

Ю.С. Яковлев, **Б.М. Тихонов**, Е.В. Елисеева

Компьютерная система типа «Процессор–в–памяти» с модифицированной кольцевой шиной

Предложена архитектурно-структурная организация компьютерной системы типа «Процессор–в–памяти» на одном кристалле с кольцевой шиной. Путем разбиения коммутационной шины на сектора и с помощью оригинального метода управления обеспечивается повышение производительности предложенной системы в сравнении с существующими системами с кольцевой шиной, краткий анализ которых приведен в статье.

The architecturally-structural organization of a computer system of a «Processor–in–memory» type on one crystal with the ring bus is suggested. Due to the breaking up of an interconnect bus on sectors and an original management method the productivity of the suggested system increased in comparison with the existing systems with the ring bus, the short analysis of which is presented in the article.

Запропоновано архітектурно-структурну організацію комп'ютерної системи типу «Процесор–в–пам'яті» на одному кристалі з кільцевою шиною. Шляхом розбиття комутаційної шини на сектори і за допомогою оригінального методу управління забезпечується підвищення продуктивності запропонованої системи в порівнянні з існуючими системами з кільцевою шиною, короткий аналіз яких наведено в статті.

Введение. Коммутация потоков информации внутри распределенной многопроцессорной системы существенно влияет на производительность системы из-за задержек сигналов при обмене информацией между процессорами и памятью, а также между блоками распределенной памяти. Стремление уменьшить эти задержки с помощью сложных коммутационных схем, обеспечивающих связи по принципу «каждый–с–каждым», часто не приводит к желаемым результатам, поскольку реализация таких коммутационных схем, способы управления ими и технология разводки большого количества входных и выходных линий сложны. С учетом этого применения простые коммутационные средства типа общей шины актуальны. Примером широко применяемой общей шины при построении средств вычислительной техники малой и средней производительности есть шина *PCI* и ее модификации. Однако такая шина имеет небольшую ширину для передачи данных (в модифицированном варианте до 64 бит) и ограничение ее длины из-за емкостных нагрузок, так что задержки сигналов при обращении процессора к памяти, подключенной в начале шины, могут существенно отличаться от задержек сигналов при обращении процессора к памяти, подключенной в конце шины. Это создает проблему синхронизации, что в итоге ограничивает частоту работы системы.

При построении высокопроизводительных вычислительных средств, таких как графические процессоры, графические ускорители и прочее, нашли применение структуры общей шины, получившие название кольцевых шин. Их применение согласно известным техническим решениям частично устраняет отмеченные недостатки.

Особенности применения кольцевой шины

Кольцевая структура шины проста для проектирования и реализации. Если кольцевая шина памяти используется внутри чипа, то, благодаря упрощению разводки (шина размещается по периметру кристалла), она удешевляет чип и менее подвержена нагреву. Как следствие, кольцевая шина может работать на более высоких тактовых частотах.

Кроме того, в набор операций кольцевой шины может быть включена операция мультиплексирования с временным уплотнением, так что данные могут передаваться по кольцевой шине в назначенные кванты времени, при этом множество кадров данных могут присутствовать на шине одновременно. Это позволяет выполнять параллельную передачу по кольцевой шине от разных каналов данных, таких как различные сектора данных. К тому же передаваемые данные проходят по всей кольцевой шине и возвращаются назад к исходному блоку, и при этом переданные данные могут чи-

таться узлом, проверяющим правильность их передачи.

Ширина кольцевой шины данных может быть уменьшена приблизительно в два раза в сравнении с разрядностью передаваемых данных за счет того, что половина разрядов данных может передаваться по шине в одном направлении, а другая половина – одновременно в другом – до соединения с первой половиной. Таким образом, применение кольцевых шин при построении распределенных компьютерных систем (КС) как с классической архитектурой, так и с архитектурой типа «Процессор–в–памяти» (PIM) актуально.

Анализ существующих КС с кольцевой шиной

Кольцевые шины достаточно широко применяются в различных модификациях графических процессоров высокой производительности серий *ATI Radeon X1000, X1300, X1600, X1800, ATI Radeon HD 2000* и других, выпускаемых зарубежными фирмами [1–2]. Каждый из этих процессоров содержит сложный перекрестный переключатель записи (*write crossbar switch*), а также блоки кольцевой остановки (*ring stop*), размещенные на кольцевых шинах. При выполнении простейших операций чтения или записи данных информация дважды проходит через сложный перекрестный переключатель (*crossbar switch*) и не менее четырех раз – через блоки *ring stop*, увеличивая тем самым минимально возможное время доступа к памяти.

