На правахрукописи

ЛЕВИН Илья Израилевич

МЕТОДЫ И ПРОГРАММНО-АППАРАТНЫЕ СРЕДСТВА ПАРАЛЛЕЛЬНЫХ СТРУКТУРНО-ПРОЦЕДУРНЫХ ВЫЧИСЛЕНИЙ

Специальности: 05.13.11 — "Математическое и программное обеспечение вычислительных машин, комплексов и компьютерных сетей"

05.13.15 - "Вычислительные машины и системы"

АВТОРЕФЕРАТ
диссертации на соискание ученой степени доктора технических наук

Таганрог-2004

Работа выполнена на кафедре "Интеллектуальные многопроцессорные системы" (ИМС) ТРТУ и Южного научного центра РАН, а также в Научно-исследовательском институте многопроцессорных вычислительных систем (НИИ МВС) Государственного образовательного учреждения высшего профессионального образования "Таганрогский государственный радиотехнический университет" (ТРТУ).

НАУЧНЫЕ КОНСУЛЬТАНТЫ:	**академик РАН, доктор технических наук, профессор Каляев Анатолий Васильевич,**
	член-корреспондент РАН, доктор технических наук, профессор Каляев Игорь Анатольевич
ОФИЦИАЛЬНЫЕ ОППОНЕНТЫ:	**член-корреспондент РАН, доктор технических наук, профессор Хорошевский Виктор Гаврилович**
	доктор технических наук, профессор Кравченко Павел Павлович
	доктор технических наук, старший научный сотрудник Аграновский Александр Владимирович
ВЕДУЩАЯ ОРГАНИЗАЦИЯ:	**Институт проблем информатики РАН (ИПИ РАН), г. Москва**

Защита состоится "8" октября 2004 г. в 14^{00}
на заседании диссертационного совета Д 212.259.05 в Таганрогском государственном радиотехническом университете по адресу: 347928, г. Таганрог, ул. Чехова, 2, корп. И, комн. 347.

С диссертацией можно ознакомиться в библиотеке ТРТУ.

Автореферат разослан " 5 " августа 2004 г.

Просим Вас прислать отзыв, заверенный печатью учреждения по адресу: 347928, г. Таганрог, Ростовская область, ГСП-17А, пер. Некрасовский, 44, Таганрогский государственный радиотехнический университет Ученому секретарю диссертационного совета Д 212.259.05 Кухаренко А.П.

Ученый секретарь диссертационного совета кандидат технических наук, доцент **А.П. Кухаренко**

ОБЩАЯ ХАРАКТЕРИСТИКА РАБОТЫ

Актуальность темы. Суперкомпьютеры предназначены для решения сложнейших проблем современности, требующих обработки гигантских объемов информации в короткие промежутки времени. Создание и использование суперкомпьютеров относится к факторам стратегического развития науки и техники и входит в первую десятку приоритетных технологий развитых стран. Без суперкомпьютеров невозможно моделировать экономические и социальные системы, прогнозировать экологические процессы и природные геофизические явления, разрабатывать важнейшие техногенные объекты, решать проблемы космонавтики, астрофизики, медицины. Развитие микроэлектроники и создание семейств высокопроизводительных микропроцессоров привело к тому, что доминирующим направлением построения суперкомпьютеров в настоящее время являются многопроцессорные вычислительные системы (МВС) с массовым параллелизмом, содержащие тысячи параллельно функционирующих микропроцессоров. О темпах развития вычислительной техники говорит тот факт, что два десятилетия назад наиболее мощные суперкомпьютеры имели производительность в несколько Гфлопс; в настоящее время эксплуатируются суперкомпьютеры с производительностью более триллиона операций в секунду, наиболее мощные суперкомпьютеры имеют производительность, превышающую 50 Тфлопс, и разрабатываются системы с производительностью от 100 до 1000 Тфлопс.

В то же время при решении сложных научно-технических задач существующие МВС с массовым параллелизмом обеспечивают достаточно низкую реальную производительность, зачастую не превышающую **10-15** % от пиковой производительности системы, вследствие того, что организация параллельных вычислений и процедуры межпроцессорных обменов требуют больше времени, чем непосредственно вычисления. Это объясняется тем, что практически все существующие методы организации параллельных вычислений в МВС с массовым параллелизмом ориентированы на решение проблем адаптации структуры последовательного алгоритма к архитектуре МВС. До сих пор эти проблемы не решены, и это не позволяет создать масштабируемые параллельные программы для МВС с массовым параллелизмом, которые могли бы быть выполнены с одинаковой эффективностью на произвольной конфигурации вычислительной системы.

Как правило, при создании параллельных программ используются эвристические методы поиска локально-оптимальных вариантов построения множества взаимосвязанных последовательных процессов, и эффективные параллельные программы удается создать лишь для определенного варианта распараллеливания. В связи с этим программирование на МВС с массовым параллелизмом до сих пор является сложным и трудоемким процессом, намного превосходящим по сложности процесс программирования на однопроцессорных машинах. Это обуславливает чрезвычайно высокую стоимость прикладного и системного программного обеспечения МВС.

От эффективности математических методов и программно-аппаратных средств организации параллельных вычислений в значительной (пенсии зависит эффективность применения многопроцессорных систем для решения научно-технических задач.

Поэтому актуальной является решаемая в диссертации научная проблема организации эффективных параллельных вычислений в многопроцессорных вычислительных системах с массовым параллелизмом, обеспечивающих их реальную производительность близкую к пиковой на широком классе задач. Решение этой проблемы обеспечит увеличение рентабельности высокопроизводительных вычислительных комплексов за счет более рационального использования оборудования многопроцессорных систем и сокращения времени, необходимого для организации параллельных вычислений, что, в свою очередь, позволит существенно сократить сроки и стоимость проведения фундаментальных исследований, а также решения важнейших прикладных задач народнохозяйственного значения и повышения обороноспособности страны.

Решением данной проблемы занимались такие видные ученые как Э. Дейкстра, Ч. Хоар, Р. Хокни, В. Хейдлер, Д. Чамберлсн, а также российские ученые: В.С. Бурцев, В.Е. Котов, В.К.Левин, Д.А. Поспелов, В.Г. Хорошевский, В.В.Воеводин, В.А. Вальковский и многие другие. Однако их исследования были ориентированы на разработку методов и аппаратно-программных средств параллельных вычислений для традиционной (жесткой) архитектуры МВС, что обеспечивает высокую реальную производительность только для некоторых классов задач.

Альтернативой существующим методам организации параллельных вычислений является организация вычислений для многопроцессорных систем с программируемой архитектурой, концепция которых была разработана академиком РАН А.В. Каляевым.

МВС с программируемой архитектурой (ПА) представляет собой множество элементарных процессоров (ЭП) и блоков (сегментов) распределенной памяти (РП), которые могут с помощью программно-аппаратных средств системы и пространственной коммутационной структуры объединяться в проблемно-ориентированные вычислители. Это позволяет не только подбирать структуру алгоритма к архитектуре системы, но и, наоборот, адаптировать структуру многопроцессорной системы к параллельному алгоритму. Для многопроцессорной системы с программируемой архитектурой используется структурно-процедурная организация параллельных вычислений, которая обеспечивает максимальную скорость обработки данных, сопоставимую со скоростью специализированных вычислителей. Однако реализация этих принципов требует разработки новых математических методов организации вычислительных процессов и создания аппаратно-программных средств для их поддержки.

Целью работы является разработка и создание формальных математических методов синтеза структурно-процедурных параллельных программ, эффективно реализуемых на различных конфигурациях многопроцессорных систем с программируемой архитектурой, и программно-аппаратных средств, обеспечивающих реальную производительность МВС ПА при решении задач различных классов, близкую к пиковой, а также линейный рост производительности при увеличении ресурсов системы.

В соответствии с поставленной целью определены задачи диссертации:

1) Провести анализ методов и моделей параллельных вычислений и на их основе разработать основные принципы эффективных структурно-процедурных вычислений.

2) Разработать формальные математические методы преобразования алгоритмов задач различных классов в структурно-процедурную форму, обеспечивающие решение за-

дач с реальной производительностью, близкой к пиковой производительности системы, для различных степеней распараллеливания.

3) Разработать средства описания структурно-процедурных вычислений на основе неявного представления параллелизма, обеспечивающие естественное и однозначное отображение параллельного алгоритма на архитектуру многопроцессорной вычислительной системы, а также эффективную реализацию масштабируемых параллельных программ.

4) Разработать технологию создания масштабируемых структурно-процедурных программ, обеспечивающую эффективную реализацию параллельных программ на различных конфигурациях МВС ПА, и рост производительности, близкий к линейному, при увеличении ресурса системы, выделенного для решения задачи.

5) Разработать аппаратно-программные средства, обеспечивающие реализацию структурно-процедурных вычислений на уровне элементной базы.

6) Разработать аппаратные средства для эффективной реализации структурно-процедурных вычислений на основе модульно-наращиваемых многопроцессорных систем.

Методы исследования. При решении поставленных задач использовались элементы теории вычислительных систем, элементы теории графов, теории множеств, а также реляционное исчисление. Теоретические исследования подтверждены вычислительными экспериментами на имитационной модели МВС со структурно-процедурной организацией вычислений, а также реализацией программных средств на созданных модульно-наращиваемых МВС ПА в интересах ряда предприятий и организаций.

Научная новизна диссертации заключается в том, что в ней впервые:

1)На основе формального определения структурно-процедурной организации вычислений разработаны методы приведения регулярных графов алгоритмов в структурно-процедурную форму, которые обеспечивают синтез семейства параллельных алгоритмов с различной степенью параллелизма и эффективностью реализации на МВС ПА, сравнимой с эффективностью специализированных вычислителей.

2) Разработаны методы приведения произвольного информационного графа алгоритма в функционально-регулярную форму и синтеза на его основе структурно-процедурного алгоритма, позволяющие в несколько раз повысить эффективность реализации на МВС ПА параллельных программ решения функционально-нерегулярных задач по сравнению с традиционными методами организации параллельных вычислений.

3)Для структурно-процедурной организации вычислений разработан язык программирования высокого уровня, ориентированный на неявное представление параллелизма с помощью объявления типов массивов переменных и индексации их элементов и, как следствие, позволяющий получать различные варианты распараллеливания задачи при модернизации структуры данных.

4) В языке программирования высокого уровня предложено следующее ограничение правила единственной подстановки, широко используемой в языках потоков данных для представления неявного параллелизма: каждая переменная в программе получает значение один раз в теле описания вычислительной структуры кадра, что обеспечивает детерминизм реализации параллельной программы и позволяет достигнуть взаимнооб-

позначного соответствия описания вычислительной структуры кадра информационному графу решения фрагмента задачи.

5) Разработаны алгоритмы трансляции с языка программирования высокого уровня, учитывающие специфику организации параллельных вычислений и методы синтеза бесконфликтных процедур обращения к распределенной памяти.

6) Разработаны принципы построения технологии индуктивных структурно-процедурных программ, представленных в виде графа минимальной конфигурации, и правил его наращивания при увеличении аппаратного ресурса системы.

7) Разработана система операций модификации кадровых структур при увеличении аппаратного ресурса, выделенного для решения задачи, и степени распараллеливания. Применение системы операций модификации кадровых структур обеспечивает близкий к линейному рост производительности при увеличении числа базовых модулей МВС, выделенных для решения задачи.

8) На основе анализа различных вариантов построения структуры распределенной памяти показано, что для многопроцессорных систем наиболее рациональной является архитектура распределяемой памяти.

9) Показано, что для структурно-процедурных вычислений возможно уже на этапе трансляции задачи, в силу детерминизма вычислительного процесса, определить множество коммутационных структур, которые необходимо последовательно реализовывать в пространственной коммутационной системе. Такое свойство позволит существенно уменьшить аппаратные затраты на построение коммутационной структуры.

Достоверность и обоснованность полученных в работе результатов подтверждается полнотой и корректностью исходных посылок, теоретическим обоснованием, непротиворечивостью математических выкладок. Теоретические исследования подтверждены вычислительными экспериментами на имитационной модели МВС ПА, а также реализацией компонентов математического обеспечения на базовом модуле МВС со структурно-процедурной организацией вычислений (МВС со СПОВ).

Научная и практическая ценность работы. Практическое использование научных результатов позволило:

1) Повысить реальную производительность многопроцессорных систем при решении ряда задач различных проблемных областей. На основе разработанных в диссертации методов создан ряд параллельных алгоритмов и параллельных программ, повышающих эффективность распараллеливания в 2-5 раз по сравнению с известными решениями.

2) Разработать и создать язык программирования высокого уровня с неявным описанием параллелизма, позволяющий сократить время создания параллельных масштабируемых программ, эффективно реализуемых на различных конфигурациях многопроцессорной системы.

3) Разработать алгоритмы планировщика индуктивных заданий, распределяющего свободный ресурс системы с минимизацией времени прохождения потока заданий. Реализация планировщика заданий показывает его высокую эффективность, в том числе, и для ограниченных коммутирующих структур, для которых преимущество технологии индуктивных структурно-процедурных программ более заметно и позволяет повысить

коэффициент использования оборудования на 40% и сократить время прохождения задач на 20-30%.

4) Разработать алгоритмы системы посттрансляции, осуществляющие модернизацию загрузочного модуля параллельной программы на более высокой степени распараллеливания и перекоммутацию пространственных связей между базовыми модулями, выделенными планировщиком заданий (ПЗ) для решения задачи.

5) Разработать и создать элементную базу для построения МВС со СПОВ, с минимальной номенклатурой и четко обозначенными функциями. Технические решения защищены четырьмя патентами РФ.

6) Разработать и создать одноплатные базовые модули с иерархической коммутационной и ортогональной коммутационной системами, аппаратно-программные средства которых позволили обеспечить повышение реальной производительности по сравнению с традиционными архитектурами на порядок, а удельной производительности - на два порядка. Набольший рост производительности достигается для наиболее сложных - сильносвязанных задач, для которых, как правило, ранее не было известно эффективного параллельного решения.

