

УДК 004.272.43

## ВЫСОКОПРОИЗВОДИТЕЛЬНЫЕ МОДУЛЬНО-НАРАЩИВАЕМЫЕ МНОГОПРОЦЕССОРНЫЕ СИСТЕМЫ НА ОСНОВЕ РЕКОНФИГУРИРУЕМОЙ ЭЛЕМЕНТНОЙ БАЗЫ

И. А. Каляев<sup>1</sup>, И. И. Левин<sup>1</sup>

Рассматриваются вопросы перспективного подхода к созданию высокопроизводительных вычислительных систем, позволяющего создавать многопроцессорные вычислительные системы, превосходящие по своим техническим характеристикам зарубежные суперкомпьютеры. Представлены описания созданных базовых модулей и многопроцессорных систем на их основе, обеспечивающих высокую удельную производительность за счет оригинальной организации параллельных вычислений. Одноплатный базовый модуль содержит 320 элементарных процессоров и обладает производительностью 50 Гфлопс. Система из четырех базовых модулей имеет производительность 200 Гфлопс и занимает объем 40 дм<sup>3</sup>. Аппаратно-программные средства оригинальных многопроцессорных систем обеспечивают линейный рост производительности при решении задач различных проблемных областей, при этом реальная производительность системы не опускается ниже 50 % от ее пиковой производительности. Статья подготовлена по материалам доклада авторов на международной научной конференции "Параллельные вычислительные технологии" (ПаВТ-2007; <http://agora.guru.ru/pavt>).

**Ключевые слова:** высокопроизводительные вычислительные системы, многопроцессорные системы, суперкомпьютеры, базовый модуль, аппаратно-программные средства.

**1. Введение.** Многопроцессорные системы необходимы для решения задач, требующих обработки больших информационных массивов в сжатые промежутки времени. Неслучайно создание высокопроизводительных вычислительных систем (суперкомпьютеров) относится к факторам стратегического развития науки и техники и входит в первую десятку приоритетных технологий развитых стран. Без суперкомпьютеров невозможно детально моделировать экономические и социальные системы, прогнозировать экологические процессы и геофизические явления, разрабатывать и управлять важнейшими техническими объектами ядерной энергетики и космонавтики. В то же время реальная производительность многопроцессорных вычислительных систем (МВС) с традиционной архитектурой, представляющих собой объединение унифицированных процессоров, при решении задач, требующих большого числа пересылок информации, не превышает 10–15 % от декларируемой пиковой производительности системы [1]. Причиной этого является низкая скорость процедур межпроцессорного обмена, а также необходимость синхронизации множества взаимосвязанных последовательных процессов, каждый из которых выполняется в отдельном процессоре системы. Поэтому основной проблемой при разработке и эксплуатации суперкомпьютеров является обеспечение высокой производительности для любых классов задач.

В последнее время успехи суперкомпьютеров обеспечивались в основном за счет увеличения технологических возможностей, в частности, за счет минимизации технологических норм изготовления микросхем и за счет увеличения вследствие этого тактовой частоты микропроцессоров, а также за счет повышения плотности компоновки вентиля на кристалле. Однако, как отмечают специалисты, данный путь развития суперкомпьютеров уже не в состоянии поддерживать темп роста производительности вычислительных систем. Разрабатываются новые технологии для создания вычислительных систем, в том числе на основе оптических технологий и квантовых методов обработки информации. Однако в этих направлениях пока ведутся лишь фундаментальные теоретические и экспериментальные исследования, которые далеки от реального промышленного воплощения.

Важнейшим направлением повышения производительности многопроцессорных систем являются архитектурные методы. Для достижения сверхвысокой производительности были разработаны и использованы в реальных суперкомпьютерах такие широко известные архитектуры, как шинные, конвейерные, векторные, векторно-конвейерные, двумерные и трехмерные сетевые (матричные), тороидальные, гиперкубовые, иерархические, кластерные, потоковые, архитектуры типа баньян и ряд других архитектур. При

<sup>1</sup> Научно-исследовательский институт многопроцессорных вычислительных систем им. академика А. В. Каляева Южного федерального университета, ул. Чехова, 2, ГСП-284, 347928, Таганрог; e-mail: [kaliaev@mvs.tsure.ru](mailto:kaliaev@mvs.tsure.ru), [levin@mvs.tsure.ru](mailto:levin@mvs.tsure.ru)

разработке и конструировании суперкомпьютеров применяются такие общие архитектурные решения, как архитектура одного потока команд и многих потоков данных (SIMD-архитектура), многих потоков команд и одного потока данных (MISD-архитектура) и наиболее эффективная архитектура многих потоков команд и многих потоков данных (MIMD-архитектура). Все перечисленные архитектуры принадлежат к классу жестких архитектур. Однако жесткие параллельные архитектуры обладают существенным недостатком: высокая производительность системы при конкретной архитектуре достигается при решении определенного класса задач, при решении других классов задач производительность того же суперкомпьютера может резко падать, уменьшаясь на порядок. Это является следствием неадекватности данной конкретной архитектуры суперкомпьютера внутренней структуре решаемой задачи.

Известно, что максимальная скорость обработки информации может быть обеспечена в специализированной вычислительной системе, структура которой синтезирована в соответствии с алгоритмом решения задачи. Однако появление новых алгоритмов обработки информации, а иногда и коррекция алгоритмов, требует создания новых проблемно-ориентированных вычислителей, что приводит к увеличению стоимости решения как отдельной прикладной задачи, так и всего комплекса решаемых задач.