Системы флэш-памяти с кольцевой шиной [3] содержат по всей ее длине специальные интерфейсы для данных, что, с одной стороны, уменьшает паразитные емкости шины на каждом выделенном участке, а с другой – усложняет синхронизацию передачи данных по шине в распределенных квантах времени при мультиплексировании сигналов с временным уплотнением и вносит дополнительную задержку при распространении сигналов по всей длине шины. Кроме того, алгоритм процесса передачи данных между узлами с использованием кольцевой шины содержит не менее семи достаточно емких последовательных процедур, включая чтение адреса, чтение и запись дан-

ных, передачу данных по кольцевой шине и так далее (всего 13 команд), что ограничивает достижение высокой производительности системы с кольцевой шиной такого типа.

Устройство с перекрестным коммутатором и кольцевой шиной [4] имеет сложный и громоздкий алгоритм реализации коммуникации сообщений между множеством компонентов сети с кольцевой шиной, включающий девять достаточно трудоемких последовательных процедур (без учета циклических операций), что, естественно, ограничивает достижимые параметры производительности системы. Сами по себе кольцевые шины – полной длины, что также снижает скорость передачи сигналов из-за значительных паразитных емкостей шины. Кроме того, реализация используемого в этом устройстве коммутатора, выполненного по принципу «каждый–с–каждым», приводит к необходимости построения сложной переключающей матрицы для одновременной коммутации множества входных портов к портам выхода и создает серьезную технологическую проблему для размещения линий связи на кристалле.

Недостатки устройства, описанного в [5], определяются применением большого количества дополнительного оборудования и соответствующих программных средств, включая отдаленный КЭШ данных и отдаленные КЭШ тегов, контроллер узла с соответствующими дополнительными функциями, а также блоки памяти для хранения признаков (тегов) для локальной разделяемой памяти процессорного узла. В результате процедура выполнения операции чтения или записи в такой системе с поддержкой когерентности КЭШа громоздка и длительна, так как включает в себя процедуры:

- обращения к отдаленному КЭШ данных и КЭШ тегов;
- анализа состояния блоков памяти каждого процессора;
- формирования процессорами необходимых пакетов запроса, пакетов подтверждения запросов и ответа на запросы.

Другим решением с кольцевой шиной является система памяти, содержащая многоуровневый двухточечный (*point-to-point*) кольце-

вой интерфейс памяти [6]. Для такой системы памяти операция чтения или записи данных по запросу от хост-машины через первичный контроллер памяти может быть выполнена не менее чем за 10 тактов процессора. При этом должны быть выполнены различные команды (чтение, запись, пересылки адреса и данных и другое) не менее 20 раз. Кроме того, увеличение времени задержки сигнала при обращении к каждому добавленному к кольцевой шине модулю памяти пропорционально логарифму от общего количества модулей, что ограничивает масштабируемость системы памяти.

Таким образом, рассмотренные системы с кольцевой шиной имеют ограничение по быстродействию из-за временных задержек при прохождении сигналов до соответствующих входов модулей памяти при выполнении операций чтения или записи данных, а также из-за невозможности одновременного обращения к памяти за несколькими данными или для параллельной записи массива данных. При этом они также имеют ограничения наращивания емкости памяти и соответствующие технологические трудности, определяемые необходимостью размещения большого количества шин (сигнальных линий) внутри каждого модуля и системы памяти в целом. Таким образом, требуется новый подход к созданию средств коммутации внутри чипа, который, с одной стороны, отличался бы простотой, а с другой – малыми временными задержками при передаче информации как между процессорами, так и между банками памяти распределенной системы.

Новый тип архитектурно-структурной организации КС типа PIM с кольцевой шиной на одном кристалле

Схема блока КС типа PIM [7], проиллюстрированная на рис. 1, содержит кольцевую шину, состоящую из кольцевой шины данных и кольцевой шины управления, разделенную n управляемыми секторными разделителями на n секторов, системный контроллер, блок управления кольцевыми шинами и секторными разделителями, множество из N банков памяти (БП), разделенное на $(n - 1)$ секторных набо-

ров, блок служебных функций, внешние входы/выходы данных системы, внешние входы/выходы управляющих сигналов.