7) Разработать и создать модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений, используемые для решения широкого класса задач в различных организация и ведомствах.

Предложенные в диссертации новые решения строго аргументированы и критически оценены по сравнению с другими известными результатами. В диссертации приведены рекомендации по использованию полученных научных выводов. Полученные научные результаты практически используются на различных предприятиях и в организациях РФ, что подтверждается соответствующими актами о реализации. В диссертации решена крупная научная проблема, которая является актуальной и имеет важное научно-хозяйственное значение. Внедрение полученных в диссертации результатов вносит значительный вклад в развитие экономики страны и повышение ее обороноспособности.

Реализация и внедрение результатов работы. Теоретические и практические результаты диссертационной работы использованы при выполнении госбюджетных и хоздоговорных НИР в НИИ МВС ТРТУ, непосредственным участником которой являлся автор диссертации. Результаты диссертации внедрены в ОАО "Научно-исследовательский центр электронной вычислительной техники" (НИЦЭВТ), г. Москва; Московском государственном институте электронной техники (технический университет), г. Москва; в/ч 26165, г. Москва; в/ч 25714, г. Курск; Таганрогском государственном радиотехническом университете, г. Таганрог, НИИ многопроцессорных вычислительных систем Таганрогского государственного радиотехнического университета, г. Таганрог; Научно-конструкторском бюро вычислительных систем Таганрогского государственного радиотехнического университета, г. Таганрог.

Апробация работы. Основные результаты работы докладывались и обсуждались на всероссийских и международных научно-технических конференциях: International Conference "Parallel Computing Technologies (PaCT)", Novosybirsk, 1991; Yaroslavl, 1997; Всероссийская научно-техническая конференция с международным участием "Теория цепей и сигналов", Новочеркасск, 1996; VI международный семинар "Распределенная

обработка информации", Новосибирск, 1998; IV, V Всероссийские научно-технические конференции '"Нейрокомпьютеры и их применение", Москва, 1998 - 1999; Международные конференции "Интеллектуальные многопроцессорные системы (ИМС'99)", Таганрог, 1999, пос. Дивноморское, 2001,2003; Международная конференция "Искусственный интеллект",-Кацивели, 2000, 2002; Международная научно-техническая конференция "СуперЭВМ и многопроцессорные вычислительные системы", Таганрог, 2002; Всероссийская научная конференция "Высокопроизводительные вычисления и их приложения", Черноголовка, 2000; Третья международная научно-техническая конференция "Электроника и информатика - XXI век", Зеленоград-Москва, МИЭТ, 2000; Между народная конференция "Параллельные вычисления и задачи управления (PACO'2001)", М: ИПУ РАН им. В.А.Трапезникова, 2001; Конференция "С.А. Лебедев и развитие отечественной вычислительной техники", Москва, ИПИ РАН, 2002; Всероссийская научная конференция "Методы и средства обработки информации", Москва, ф-т вычислительной математики и кибернетики МГУ им. М.В. Ломоносова, 2003.

Основные положения, выносимые на защиту:

1)Доказательство более высокой эффективности (отношение ускорения, достигаемого на МВС по сравнению с однопроцессорной ЭВМ, к числу процессоров) структурно-процедурных вычислений по сравнению с традиционными методами организации параллельных вычислений.

2) Принципы и методы преобразования алгоритмов в структурно-процедурную форму, эффективно реализуемую в МВС ПА для различных степеней распараллеливания, обеспечивающие линейный рост производительности при увеличении ресурсов системы.

3) Язык программирования высокого уровня для описания структурно-процедурных вычислений на основе неявного представления параллелизма за счет объявления типов массивов переменных и индексации их элементов, обеспечивающий естественное и взаимодополняющее отображение параллельного алгоритма в структуру многопроцессорной вычислительной системы.

4) Технология индуктивных программ, обеспечивающая высокую эффективность решения структурно-процедурных программ на различных аппаратных ресурсах и сокращение времени прохождения потока заданий.

5) Программно-аппаратные средства поддержки структурно-процедурных вычислений на уровне элементной базы, позволяющие достигать реальной производительности системы, близкой к пиковой, для широкого класса задач и повышающие удельную производительность системы.

6) Аппаратные средства, обеспечивающие реализацию структурно-процедурных вычислений на основе модульно-наращиваемых многопроцессорных систем для различных вариантов распараллеливания задачи различных проблемных областей.

Личный вклад автора. Все научные результаты, полученные при решении крупной научной проблемы создания методов структурно-процедурного программирования и программно-аппаратных средств, обеспечиваюших реальную производительность МВС при решении задач различных классов, близкую к пиковой, а также линейный рост производительности при увеличении ресурсов системы, получены автором лично.

Публикации. По теме диссертации опубликованы 102 печатные работы, в том числе, 1 монография, 30 статей в центральной печати, из них 10 - в изданиях, входящих в "Перечень ведущих научных журналов и изданий, выпускаемых в Российской Федерации" ВАК, получено 4 патента на изобретение.

Структура и объем диссертации. Диссертация состоит из введения, шести разделов, заключения и списка литературы из 232 наименований. Она содержит 363 страницы текста, 11 таблиц, 156 рисунков, 62 страницы приложений.

КРАТКОЕ СОДЕРЖАНИЕ РАБОТЫ

Во введении обосновывается актуальность проблемы, проанализирован научный уровень методов организации параллельных вычислений, сформулированы цель и задачи исследования, дан краткий обзор содержания диссертации, перечислены полученные научные результаты и приведены сведения о практической ценности работы.

В первой **главе** проведен аналитический обзор методов организации параллельных вычислений. Показаны нерешенные проблемы в области создания параллельных алгоритмов. Разработаны основные принципы преобразования задач различных классов в структурно-процедурную форму.

Большинство существующих МВС использует мультипроцедурную организацию вычислений, когда в каждом процессоре реализуется последовательная программа, а для обмена данными между процессорами в системе организуются специальные процедуры. Существующие системы распараллеливания для таких МВС ориентированы на решение определенного класса задач или вносят большие накладные расходы, в результате параллельные программы решения сложных задач имеют низкую реальную производительность.

В последнее время появились реконфигурируемые системы, позволяющие синтезировать вычислительные структуры, адекватные решаемой задачи. Однако такие системы перестраиваются с одной структуры на другую за очень большое время, что ограничивает их применение для различных классов задач, таких как система HC Starbridge, или применяются только для систолических вычислений и сводимых к ним задач, например, система Xputer.

Альтернативой существующим МВС с массовым параллелизмом являются многопроцессорные вычислительные системы с программируемой архитектурой, концепция которых разработана в Научно-исследовательском институте многопроцессорных систем ТРТУ под руководством академика РАН А.В. Каляева. МВС с программируемой архитектурой обеспечивает пользователю возможность формировать в универсальной системе виртуальные специализированные вычислители, адекватные структуре решаемой задачи, что выводит на передний план "внутренний" параллелизм задачи. В тоже время формальные методы синтеза структурно-процедурных решений для задач различных классов не созданы.

Для более детального анализа целесообразно рассмотреть общую структуру задачи - информационный граф алгоритма. В информационном графе вершины соответствуют

исходным данным, операциям над данными и результатам вычислений. Дуги информационного графа соответствуют информационным связям алгоритма.

Если множество вершин V информационного графа G может быть представлено в виде объединения некоторых непересекающихся подмножеств

$$V = \bigcup_{i=1}^{k} V_i \qquad (1)$$

таких, что если вершина $u \in V_i$, а вершина $v \in V_j$, $i<j$,, и существует дуга $\langle u, v \rangle$, то такое разбиение называется параллельной формой алгоритма. При описании задачи информационным графом все ее циклические участки итерированы столько раз, сколько должен повторяться этот участок. Если количество итераций неизвестно, то можно представить некоторую итерированную структуру с неизвестным числом повторений. При этом предполагается, что операции алгоритма разбиты на группы, упорядоченные так, что каждая операция любой группы зависит либо от начальных данных алгоритма, либо от результатов выполнения операций, находящихся в предыдущих группах.

Для мультипроцедурных методов организации параллельных вычислений широкое распространение получили ярусно-параллельные формы (ЯПФ). Представление графа алгоритма в ярусно-параллельной форме заключается в упорядочении вершин графа по подграфам V_i разбиения (1). Упорядоченные подмножества называются ярусами. Множество нулевого яруса V_0 определяет совокупность входных вершин X. В общем случае, вершина d принадлежит i-му ярусу тогда и только тогда, когда существует, по крайней мере, хотя бы одна дуга $A=\langle c, d \rangle$, где c — вершина, принадлежащая ярусу с номером i-1. Данное утверждение можно записать в реляционном исчислении следующим образом:

$$V_i = \{c | A(\langle c, d \rangle) \wedge V_{i-1}(d)\}. \qquad (2)$$

При соблюдении только условия (2) формируется ярусно-конвейерный граф, в котором разрешены обратные связи. Достоинством параллельно-конвейерной формы представления задач является ее большая компактность по сравнению с ярусно-параллельной формой. Однако если необходимо перейти к другому варианту распараллеливания задачи, требуется разрывать рекурсивные связи, что представляет сложную проблему при реализации эффективных параллельных программ.

Поэтому для преобразования задачи в параллельную форму, эффективно реализуемую для различных вариантов распараллеливания, необходимо использовать альтернативные способы представления параллельного алгоритма. В общем, информационный граф можно представить в виде:

$$G=\langle (X, A_X), (Q, A_Q), (Y, A_Y) \rangle, \qquad (3)$$

где X- множество входных вершин графа,

Y- множество выходных вершин,

Q - множество операционных вершин.

Объединение множеств входных дуг A_X, операционных дуг A_Q и выходных дуг A_Y описывает информационные связи в задаче. Для любой дуги $\langle u,v \rangle$, принадлежащей к A_X, вершина $u \in X$, а вершина $v \in Q$. Для любой дуги $\langle u,v \rangle$, принадлежащей к A_Q, вершина $u \in Q$ и вершина $v \in Q$. Для любой дуги $\langle u,v \rangle$, принадлежащей к A_Y, вершина $u \in Q$, вершина $v \in Y$.

Если число элементарных процессоров в системе не меньше числа операционных вершин, и все дуги информационного графа задачи (в том числе, обуславливающие связь между информационными и операционными вершинами) можно реализовать в пространственной коммутационной системе, то информационный граф задачи тривиальным образом реализуется в многопроцессорной системе. Операционная вершина графа будет соответствовать процессору, настроенному на выполнение определенной арифметико-логической операции. Входная (выходная) информационная вершина должна соответствовать сегменту операционной памяти, в которой расположены исходные данные (результаты вычислений). Такую организацию вычислений будем называть **структурной организацией** вычислений. На практике обеспечить структурную организацию вычислений не представляется возможным, так как для се реализации требуется чрезвычайно большое количество процессоров. В этой связи возникает проблема, решаемая в диссертации, заключающаяся в сегментации информационного графа на пересекающиеся подграфы, которые структурно реализуются в системе, при этом обеспечивается минимальное время решения задачи.

Задача, описываемая в виде информационного графа *G,* может быть представлена в виде множества крупных функционально законченных вычислительных фрагментов P_j:

$$G=\bigcup_{i=1}^{M} P_i, \qquad (4)$$

таких, что если вершина $a \in P_i$ и вершина $b \in P_j$ и существует хотя бы одна дуга $\langle a, b\rangle$, то $i \le j$. Будем называть дуги, соединяющие подграфы, - внешними, а дуги, принадлежащие подграфу P_i, - внутренними. Данное разбиение коренным образом отличается от ранее рассмотренных разбиений (1, 2) возможностью связей внутри фрагментов.

Подграфы информационного графа задачи, последовательно реализуемые на многопроцессорной системе, принято называть кадрами, а механизм последовательного обхода информационного графа задачи подграфами - **структурно-процедурной программой.** Кадр представляет собой структурно-реализованный подграф задачи, через который следует поток данных. Подобная организация вычислений обеспечивает максимальную скорость обработки данных, сопоставимую со скоростью специализированных вычислителей. Вычисления выполняются по принципу управления потоком данных и не требуют синхронизации. Настройка на кадры производится по единой управляющей программе, что обеспечивает фон-неймановский детерминизм. Кадр часто описывается как объект, определенный в виде тройки $<Q,I,O>$, где Q - функциональный граф кадра; I, O - множество информационных потоков. На практике зачастую понимается, что информационные потоки, входящие в кадр, однозначно определены и соответствуют появлению входных данных в заданном временном масштабе. Однако, как показали проведенные исследования, такой взгляд на информационные потоки кадра является частным случаем. На рис.1, представлено графическое изображение кадра. Несложно заметить, что вышеприведенное определение кадра соответствует кортежу элементарных кадров $\langle K_1, K_2, \ldots, K_l \rangle$, где l - размер информационных потоков на входе (выходе) кадра.

Элементарным кадром будем называть кадр, в котором на каждый вход функционального графа кадра подается только один операнд и с каждого выхода снимается толь-

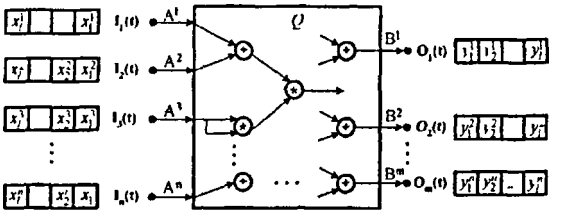


Рис 1. Графическое изображение кадра.