Для универсальных многопроцессорных систем проблема адекватности архитектуры системы по отношению к структуре решаемых задач является далеко не тривиальной. Особенно актуальное значение данная проблема имеет для наиболее перспективных суперкомпьютеров с массовым параллелизмом, в которых организуется параллельная работа сотен, тысяч, а иногда и десятков тысяч высокопроизводительных микропроцессоров, в процессе параллельной работы обменивающихся информацией как между собой, так и с распределенной памятью. В то же время, если пользователю будет обеспечена возможность программировать, а также автоматически в динамике вычислений перестраивать архитектуру суперкомпьютеров под структуру решаемой задачи или ее фрагмента, то проблема обеспечения высокой реальной производительности универсальных суперкомпьютеров будет решена. Фактически это означает, что пользователю должна быть предоставлена возможность программировать в рамках универсального суперкомпьютера виртуальные проблемно-ориентированные многопроцессорные системы с параллельной структурой, адекватной решаемой в текущий момент задачи (фрагмента задачи).

**2. Концепция построения многопроцессорных вычислительных систем с программируемой архитектурой.** В последней четверти XX века в НИИ многопроцессорных вычислительных систем Таганрогского государственного радиотехнического университета (НИИ МВС ТРТУ) под руководством академика РАН А. В. Каляева была разработана, теоретически и экспериментально подтверждена концепция многопроцессорных систем с программируемой архитектурой [2]. В процессе конструирования архитектура таких многопроцессорных систем не формируется окончательно и остается в определенном смысле незавершенной и открытой. Аппаратно-программные свойства МВС предоставляют пользователю возможность как на этапе трансляции задачи, так и в процессе ее решения программировать и настраивать архитектуру системы на виртуальные вычислители, адекватные информационной структуре решаемой задачи или ее фрагмента [3].

Аппаратно-программные средства МВС с программируемой архитектурой (ПА) позволяют синтезировать произвольные соединения между компонентами системы. Совокупность команд МВС ПА образует в рамках архитектуры системы виртуальную проблемно-ориентированную вычислительную структуру, адекватную информационному графу (подграфу) задачи [2]. Входные и выходные вершины графа соответствуют блокам распределенной памяти, внутренние (операционные) вершины графа соответствуют процессорам, настроенным на выполнение определенной операции. Дуги информационного графа должны быть реализованы в пространственной коммутационной среде МВС ПА. Такая организация вычислений называется структурной организацией.

Если ресурсов МВС ПА недостаточно для покрытия информационного графа задачи, то граф задачи необходимо разрезать на отдельные фрагменты. Одним из эффективных методов реализации параллельных вычислений в многопроцессорных системах с программируемой архитектурой является структурно-процедурная организация вычислений [4]. При реализации данного метода информационный граф задачи разбивается на кортежи изоморфных подграфов, которые являются информационно-независимыми или непосредственно зависящими друг от друга. Кортеж в общем случае взаимосвязанных информационных подграфов называется кадром. Можно сказать, что кадру соответствует подграф задачи, через который следует поток операндов. При этом каждая группа операндов (результатов) соответствует входным (выходным) вершинам определенного подграфа кортежа. Подграф кортежа выбирается таким образом, чтобы ресурсов системы было достаточно для его структурной реализации. Можно сказать, что при структурно-процедурной организации вычислений информационный граф задачи разбивается на функционально законченные фрагменты, каждый из которых аппаратно (структурно) реализуется в суперкомпьютере с

программируемой архитектурой (рис. 1).