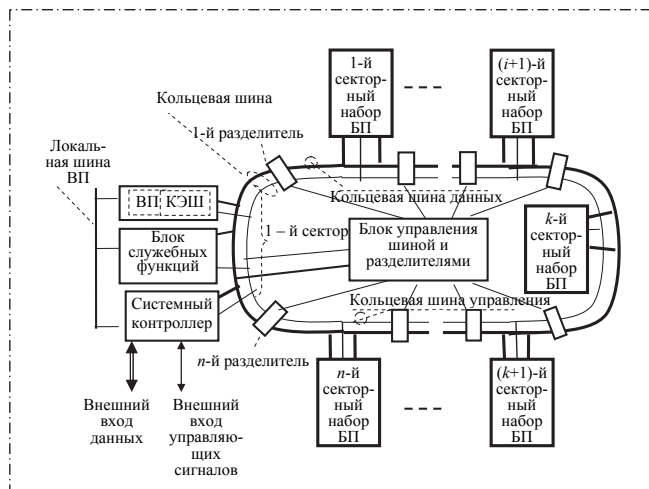


Рис. 1. Схема КС типа PIM с модифицированной кольцевой шиной

В состав КС типа PIM также входит ведущий процессор ВП с КЭШ-памятью, локальная шина ВП, а каждый секторный набор включает процессоры памяти (процессорные ядра – ПЯ) и контроллер секторного набора (рис. 2), объединенные локальной шиной.

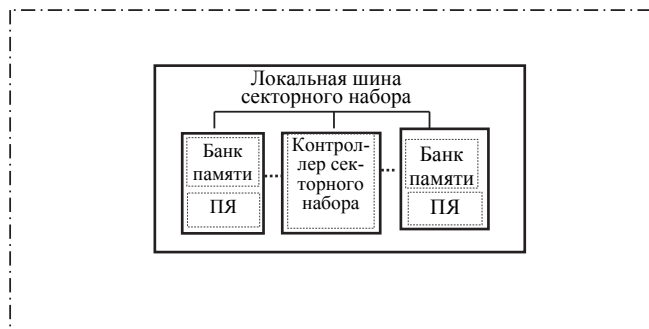


Рис. 2. Состав секторного набора БП КС типа PIM с кольцевой шиной

Системный контроллер управляет работой всей системы интеллектуальной памяти, обеспечивая ее применение в качестве:

- основной или дополнительной памяти любой компьютерной системы с реализацией классических функций памяти (запись, хранение и чтение информации);

- средства обработки типа «Процессор–в–памяти» («Processor-in-memory») вместе с имеющимися внутри системы устройствами памяти с реализацией функций параллельной обработки.

Набор основных функций, реализуемых этим контроллером, определяется управляющим пакетом, который поступает от хост-машины по локальной шине ВП на его соответствующие входы/выходы. Вариант структуры пакета, содержащего поля, согласно [8], имеет вид (см. рис. 3):



При этом приняты следующие обозначения: СДП_{оп} – спецификатор действия пакета, определяющий операции, которые должны быть выполнены объектом–адресатом после получения пакета; ПД₁ – поле значения параметров (данных), которые могут использоваться при выполнении текущего действия, или могут быть частью процесса, который переносит значения в следующие местоположения; КОП_{стр} – код строки явных операций, определяющий действие пакета; ПР_{реж} – поле признака, отражающего характер и последовательность действий в различных режимах работы системы памяти, в том числе – в качестве обычной памяти, а также в качестве «Процессора–в–памяти»; ПР_{рес} – поле признака использования ресурсов обработки информации собственной системы или дополнительных ресурсов за счет других систем, подключенных через соответствующие интерфейсы; КЗ_{сп} – коды запуска сервисных программ для выполнения функций: распределения памяти и размещения данных, разбиения задачи на параллельные фрагменты, формирования нового управляющего пакета, запуска библиотеки стандартных подпрограмм, загрузки данных и др.; КР_{кш} – код для управления режимами работы кольцевых шин (широковещательная передача по кольцевым шинам данных или команд, выделение секторных наборов БП и соответственно секторов кольцевых шин для параллельной обработки и др.); КК_{чип} – код для управления коммутацией внутри чипа (блока интеллектуальной памяти); КК_{м/чип} – код для управления передачей за пределы чипа; ЧТ_{в.адр} – поле адреса для чтения отдаленного, виртуального адресного обращения за пределы системы; ПД_{м.чип} – поле адреса для передачи блоков данных между различными секторными наборами БП; П_{доп} – дополнительные (вспомогательные) поля, необходимые для надежной транспортировки, обнаружения ошибок, маршрутизации и управления контекстом; ПД_{зак} – заключительный элемент пакета – поле, определяющее последующие действия после окончания выполнения данного пакета, например, могут быть созданы один или более дочерних пакетов.

