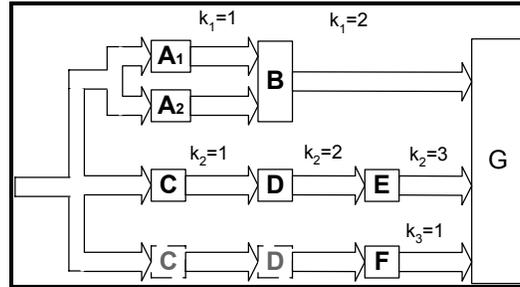


Рис. 10. ПСПД для трёх алгоритмов с независимыми частями A_1A_2B , CDE и CDF .

Используя ПСПД (рис. 10) для различных вариантов разбиения между ПЛИС и ЦСП частей алгоритмов тех детекторов, которые нужны для создания конкретной СТЗ, можно определить необходимые параметры ПЛИС и ЦСП. Прежде всего, необходимо исключить повторяющиеся в разных алгоритмах независимые части, работающие с одними и теми же данными (независимые части C и D на рис. 10). Затем для независимых частей, использующих буферизирующий способ работы с внутренней памятью и обрабатывающих одни и те же данные, обнуляются требуемые объёмы внутренней памяти, за исключением части, требующей наибольший объём. Необходимые производительность и объём внутренней памяти ПЛИС определяются суммированием K_i независимых частей алгоритма каждого детектора i , которые должны или могут выполняться на ПЛИС (5, 6). Для ЦСП требуемые характеристики определяются суммированием параметров L_i-K_i частей (7, 8).

$$OP_{\text{ПЛИС}} = F_{\text{кадр}} \cdot \sum_{i=1}^I \sum_{k=1}^{K_i} C_k \cdot R_k \cdot OP_k, \quad (5)$$

$$M_{\text{ПЛИС}} = \sum_{i=1}^I \sum_{k=1}^{K_i} M_k. \quad (6)$$

$$OP_{\text{ЦСП}} = F_{\text{кадр}} \cdot \sum_{i=1}^I \sum_{k=K_i+1}^{L_i} C_k \cdot R_k \cdot OP_k, \quad (7)$$

$$M_{\text{ЦСП}} = \sum_{i=1}^I \sum_{k=K_i+1}^{L_i} M_k \quad (8)$$

Здесь $F_{\text{кадр}}$ – частота кадров обрабатываемого изображения, C_k , R_k – количество столбцов и строк обрабатываемых данных на входе k -ой части алгоритма каждого детектора, OP_k – количество операций на один пиксел в k -ой части алгоритма каждого детектора.

В свою очередь требование к пропускной способности интерфейса и канала передачи видеоданных между ПЛИС и ЦСП определяются из:

$$D_{\text{ПЛИС-ЦСП}} = F_{\text{кадр}} \cdot \sum_i Com_{K_i+1}, \quad (9)$$

где Com_{K_i+1} – поток данных между частями алгоритма K_i и K_i+1 .

$$Com_{K_i+1} = C_{K_i+1} \cdot R_{K_i+1} \cdot B_{K_i+1}, \quad (10)$$

где B_{K_i+1} – разрядность выходных данных K_i -ой части алгоритма.

Значение параметра Com_{K+1} можно вычислить и другим путём - через коэффициенты KD изменения объёмов потоков данных в каждой из частей алгоритма (пример вычисления KD показан на рис. 8). Пусть $Pre(K)$ – множество независимых частей, непосредственно предшествующих части K , замыкающей перечень алгоритмических частей, исполняемых в ПЛИС, тогда