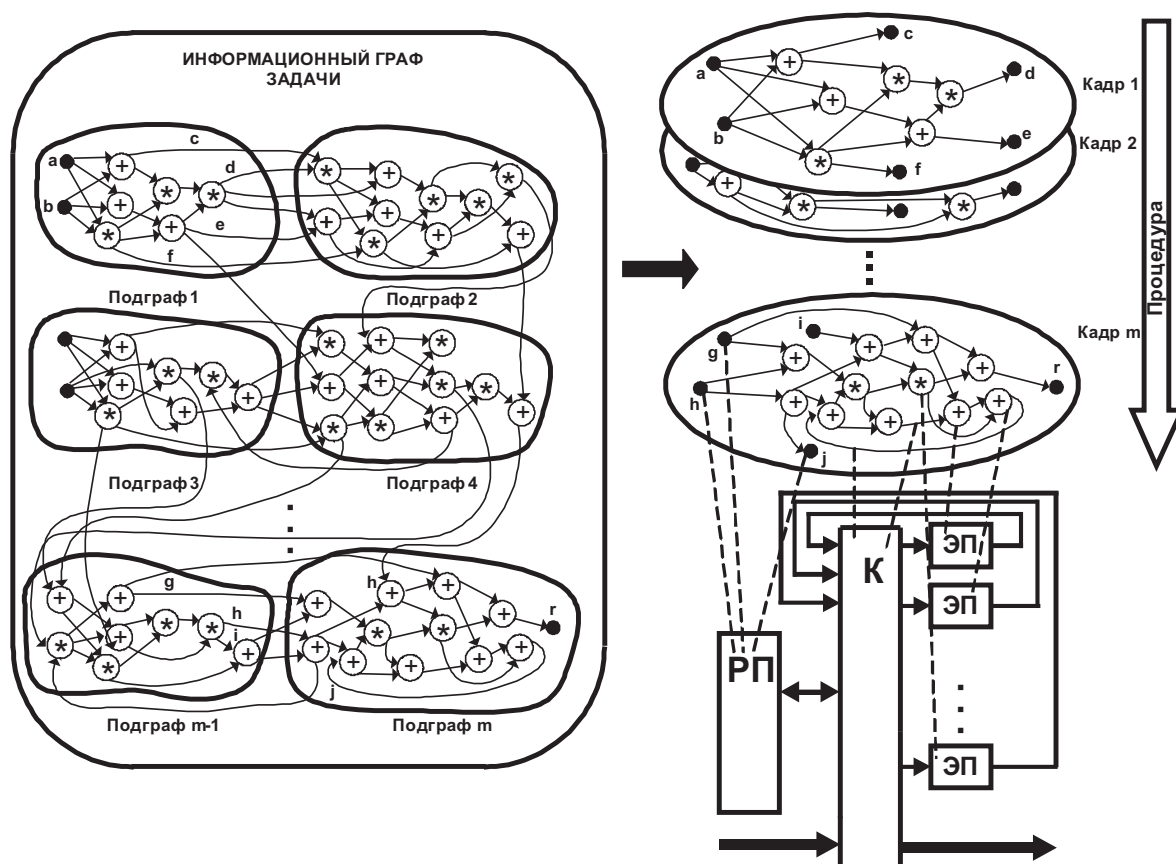


Рис. 1. Процесс преобразования задачи в структурно-процедурную форму

При этом имеет место соответствие вершин информационного графа задачи элементарным процессорам и данным в каналах распределенной памяти. Множество дуг информационного графа реализуется пространственной коммутационной системой. Настройка на кадры производится по единой управляющей программе, что обеспечивает фон-неймановский детерминизм вычислительной процедуры.

Процедура представляет собой последовательность вызовов кадров. Вычисления в теле кадра выполняются по принципу управления потоком данных и не требуют синхронизации. По существу организация вычислительного процесса в кадре осуществляется по принципу машины потока данных. Можно утверждать, что суперкомпьютер с программируемой архитектурой является своеобразным гибридом фон-неймановской архитектуры и архитектуры машин потоков данных и сочетает в себе их достоинства: детерминизм программирования, присущий фон-неймановским ЭВМ, и высокую реальную производительность, характерную для конвейерных машин потоков данных.

### 3. Развитие концепции многопроцессорных систем с программируемой архитектурой.

Несмотря на то что концепция МВС с программируемой архитектурой была разработана в начале 80-х годов прошлого века, реальные образцы универсальной МВС ПА были созданы только в конце 90-х годов. Основной причиной этого является необходимость создания специальной элементной базы. Такая элементная база — макропроцессорный комплект СБИС, разработанный в НИИ МВС ТРТУ, была создана в СКБ «Белмикросистем» НПО «Интеграл», г. Минск в 1992–1994 гг. Макропроцессорный комплект состоял из трех СБИС, каждая из которых выполняла строго определенную функцию:

- макропроцессор, представляющий собой совокупность элементарных процессоров, объединенных внутренним коммутатором по полному графу, предназначен для структурной реализации крупных функционально законченных операций (макроопераций);
- мультиконтроллер распределенной памяти, предназначен для управления процессом структурно-процедурных вычислений и организации информационных потоков между блоками распределенной памяти и другими элементами системы;
- макрокоммутатор, предназначен для динамической коммутации информационных каналов базово-

го модуля.

В 1996 г. на основе оригинальной макропроцессорной элементной базы был создан базовый модуль универсальной МВС с программируемой архитектурой. На одной плате было реализовано 32 элементарных процессора (два макропроцессора), связанных между собой по полному графу.

В течение нескольких десятилетий в России и за рубежом развивались концепции построения реконфигурируемых вычислительных систем, которые, как правило, строились из небольших вычислительных элементов, предназначенных для преобразования минимальных информационных единиц — битов. Из подобных блоков пользователь мог синтезировать процессорные элементы и далее вычислительную систему для решения определенного алгоритма задачи. В отличие от МВС ПА, архитектура которой включала в себя ряд крупных неизменных компонентов, что обеспечивало, с одной стороны, более простое программирование, а с другой стороны, возможность изменения структуры вычислителя в процессе решения задачи, архитектура реконфигурируемых систем требовала очень сложного и детального программирования, кроме того, практически исключала возможность реконфигурирования вычислителя в процессе решения задачи. Пользователь практически разрабатывал многопроцессорный вычислитель, начиная с самого низкого уровня, что приводило к высокой стоимости прикладных программ для реконфигурируемых систем.

Появившиеся в последние годы средства построения систем с реконфигурируемой структурой — программируемые логические интегральные схемы (ПЛИС) — позволили упростить процесс построения и программирования реконфигурируемых систем. Хотя на практике ПЛИС в основном используются в качестве интерфейсных блоков, они обладают высоким потенциалом для реализации высокоскоростных вычислений. Можно сказать, что в реконфигурируемых устройствах на основе ПЛИС реализуется своеобразный аналог программирования, только в отличие от традиционных процессоров программирование для ПЛИС выполняется не во времени, а в пространстве.

Как показал опыт эксплуатации реконфигурируемых вычислителей, построенных на основе ПЛИС, большинство подобных систем характеризуется теми же недостатками, что и реконфигурируемые системы в целом: длительным временем программирования, сложностью внесения исправлений в структурный алгоритм. Кроме того, скорость реконфигурирования ПЛИС на несколько порядков меньше скорости обработки информации, что исключает возможность реконфигурации ПЛИС в процессе решения задачи.

Сочетание принципов программирования архитектуры и ПЛИС-технологии позволило создать реконфигурируемый вычислитель нового качества. Такая многопроцессорная система с программируемой архитектурой строится из унифицированных базовых модулей, которые включают множество ПЛИС высокой степени интеграции.

Для подобной системы целесообразно создавать узлы и блоки системы, которые постоянно используются и структура которых не меняется, по крайней мере при решении задач одной проблемной области, и определить правило их программирования. Создание подобных вычислительных и управляющих компонентов неизбежно приводит к необходимости помимо структурного программирования (организации связей между блоками, характерной для ПЛИС), осуществлять программирование устройств в традиционном смысле этого слова, т.е. создавать для некоторых блоков, входящих в состав ПЛИС, вычислительные процедуры. Необходимость программирования во времени для реконфигурируемых систем обосновывается ограниченным ресурсом системы, которого может не хватить для структурной реализации задачи в полном объеме. Кроме того, вычислительные задачи могут включать в себя фрагменты, которые сложно реализовать структурно или которые требуют для подобной реализации чрезвычайно больших аппаратных затрат.