ко один результат. Элементарный кадр определяется тройкой $\langle Q, X_i, Y_i \rangle$, где X_i и Y_i - это множества входных и выходных вершин, соответствующие информационному потоку в определенный момент времени. Множества входных и выходных вершин элементарного кадра K_i можно определить следующим образом: $X_i = I(t_i), Y_i = O(t_i)$. Следует отметить, что функциональный граф всех элементарных кадров кортежа - изоморфный. На рис.2 представлено графическое изображение кортежа изоморфных кадров. Доказано, что если имеется кортеж элементарных кадров, каждый элемент которого обладает изоморфным функциональным графом, то такой кортеж можно представить в виде исполняемого кадра.

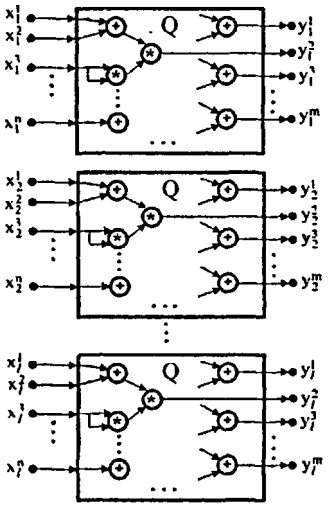


Рис.2. Кортеж изоморфных кадров.

Обращаясь к кортежам по индексу, можно получить требуемый элемент и, следовательно, информационные потоки можно определить как функции обращения к информационным массивам. Такой подход позволяет значительно расширить применение структурных и структурно-процедурных методов организации параллельных вычислений для задач различных проблемных областей и увеличить эффективность распараллеливания, а, как следствие, и существенно сократить время и стоимость решения задачи на МВС. В общем случае, информационные потоки, поступающие на входы кадровых структур, определяются не временными соотношениями задачи, а информационными зависимостями алгоритма, в которых временные зависимости являются только частными случаями.

Под информационной зависимостью двух подграфов G_1 и G_2 графа задачи понимается наличие пути в информационном графе задачи, соединяющем вершину $a \in G_1$ и вершину $b \in G_2$. Следовательно, два подграфа будут информационно независимыми, если в информационном графе задачи не существует ни одного пути, соединяющего любые вершины, принадлежащие G_1 и G_2. Множество информационно-независимых изоморфных подграфов представлено в виде кадра. Доказано, что правило упорядочивания элементарных кадров в кортеж однозначно определяет порядок появления входных и выходных информационных вершин. Дополнительным условием, при котором кортеж подграфов может быть представлен в виде кадра, является наличие рекуррентных правил отображения входных и выходных дуг подграфов g_i $(i=1,2,...,l-1)$: $F_R : A_{X_i} \to A_{X_{i+1}}$ и $F_W : A_{Y_i} \to A_{Y_{i+1}}$.

Изоморфные графы, которые преобразуются в кадр, должны быть реализуемыми, т.е. к моменту инициализации кадра все информационные элементы, соответствующие входным вершинам всех информационных графов, должны быть получены и будут находиться в каналах распределенной памяти.

Полученные выражения расширены на наиболее общий случай, когда объединяются информационно зависимые подграфы, но дуги зависимости соединяют непосредственно информационные подграфы без дополнительных операционных вершин.

Если существуют два изоморфных подграфа G_1 и G_2, не являющиеся информационно независимыми, и существует единственная дуга, начальная вершина которой является $a_1 \in G_1$, а конечной вершиной является $b_2 \in G_2$ и не существует другого пути информационной зависимости между подграфами G_1 и G_2, то подграфы G_1 и G_2 можно упорядочить в кадр, при этом имеется единственное правило упорядочивания $<G_1,G_2>$, определенное дугой $<a_1,b_2>$. В этом случае производится дополнительное преобразование информационного графа. Формируется дуга, соединяющая вершину a и вершину b. На эту дугу вводится начальный элемент, обеспечивающий реализацию обратной связи. Начальное значение определяется начальной вершиной дуги, входящей в граф G_1. Вершина b_1 соответствует вершине b_2 (графы G_1 и G_2 изоморфны). Дуги $<x_{h_1},b_1>$ и $<a_1,b_2>$ удаляются из графа кадра. Пример преобразования представлен на рис.3, где а) исходный подграф задачи; б) граф кадра.

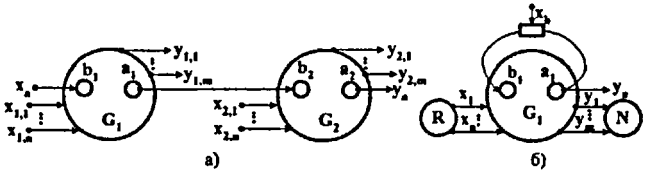


Рис.3. Преобразование связанных информационных графов в кадр.

Данные преобразования расширены на произвольное количество подграфов. Если имеется множество изоморфных подграфов $G_1,G_2,...G_n$, являющихся информационно-зависимыми только по смежным дугам α_1, соединяющим подграфы G_i и G_{i+1} (начальной вершиной дуг смежности является $a_i \in G_i$, конечной вершиной - $b_i \in G_{i+1}$ и вершина a_1 является соединенной с входной вершиной X_0), то подграфы $G_1,G_2,...G_n$ могут быть преобразованы в кадр, причем, правило упорядочивания подграфов определяется путем, связующим подграфы $<G_1,G_2,...G_n>$. Пример такого графа показан на рис.4.

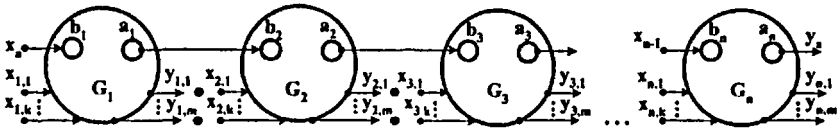


Рис.4. Информационный граф, содержащий изоморфные взаимосвязанные подграфы.

При объединении графа дуга α_i исключается и формируется обратная связь $<a,b>$, инициализируемая начальной вершиной, смежной по дуге α_1 с b_1, т.е. x_0. Структура графа кадра будет такая же, как на рисунке 3б).

Показано, что информационных дуг, соединяющих подграфы, может быть множество (связи могут следовать не только через смежные фрагменты), но все они должны

быть описаны функциональной зависимостью между подграфами. Количество инициализаций (начальных значений по обратным связям) определяется разностью индексов между индексом начальной вершины $a_i \in G_i$ и конечной в е р ш и н о й $b_{i+\Delta i} \in G_{i+\Delta i}$. В более общем случае выделяется функциональная зависимость между начальной вершиной $a_i \in G_i$ и конечной вершиной $b_{f(i)} \in G_{f(i)}$. Следует отметить, что связь может быть не только между одиночными подграфами, но и некоторой совокупностью подграфов (слоев) между собой.

Очевидно, что связь между последовательно реализуемыми кадрами осуществляется только через оперативную память, которая для МВС со структурно-процедурной организацией вычислений является распределенной и многоканальной. Каждой входной дуге a_j кадра ставится в соответствие функция чтения r_j, отображающая моменты времени (такты работы системы) на множество операндов, т.е. обеспечивающая поступление информационных потоков на входы кадра из каналов распределенной памяти. Аналогично, каждой выходной дуге b_j кадра ставится в соответствие функция записи w_j, отображающая моменты времени (такты работы системы) на множество результатов, т.е. обеспечивающая сохранение информационных потоков с выходов кадра в каналах распределенной памяти.

Будем называть информационным кадром следующую тройку: $K = < R, Q, W >$, где R - узел чтения; Q- структурно-реализованный подграф задачи; W- узел записи. Здесь

$$R = \bigcup_{j=1}^{N(A)} < a_j, r_j >, \tag{5}$$

где $N(A)$ - мощность множества входных дуг кадра A.

Аналогично узел записи представляет собой объединение упорядоченных пар:

$$W = \bigcup_{j=1}^{N(B)} < b_j, w_j >, \tag{6}$$

где $N(B_j)$ - мощность множества выходных дуг кадра.

В диссертации определены дополнительные условия разбиения информационного графа задачи на структурно-реализуемые подграфы. Каждый фрагмент должен быть представлен в виде множества подграфов

$$P_i = \bigcup_{j=1}^{N} S_{i,j} \tag{7}$$

таких, что если существует хотя бы одна дуга $\langle a,b \rangle$ и вершина $a \in S_{ki}$, а вершина $b \in S_{kj}$, то $i \le j$. Подграфы S_{ij} будем называть слоями. При этом дуги, соединяющие слои, будем называть межслойными дугами, а дуги, соединяющие операционные вершины одного слоя, будут внутрислойными дугами. Наконец, каждый подграф $S_{i,j}$ можно представить в виде множества подграфов

$$S_{i,j} = \bigcup_{k=1}^{K} g_{i,j,k}, \tag{8}$$

при этом все подграфы $g_{i,j,k}$ должны быть изоморфны друг другу. Подграфы $g_{i,j,k}$ будем называть базовыми подграфами.

Кроме того, выполняются следующие условия. Внутрислойные дуги должны прина-

длежать какому-нибудь базовому подграфу. Если существует дуга $\langle a,b\rangle \in S_{i,j}$, то вершина $a \in g_{i,j\,k}$ и вершина $b \in g_{i,j\,k}$. Как следствие, не существует дуг, соединяющих вершины, принадлежащие базовым подграфам одного слоя. Базовый подграф $g_{i,j\,k}$ есть $S_{i,j}$ тогда и только тогда, когда существует, по крайней мере, одна дуга $\langle a,b\rangle$ такая, что $a \in q_{ij\cdot 1/n}$ и $b \in q_{ijk}$.

Подобное представление графа в виде множества подграфов будем называть секвенцией графа. Секвентированный граф вычислительного процесса решения задачи может быть представлен в виде

$$P_i = \Phi_{i,j}(F_{i,j,k}(g_{i,j,k})), \tag{9}$$

где $F_{i,j\,k}$ — функция, отображающая дуги базового подграфа $g_{i,j\,k}$ на базовый подграф $g_{i,j,k+1}$, $\Phi_{i,j}$ - функция, отображающая множество дуг слоя $S_{i,j}$ на множество дуг слоя $S_{i,j+1}$. Множество функций F является описанием параллельного выполнения базовых подграфов. Множество функций Φ определяет информационную зависимость между слоями.

Если в результате выполнения секвенции в подграфе P_m подграфы $g_{i,n,k} \in S_{i\,n}$ и $g_{i,m,k} \in S_{i,m}$ изоморфны для всех значений k, n и m и существуют обобщающие функции отображения базовых подграфов внутри слоя и функции отображения слоев друг на друга, то такое разбиение графа вычислительного процесса решения задачи G на подграфы P_i будем называть функциональным секвентированием.

При функциональном секвентировании фрагменты P_i являются функционально-регулярными $P_i = \Phi_i(F_i(g_i))$. Информационный граф алгоритма задачи, представленный в этом виде, будет называться функционально-регулярной формой графа вычислительною процесса решения задачи.

Показано, что кажущееся ограничение на возможность представления произвольного графа в функционально-регулярной форме не соответствует действительности. Любой граф вычислительного процесса может быть представлен, по крайней мере, в тривиальной функционально-регулярной форме, когда фрагмент графа содержит единственный слой, а слой содержит единственный базовый подграф. Тогда фрагмент представляется в виде элементарного кадра. Для увеличения эффективности параллельных программ целесообразно осуществлять секвенцию информационного графа в функционально-регулярную форму так, чтобы мощности множества слоев фрагмента и множества базовых подграфов слоя были как можно большими.

При формировании кадра происходит объединение множеств входных (выходных) информационных вершин в функциональный узел чтения (записи), дугам которого поставлен в соответствие кортеж функций. Элемент кортежа функций чтения ставит в соответствие моменту времени t множество (группу) входных информационных вершин x. Формируется операционная структура кадра, которая описывается графом $Q=(V_Q, A_K)$.

Показано, что для эффективной реализации параллельных вычислений целесообразно, чтобы память МВС состояла из множества сегментов (каналов), при эти данные, требующие параллельного обращения, должны быть расположены в разных сегментах памяти. В результате массивы данных являются двумерными, при этом первое измерение определяет номер сегмента (канала), а второе — адрес в ячейке канала. Важной проблемой является синтез единой для всех кадров задачи структуры данных. Чтобы реализовать бесконфликтное параллельное обращение к каналам памяти многопроцессорной

системы, сформулирована система ограничений, позволяющих выбрать ограниченное количество вариантов размещения данных в сегментированной памяти в соответствии с выбранным вариантом распараллеливания задачи и конфигурацией МВС.

Система ограничений включает в себя фундаментальные ограничения, невыполнение которых не позволяет реализовать выбранный вариант распараллеливания в архитектуре вычислительной системы, а также дополнительные ограничения, невыполнение которых лишь уменьшает эффективность реализации задачи.

Фундаментальное ограничение параллельного доступа требует, что если функция чтения (записи) ставит в соответствие моменту времени t два операнда, то они должны быть расположены в разных каналах распределенной памяти. Кроме того, существует фундаментальное ограничение на количества каналов РП и элементарных процессоров.

Одной из основных задач организации эффективных структурно-процедурных вычислений является уменьшение времени решения задачи, что достигается минимизацией числа кадров и максимизацией размеров потока данных, обрабатываемых в кадрах, и минимизацией времени поступления операнда. Дополнительное ограничение регулярности накладывается на размещение элементов в сегментированной памяти многопроцессорной системы таким образом, чтобы программные затраты на реализацию функций чтения и записи были минимизированы.

В соответствии с системой фундаментальных ограничений выбирается рациональный вариант размещения элементов многомерных массивов в распределенной памяти МВС. На основании функций чтения и сформированной структуры данных реализуются рекуррентные выражения для функции адресации и коммутации в каждом кадре.