$$Com_{K+1} = KD_{K+1} \cdot \sum_{k \in Pre(K)} Com_k . \quad (11)$$

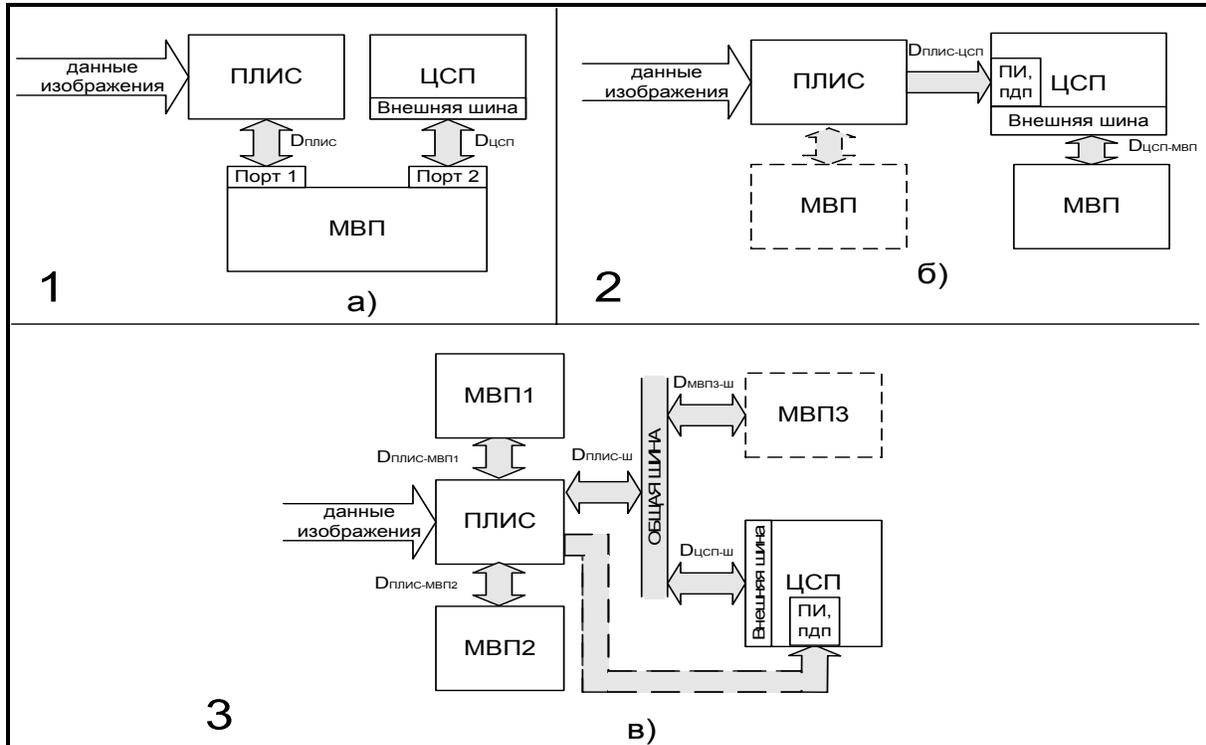


Рис. 11. Архитектуры СТЗ, состоящих из ПЛИС, ЦСП и МВП.

Здесь D - объём передаваемой информации между микросхемами в секунду, ПИ – периферийный интерфейс, ПДП – канал прямого доступа к памяти.

Таким образом, задача параллельно-последовательного распределения частей алгоритмов детекторов сводится к определению таких значений $k=K_i$, которые в совокупности минимизируют критерий (4) с учётом условия выполнимости алгоритмических частей с индексами меньшими k , определяемого ограничениями возможностей ПЛИС.

В процессе решения этой задачи следует выбрать структуру соединений микросхем ПЛИС, ЦСП и МВП. Важные ограничения в структуре их соединения связаны с каналами передачи данных. При одновременном обращении к памяти и ПЛИС, и ЦСП возможна ситуация недостаточной пропускной способности их общей шины. Наиболее простой выход - архитектура с использованием двухпортового МВП. Такая архитектура позволяет организовать конвейерную работу с обменом данными через обобщённые участки памяти, не затрагивая трафик через шину данных каждого из вычислительных модулей. Единственным

недостатком этого варианта является высокая цена на двухпортовые модули памяти, превосходящая цену однопортовых динамических модулей памяти такого же объёма почти в 100 раз. Этот вариант архитектуры (рис.11-а) целесообразен при высоких частотах видеоданных.