Структурно-процедурная организация вычислений обеспечивает детерминизм при различных запусках параллельной программы как порядка вызова кадров, происходящих по управляющей, по сути дела фон-неймановской, программе, так и порядка обработки операндов в кадре, определяемых аппаратно реализованным графом информационной зависимости. Детерминизм реализации структурно-процедурных вычислений позволяет уже на этапе трансляции алгоритма определить все необходимые в процессе решения задачи коммутационные структуры и порядок их реализации в системе. Данное свойство структурно-процедурной организации вычислений позволило сократить коммутационную среду за счет предварительной разработки на этапе трансляции задачи всех возникающих коммутационных структур. Было предложено для построения МВС ПА использовать ортогональную систему коммутации.

Базовый модуль с ортогональной системой коммутации был разработан и создан в НИИ МВС ТРТУ в 2000 г. и обладал приведенной производительностью  $2,5 \times 10^{10}$  операций в секунду. Производительность базового модуля за счет применения более совершенных архитектурных решений выросла более чем в 150 раз. Число элементарных процессоров возросло в восемь раз, а количество одновременно обрабатыва-

емых разрядов выросло в 16 раз. При этом благодаря созданным средствам автоматического построения коммутационных структур сложность программирования базового модуля практически не увеличилась.

Базовый модуль, созданный на основе ПЛИС-технологии в 2000 г., был использован для построения различных проблемно-ориентированных МВС, построенных на основе принципа модульного наращивания. Один и тот же базовый модуль был использован для решения задач: криптографии, цифровой обработки сигналов, цифровой обработки изображений, а также построения нейрокомпьютерных систем.

В то же время анализ построения вычислительных структур на основе базового модуля показал недостаточную эффективность МВС с программируемой архитектурой. Разнообразие вычислительных структур, которые возникают при решении задач различных проблемных областей, потребовало изменения функционального назначения объектов системы. Возможности ПЛИС-технологии реализуют подобное преобразование тривиальным образом путем загрузки определенного конфигурационного файла объекта в ПЛИС. Кроме унифицированных объектов часто возникает необходимость создания специальных функциональных устройств для повышения эффективности вычислений. Функциональные устройства создаются на более низком уровне, чем вычислительные структуры, созданные на основе унифицированных объектов. Вычислительные структуры проектируются на уровне команд элементарных процессоров и коммутации информационных каналов между ними. Функциональные устройства создаются на уровне таблиц истинности ячеек ПЛИС и их соединения между собой.

Таким образом, возникает необходимость двухуровневого программирования архитектуры многопроцессорной системы. На первом уровне (более низком) создаются объекты системы, которые располагаются в определенных ПЛИС базового модуля. Как правило, местоположение объекта и его функциональное назначение остаются неизменными не только в процессе решения отдельной задачи, но и в процессе решения задач определенного класса. Можно утверждать, что подобным образом производится настройка многопроцессорной системы на конкретную проблемно ориентированную архитектуру.

На втором уровне с помощью команд объектов (элементарных процессоров, контроллеров распределенной памяти, динамических коммутаторов) и множества функциональных устройств, объединенных распределенной коммутационной системой в единое коммутационное устройство, программируются и создаются вычислительные структуры для решения отдельных фрагментов задачи (кадров). Такая организация программирования архитектуры позволяет значительно сократить сроки создания эффективных параллельных программ.

#### **4. Принципы построения многопроцессорной системы с макрообъектной архитектурой.**

Многопроцессорная система с макрообъектной архитектурой строится из унифицированных базовых модулей. Базовый модуль состоит из множества макрообъектов, каждый из которых представляет собой ПЛИС высокой степени интеграции с несколькими миллионами эквивалентных вентилей.

При построении таких систем пользователь, как правило, сталкивается с серьезной проблемой: создание (программирование в пространстве) реконфигурируемого вычислителя под каждую конкретную задачу требует колоссального времени. В этой связи целесообразно выделять некоторые узлы и блоки, которые постоянно используются, по крайней мере при решении задач одной проблемной области. Выделение подобных вычислительных и управляющих компонентов неизбежно приводит к необходимости помимо структурного программирования осуществлять программирование в традиционном смысле этого слова, т.е. создавать для некоторых узлов и блоков вычислительные процедуры.

Необходимость программирования во времени для реконфигурируемых систем также обосновывается ограниченным ресурсом системы, которого может не хватить для реализации задачи в полном объеме. Кроме того, зачастую вычислительные задачи включают в себя фрагменты, которые сложно реализовать структурно или которые требуют для подобной реализации чрезвычайно больших аппаратных затрат. К числу подобных фрагментов относятся условные операторы (прежде всего, многократно вложенные), так как для их структурной реализации необходимо выполнять все альтернативные ветви, а не единственную ветвь, как при процедурных вычислениях. Так же структурно неэффективно выполняются рекурсивные вычисления с глубокими обратными связями и вычислительные фрагменты, содержащие множество информационно незначимых пересылок информации, каждая из которых при структурной реализации требует, по крайней мере одного регистра.

Таким образом, необходимо помимо структурных вычислений в реконфигурируемых системах использовать процедурные вычисления.