При работе с двумерными массивами индексы обоих переменных, в общем случае, зависят друг от друга В то же время для каждого канала распределенной памяти необходимо выполнить фиксированную процедуру обращения к данным, расположенным в канале, независимо от коммутации, для чего следует преобразовать функции адресации так, чтобы они зависели только от номера канала и момента времени обращения к данным. Для организации независимых процедур доступа к ячейкам РП (ar, aw) в канале в диссертации предложено осуществить переход от кортежей функций адресации и коммутации к кортежу функций исполнительной адресации чтения FE_R и кортежу функций исполнительной адресации записи FE_W в каналах РП. С-й элемент er_C кортежа FE_R определяется следующим образом:

$$er_c = \bigvee_{k=0}^{'} \left(Dr_{k,c}\right) \cdot t_k = \bigvee_{k=0}^{'} A_c\left(x^J_{k,\alpha(k,c)}\right) \cdot t_k = ar_{\alpha(k,c)}, \qquad (10)$$

где $Dr_{k,c}$ - адрес чтения в канале РП с логическим номером c в момент времени t_k;

$\alpha(k,c)$ - номер элемента вектора коммутации чтения, значение которого в момент времени t_k равно c.

Аналогично, c-й элемент ew_C кортежа функций исполнительной адресации записи FE_W определяется следующим образом:

$$ew_c = \bigvee_{k=0}^{'} \left(Dw_{k,c}\right) \cdot t_k = aw_{\beta(k,c)} \cdot t_k, \qquad (11)$$

где $Dw_{k,c}$ - адрес записи в канале РП, имеющем логический номер c в момент времени t_k;

$\beta(k,c)$ - номер элемента кортежа функций коммутации записи, значение которого в момент времени t_k равно c.

После преобразования информационных подграфов в кадровую форму задача представляется в виде графа кадров, в котором каждой вершине ставится в соответствие кадр. Если между фрагментами задачи P_k и P_m существует информационная зависимость, описываемая множеством дуг $D_{k,m}$, то при преобразовании графа задачи в граф кадров множество дуг $D_{k,m}$ заменяется дугой информационной зависимости (K_k, K_m). Граф кадров может быть, в свою очередь, представлен в параллельной форме.

Структурно-процедурная организация вычислений накладывает на реализацию графа кадра следующее ограничение: в МВС одновременно может быть реализован только один кадр, что обеспечивает фон-неймановский детерминизм программы. При такой организации вычислительного процесса легко преобразовать граф кадров в структурно-процедурную форму. В самом деле, кадр можно рассматривать как некоторый функциональный оператор. Тогда можно выделить повторяемые (итерируемые) участки вычислений и реализовать их в виде циклов.

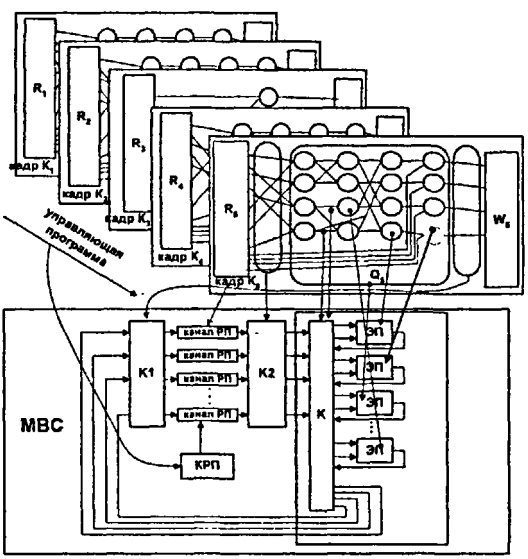


Рис.5. Отображение кадровой структуры на архитектуру МВС ПА.

Появляются альтернативные ветки алгоритма и т.д. В целом процесс преобразования задачи из графа кадра в структурно-процедурную программу соответствует синтезу последовательной программы для однопроцессорной ЭВМ, и здесь можно воспользоваться всем опытом подобного синтеза. Компоненты кадров задачи естественным образом отображаются в структуру МВС ПА (рис.5). Так, внутренние (операционные) дуги базового подграфа отображаются на множество коммутационных связей, реализуемых в коммутаторе K, множество операционных вершин базового фрагмента - во множество элементарных процессоров. Множество входных и выходных луг базового фрагмента реализуется во множестве коммутационных связей коммутаторов K_1 и K_2.

Массивы данных отображаются на множество адресов распределенной памяти.

Управляющая программа смены кадров и процедуры обращения к каналам распределенной памяти осуществляется в контроллере распределенной памяти (КРП).

Во второй главе на основании принципов организации структурно-процедурных вычислений разрабатываются методы преобразования алгоритмов решения задачи в

структурно-процедурную форму. Показано, что функционально-регулярный граф может быть реализован в виде единственного кадра, который представляет собой структурно-реализованный базовый подграф или множество взаимосвязанных базовых подграфов, через который следует поток данных. При этом элемент потока данных соответствует множеству информационных вершин на входах структурно-реализуемого подграфа.

При объединении базовых подграфов в кадр формирование обратных связей происходит не во всех случаях, а только когда ресурса системы достаточно для структурной реализации множества подграфов и время следования операндов одного слоя через структуру меньше времени заполнения конвейера. Определены условия формирования рекурсивных связей и время выполнения кадра.

Время выполнения кадра можно определить по следующей формуле:

$$T_k = t_з(Q) + t_н + \tau \cdot N, \tag{12}$$

где $t_з(Q)$ - время заполнения конвейера графа кадра;

$t_н$ - время настройки на кадр;

N - размер потока данных, следующих через кадр;

τ - время следования операнда.

Время следования операнда можно оценить по формуле:

$$\tau = \max(t_l(Q), t_0), \tag{13}$$

где t_0 - время следования операнда в потоке; t_l - максимальная величина задержки по обратной связи в графе Q.

При большой величине N временем заполнения конвейера и временем настройки на кадр можно пренебречь. Время вычисления кадра можно оценить как

$$T_k \approx t_0 \cdot N. \tag{14}$$

Однако если существует рекурсивная связь, то время выполнения кадра замедляется до величины

$$T_k \approx t_l(Q) \cdot N. \tag{15}$$

При достаточно больших структурных фрагментах это время может многократно превысить минимальное время выполнения кадра. Величину $t_l(Q)$ можно оценить как $m \cdot t_0$, где $m \sim \sqrt{M}$, здесь M - число вершин в графе кадра.

Когда слой фрагмента информационного графа, преобразуемого в кадр, содержит n базовых подграфов, требование рекурсивной связи можно записать как выполнение условия

$$t_0 \geq \frac{T_p}{\lceil n/p \rceil}, \tag{16}$$

где p - число одновременно реализуемых базовых подграфов.

Поэтому целесообразно повысить степень распараллеливания p только до тех пор, пока справедливо неравенство (16). В противном случае время решения задачи существенно увеличится. В диссертации на примере задачи математической физики показано применение методов структурно-процедурной организации вычислительного процесса для функционально-регулярных задач.

Выделено несколько основных типов информационных графов процесса решения задачи, для отображения которых в структурно-процедурную форму требуется выполнение специальных информационно-эквивалентных преобразований. К таким графам сле-

дуст отнести информационные графы (подграфы) решения задачи, которые хоть и имеют детерминированную структуру и могут быть представлены в виде кортежа информационно-зависимых подграфов (слоев), но функциональная зависимость между базовыми подграфами явно не выражена. Второй тип графа представляют собой задачи, у которых количество базовых подграфов, принадлежащих слою, одинаково, но правила отображения базовых подграфов внутри каждого слоя различны, а правило отображения слоев определяется конкретной реализацией данной задачи. Кроме того, существуют задачи, графы вычислительного процесса решения которых содержат различное и заранее неизвестное количество базовых подграфов, принадлежащих слою. При этом мощность слоя, содержащего базовые подграфы, зависит от результатов вычислений на предыдущих слоях графа (подграфа) задачи.

Для нерегулярных задач предложен комплекс преобразований. На первом этапе алгоритм задачи представляется в параллельной форме - в виде графа вычислительного процесса решения задачи. На втором этапе устраняется функциональная избыточность графа и формируется промежуточный функционально-нерегулярный граф. На третьем этапе выполняется специальное преобразование - векторизация информационных вершин, при выполнении которого множество непересекающихся подмножеств информационных вершин, в каждом из которых существует ациклический путь, соединяющий в определенном порядке информационные вершины, или другое правило детерминированного упорядочивания вершин, заменяется специальным объектом — вершиной-массивом. Данная вершина соответствует расположению структуры данных в каналах распределенной памяти МВС. Каждое подмножество информационных вершин, соединенных ациклическим путем в информационном графе или отобранное другим правилом упорядочивания, будет расположено в отдельном канале распределенной памяти. Адреса, по которым будут расположены вершины в канале, взаимно однозначно соответствуют номеру вершины на пути, соединяющему эти вершины в графе, или номеру в кортеже упорядочивания. После выполнения операции векторизации информационные дуги, источниками или приемниками которых являлись векторизированные информационные вершины, исключаются и заменяются дугами, источником или приемником которых является вершина-массив. В векторизированном графе появляются адресные дуги, которые формируются по следующему правилу: началом адресной дуги является информационная адресная вершина с уникальным именем, соответствующим адресу информационной вершины, а концом - метка информационной дуги, выходящей (входящей) из вершины массива. Адрес соответствует номеру канала распределенной памяти, т.е. номеру упорядоченного в кортеж подмножества информационных вершин. Каждая адресная дуга имеет признак t, который соответствует временной метке обращения по адресу ячейки в канале распределенной памяти. На рис.6 представлен граф процедуры построения гистограммы, которая широко используется при решении многих задач обработки сигналов и изображений. После преобразования информационного избыточного графа операция над данными производится по адресу элемента массива гистограмм.

Адресом служит функция $F(x_i)$, которая определяет: к какому элементу гистограммы относится пришедший операнд x_i. Полученные результаты записываются в промежуточную вершину-массив WX.

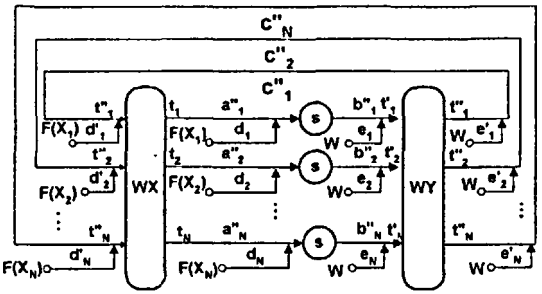


Рис.6. Пример *W*-графа, после выполнения векторизации информационных вершин.

Однако при решении ряда задач, например, задачи трассировки, применения однократной векторизации информационных вершин недостаточно, поэтому необходимо использовать процедуру вторичной векторизации.

Поэтому в отличие от первичной векторизации здесь предложено в вершину-массив преобразовывать адресные признаки ***F(x)*** и ***F(y),*** которые можно рассматривать как информационные адресные вершины.

В результате вторичной векторизации адресных вершин (рис. 7) формируется функционально-регулярный граф, в котором правила отображения дуг подграфов определены через правила отображения адресных вершин.

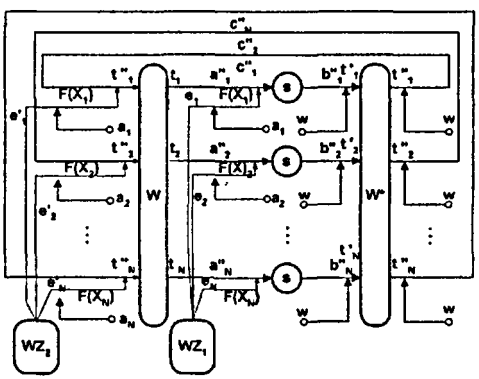


Рис.7. Вторичная векторизация адресных вершин.

Существуют задачи, информационные графы которых не структурированы и определяются некоторым внешним описанием. Эффективно реализовать в структурно-процедурной форме задачи подобного рода ранее описанными методами не представляется возможным. К числу таких задач относятся задачи логического и функционального моделирования. Несмотря на то что имеется определенный цикл повторения крупных участков задачи, соответствующих состоянию моделируемой схемы в определенный момент времени, взаимосвязь между элементами для каждой моделируемой схемы различна. Разработан алгоритм мажорирования таких графов. В результате предложенного в диссертации преобразования формируется параллельный информационный граф. При этом связи между слоями параллельного информационного графа однозначно соответствуют коммутации между процессорными элементами (номерами, которыми отмечены операционные вершины). Данная коммутация должна меняться каждый такт.

С помощью разработанных методов возможно преобразовать в структурно-процедурную форму даже нерегулярные информационные графы задач, таких как логическое моделирование, хотя МВС со структурно-процедурной организацией вычислений наиболее эффективны для сильносвязанных функционально-регулярных задач.

Как известно, распараллеливание еще не гарантирует увеличение производительности. Поэтому важным является вопрос о выборе оптимального варианта распараллеливания. В диссертации проведено сравнение эффективности структурно-процедурных и мультипроцедурных методов организации вычислительных процессов. Для простоты

рассуждения будем считать, что каждый слой содержит n базовых подграфов. Общее число слоев фрагмента равно m

Время решения задачи определятся суммой времен выполнения кадров:

$$T = \sum_{i=1}^{Z} T_{Ki} . \qquad (17)$$

Время выполнения кадра существенным образом зависит от степени распараллеливания. Когда число процессоров, отведенных для решения задачи, не превышает K - число арифметико-логических операций в базовом подграфе, то время выполнения кадра на N процессорах определяется по следующей формуле:

$$T_{СП}(N) = mn\left\lceil \frac{K}{N} \right\rceil (t_н + t_з(K) + \tau). \qquad (18)$$

Если $N \geq K$, то

$$T_{СП}(N) = t_н + t_з(K) + \tau m\left\lceil \frac{n}{\lfloor N/K \rfloor} \right\rceil . \qquad (19)$$

Эффективность распараллеливания $E(N) = \frac{T(1)}{T(N) \cdot N}$ можно оценить

$$E(N) = \frac{m(t_н + t_з + \tau \cdot n)}{t_н + t_з(K) + \tau \cdot m \cdot n} . \qquad (20)$$

Эффективность структурно-процедурной организации вычислений существенно возрастает, как только число процессоров сравняется с числом арифметико-логических операций в базовом подграфе. В этом случае происходит скачок производительности, т.к. можно осуществлять конвейер не только по базовым подграфам одного слоя, но и по базовым подграфам всех слоев структурно-реализуемого фрагмента, и тогда эффективность превысит единицу, т.е. производительность системы вырастает в большее число раз, чем число процессоров.