Рис. 3. Вариант структуры управляющего пакета предложенной КС типа *PIM*

Пакет может изменяться по длине, таким образом обеспечивая эффективную обработку простых операций с маленькими пакетами и эффективным использованием полосы пропускания для перемещения больших блоков данных.

Управляющий пакет, помимо указанных выше полей, может содержать поля и признаки, отражающие характер и последовательность

действий, обеспечивающих настройку ресурсов системы (П_{настр}) перед запуском ее на решение конкретной задачи, а также, при необходимости, перестройку ресурсов в процессе работы.

Отметим, что состав полей управляющего пакета и разрядность каждого поля определяется типом и функциональными возможностями КС типа *PIM*, а также особенностями решаемых задач. При этом длина (количество разрядов) управляющего пакета может быть равна длине строки хранимых данных (например, 256 бит), и поэтому пакеты могут быть сохранены в строке широких регистров и обработаны непосредственно АЛУ КС с разрядностью на всю длину строки.

Секторный набор банков памяти (БП) (рис.2) предназначен для записи, чтения и хранения данных, а также для обработки информации внутри секторного набора БП согласно режиму работы типа «Процессор–в–памяти». При этом секторный контроллер управляет режимами работы банков памяти, а также выполняет анализ, формирование и выдачу сигналов загрузки ПЯ, освобождения банков памяти в каждом секторе и выдачу этих сигналов через соответствующие входы/выходы секторного контроллера на шину управления первого сектора кольцевой шины для дальнейшего их использования блоком управления шиной и секторными разделителями.

Секторные разделители предназначены для передачи данных и управляющих сигналов предыдущего сектора кольцевой шины на последующий сектор через соответствующие входы/выходы секторного разделителя как по часовой, так и против часовой стрелки. При соответствующем значении кода управляющего сигнала на управляющем входе секторного разделителя (например, код 00) передача информации по кольцевой шине через секторный разделитель запрещена, тем самым имеется возможность блокировки выбранного сектора кольцевой шины с двух сторон, обеспечивая обработку данных соответствующим секторным набором независимо от других.

Два соседних секторных разделителя выделяют сектор кольцевой шины, длина которого

меньше длины всей кольцевой шины в n раз, где n – количество секторов, при этом емкостная нагрузка на секторную часть кольцевой шины также уменьшается в n раз. При широкорешательной передаче сигналов по кольцевой шине емкостная нагрузка всей кольцевой шины также существенно уменьшена, так как каждый секторный разделитель при сквозном прохождении через него сигналов усиливает их, восстанавливая форму его импульса. Кроме того, возможность блокировки с помощью секторных разделителей каждого сектора кольцевой шины от влияния других оставшихся секторов позволяет реализовать параллельные вычисления в секторных наборах заблокированных секторов, что также существенно увеличивает производительность системы в целом. Таким образом, добавление новых секторов кольцевой шины вместе с секторными наборами БП не может существенно увеличить емкость кольцевой шины, при этом каждый ПЯ в добавленном секторном наборе «видит» шину только своего секторного набора, тем самым увеличивая пределы масштабируемости системы в целом.

Блок управления кольцевой шиной и разделителями предназначен для управления кольцевой шиной при передачах по ней информации, поступающей из блока служебных функций и системного контроллера при выполнении операции записи, либо из БП любого секторного набора при выполнении операции чтения и передачи полученных данных на первый сектор кольцевой шины и далее на соответствующие входы/выходы системного контроллера и блока служебных функций. При этом блок управления выполняет следующие функции:

- фиксацию (например, в виде соответствующих индексов или табличного описания) компонентов и связей кольцевой шины с указанием количества секторных разделителей и секторов с выделением первого сектора кольцевой шины для связи с системным контроллером и блоком служебных функций, а остальных секторов – с соответствующими секторными наборами БП;