В ряде случаев (см. табл.1) возможна детекторная обработка непосредственно видеоканальных данных без предварительной буферизации всего кадра в МВП. Данный способ, во-первых, позволяет уменьшить трафик данных через шину присоединения МВП для сохранения изображения, во-вторых, он позволяет уменьшить задержку в обработке очередного кадра и, в третьих, - не использовать адресные линии (это упрощает разводку проводников печатной платы и уменьшает количество занятых ножек ввода/вывода в ПЛИС). Поэтому в пределах, определяемых допустимыми нагрузками трафика через шину данных, эффективно использовать наиболее дешёвый вариант архитектуры, показанный на рис. 11-б. Здесь трафик данных является однонаправленным от ПЛИС к ЦСП. В этом варианте достаточно просто можно динамически задавать адреса памяти для хранения данных, выходящих из ПЛИС. Эта информация через элементы ЦСП (ПИ и канал ПДП) попадает на шину ЦСП-МВП. Кроме того, через эту шину осуществляется кэширование программного кода и работа с хранимыми в МВП данными.

В общем случае, ПЛИС позволяет довольно гибко синтезировать архитектуры с различными встроенными в ПЛИС коммутаторами шин. Ценой такой гибкой коммутации является увеличение задержки начала процесса передачи данных T_{RL} (пропускная способность при этом остаётся той же). С использованием "ПЛИС-коммутаторов" можно сократить потери эффективности исполнения алгоритма, связанные с большими объёмами передачи видеоданных. В этом случае существует возможность выполнить требования трафика через шину МВП, при работе в темпе реального времени, и, в то же время, обеспечить передачу данных из ПЛИС в ЦСП. Вариант такой реализации показан на рис. 11-в. В этой архитектуре, поступающие кадры поочередно занимают одну из двух МВП, присоединённых к ПЛИС, в то время как с другой МВП через контроллер, реализованный на ПЛИС, работает ЦСП. В главе показано, что этот вариант не эффективен в случае необходимости использования СЗА обращения к памяти при выполнении алгоритма на ЦСП. Показано также, что снижение трафика через общую шину данной архитектуры можно достичь, используя ПИ ЦСП и канал ПДП. В таком варианте ЦСП заказывает копирование блоков изображения из МВП1, или - МВП2. ПЛИС осуществляет посылку заказанных данных в ЦСП через ПИ и канал ПДП. Это исключает операции чтения больших объёмов данных из МВП1 и МВП2 по общей шине, где остаются только операции записи в МВП3.

Каждая из приведенных архитектур соединения микросхем ПЛИС, ЦСП и МВП по-своему хороша (или плоха) для реализации алгоритмов обработки

видеоданных. В главе оценивается возможность и преимущества использования каждой из трёх описанных видов архитектур рефлексных камер для реализации различных типов детекторов простых элементов изображения. Полученное отображение множества алгоритмов детектирования на множество архитектур показано в табл. 2.

Результат синтеза архитектур детекторов простых элементов изображения. Табл. 2.

№ детектора (рис.1)	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
№ архитектуры	2	2	2	1	2	2	2	2	2	2	2	2	3	2	2

Из таблицы видно, что для большинства детекторов наиболее подходит архитектура, показанная на рис. 10-б. Исключения составляют лишь детектор контуров Канни и детектор Хака для поиска прямых линий, присутствующих в кадре изображения. В главе содержится анализ причин целесообразности выбора другой архитектуры для реализации этих детекторов и приводятся рекомендации по выбору архитектуры, наиболее подходящей для реализации заданного множества детекторов, содержащих в своём составе детекторы Канни и Хака.

Таким образом, формальное описание метода синтеза специализированной архитектуры для реализации заданного алгоритма обработки видеоданных на ПЛИС и ЦСП состоит из последовательности следующих действий:

1. Алгоритмы разбиваются на независимые части.
2. Строятся ЗАП-графы независимых частей алгоритмов.
3. Вычисляются необходимые объёмы внутренней памяти для “оконных” (2), (3) и для других используемых операций.
4. Строится ПСПД.
5. Выделяется подмножество совпадающих независимых частей разных детекторов, работающих с одинаковыми данными. Они исключаются из всех детекторов кроме одного.
6. Выделяется подмножество независимых частей, использующих внутреннюю память буферизирующим способом и работающих с одинаковыми входными данными. Обнуляются требуемые объёмы внутренней памяти во всех независимых частях, за исключением части, требующей наибольший объём.
7. Выбирается одна из трёх архитектур их соединения, исходя из выработанных рекомендаций главы 4.
8. Оцениваются трафики через шины и вычисляются значения критериев k_i , используя зависимости (4 - 11).
9. Проверяется выполнимость циклов ЗАП-графа в выбранных микросхемах и архитектуре. Если цикл не выполняется, то следует видоизменить алгоритм или выбрать более удачные микросхемы, минимизируя время выполнения операторов цикла.
10. Оцениваются необходимые объёмы внутренней памяти ПЛИС для группировки обработанных данных в блоки и передачи по интерфейсу.
11. Окончательно выбираются микросхемы.