Таким образом, если справедливо неравенство

$$m(t_н + t_з) > t_н + t_з(K), \qquad (21)$$

то эффективность распараллеливания задачи будет больше единицы. Для МВС со структурно-процедурной организацией вычислений для большинства задач неравенство (21) справедливо. В частности, для задачи математической физики число узлов сетки существенно превышает число операций в узле сетки. Для задач цифровой обработки сигналов число отсчетов много больше, чем число операций по преобразованию сигналов. Для задач САПР число обрабатываемых элементов в тысячи раз больше, чем количество операций по преобразованию элементов.

Для МВС традиционных архитектур такая эффективность распараллеливания не достижима. Время выполнения задачи для мультипроцедурного метода распараллеливания

$$T_m(N) = \begin{cases} T_{m1}(N), \text{ если } N \leq n, \\ T_{m2}(N), \text{ если } N > n. \end{cases} \qquad (22)$$

Здесь $T_{m1}(N) = m \cdot K(t_н + t_з + \tau)\left\lceil \frac{n}{N} \right\rceil$, $T_{m2}(N) = \frac{m \cdot K}{\left\lceil \frac{N}{n} \right\rceil}(t_н + t_з\left(\left\lfloor \frac{N}{n} \right\rfloor\right) + \tau)$.

На рис 8 представлены сравнительные |рафические зависимости времени решения *(T)* и эффективности распараллеливания *(E)* от различного числа процессоров *(N)*. Здесь кривые, обозначенные цифрой 1, соответствуют мультипроцедурной организации вычислений; кривые, обозначенные цифрой 2, - структурно-процедурной организации вычислений.

При небольшом числе процессоров эффективность мультипроцедурной организации вычислений превосходит эффективность структурно-процедурных вычислений, но как только число процессоров системы достигнет количества вершин базового подграфа, происходит скачок производительности для структурно-процедурной организации вычислений. Легко заметить, что зависимость эффективности представляет собой пилообразную кривую, приближающуюся к линейной зависимости роста производительности при увеличении числа процессоров.

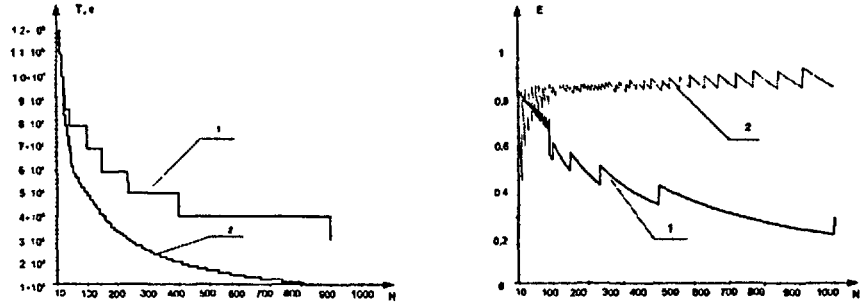


Рис.8. Зависимости времени решения и эффективности от числа процессоров.

В третьей главе представлены разработанные средства описания структурно-процедурных вычислений на основе языка программирования высокого уровня с неявным представлением параллелизма. В отличие от языков потоков данных предложено ограничение: правило единственной подстановки действует в специальных конструкциях языка, описывающих вычислительные структуры, задаваемые пользователем. Кадр языка соответствует совокупности арифметико-логических команд, выполняемых на различных ЭП и контроллерах распределенной памяти, соединенных между собой в соответствии с информационной структурой алгоритма. В теле кадра может находиться несколько операторов присваивания; программист не выделяет в них параллельные участки вычислений; в результате преобразования программы транслятором параллелизм, содержащийся в описании ВС, извлекается из нее естественным образом на основании взаимосвязей между данными. Описание кадра: CADR name,P; END name; где P - список внутренних операторов кадра. Внутренние операторы кадра реализуются структурно, внешние - процедурно. При этом обеспечивается биективное соответствие программе на языке и граф-схемам.

В предлагаемом языке переменные разделяются по способу хранения на мемориальные, коммутационные и регистровые. Мемориальной (MEM) переменной называется величина, хранящаяся в ячейке памяти и, следовательно, сохраняющая свое значение до очередного переприсваивания. Коммутационной переменной (COM) называется величина, служащая для описания каналов между элементами МВС, значения переменной дан-

ного типа недоступны пользователю. Введение переменной данного типа позволяет упростить графы вычислительных структур. На рис.9 приведены программы и эквивалентные им графы вычислительных структур, иллюстрирующие минимизацию оборудования при введении коммутационной переменной.

```
DCL A,B,Q,C,D,K,L,S) MEM;
CADR n1;
A=B*C+D-K/L+S; Q=B*C+D-K/L-S;
END n1;
```

```
DCL (A,B,Q,C,D,K,L,S) MEM,Z COM;
CADR n2;
Z=B*C+D-K/L; A=Z+S; Q=Z-S;
END n2;
```

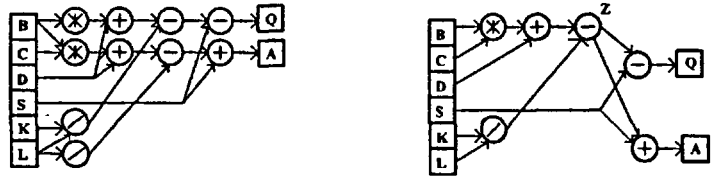


Рис.9. Использование мемориальной и коммутационной переменных.

В данном языке массивы различаются по способу обработки на векторы и потоки. Потоками являются массивы, элементы которых могут быть обработаны только последовательно. Векторами являются массивы, элементы которых могут быть обработаны параллельно. На рис. 10 представлены программы, являющиеся граничными примерами извлечения параллелизма, и графы вычислительных структур, в которые они будут отранслированы.

```
DCL (A,B,C) MEM [10 VEC];
CADR summap;C=A+B; END summap;
```

```
DCL (A,B,C)[10 STR];
CADR summas,C=A+B; END summas;
```

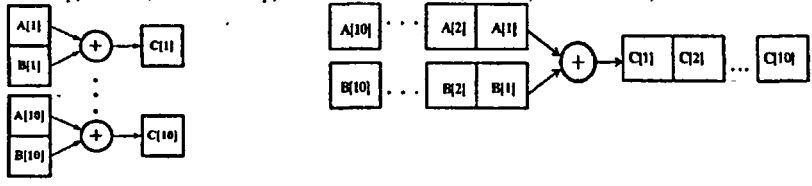


Рис.10. Параллельное и последовательное сложение массивов.

Все записи, находящиеся во внутреннем операторе цикла FOR, мультиплицируются в соответствии с параметрами цикла на вычислительную структуру для векторов или осуществляется периодическое поступление элементов массивов по заданной оператором цикла процедуре для потоков. Внешний по отношению к описанию вычислительной структуры оператор FOR осуществляет циклическое выполнение операторов, в том числе, и кадров, находящихся в теле цикла. На рис.11 представлен пример оператора цикла.

Семантика синтаксически одинаковых конструкций кадра определяется структурой данных, однозначно диктующих правило доступа к элементам массивов.

```
DCL (A,B,C) [10 VEC];
FOR I=1 TO 10 BY 2 DO;
 CADR summa;
  FOR J=I TO I+1 DO;
    C[J]=A[J]+B[J];  END;
END summa;END;
```

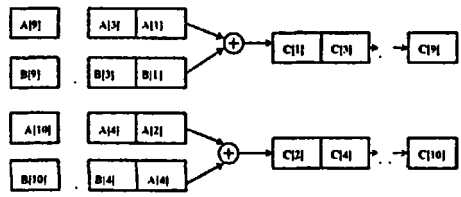
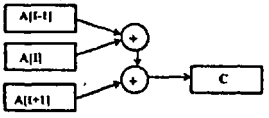


Рис. 11. Использование циклических операторов.

Если в одном кадре для вычисления переменной происходит обращение к нескольким элементам массива, то для вектора это означает определенным образом заданную коммутацию каналов распределенной памяти, а для потоков - запаздывание между элементами массива (неявно заданное использование регистровых переменных для буферизации). На рис.12 приведены программы и графы вычислительных структур, которые демонстрируют отличие индексации обращения к элементам потоков и массивов.

```
DCL A[10] VEC;
CADR a;
FOR I=2 TO 9 DO; C=A[I]+A[I-1]+A[I+1]; END;
END a;
```

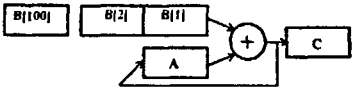



```
DCL A[10] STR;
CADR b;
FOR I=2 TO 9 DO; C=A[I]+A[I-1]+A[I+1; END;
END b;
```

Рис.12. Обращение к элементам векторов и потоков.

Использование правила единственной подстановки в теле кадра приводит к трудностям при организации рекурсивных вычислений. Рассмотрим пример: пусть требуется осуществить сложение элементов массива. Эту задачу предложено решить наиболее эффективно с использованием регистровой переменной, для которой правило единственной подстановки не выполняется. На рис.13 представлены программы и эквивалентные им графы, производящие последовательное и параллельное суммирование элементов массива B.

```
DCL A REG, B[100] STR, C,I;
CADR a; A=0; SAVE A; END a;
CADR b; FOR I=1 TO 100 DO;
A=A+B[I], C=A; END: END b;
```




```
DCL A REG INIT(0), B[4 VEC, 25 STR], C,I;
CADR b; FOR I=1 TO 25 DO;
A=A+((B[I,1]+B[I,2])+(B[I,3]+B[I,4])),
END b;
```

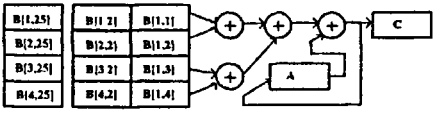


Рис.13. Последовательное и параллельное сложение элементов массива.

Для реализации ветвления в программах используется условный оператор IF. Семантика внешнего условного оператора совпадает с семантикой условных операторов процедурных языков программирования высокого уровня. Внутренний условный оператор выполняется следующим образом: группа операторов **a и b,** представляющих подграфы графа кадра, выполняется параллельно; результаты вычислений обеих групп поступают на переключатель, работа которого определяется логическим выражением q. Другими словами, внутренний условный оператор реализует структурный аналог условного перехода. Пример внутреннего условного оператора представлен на рис. 14.

Разработаны алгоритмы трансляции типовых конструкций языка и учитывающих особенности реализации структурных вычислений, разработаны алгоритмы компоновки графа кадра на макропроцессоры.

```
DCL (A,B) [100] STR;
CADR b;FOR I=1 TO 100 DO;
IF A[I]>0 THEN B[I]=sin(A[I]);
ELSE B[I]=exp(A[I]**2),END;
END b;
```

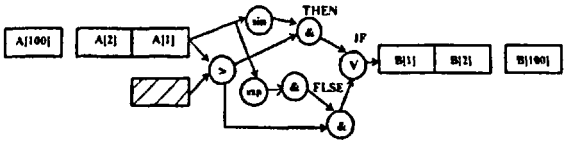


Рис. 14. Структурный аналог условного перехода

В четвертой главе на основании разработанных методов структурно-процедурных вычислений и средств описания программ разрабатывается технология индуктивных параллельных программ, обеспечивающая реализацию параллельных программ для различных конфигураций МВС.

Для высокопроизводительных многопроцессорных систем с массовым параллелизмом необходимо обеспечить эффективную организацию многозадачного режима. При параллельном программировании задач на определенные аппаратные компоненты многопроцессорной системы со структурно-процедурной реализацией вычислений неизбежны простои в процессе реализации потока заданий. Очередное задание может быть отправлено на выполнение тогда и только тогда, когда свободны все аппаратные компоненты, требуемые заданию. Более совершенной является организация, когда каждая задача может быть решена практически на любом аппаратном ресурсе. Поскольку в кадр (рис.15) преобразуется кортеж базовых подграфов, то для структурно-процедурной организации вычислений можно синтезировать масштабируемые параллельные программы.

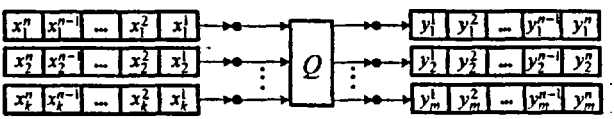


Рис.15. Исходный кадр.

Функции отображения базовых подграфов в одном слое могут в зависимости от степени распараллеливания реализоваться как последовательно, так и параллельно. Функции отображения слоев реализуются исключительно последовательно.

При увеличении аппаратного ресурса системы для решения задачи кадр можно естественным образом распараллелить. Граф кадра мультиплицируется столько раз, во сколько раз ресурс системы превышает ресурс, необходимый для структурной реализации графа *Q,* а также мультиплицируются каналы распределенной памяти, в которых хранятся исходные данные кадра и результаты. Формируется единый кадр (рис. 16), функциональный граф которого содержит несколько подграфов *Q,* что обеспечивает неизменность единой управляющей программы. Все элементы системы перестраиваются согласованно.

Поступившие задания рассматриваются планировщиком заданий, который реализует функции параллельных индуктивных программ и планирует загрузку МВС для различных вариантов распараллеливания задания.