- управление широкорешательным режимом передачи, когда все секторные разделители открыты для передачи по кольцевой шине информации от каждого предыдущего сектора к каждому последующему и наоборот;

- управление адресным режимом передачи информации, когда открыты секторные разделители только по пути передачи информации либо из одного секторного набора БП в другой, либо из секторного набора БП в первый сектор кольцевой шины и далее на соответствующие входы блоков системного контроллера и блока служебных функций;

- управление режимом секторной блокировки, когда один или несколько секторов кольцевой шины в соответствии с адресной зоной секторных наборов БП блокируются секторными разделителями от других секторов шины, предоставляя тем самым возможность выделенным (заблокированным) секторным наборам БП параллельно выполнять приписанные им участки алгоритмов.

При реализации указанных функций блок управления кольцевой шиной и разделителями использует следующие поля управляющего пакета, выделенные системным контроллером и переданные на его входы/выходы через первый сектор кольцевой шины управления: $KP_{кш}$, $PP_{реж}$, $KZ_{сп}$, $KK_{м/чип}$, $PD_{м.чип}$, $P_{доп}$ и их комбинации.

Блок служебных функций КС типа *Р1М* предназначен для распределения ресурсов памяти в целом, включая распределение всего адресного пространства памяти секторных наборов БП, соответствующего размещения данных, разделение программ пользователя по секторным наборам на параллельно выполняемые части, а также для загрузки средств обработки секторных наборов БП (рис. 4).

При этом блок служебных функций может работать в двух основных режимах: режим 1 и режим 2. В режиме 1 блок осуществляет исходную настройку системных ресурсов для оптимального решения пользовательской задачи. В режиме 2 он осуществляет перераспределение ресурсов в процессе работы системы. При этом алгоритмы реализации перечисленных выше

функций распределения ресурсов с помощью блока служебных функций в общем случае основываются на способах, опубликованных в [9–14].

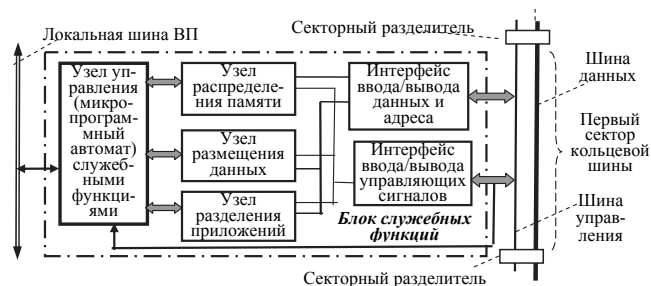


Рис. 4. Основные компоненты блока служебных функций

Однако, если все эти способы ориентированы на чисто программную реализацию с помощью специальных программных пакетов, то применительно к системе (рис. 1) каждая из функций распределения ресурсов делится на две части: одна часть, касающаяся анализа алгоритма решаемой задачи и определения его параметров (типы и частота встречаемости команд, количество одинарных и вложенных циклов, типы и характер взаимосвязи операторов: сильносвязанные, слабосвязанные и несвязанные и другие параметры) реализуется хост-машиной, и полученные параметры поступают через локальную шину ВП на входы/выходы блока служебных функций, где запоминаются в специально отведенной для этого области памяти, входящей в состав блока. Вторая часть реализации служебных функций, относящаяся непосредственно к алгоритмам распределения памяти, размещения данных и разделения приложений переложена на аппаратно-программные средства специально выделенных для этой цели узлов, входящих в блок служебных функций. Такое разделение позволяет оперативно реагировать на эффективность использования ресурсов при реализации режима 2 и в первую очередь – на баланс загрузки процессоров в каждом секторе БП, что положительно сказывается на повышении производительности всей системы. Наличие специального аппаратно-программного блока служебных функций в составе системы вместо реализации этих функций только известными программными способами позволяет в начале работы системы

или в любой другой момент ее функционирования настроить (либо подстроить) ресурсы системы под решение конкретной задачи, что существенно повышает производительность системы.