При программировании архитектуры системы на макроуровне — на уровне элементарных процессоров, каналов распределенной памяти и коммутации связей между ними удастся выделить три функциональных блока, каждый из которых имеет неизменные функции при решении задач из любых проблемных областей. Макропроцессоры реализуют обработку информации, макропамять обеспечивает хранение

информационных массивов, а также скоростной параллельный доступ к данным. Макрокоммутатор реализует и изменяет в процессе решения задачи соединение компонентов системы. Однако при программировании системы на низком уровне неизбежно возникает такая ситуация, что функциональное назначение узлов и блоков многопроцессорной системы необходимо изменять не только при переходе от одной проблемной области к другой, но и при решении различных задач одной и той же проблемной области.

Для адаптируемых систем даже функции объекта могут изменяться в процессе решения задачи. Устройство обработки информации может в следующий момент времени быть преобразовано в память, а коммутатор стать функциональным устройством. Разумеется, кроме ПЛИС базовый модуль должен содержать стандартные микросхемы памяти, которые фиксированно связаны с одной или несколькими ПЛИС. Однако архитектура базового модуля и системы в целом может произвольным образом изменяться, в том числе и динамически.

Следует подчеркнуть, что эффективность вычислительного процесса при реконфигурации архитектуры системы на низком уровне может быть повышена от трех до ста раз по сравнению с МВС, архитектура которой реконфигурируется на макроуровне. Это делает, с одной стороны, чрезвычайно привлекательной реконфигурируемые на низком уровне системы, а с другой стороны, их программирование становится по сложности сопоставимым с созданием новой вычислительной системы.

Все это требует новых методов и средств организации параллельных вычислительных процессов.

Предлагается для реконфигурируемых МВС со структурно-процедурными вычислениями ввести понятие “макрообъект”. В отличие от ранее определенных макроэлементов (макропроцессора, макропамяти и макрокоммутатора) макрообъект может изменять свое функциональное назначение. Введение макрообъекта, прежде всего, продиктовано ростом интеграции ПЛИС. Если в МВС с программируемой архитектурой 1998 г. присутствовали все три функциональных макроэлемента, что было необходимо при интеграции элементной базы на уровне 50–100 тыс. логических вентилей, то в МВС, созданных в 2000–2004 гг. как самостоятельный макроэлемент, макрокоммутатор полностью исчезает из системы. Его функции реализуют макропамять нового типа, включающая внутреннее коммутационное устройство, и макропроцессор. Подобная интеграция осуществлялась на элементной базе, содержащей 300–500 тыс. вентилей. Дальнейший рост числа элементов ПЛИС требует неизбежного объединения в единый комплекс вычислительных функций макропроцессора и управляющих функций, а также средств доступа к распределенной памяти, реализованной в макропамяти. При уровне в 6–10 млн. вентилей на кристалле такое объединение становится уже неизбежным.

В настоящее время в НИИ МВС ТРТУ разрабатываются базовые модули для построения макрообъектной вычислительной системы для решения задач различных проблемных областей. Базовый модуль представляет собой совокупность макрообъектов, соединенных между собой по принципу близкого действия. Большая часть макрообъектов сопряжена с внешней распределенной памятью и имеет высокоскоростные каналы для обмена информацией с другими базовыми модулями.

Каждый макрообъект, входящий в состав базового модуля, может содержать множество разнородных объектов, объединенных между собой некоторой внутренней пространственной коммутационной структурой, которая позволит соединить объекты в функционально законченный блок, а также обеспечивает транзит информации между различными макрообъектами. Кроме того, каждый макрообъект обязательно содержит интерфейсную схему, которая обеспечивает корректное взаимодействие между объектами, принадлежащими различным макрообъектам.

Структура унифицированного макрообъекта представлена на рис. 2. В состав макрообъекта входит интерфейсный блок (И), обеспечивающий высокоскоростной обмен информацией между макрообъектами, блок распределенной памяти (БРП) и вычислительный узел (ВУ).

Проведенные исследования показали, что объектами могут быть: элементарный процессор (ЭП), контроллер распределенной памяти (КРП), преобразователь информации (ПИ), динамический коммутатор

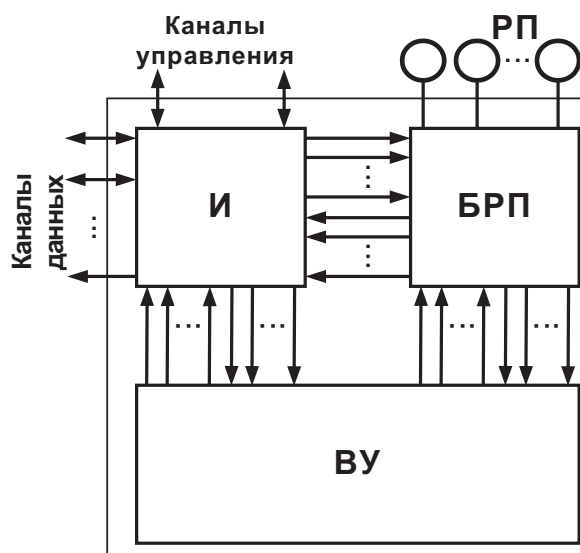


Рис. 2. Структура унифицированного макрообъекта

(ДК), блоки памяти различного назначения (П), процедурное арифметико-логическое устройство (АЛУ) и локальное устройство управления (ЛУУ).

Элементарный процессор осуществляет обработку пары операндов в соответствии с кодом операции, а также обеспечивает синхронизацию потоков операндов и реализацию структур операций (операции с константой, с обратной связью, с запаздыванием).

Процедурное АЛУ выполняет те вычисления, которые неэффективно реализовывать структурно. При этом их число должно быть настолько велико, чтобы обеспечить согласованный разбор элементов информационного потока, выходящих (входящих) из структурных компонентов макрообъектов. Число процедурных АЛУ может достигать нескольких сотен, что требует существенного сокращения системы команд подобных элементов и, как следствие, уменьшение аппаратных затрат на их реализацию. Предполагается, что процедурные АЛУ и ПИ будут проектироваться для каждой задачи или по крайней мере для проблемной области.