Минимальное время отклика операционной системы, а, следовательно, и минимальное снижение эффективности решения задач на многопроцессорной системе обеспечивается при создании планировщика заданий по принципу разделения аппаратного ресурса. В этом случае заданию на время его выполнения выделяется неизменный фраг-

мент аппаратного ресурса, на котором задание выполняется практически независимо от других заданий, одновременно реализуемых в многопроцессорной системе.

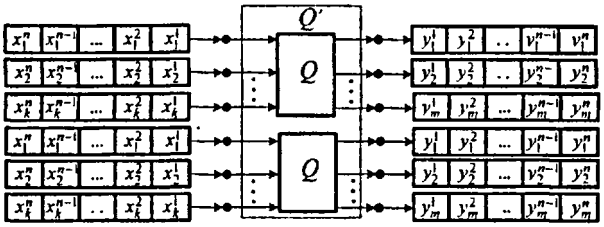


Рис. 16. Распараллеливание кадра

Параллельные индуктивные программы описываются четверкой $<P_o,X_o,Fp,Fx>$, где P_o параллельная структурно-процедурная программа, решаемая на минимальном ресурсе X_o; Fp-функция, ставящая в соответствие варианты распараллеливания задачи: $P_i=Fp(P_{i-1})$; Fx-функция, ставящая в соответствие варианты конфигурации МВС: $X_i=Fx(X_{i-1})$.

Алгоритм планировщика заданий представлен ниже.

1°. Если пришло новое задание J_i, то оно ставится в очередь Q_i: $SQ_i= SQ_i+1$; $Q_i[SQ_i]=J_i$.

2°. Если задание J_a в очереди активных заданий Qa завершилось, то перейти к **3°**, иначе к **5°**.

3°. Переводится задание в выходную очередь Q_o: $SQ_o=SQ_o+1$, $Q_i[SQ_o]=J_a$; $SQ_a=SQ_a-1$, $Q_a[SQ_a]=E$.

4°. Множество базовых модулей $L(J_a)$, которые занимало задание J_a, освобождается. Модифицируется свободный ресурс системы F: $F = F \cup L(J_a)$.

5°. j=1.

6°. Отбирается ресурс, удовлетворяющий требованиям j-го задания во входной очереди $G=Q_i[j]$: $L(G)=\{l \mid F(l) \& \sim Ra(G) \& Sa(l) \& Rc(G)<Sc(l)\}$, где $Sa(l)$ - характеристики *l*-го модуля системы, $Sc(l)$ •• описание связей *l-го* блока системы. Если m= M(L(G))>0, то перейти к **7°**, иначе к **8°**. M(L(G)) - мощность множества L(G), m - степень распараллеливания (число базовых модулей для G).

7°. Задание снимается с входной очереди $Q_i[i]=E$.

8°. Модифицируется свободный ресурс F = F / L(G).

9°. Инициализируется программа Pc(G,m), по окончании которой задание ставится в очередь активных заданий: $SQ_a = SQ_a +1$, $Q_a[SQ_a]=G$.

10°. j=j+1. Если $SQ_i>j$, то перейти к **6°**, иначе к **11°**.

11°. Просматривается очередь выходных заданий. Если к заданию q поступил запрос от пользователя, то задача пересылается пользователю и исключается из выходной очереди: $SQ_o= SQ_o - 1$, $Q_o[SQ_o]=E$.

В диссертации также разработаны алгоритмы планировщика заданий с учетом ограничений топологии коммутационной структуры МВС, приоритетов и других дескрипторов параллельных заданий. Анализ эффективности планировщика заданий проводился по четырем параметрам: коэффициенту использования оборудования *U,* степени распараллеливания задачи *R,* доле времени вычислений *C* (отношение времени исполнения задачи к общему времени прохождения задачи) и времени прохождения потока заданий

T (рис.17). При проведении вычислительного эксперимента интенсивность обслуживания была зафиксирована и варьировалась интенсивность поступления. Интенсивность поступления $b=1/t$, t - среднее время между заданиями. Интенсивность обслуживания $a=1/t_p$, где t_p - время решения задачи.

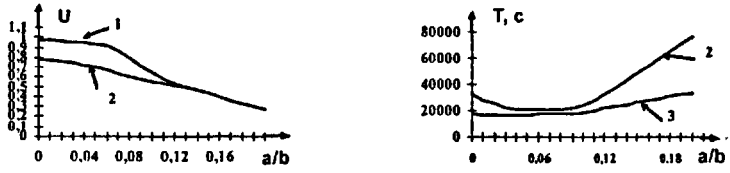


Рис. 17. Эффективность планировщика индуктивных заданий.

Здесь цифрой 1 помечены кривые, соответствующие потоку индуктивных заданий, цифрой 2 - кривые, соответствующие потоку заданий с фиксированной степенью распараллеливания. Анализируя полученные результаты, можно отметить следующее. Технология параллельных индуктивных заданий позволяет существенно повысить загрузку оборудования системы и, следовательно, уменьшить стоимость решения задачи. Причем, для коммутационных сетей с меньшими возможностями по организации связей преимущество индуктивных параллельных заданий будет больше. Так, для плотно поступающих заданий для коммутационной сети, построенной по полному графу, выигрыш в коэффициенте использования оборудования для индуктивных заданий составляет 20-25%; для коммутационной сети, построенной по графу типа "бинарное дерево" - 40-43%. При этом коэффициент использования оборудования для потока индуктивных заданий снижается незначительно. Для коммутационной сети по полному графу коэффициент использования оборудования для индуктивных заданий равен 0,97, для идеальных индуктивных заданий коэффициент использования оборудования равен 0,97, для коммутационной сети, построенной по графу типа "бинарное дерево" - 0,94. Примерно такие же соотношения имеют место для времени прохождения потока заданий *T*.

Одним из важнейших компонентов операционной системы, поддерживающих обработку потока параллельных индуктивных заданий, является разработанная в диссертации система посттрансляции, структура которой представлена на рис. 18.

Исходными данными для системы посттрансляции являются: задание, сформулированное в терминах технологии индуктивных программ, и номера базовых модулей МВС, выделенных для решения задачи планировщиком заданий.

Задание представляет собой тройку: $<Z,M_0,W>$, Z -загрузочный модуль; M_0 - граф минимальной конфигурации вычислительного ресурса, необходимого для структурно-процедурного решения задачи; W— секция правил наращивания графа конфигурации вычислительного ресурса при увеличении степени распараллеливания задачи. Задание должно быть запрограммировано в соответствии с технологией индуктивных параллельных программ. Показано, что в кадр преобразуется функционально-регулярный подграф задачи $K_i=F_i(\Phi_i(g))$, где K_i -базовый подграф, который тиражируется для параллельного и последовательного выполнения; F_i- правило наращивания базового подграфа при параллельной реализации, Φ_i - правило наращивания базового подграфа при последовательной реализации. Можно сказать, что функции F_i соответствует коммутационная

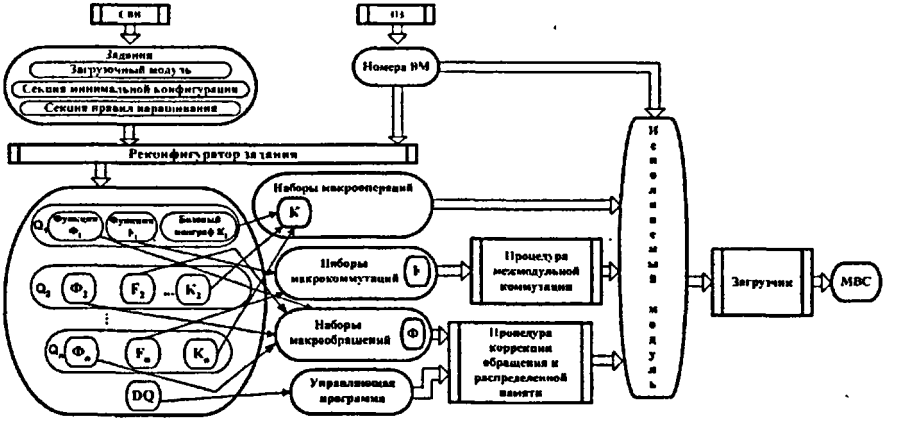


Рис.18. Структура системы пост трансляции.

составляющая, функции Φ_i - составляющая обращения к каналам распределенной памяти, реализуемая последовательно во времени. Множество базовых подграфов всех кадров $K=\{K_1, K_2, ..., K_N\}$ образует наборы макроопераций. Множество функций $F=\{F_1, F_2, ..., F_n\}$ образует набор макрокоммутаций. Множество функций $\Phi=\{\Phi_1, \Phi_2, ..., \Phi_N\}$ образует набор обращений каналам РП.

Если оборудования МВС достаточно для того, чтобы обеспечить максимальное распараллеливание не только в кадре K_i, но и в кадре K_j, который непосредственно связан с кадром K_i (расположен с ним на одном уровне графа кадров G_K или на смежных ярусах), то компоненты кадров K_i и K_j могут объединяться, образуя единый кадр K. Подобное объединение можно описать с помощью операции параллельного соединения. Для результирующего кадра формируются единый узел чтения и единый узел записи. При формировании объединенного кортежа функций чтения FR^* в него включаются все функции, принадлежащие кортежу FR_i, а также те элементы кортежа FR_j, которые не состоят в отношении информационной зависимости с элементами кортежа функций чтения FR_i и с элементами кортежа функций записи FW_i. Здесь и далее будем говорить, что две функции f и ψ информационно зависимы, если для всей области их определения $t \in [t_1, t_2]$ наблюдается равенство получаемых информационных вершин, формируемых выражениями $f(t)$ и $\psi(t+\Delta)$, где Δ - константа. В этом случае говорится о том, что существует отношение информационной зависимости $\mu(f, \psi, \Delta)$. Подобная зависимость может быть легко реализована с помощью элементов задержки, при этом дуги a и b, начало или конец которых связаны с функциями f и ψ, связаны отношением $\nu(a, b, \Delta)$. Причем, если в отношении ν указана входная дуга, то с функцией чтения связано начало дуги, а если - выходная, то с функцией записи связан конец дуги. Аналогично формируется кортеж функций чтения объединенного узла записи. Исходные кадры показаны на рис.19. Здесь функция чтения, соответствующая источнику дуги a_i, находится в отношении μ с функцией чтения, соответствующей началу дуги b_i, а функция записи, соответствующая кон-

цу дуги a_2, находится в отношении μ с функцией чтения, соответствующей началу дуги b_2. Данные дуги модернизируются: изменяется источник дуги, и на дугу включается буфер с соответствующей задержкой операндов. Результат объединения кадров представлен на рис.20.

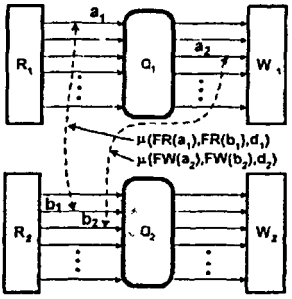


Рис.19 Исходные кадры.

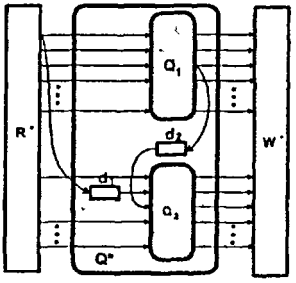


Рис 20 Объединение кадров

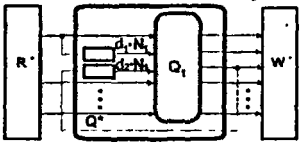


Рис.21. Свертка кадров

Если оборудование системы не позволяет реализовать кадр K даже для минимальной степени распараллеливания, то такой кадр необходимо разделить на два кадра: $K_1=(R_1,Q_1,W_1)$ и $K_2=(R_2,Q_2,W_2)$, которые выполняются последовательно. Операция разъединения позволяет для некоторого подграфа $VQ_1 \subset VQ$ получить кадровую структуру, которая может быть реализована в МВС. Область определения функций чтения и функций записи в полученных в результате выполнения данной операции двух кадрах совпадает с областью определения функций исходного кадра. Если в результате выполнения множества операций разъединения оказывается, что существуют два кадра K_1 и K_2, функциональные графы которых Q_1 и Q_2, и данные кадры расположены на одном уровне графа кадров, либо непосредственно связаны, то кадры K_1 и K_2 можно соединить последовательно, используя для этого операцию свертки (рис.21). При выполнении операции свертки размеры буферов, включаемых на дуги, находящиеся в исходных кадрах в отношении μ, умножаются на величину N_1 - размер информационного потока первого кадра. Операция свертки некоммутативна.

Программная реализация функций индуктивных программ содержит секцию описания графа минимальной конфигурации базового ресурса M_0 и секцию правил наращивания ресурса. Описание графа минимальной конфигурации представляет собой совокупность описания переменных, внешних связей графа и связей блоков. Предполагается возможность описания блоков по иерархическому принципу. Пример описания графа минимальной конфигурации приведен ниже.