В пятой главе описанный метод синтеза специализированных архитектур демонстрируется на примере выбора элементной базы и архитектуры СТЗ для распознавания штриховых кодов (ШК).

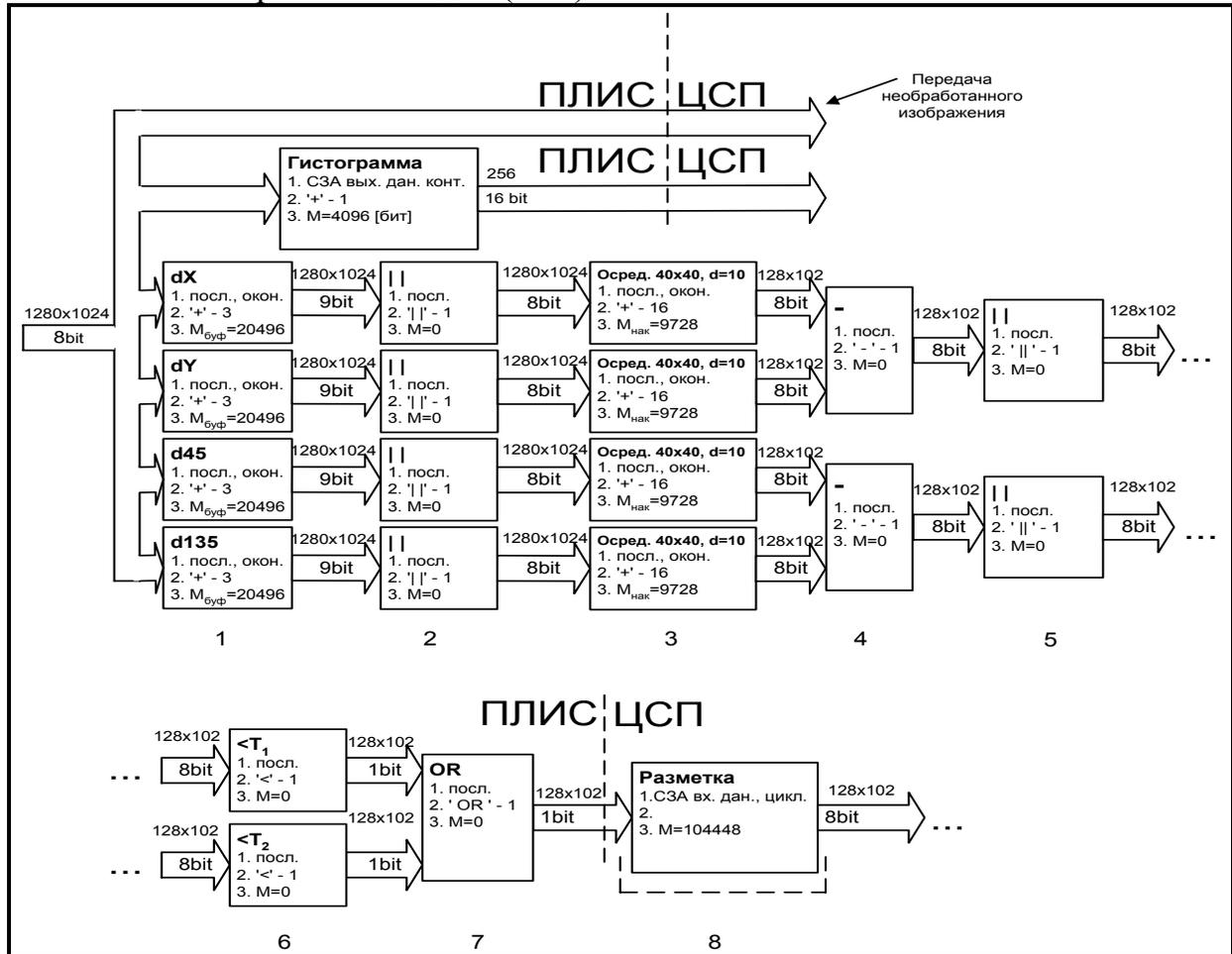


Рис. 12. ПСПД алгоритма локализации штрихового кода.