Преобразователь информации обеспечивает реализацию фиксированной функции, которая не может быть изменена. Следует отметить, что функциональное преобразование может быть выполнено над группой операндов, например, скалярное произведение векторов, численное интегрирование, умножение матриц, решение системы логических уравнений.

Совокупность элементарных процессоров, настроенных на выполнение определенных арифметико-логических операций и преобразователей информации, входы которых соединены внутренней пространственной коммутационной структурой в соответствии с информационным графом решаемой задачи, образуют вычислительную структуру, реализующую обработку информационных массивов в соответствии с алгоритмом решения задачи (фрагмента решения задачи). В вычислительную структуру могут быть включены дополнительные узлы и блоки, например, буферы данных для синхронизации информационных потоков или компараторы-переключатели.

На рис. 3 представлена структура вычислительного узла, в состав которого входит множество унифицированных элементарных процессоров, а также множество специально созданных для решения задачи или ее фрагментов функциональных устройств (ФУ), структура которых продолжительное время не изменяется в отличие от элементарных процессоров, структура которых модернизируется при настройке на определенную арифметико-логическую операцию.

Помимо вычислительных элементов (элементарных процессоров и функциональных устройств) в вычислительный блок входят коммутационные элементы: статические коммутаторы К1, К2 и множество блоков динамической коммутации (БДК). Статические коммутаторы обеспечивают соединение всего информационного массива, остающегося неизменным в течение времени выполнения кадра. Динамические коммутаторы обеспечивают возможность изменения связи, в том числе и каждый такт (с каждым операндом) для определенной группы информационных каналов. Совокупность статических и динамических коммутаторов обеспечивает возможность создания различных вычислительных структур для решения задач различных проблемных областей.

Информационные потоки в вычислительной структуре образуются с помощью контроллеров распределенной памяти, которые организуют скоростные параллельные обращения к сегментированной внутренней (буферной) и внешней памяти макрообъекта.

Структура блока распределенной памяти показана на рис. 4.

Блок распределенной памяти (БРП) содержит  $m$  адресных процессоров (АП), каждый из которых по отдельной независимой процедуре осуществляет доступ к блоку памяти данных (ПД) с помощью блока

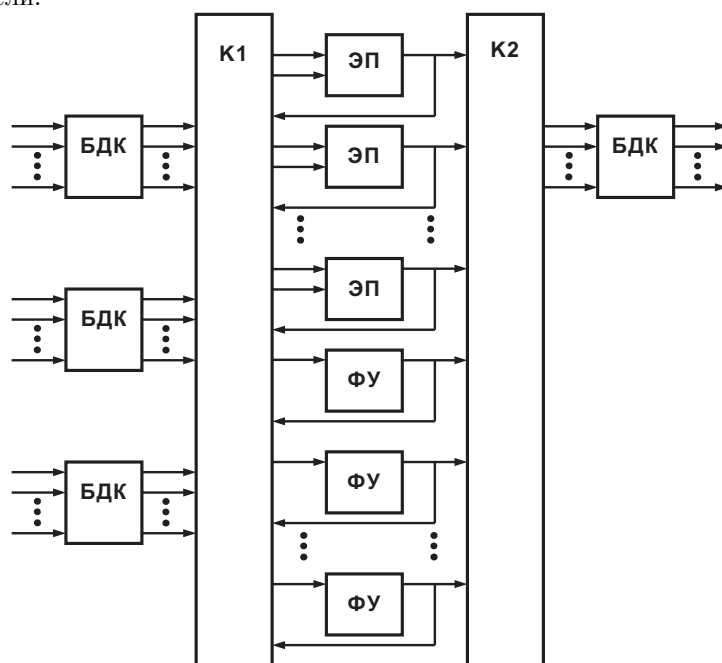


Рис. 3. Структура вычислительного блока



адресного интерфейса. В ряде случаев для организации сложных процедур обращения к распределенной памяти необходимо осуществлять коммутацию адресов, для этого в БРП вводятся специальные адресные коммутаторы (АК).

Совокупность адресных процессоров БРП обеспечивает высокоскоростной параллельный и бесконфликтный доступ к распределенной памяти. Для согласования и управления работой адресных процессоров в состав БРП был введен блок управления (БУ), который функционирует по управляющей программе. Эта программа находится во внутренней памяти БРП, более точно — в сегменте операторов (СО).

Внутренняя память БРП подразделяется на ряд сегментов, каждый из которых имеет определенное функциональное назначение.

Сегмент операторов содержит управляющую программу.

В сегменте параметров находятся параметры операторов группового обращения к блокам распределенной памяти. Параметры передаются и обрабатываются адресными процессорами. Динамические коммутаторы обеспечивают переключение информационных потоков между источниками и приемниками, причем изменение адреса источника (приемника) может производиться произвольно с каждым данным.

Информационная связь между базовыми модулями осуществляется коммутационной пространственной системой, которая реализована внутри базового модуля. Коммутационная система базового модуля реализует не только информационные связи между компонентами базового модуля, а также и информационные связи между БМ. Подобная организация коммутационной структуры обеспечивает однородность вычислительной системы (система состоит только из базовых модулей) и высокую технологичность организации вычислительных процессов. Обмен информацией между объектами, принадлежащими разным базовым модулям, организуется точно так же, как и обмен между объектами внутри базового модуля. Многопроцессорная система адаптирована к структуре задачи и, с одной стороны, обеспечивает высокую скорость обработки информации за счет высокой удельной производительности и конвейерной обработки данных, а с другой стороны, реализует фон-неймановский детерминизм вычислений и может быть основой для создания сверхвысокопроизводительных вычислительных систем.