Работу КС типа *PIM* (рис. 1) можно кратко описать следующим образом. При первом запуске системы сначала обнуляют содержимое всех регистров (цепи сброса на рисунках не показаны). Затем выполняют настройку секторных разделителей кольцевой шины на заданный режим работы, и для режима 2 (режим *PIM*-системы) – настройку ресурсов под параметры реализуемой пользовательской задачи. Для этого от хост-машины через локальную шину ВП на соответствующие входы/выходы блока системного контроллера и блока служебных функций поступает управляющий пакет, сформированный либо хост-машиной, либо системой подобного типа. Пакет запоминается в этих блоках, где происходит анализ содержательной сущности каждого его поля. Если код поля $PP_{реж}$ указывает на обычный классический режим использования системы с реализацией только стандартных для памяти операций записи, хранения и чтения данных, системный контроллер памяти по кодам полей $PP_{реж}$, $PP_{рес}$, $KP_{кш}$, $СДП_{оп}$ вырабатывает соответствующий управляющий код, поступающий через первый сектор управляющей шины на соответствующий вход/выход блока управления кольцевой шиной, который вырабатывает кодовые комбинации 01 (передача данных по кольцевой шине через секторный разделитель в одном направлении, например, по часовой стрелке) либо 10 (передача против часовой стрелки), поступающие на управляющие входы соответствующих секторных разделителей. При кодовой комбинации 00 любая передача данных через кольцевой разделитель блокируется. Таким образом, обеспечена возможность как широкоэвентальной передачи информации по кольцевой шине данных и кольцевой шине управления в двух направлениях, так и возможность передачи информации к заблокированному (выбранному) сектору, например, с целью записи данных и управляющей инфор-

мации в БП выбранного секторного набора. Информация для записи (коды адреса и данные) и соответствующие управляющие сигналы поступают либо с входов/выходов системного контроллера, либо с соответствующих входов/выходов блока служебных функций через первый сектор кольцевой шины. При этом код адреса и данные могут быть также, при необходимости, обозначены идентификаторами поля ПД₁ управляющего пакета.

Информация, необходимая для записи в БП выбранного сектора (рис. 1), в соответствии с кодом адреса поступает через первый сектор кольцевой шины по кольцевой шине данных на соответствующие входы БП (данные) и соответствующие входы контроллера секторного набора (сигналы управления), который формирует сигналы, необходимые для выполнения операций записи данных, поступающих на соответствующие входы выбранного банка памяти. Операцию чтения данных из любого банка памяти любого секторного набора БП выполняют аналогично операции записи, при этом считанные данные по той же цепочке (только в обратном порядке) могут быть переданы либо на хост-машину через локальную шину ВП, либо на внешние устройства через системный контроллер и внешние входы/выходы.

При применении предложенной КС типа РИМ в качестве эффективного средства обработки информации по месту ее хранения, т.е. в режиме работы РИМ-системы («Процессор–в–памяти»), сначала осуществляют исходную настройку ресурсов на решаемую пользователем задачу. Для этого через локальную шину ВП на соответствующие входы/выходы блока служебных функций (рис. 1) поступает информация о параметрах решаемой пользовательской задачи (типы циклов и их количество, требуемых операций и их частоте встречаемости, типах взаимосвязи операторов и другое), запоминающаяся на соответствующих регистрах этого блока. Эта информация заранее готовится хост-машиной, используя для этого соответствующий управляющий пакет. После этого по той же цепи от хост-машины на локальную шину ВП поступает управляющий пакет, коды

полей которого ПР_{реж}, ПР_{рес}, КЗ_{сп}, ПД_{зак} записываются в соответствующие регистры блока служебных функций, а коды полей ПР_{реж}, КР_{кш}, КК_{чип}, КК_{м/чип}, ПД_{м.чип}, П_{доп}, ПД_{зак} СДП_{оп} – в системный контроллер. В зависимости от значения кода поля КЗ_{сп} процессы распределения памяти, размещения данных и разделения приложений могут осуществляться с помощью специальных сервисных программ, реализованных аппаратно-программным способом в соответствующих узлах либо интуитивно самим пользователем, используя для этого хост-машину или через системный контроллер. Результаты выполнения указанных служебных функций представляются в виде таблиц в соответствующих узлах:

- распределения всего адресного пространства памяти по секторным наборам и БП каждого набора;
- идентификаторов размещения данных по БП каждого секторного набора;
- идентификаторов разделения приложений по БП секторного набора.

Затем данные таблиц и разделенные по БП фрагменты приложений передаются в соответствующий блок служебных функций. Разделенные фрагменты приложений, таблицы распределения памяти, размещения данных по банкам памяти передаются по соответствующим связям в КЭШ-память ведущего процессора и в соответствующие БП всех секторов.