Секция правил наращивания ресурса представляется следующим образом. В начале следует оператор описания переменных, задействованных в программе. Далее следуют операторы описания присвоения входных/выходных переменных графа минимальной конфигурации друг другу в зависимости от значения степени распараллеливания **_Par** (специальное служебное имя) Кроме оператора присваивания в секции могут находитьдиться условные операторы и операторы организации циклов. Пример описания графа минимальной конфигурации представлен на рис.22, секций наращивания - на рис 23, 24.

```
BEGIN Graph;
DCL a 18, g·3, d.4, k:18, b:4, q·3,
 ca·10, cd 2, ac:5,acd 8,ad 1,ab 4,ba 6;
INPUT (a,g,d);
OUTPUT (k,b,q);
A(IN(a,ca,ba),OUT(ac,acd,b,ad,ab),)
B (IN (ab,g); OUT(ba); );
C (IN (acd,ac);  OUT(cd), );
D(IN(ad,acd,d);OUT(q,g););
END Graph;
```

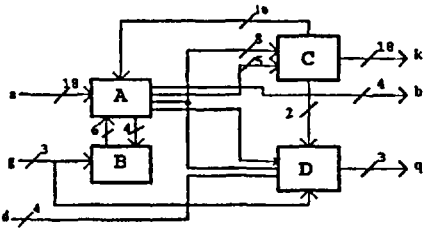


Рис.22. Пример графа минимальной конфигурации.

```
DCL k; FOR i=0 TO _Par-1 DO;
k=cadd(i,-1,_Par-1);
a[i]:=k[k],b[i]:=q[k]; g[i]:=d[k];
     END;END PROGRAM.
```

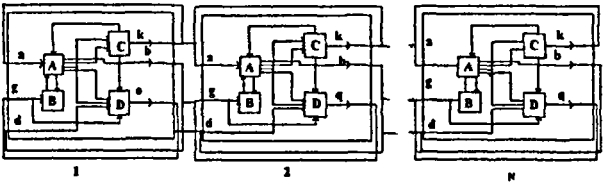


Рис.23. Пример секции линейного наращивания.

```
DCL a,b,i,j,k,l;
_Par=a·b; a:=log(_Par)/log2,b=a;
IF a - int(a) = 0 BREAK;
a:=a+1; b:=N/a;
IF b - int(b) = 0 BREAK; b:=b-1; a:=N/b,c:=b/4;
 FOR k=1 TO a DO; n:=cadd(k,-1,a); s:=0;
 FOR j=1 TO b DO;
FOR i=1 TO 4 DO; s:=s+1; l.=s mod 4;  m:=s / 4; Wi[k,j,i].=Wo[n,l,m];
 END; END; END;
                    END PROGRAM.
```

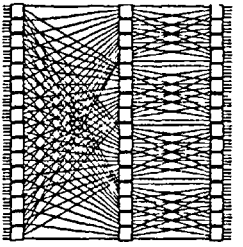


Рис.24. Пример секций наращивания ресурса в соответствии с косвенным бинарным графом для процедуры БПФ.

В пятой главе представлены разработанные программно-аппаратные средства, поддерживающие структурно-процедурные вычисления на уровне элементной базы.

Для обеспечения высокой производительности архитектура МВС должна воплощать основные принципы построения системы на всех уровнях, включая уровень элементной базы. С учетом этого в основу создания элементной базы для многопроцессорных вычислительных систем с массовым параллелизмом положены следующие принципы: унифицируемость; минимальная номенклатура; аппаратная поддержка синхронизации вычислений; аппаратная реализация системы команд; совмещение процессов обработки и передачи данных.

Для МВС ПА основными функциональными устройствами являются: макропроцессор, блок распределенной памяти, макрокоммутатор.

Функция обработки информации осуществляется в макропроцессоре (МАП), который содержит группу элементарных процессоров и коммутационную структуру, соединяющую ЭП по полному графу.

Макропроцессор предназначен для построения высокопараллельных процессорных полей со структурной реализацией крупных операций (макроопераций) графов вычислительного процесса. Разработаны и созданы различные варианты МАП на основе ПЛИС-технологии, реализующие вычисления в знакоразрядном и стандартном кодах.

В диссертации разработаны средства программирования макроопераций, представляющих собой программно-неделимую совокупность команд элементарных процессоров, образующих структурно-реализуемую функционально законченную математическую операцию. Директива описания макроопераций имеет следующий формат.

Macrooperation *имя (***in** *список формальных параметров входов,* **processors** *список формальных имен ЭП,* **out** *список формальных параметров выходов) Совокупность команд ЭП,* **Endmacro;**

Команда ЭП имеет вид: **P = φ1 (φ2(X) Θ φ3(Y));** где P - идентификатор ЭП; X,Y - идентификатор **s** операндов, **φI** —префиксные (постфиксные) функции, **Θ** - мнемоника операции.

Для удобства пользователей разработана и создана система программирования макроопераций, включающая в себя среду визуального программирования макроопераций (рис 25) и базу данных, в которой хранятся макрооперации для различных проблемных областей, например, для задач математической физики, для задач САПР СБИС, для процедур цифровой обработки сигналов

Функция хранения информации реализуется в блоке многоканальной распределенной памяти (макропамяти), которая может реализовать различные операции параллельного доступа к необходимой информации, хранящейся в памяти.

Блок распределенной памяти предназначен для создания систем распределенной памяти МВС и конструктивно реализуется с помощью контроллера распределенной памяти (КРП), который обеспечивает управление модулями распределенной памяти большой размерности и серийных ОЗУ. В модуле распределенной памяти размещаются программы и совокупность данных, представляющих исходные значения величин, промежуточные и окончательные результаты Оперативная память подразделяется на память программ и память данных. В памяти программ хранятся операторы, векторы начальных адресов и адресных приращений. КРП осуществляет чтение/запись потоков данных, реализует сложные процедуры параллельного доступа к распределенной памяти, выполняет модификацию параметров операторов В диссертации разработаны функциональные схемы КРП и система команд, а также средства программирования параллельного обращения к распределенной памяти.

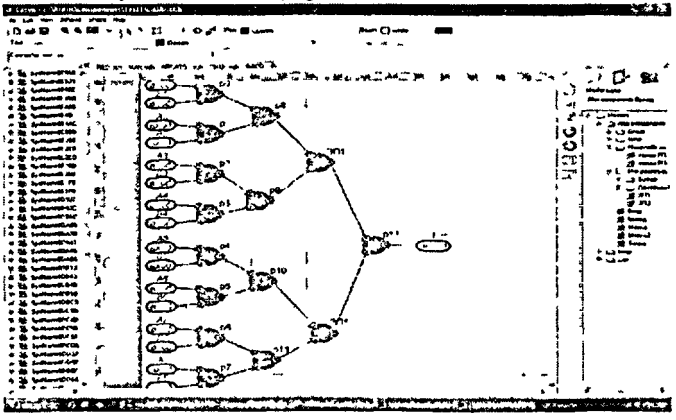


Рис 25 Среда синтеза макроопераций

Функция передачи информации реализуется в макрокоммутаторах, представляющих собой многоканальный программируемый динамический переключатель каналов

Разработаны функциональные схемы макрокоммутатора, его система команд, атак-

же средства описания макрокоммутаций Макрокоммутации используются для программирования каналов распределенной памяти и для каналов связи между РП и макропроцессорами

Macromemory *имя макрообращения* **(in** *список формальных параметров входов,* **channel** *список имен выходов каналов РП,* **out** *список формальных параметров выходов) Совокупность описания каналов РП,* **Endmacro;**

Ниже приведен пример определения доступа к каналам распределенной памяти. Структура макрообращения показана на рис 26

Macromemory A (in a,b,c,d, channel A,B, out q,v,w,u)
A<<a,B<<c(b), q<<d, v<<A, w<<B,u<<c, Endmacro;

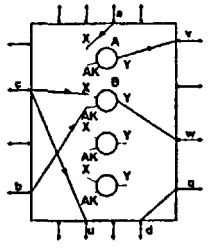


Рис 26 Структура макрообращения.

Описание коммутаций используется для построения кадров:

Declare Cadr *имя кадра* **(in** *список формальных параметров входов,* **out** *список формальных параметров выходов,* **Com** *список коммутационных переменных) Обращения к структурным конструкциям,* **Endcadr;**

Разработаны и созданы на основе ПЛИС-технологии различные варианты макрокоммутатора и контроллера распределенной памяти. Совокупность команд макрокоммутатора и КРП обеспечивает эффективную реализацию структурно-процедурных вычислений, что подтверждено реализацией параллельных структурно-процедурных программ для решения задач различных проблемных областей.

Технические решения по элементной базе многопроцессорных вычислительных систем со структурно-процедурной организацией вычислений защищены четырьмя патентами РФ на изобретение.

В шестой главе предлагаются варианты построения аппаратных средств МВС ПА, эффективно реализующих структурно-процедурные вычисления на основе модульной наращиваемости.

На основе анализа различных вариантов построения структуры распределенной памяти (РП) показано, что для многопроцессорных систем наиболее рациональной является архитектура с распределяемой памятью, когда элементы (каналы) распределенной памяти могут с помощью пространственной коммутационной системы подключаться к любому элементарному процессору (рис.27) Такая структура МВС позволяет повысить эффективность параллельных вычислений за счет более рационального взаимодействия компонентов системы при структурной реализации процесса решения задачи и снижения доли времени, необходимого для организации параллельных вычислений.

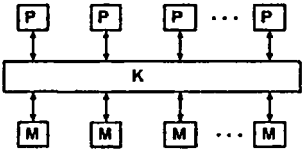


Рис 27. Структура МВС СПОВ

На основе анализа коммутационных структур показаны преимущества коммутаций иерархических (рис.28) и ортогональных (рис.29), требующих минимальных аппаратных затрат при реализации функций системы. Разработаны структуры МВС с многоуровневой коммутацией, эффективно реализующей решение задачи математической физики и бинарного N-графа, эффективно реализующего вычисления БПФ. На основе ортогональной и иерархической систем коммутации разработаны и созданы семейства базовых модулей (БМ), структурные схемы которых показаны на рис.30 и рис.31. Предложены одномерные и двумерные структурные

схемы синхронизации системы, вторая обеспечивает более скоростную согласованную перестройку системы. Базовые модули реализованы на основе ПЛИС-технологии.

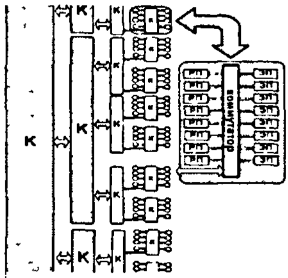


Рис.28. Иерархическая коммутационная система

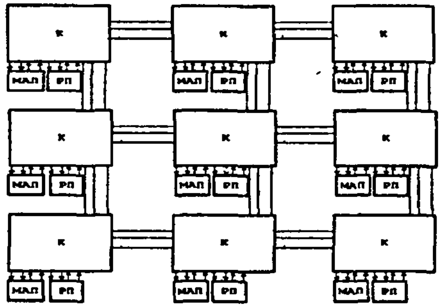


Рис.29. Ортогональная коммутационная система.

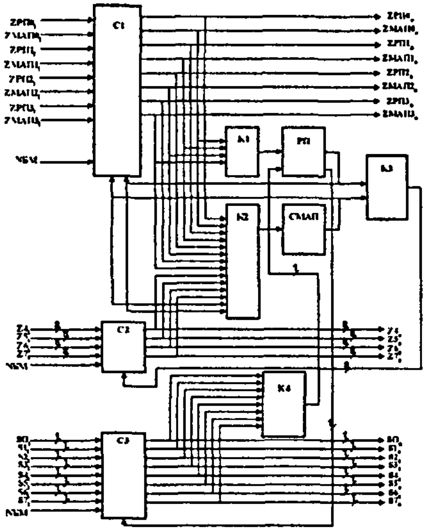


Рис.30. БМ с иерархической коммутационной системой.

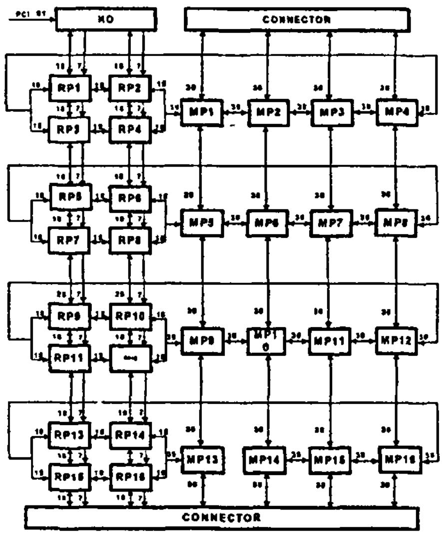


Рис 31. БМ с ортогональной коммутационной системой.

На основе БМ создан ряд модульно-наращиваемых МВС (рис.32-34).

В частности, модулью-наращиваемая МВС со структурно-процедурной организацией вычислений, созданная в рамках Программы Союзного государства России и Беларуси "Разработка и освоение в серийном производстве семейства высокопроизводительных вычислительных систем с параллельной архитектурой и создание прикладных программно-аппаратных комплексов на их основе (шифр "СКИФ")", имеет следующие характеристики:

Количество БМ	8	Количество каналов РП	256
Количество ЭП	512	Производительность,	$2\cdot10^{11}$ оп/с

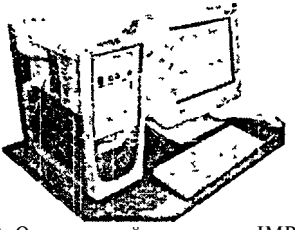


Рис 32. Одноплатный ускоритель IMB PC.
Количество ЭП 64
Количество каналов РП 16
Производительность $2{,}5 \cdot 10^{10}$ оп /с

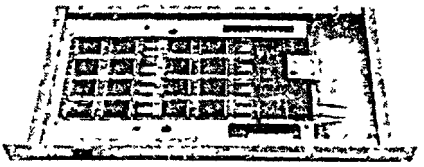
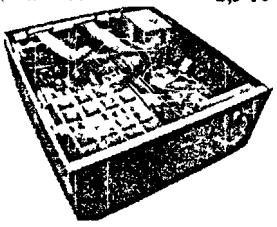


Рис 33 Автомат сопровождения цели.
Количество ЭП 64
Время обработки информации, не более 20мс
Размеры области анализа 256х256 пиксел



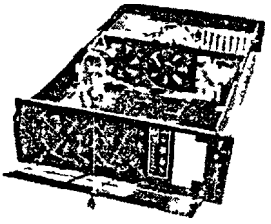


Рис.34. Многопроцессорные вычислители.

Показано, что для структурно-процедурных вычислений возможно уже на этапе трансляции задачи, в силу детерминизма вычислительного процесса, определить множество коммутационных структур, которые необходимо последовательно реализовывать в пространственной коммутационной системе. Такое свойство позволит существенно уменьшить аппаратные затраты на построение коммутационной структуры.