Информационная связь между базовыми модулями осуществляется коммутационной пространственной системой, которая реализована внутри базового модуля. Коммутационная система базового модуля реализует не только информационные связи между компонентами базового модуля, а также и информационные связи между БМ. Подобная организация коммутационной структуры обеспечивает однородность вычислительной системы (система состоит только из базовых модулей) и высокую технологичность организации вычислительных процессов. Обмен информацией между объектами, принадлежащими разным базовым модулям, организуется точно так же, как и обмен между объектами внутри базового модуля. Многопроцессорная система адаптирована к структуре задачи и, с одной стороны, обеспечивает высокую скорость обработки информации за счет высокой удельной производительности и конвейерной обработки данных, а с другой стороны, реализует фон-неймановский детерминизм вычислений и может быть основой для создания сверхвысокопроизводительных вычислительных систем.

**5. Базовые модули многопроцессорной системы с макрообъектной архитектурой.** На осно-

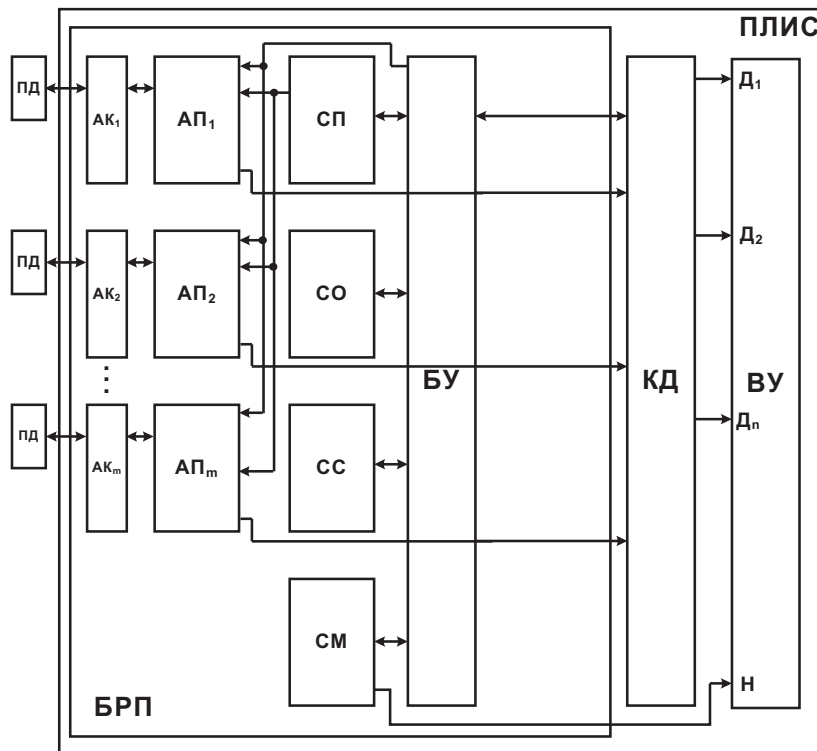


Рис. 4. Структура блока распределенной памяти



Рис. 5. Базовый модуль в конструктиве Евромеханика 6U



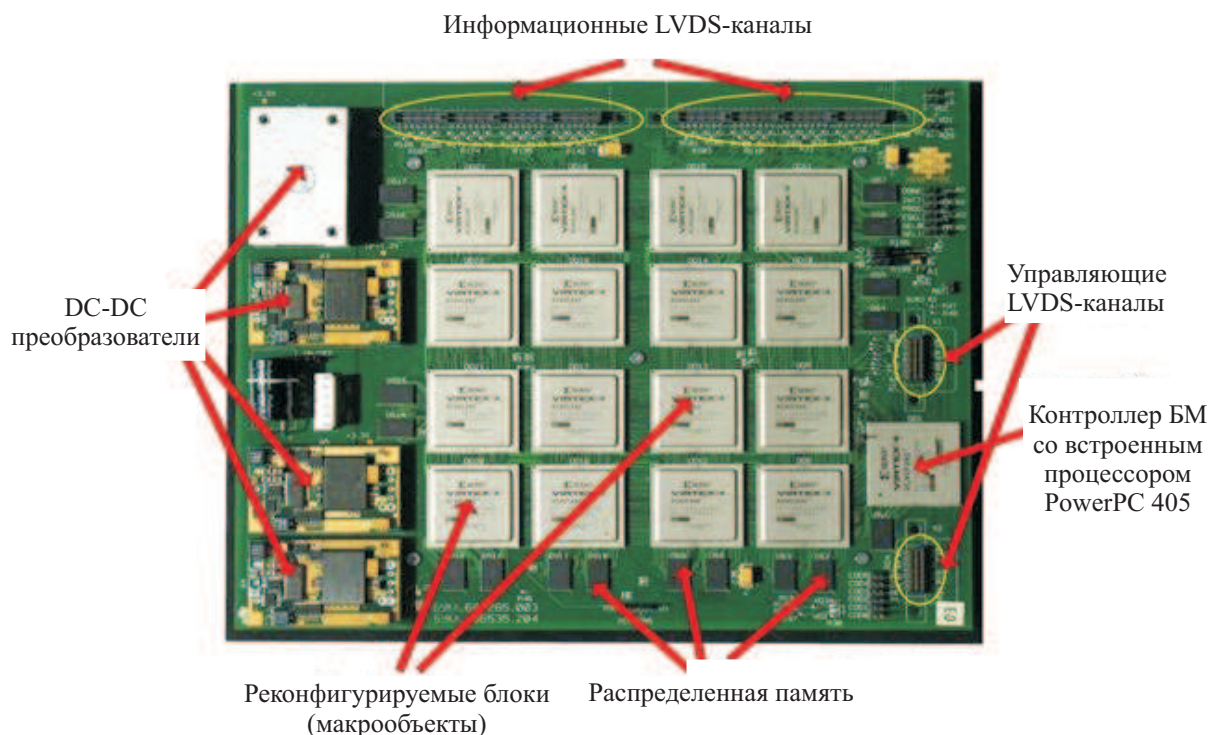


Рис. 6. Базовый модуль нового поколения

ве указанных принципов разработан и создан ряд базовых модулей, а также многопроцессорных систем.