Задача распознавания и алгоритм её решения описываются в начале главы. Особенности оптической части делают критичным ко времени исполнения алгоритм локализации ШК, который должен производить обработку изображения разрешением SXGA (1280x1024) с частотой 30 кадров/сек. После обнаружения ШК происходит его декодирование, в течение этого периода времени локализация ШК на вновь поступающих кадрах изображения продолжается, в МВП сохраняются только те кадры, на которых найдены ШК. Отсутствие требований к высокому быстродействию алгоритмов декодирования ШК позволяет реализовать их на ЦСП. Таким образом, производится разложение между ПЛИС и ЦСП алгоритма локализации ШК, ПСПД которого показана на рис. 12. На этой схеме показаны также параметры задачи передачи необработанного изображения в МВП и формирования гистограммы яркостей элементов изображения.

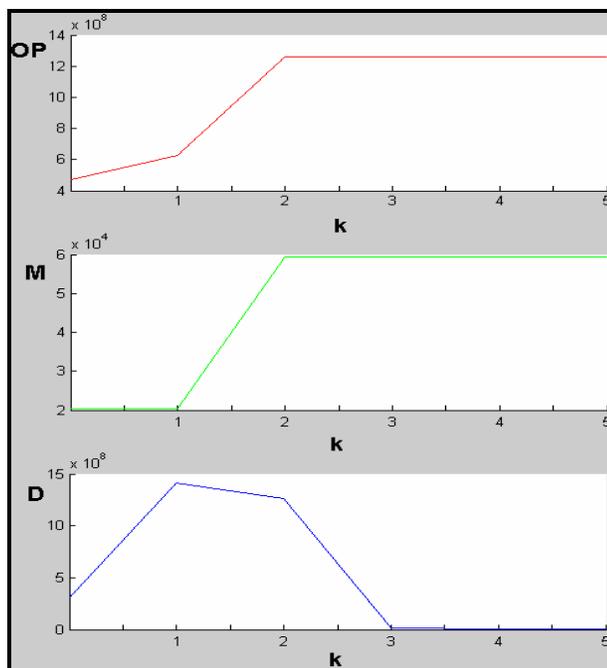


Рис. 13. Требуемые от ПЛИС в задаче локализации ШК ресурсы производительности – **OP**, объём внутренней памяти – **M**, объём потока данных между ПЛИС и ЦСП – **D**, в зависимости от места разбиения алгоритма между ПЛИС и ЦСП - **k**.

Независимые части 1..7 алгоритма локализации ШК имеют характеристику последовательного доступа к памяти, следовательно, они могут работать непосредственно с видеоканальными данными. Независимая часть 1 имеет признак “оконности”, и при этом шаг окна $d=1$ составляет один пиксель, что означает высокую эффективность буферизирующего метода использования внутренней памяти ($M_{буф}=20496$ бит или 5 блоков памяти ПЛИС, т.е. 23040 бит). Независимая часть 3 также имеет признак “оконности”, но шаг окна $d=10$, следовательно, более эффективен накапливающий метод использования внутренней памяти ($M_{нак}=4*9728$ бит или 4*3 блока памяти ПЛИС, т.е. 4*13842 бит). На рис. 13 показаны требуемые ресурсы ПЛИС, в зависимости от места разбиения k алгоритма между ПЛИС и ЦСП.

Часть 8 имеет СЗА входных данных, что делает невозможным работу с видеоканальными данными и требует сохранения изображения в МВП (нужно отметить, что СЗА можно избежать за счёт изменения алгоритма, предусматривающего несколько проходов по изображению и генерацию большого блока вспомогательных данных). Поскольку эта часть алгоритма не использует преимуществ ПЛИС и сложна в аппаратной реализации, то она реализована в ЦСП.