Созданные в рамках диссертации программные средства и многопроцессорные вычислительные системы демонстрировались на ряде международных научно-технических выставок, в том числе: на Всемирных выставках оргтехники, информации и телекоммуникаций CeBIT, Ганновер, Германия, 1997, 1998, 2001; на Международных выставках SIMO TCI, Мадрид, Испания, 1997, 1998; на Международной выставке-ярмарке Hannover Messe, Ганновер, Германия, 2002.

В приложениях приведены примеры решения задач структурно-процедурным методом распараллеливания, акты внедрения результатов работы.

Основной научный результат диссертации заключается в решении единой крупной научной проблемы развития теоретических основ организации эффективных параллельных вычислений, образованной совокупностью взаимосвязанных проблем разработки математических методов организации эффективных параллельных структурно-процедурных вычислений и создания программно-аппаратных средств, обеспечивающих реальную производительность МВС при решении задач различных классов, близкую к пиковой, а также линейный рост производительности при увеличении ресурсов системы.

При проведении исследований и разработок по теме настоящей работы получены следующие новые теоретические и прикладные результаты

1) Разработаны методы и программно-аппаратные средства, обеспечивающие сокращение времени решения на многопроцессорных системах с программируемой архитектурой задач различных проблемных областей, а также снижение стоимости и повышение эффективности эксплуатации дорогостоящих высокопроизводительных вычислительных комплексов.

2) Создан комплекс математических и программных средств, обеспечивающих эффективную реализацию на различных конфигурациях многопроцессорных систем с программируемой архитектурой задач различных проблемных областей.

3) Разработаны принципы преобразования задачи в структурно-процедурную форму с помощью сегментации информационного графа алгоритма в функционально-регулярный граф.

4) Доказана возможность построения методов и аппаратно-программных средств, обеспечивающих существенное повышение реальной производительности при решении на МВС ПА задач различных классов, а также пропорциональное увеличение производительности системы при увеличении числа процессоров на основе структурно-процедурной организации вычислений.

5) Разработаны методы приведения нерегулярных подграфов в эффективную функционально-регулярную форму, основанные на процедуре векторизации и мажорирования информационных подграфов.

6) Доказана более высокая эффективность структурно-процедурной организации вычислений по сравнению с традиционными мультипроцедурными методами организации параллельных вычислений для широкого класса задач.

7) Разработаны средства описания СПОВ на основе языка программирования высокого уровня, позволяющего описать различные степени распараллеливания и обеспечить однозначное отображение в структуру МВС ПА параллельных алгоритмов, а также детерминизм выполнения параллельных программ.

8) Разработаны теоретические принципы, методы и средства технологии индуктивных параллельных программ, обеспечивающие реализацию параллельных структурно-процедурных программ для различных конфигураций МВС со СПОВ, когда каждое задание может быть реализовано на любом количестве и произвольном сочетании базовых модулей системы.

9) Разработаны алгоритмы построения планировщика индуктивных заданий и системы посттрансляции, обеспечивающие за счет применения оригинальных методов преобразования кадровых структур возможность повышения степени распараллеливания задачи в зависимости от выделенного ресурса.

10) Разработаны и созданы программно-аппаратные средства, поддерживающие структурно-процедурные вычисления на уровне элементной базы, макропроцессора, структурно-реализующего вычисления крупных функционально-законченных программно-неделимых фрагментов задачи, элемента коммутирующей системы - макрокоммутатора и контроллера распределенной памяти, обеспечивающего высокоскоростной параллельный бесконфликтный доступ к каналам распределенной памяти и управление вычислительным процессом.

11) Разработаны и созданы эффективные аппаратные средства, поддерживающие структурно-процедурные вычисления, базирующиеся на принципах модульной наращиваемости, на основе которых разработаны и созданы базовые модули МВС со СПОВ.

12) Разработаны и созданы модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений, позволяющие повысить удельную производительность в несколько раз по сравнению с традиционными архитектурами многопроцессорных систем.

Основные публикации по теме диссертации

1. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. - М.: Янус-К, 2003. - 380 с. (опубликована при поддержке РФФИ, грант № 02-07-95004-д).

2. Левин И.И., Пономарев И.М. Реализация БПФ на многопроцессорной системе со структурной организацией вычислений //Электромеханика, 1995. -№ 4. - С.72 - 74.

3. Левин И.И., Гузик В.Ф., Сафронов О.О. Представление параллелизма в программах для многопроцессорной системы с программируемой архитектурой // Известия ВУЗов. Северокавказский регион. Технические науки, 1996. - № 2. - С. **4-15.**

4. Каляев А.В., Фрадкин Б.Г.Левин И.И., Унифицированная элементная база для построения реконфигурируемых под задачу вычислительных систем // Известия вузов. Электроника, 1997. -№ 1. - С.75-83.

5. Каляев А.В., Каляев И.А., Левин И.И., Пономарев И.М. Базовый модуль для построения реконфигурируемых под задачу вычислительных систем // Известия вузов. Электроника, 1998. - № 4. - С. 67-74.

6. Каляев А.В., Левин И.И. Многопроцессорные системы с перестраиваемой архитектурой; концепции развития и применения //Наука - производству, 1999.-№11.-С.11-19.

7. Каляев А.В., Левин И.И., Пономарев И.М. Базовый модуль многопроцессорной вычислительной системы с программируемой архитектурой для эффективного решения исследовательских и производственных задач //Наука - производству, 1999.-№ 11.-С.33-39.

8. Левин И.И., Пономарев И.М. Структурно-процедурная реализация кластерной группировки данных в задачах обработки изображений //Электромеханика, 1999.-№2.-С.54-57.

9. Левин И.И. Структурно-процедурная реализация электрического моделирования на многопроцессорной системе//Известия ВУЗов. Электромеханика, 2002. -№1. - С. 27-30.

10.Левин И.И., Коробкин В.В., Чернов Е.И. Об одном подходе к проблеме повышения надежности управляющего вычислительного комплекса с использованием технологии МВС ПА // Мехатроника, автоматизация, управление. -М.: Новые технологии, 2003.-№4.-С.40-43.

11.Левин И.И., Трунов Г.Л. Отказоустойчивое функционирование реконфигурируемых МВС // Электромеханика, 2004. -№ 3. - С. 11-15.

12. Каляев В.А., Левин И.И., Фомин С.Ю. Об оценке эффективности решения задач математической физики на многопроцессорных системах // Электронное моделирование, 1989.-№6.-С.11-15.

13. Kalyaev A.V., Kaliaev I.A., Levin I.I. The Base Module of Multiprocessor System with Structural-Procedural Organization of Computing // Parallel Computing Technologies. Proceedings of 4-th International Conference, PaCT-97. Yaroslavl, Russia, 1997. - Pp.394-396.

14. Левин И.И. Язык параллельного программирования высокого уровня для структурно-процедурной организации вычислений // Труды Всероссийской научной конференции "Высокопроизводительные вычисления и их приложения", Черноголовка, 2000. - М:Изд-во МГУ.-С.108-112.

15. Каляев А.В., Левин И.И. Структурно-процедурная организация параллельных вычислений // Труды межд. конф. "Параллельные вычисления и задачи управления (PACO'2001)". -М: ИПУ РАН им. В.А.Трапезникова, 2001. -Т. 5. - С. 112-121.

16. Левин И.И., Штейнберг Б.Я. Сравнительный анализ эффективности параллельных программ для различных архитектур многопроцессорных систем // Искусственный интеллект. - Донецк: Наука і освіта, 2001. - №3. -С.234-242.

17. Левин И.И., Шахов Р.В., Шматок А.В., Сластен Л.М. Математическое обеспечение многопроцессорных вычислительных систем с программируемой архитектурой // Искусственный интеллект. - Донецк: Наука і освіта, 2002. - №3. - С.286 294.

18. Левин И.И. Многопроцессорная система с программированием архитектуры на нескольких уровнях // Труды Первой Всероссийской научной конференции "Методы и средства обработки информации". -М.: МГУ, 2003. - С. 111-118.

19. Левин И.И., Пономарев И.М. Структурно-процедурная реализация задачи трассировки // Искусственный интеллект. Донецк: Наука і освіта, 2003. - №3. - С.121-129.

20. Левин И.И. Ресурсонезависимое параллельное программирование // Искусственный интеллект. - Донецк: Наука і освіта, 2002. - №3. - С.277-285.

21. Левин И.И. Модульно-наращиваемая многопроцессорная вычислительная система со структурно-процедурной организацией вычислений на основе ПЛИС-технологии // Искусственный интеллект. - Донецк: Наука і освіта, 2003. - №4. -С.446-453.

22. Матричный коммутатор: Патент РФ № 2059288 на изобретение по заявке № 94029855/09. / Левин И.И., Ерохин А.В., Рыжих О.А., Фрадкин Б.Г. - Заявл. 05.08.1994; Опубл. 27.04.1996. - Бюлл. № 12. - 5 с.

23. Матричный коммутатор: Патент РФ № 2103729 на изобретение по заявке № 94029856/09. / Левин И.И., Ерохин А.В., Рыжих О.А., Фрадкин Б.Г. - Заявл. 05.08.1994; Опубл. 27.01.1998. - Бюлл. № 3.-7 с.

24. Макропроцессор: Патент РФ № 2210808 на изобретение по заявке № 2001100388. / Каляев А.В., Левин И.И., Виневская Л.И., Станишевский О.Б. - Заявл. 05.01.2001; Опубл. 20.08.2003. - Бюлл. № 23. - 26 с.

25. Мультиконтроллер распределенной памяти: Патент РФ № 2210804 на изобретение по заявке № 2001122359. / Каляев А.В., Левин И.И., Виневская Л.И., Дмитренко Н.Н. -Заявл. 08.08.2001; Опубл. 20.08.2003.-Бюлл. №23.-44 с.

26. Левин И.И., Каляев В.А., Фомин С.Ю. Многопроцессорная система для оперативного моделирования гидрофизических процессов. // Сб. "Программные и аппаратные средства машинного моделирования", Москва, 1988.

27. Левин И.И. Сопроцессор для решения задач математической физики структурно-процедурным методом распараллеливания. // Сб. "Анализ эффективности вычислительных систем". -Львов, 1991. - Препринт НТЦ "Интеграл". -№ 9-16.

28. Левин И.И., Рвачев В.Л., Шевченко А.Н., Кошеленко А.И. Software and Hardware of Simulation of Phisical Mechanical Fillds. // Сб. "Parallel Computing Technologies Word Scientific". - Новосибирск, 1991.

29. Левин И.И., Пономарев И.М. Реализация алгоритма волновой трассировки на многопроцессорной системе со структурной организацией вычислений. - М., 1995. - Деп. ВИНИТИ, № 1553-В95. - 49 .

30. Левин И.И. Высокоэффективный алгоритм структурно-процедурной реализации задачи логико-временного моделирования на многопроцессорной системе. - М., 1995. - Деп. ВИНИТИ, № 1445-В95. - 31 с.

31. Левин И.И. Структурно-процедурная реализация функционального моделирования на многопроцессорной системе. - М., 1995. - Деп. ВИНИТИ, № 2043-В95. - 25 с.

32. Левин И.И., Пономарев И.М., Фрадкин Б.Г. Анализ эффективности структурно-процедурной организации вычислительного процесса при решении прикладных задач на МВС. // Сборник аннотаций и научных статей по результатам исследований, проведенных в ходе выполнения межвузовской научно-технической программы "Фундаментальные исследования в области прикладной физики и математики в технических ВУЗах России". ФИЗМАТ 1992-1995. -М.:ГКРФ по Высшему образованию, 1995.

33. Левин И.И., Виневская Л.И., Дмитренко Н.Н., Логвинов С.А. Элементная база для построения высокопроизводительных систем. // Труды международной конференции "Интеллектуальные многопроцессорные системы". -Таганрог: Изд-во ТРТУ, 1999.-С. 93-97.

34. Левин И.И., Шматок А. В. Технология параллельных индуктивных программ. // Труды международной конференции "Интеллектуальные многопроцессорные системы (ИМС99)". -Таганрог: Изд-во ТРТУ, 1999. - С. 142-146.

35. Левин И.И. Методы построения самонастраиваемой элементной базы. // Тезисы международной конференции "Интеллектуальные и многопроцессорные системы-200Г. - Таганрог: Изд-во ТРТУ, 2001. - С. 126-129.

36. Левин И.И., Трунов Г.Л. Отказоустойчивое функционирование многопроцессорных систем с массовым параллелизмом. // Искусственный интеллект. Донецк: Наука і освіта, 2002. - №3. - С.295-302.

37. Левин И.И. Отображение структурно-процедурной формы задачи на архитектуру многопроцессорной системы. // Материалы Международной научно-технической конференции "СуперЭВМ и многопроцессорные вычислительные системы". - Таганрог: Изд-во ТРТУ, 2002. - С. 95-98.

38. Каляев А.В., Каляев И.А., Левин И.И. Модульно-наращиваемые многопроцессорные системы с программируемой архитектурой и структурно-процедурной организацией вычислений. // Сборник докладов конференции "С.А. Лебедев и развитие отечественной вычислительной техники". -М., 2002. - С. 112-116.

40. Левин И.И., Шахов Р.В. Алгоритмы трансляции структурно-реализуемого фрагмента задачи для многопроцессорной системы с программируемой архитектурой. // Искусственный интеллект. -Донецк: Наука і освіта, 2003. - №3. - С.138-146.

ЛР № 020565 от 23 июня 1997 г. Подписано к печати 02.08.2004 г.
Формат 60x84 1/16. Бумага офсетная Печать офсетная
Усл. п.л -2,0. Уч. - изд л. - 1,8.
Заказ N° 271. Тираж 100 экз

ГСП 17А, Таганрог, 28, Некрасовский, 44
Типография Таганрогского государственного радиотехнического университета.