Для решения задач цифровой обработки сигналов мониторинга связи был разработан малогабаритный базовый модуль в конструктиве Евромеханика 6U, представленный на рис. 5.

Данный модуль сопрягается с host-машиной по расширенной шине VME. Модуль построен с использованием ПЛИС серии Virtex II-Pro фирмы Xilinx, содержащих 10 миллионов эквивалентных вентилей (XC2VP100). Конструктив базового модуля обеспечивает возможность наращивания базового модуля с помощью соединения специальных разъемов зеркальной мезонинной конструкции, что обеспечивает возможность увеличения производительности в два раза, без привлечения дополнительных коммутационных устройств. Модуль имеет следующие характеристики:

Объем памяти, Гбайт	0,5
Производительность, Гфлопс	32
Частота ПЛИС, МГц	200
Частота платы, МГц	200
Потребляемая мощность, кВт	0,22

По государственному контракту по теме “Разработка технологии создания высокопроизводительных модульно-наращиваемых многопроцессорных вычислительных систем с программируемой архитектурой на основе реконфигурируемой элементной базы”, выполняемому в рамках федеральной целевой научно-технической программы “Исследования и разработки по приоритетным направлениям развития науки и техники на 2002–2006 гг.” разработан базовый модуль нового поколения на основе ПЛИС серии Virtex IV. Фотография БМ приведена на рис. 6.

Базовый модуль нового поколения имеет следующие характеристики:

Число ПЛИС Virtex IV, шт.	16
Объем памяти, Гбайт	1
Производительность, Гфлопс	50
Рабочая частота БМ, МГц	160
Габариты, мм	280×233

Базовый модуль нового поколения предназначен для построения модульно-наращиваемых многопроцессорных вычислительных систем (МНМС) в конфигурации до 16 БМ.

Разработана и создана модульно-наращиваемая многопроцессорная вычислительная система, содержащая четыре БМ, фотография которой представлена на рис. 7.

МНМС в составе четырех БМ имеет следующие характеристики:

Количество унифицированных базовых модулей	4
Число вычислительно реконфигурируемых блоков	64
Распределенная память, Гбайт	4
Производительность пиковая, Гфлопс	200
Рабочая частота, МГц	160
Потребляемая мощность, КВт	0,9
Объем, дм <sup>3</sup>	40

**6. Заключение.** Результаты теоретических исследований и практического использования показывают, что многопроцессорные вычислительные системы с программируемой архитектурой и структурно-процедурной организацией вычислений:

- обеспечивают такую производительность, которая близка к пиковой производительности на любом классе решаемых задач;
- дают возможность программировать архитектуру, включая прямые каналы коммуникаций, наборы макроопераций, внутренний язык высокого уровня и структуру распределенной памяти;
- обеспечивают практически линейный рост производительности пропорционально числу параллельно функционирующих супертранспьютеров;
- обеспечивают за счет оригинальных аппаратно-программных средств и модульного принципа организации масштабируемость системы.

Все это открывает широкие перспективы для создания отечественных суперкомпьютеров, отвечающих, с одной стороны, потребностям обеспечения информационной безопасности нашей страны, а с другой стороны, конкурентоспособности на мировом рынке.

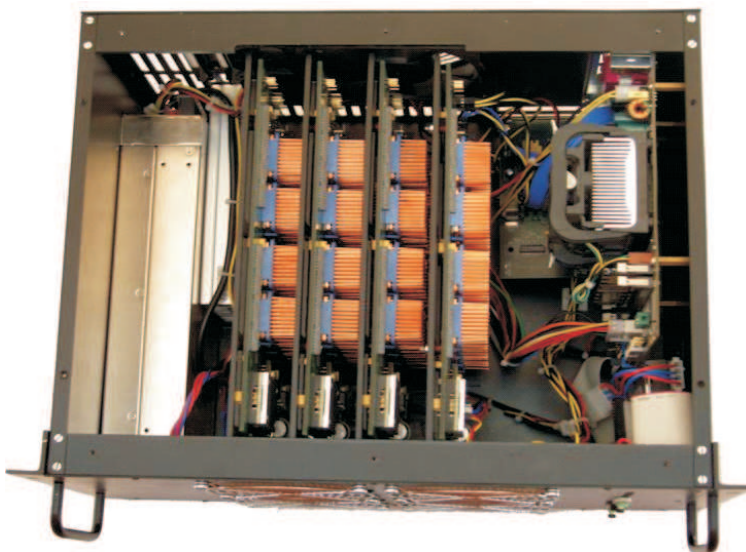


Рис. 7. МНМС в составе четырех БМ нового поколения

СПИСОК ЛИТЕРАТУРЫ

1. Аладышев О.С., Дикарев Н.И., Овсянников А.П., Телегин П.Н., Шабанов Б.М. СуперЭВМ: области применения и требования к производительности // Изв. ВУЗов. Электроника. 2004. № 1. 13–17.
2. Каляев А.В. Многопроцессорные вычислительные системы с программируемой архитектурой. М.: Радио и связь, 1984.
3. Каляев А.В., Левин И.И. Многопроцессорные системы с перестраиваемой архитектурой: концепции развития и применения // Наука — производству. 1999. № 11. 11–19.
4. Каляев А.В., Левин И.И. Модульно-наращиваемые многопроцессорные системы со структурно-процедурной организацией вычислений. М.: Янус-К, 2003.

Поступила в редакцию  
27.02.2007