Задача вычисления гистограммы распределения яркостей пикселей изображения имеет ЗАП-граф, показанный на рис. 14. Важно отметить наличие цикла из операторов $R_{вых}$, P и $W_{вых}$. Для работы в темпе поступления видеоданных необходимо выполнение условия не превышения времени работы

этих операторов периода поступления очередной яркости: $T_{R_{ВЫХ}} + T_P + T_{W_{ВЫХ}} < T_{p_{ix}}$. По условиям задачи $T_{p_{ix}} \approx 20,8 nS$. Минимально возможная задержка чтения составляет 2 такта работы памяти, записи – 1 такт, и 1 такт будет потрачен на выполнение оператора **P**. Следовательно, 1 такт примерно равен 5,2 nS, а частота работы памяти и вычислителя должна быть не менее 192 МГц. Что удовлетворяет параметрам АЛУ и внутренней памяти предположительных ПЛИС и ЦСП. При этом объём необходимой внутренней памяти – $256 * 16 = 4096$ бит. Вычисление гистограммы, непосредственно используя видеоканальные данные, не требует использование операторов $A_{ВХ}$, $R_{ВХ}$ и $A_{ВЫХ}$.

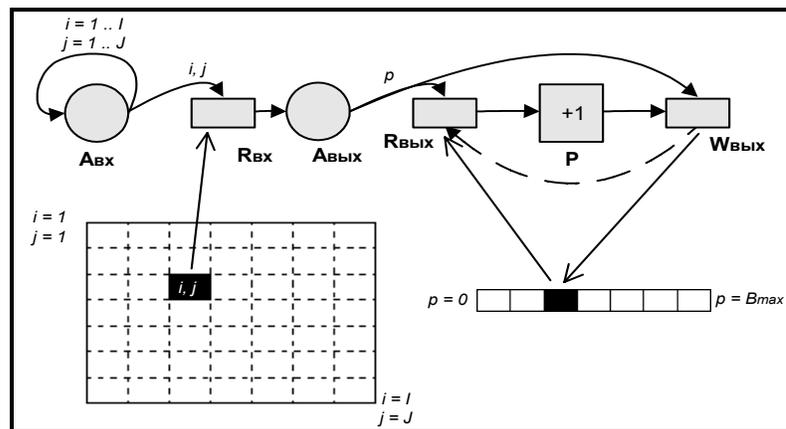


Рис. 14. ЗАП-граф алгоритма вычисления гистограммы распределения яркостей пикселей изображения.

Характеристики и цена ПЛИС фирмы Altera семейства Cyclone. Табл. 3.

	EP1C3	EP1C4	EP1C6	EP1C12
Количество ЛЭ	2910	4000	5680	12060
Частота работы ЛЭ (градация скорости микросхемы – С8), МГц	275	275	275	275
Общее количество Кбит блоков памяти ОЗУ	60	78	92	239
Частота работы блоков памяти (градация скорости микросхемы – С8), МГц	197.01	197.01	197.01	197.01
Макс. возможное количество 8-битных сумматоров	363	500	710	1507
Макс. возможное количество миллионов 8-битных сложений в секунду, MACs	99 825	137 500	195 250	414 425
Цена ¹ при градации скорости микросхемы – С8	264 руб	474 руб	533 руб	965 руб

Микросхемы ПЛИС выбирались из семейства Cyclone фирмы Altera, как наиболее новых и дешёвых микросхем на момент разработки устройства. Изменение параметра разбиения k охватывало ПЛИС этого семейства в диапазоне EP1C3, EP1C4, EP1C6 и EP1C12 (см. табл. 3). Со стороны ЦСП были представлены процессоры семейства BlackFin фирмы Analog Devices.

¹ По данным сайта www.digikey.com на январь 2008 года

Выбор был сделан в пользу EP1С6 и VF-533. Полученное разбиение алгоритмов показано на рис. 12. Архитектура выбрана по типу изображённой на рис. 11-б, так как большая часть обработки сосредоточена на видеоканальных данных, и трафик, дополнительно создаваемый информацией от ПЛИС, составляет 39 Мбайт/сек, что примерно соответствует 15% максимально возможного трафика через шину ЦСП-МВП. В качестве МВП использован SDRAM 16MB с 16-ти разрядной шиной, работающей на частоте 133 МГц.

Разработанная по этому методу рефлексная видеокамера и её электронная часть показаны на рис. 15 и 16.

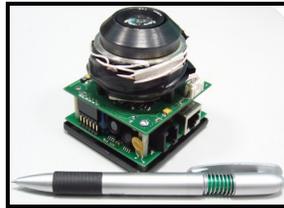


Рис. 15. Видеокамера, обеспечивающая программную реализацию на ПЛИС и ЦСП алгоритмов первичной обработки видеоданных.

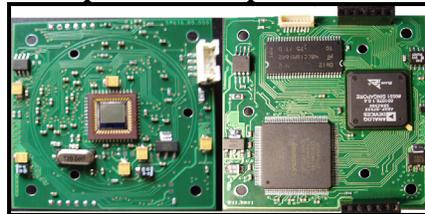


Рис. 16. Реализация 2-го варианта архитектуры соединения ПЛИС, ЦСП и МВП.

В Приложении приводятся список сокращений и тексты упомянутых патентов.

Основные результаты диссертационной работы

- Проведён анализ алгоритмов обработки изображения нижнего и среднего уровней. Предложен способ графико-параметрического описания алгоритмов преобразований цифровых изображений, позволяющий определить возможность их параллельно-конвейерного исполнения с учётом особенностей требуемых обращений к памяти видеоданных.
- Для детекторов простых элементов изображения предложены и построены их параметризованные схемы потоков данных, определяющие метод выбора их параллельно-последовательной программной реализации и наилучшей архитектуры нужных аппаратных частей.
- Построено формальное описание процедуры разложения алгоритма на параллельно-последовательные составляющие и выбора архитектуры средств обработки видеоданных по критерию цены при заданном быстродействии. Построено три архитектурных решения, обеспечивающих программно-аппаратную реализацию алгоритмов нижнего и среднего уровней обработки изображения.

- Предложены программно-аппаратные архитектуры рефлексного видеосенсора, обеспечивающие встроенную возможность программной реализации алгоритмов первичной обработки изображений в непосредственной близости от приёмного видеодатчика, что резко удешевляет канал передачи видеоданных на удалённый компьютер.

Публикации по теме диссертации

1. Краснобаев А.А. Выделение трёхмерных объектов на сцене методом их моделирования. Сборник трудов конференции Мобильные роботы 2003. –М., МГУ, 2003. С. 309-317.
2. Краснобаев А.А. Алгоритм локализации линейного штрихового кода в изображении сцены. Сборник трудов молодых учёных, аспирантов и студентов "Информатика и системы управления в XXI веке". –М., МГТУ, 2004. С. 228-234.
3. Краснобаев А.А. Алгоритмы распознавания штриховых кодов, Препринт Института прикладной математики им. М.В. Келдыша РАН, 2004, №84, 29 с.
4. Краснобаев А.А. Алгоритмы работы системы технического зрения для распознавания штриховых кодов. Сборник трудов конференции "Мобильные роботы 2005". –М., МГУ, 2005. С. 22-32.
5. Краснобаев А.А. Обзор алгоритмов детектирования простых элементов изображения и анализ возможности их аппаратной реализации, Препринт Института прикладной математики им. М.В. Келдыша РАН, 2005, №114, 20 с.
6. Краснобаев А.А., Платонов А.К. Классификация алгоритмов детектирования простых элементов изображения и анализ возможности их аппаратной реализации. 9-ая Всероссийская научн.-практич. конференция “Экстремальная робототехника”, Том 5, СПб.: НПО Спец. материалов, 2006г.
7. Краснобаев А.А., Платонов А.К. Метод параллельно-последовательного детектирования элементов изображения в системах технического зрения // Теория и системы управления, 2007, №4, С. 96-109.
8. Видеосканер штриховых кодов и расчётно-кассовый узел с его использованием : пат. 65267 Рос. Федерация. №2007111328/22,. ; заявл. 28.03.07 ; опубл. 27.07.07, Бюл. № 21.
9. Способ нахождения штриховых кодов в кадре видеоизображения : пат. 2314564 Рос. Федерация. № 2006123894; заявл. 05.07.06 ; опуб. 10.01.08, Бюл. № 1