На основе анализа поля СДП_{оп} системный контроллер памяти вырабатывает управляющие сигналы инициализации процессов обработки информации, которые с его выхода через первый сектор кольцевой шины управления поступают на соответствующий вход ведущего процессора ВП, а также по кольцевой шине управления через секторные разделители – на вход контроллера каждого сектора, управляющие сигналы с которого поступают на входы банков памяти, запуская в работу соответствующие процессоры (ПЯ) банков памяти. Так как в предлагаемом устройстве имеется возможность блокировки любых выбранных секторов кольцевой шины от влияния других с помощью секторных разделителей, то обработка фраг-

ментов пользовательских задач в выбранных секторах может осуществляться одновременно, что повышает производительность системы. Результат обработки фрагмента задачи в каждом БП по соответствующим связям поступает на секторный контроллер и через выход БП на соответствующий сектор кольцевой шины данных (рис. 1), и далее через секторные разделители – на первый сектор кольцевой шины данных, а с него через системный контроллер – на внешний ввод/вывод системы памяти к внешнему устройству или внешней системе.

После выполнения приписанной каждому сектору фрагмента задачи, контроллер секторного набора выдает на соответствующий сектор кольцевой шины управления сигнал об окончании работы, который через соответствующие входы/выходы секторного разделителя поступает на вход блока управления шиной и разделителями, а также на вход блока служебных функций, инициируя процесс перераспределения ресурсов системы. При этом остальные секторные наборы банков памяти одновременно могут обрабатывать информацию, используя блокировку соответствующих секторов.

Заключение. Распределенная компьютерная система типа «Процессор–в–памяти» (*PIM*-система), выполненная на одном кристалле, вследствие особенностей архитектурно-структурной организации может обеспечить широкую полосу пропускания по каналу процессор–память. Однако при реализации таких возможностей коммутационная среда, обеспечивающая связь между процессорами и банками памяти, а также непосредственно между процессорами, не должна вносить существенных задержек при передаче по ее цепям информации и тем самым – не снижать эффект использования широкой полосы пропускания. Применение сложных коммутационных систем, построенных по принципу «каждый-с-каждым» не является оптимальным решением, поскольку при этом возникают серьезные проблемы технологического характера, связанные с размещением на кристалле большого количества пересекающихся проводников. С этой точки зрения применение общей шины (в современных ее модификаци-

ях) ввиду простоты и достаточно высокой скорости передачи информации особенно привлекательно, однако ввиду увеличения емкостной нагрузки при увеличении ее длины, такая шина ограничивает пределы масштабирования и скорость передачи данных между элементами *PIM*-системы, снижая тем самым ее производительность. Применение кольцевой общей шины в известных распределенных компьютерных системах частично снимает эту проблему, однако реализация операций с памятью вследствие особенностей архитектурно-структурной организации таких систем с кольцевыми шинами требует большого количества тактов процессора. Предложенный авторами модифицированный вариант кольцевой шины, разделенной с помощью секторных разделителей на n секторов с подключенными к каждому сектору ($n - 1$) секторных наборов БП вместе с процессором памяти (процессорным ядром – ПЯ) обеспечивает возможность одновременной работы всех секторных наборов. При этом обработка и передача информации в каждом секторном наборе происходит существенно быстрее, так как для каждого из них паразитная емкость кольцевой шины уменьшена в n раз в сравнении с традиционной кольцевой шиной. Тем самым снимаются пределы масштабирования системы, так как компоненты каждого секторного набора «видят» паразитную емкость шины только своего сектора.

1. Пахомов С. Современные видеокарты для ПК // КомпьютерПресс, 2005. – № 12. – <http://www.compress.ru/article.aspx?id=15082&iid=713>
2. Чеканов Д. Radeon X1300, Radeon X1600 и Radeon X1800: тест. – <http://www.thg.ru/graphic/20051006/index.html>
3. Pat. App. Pub., 2006031593 (A1), US, G06F15/16. Ring bus structure and its use in flash memory systems. / Sinclair Alan W. // 2006. – 38 с.
4. Pat., 6,728,206, US, 370/222, 370/423; EU, H04L 12/56; H04L 12/427; H04L. Crossbar switch with communication ring bus / Carlson, John R. // 2004. – 19 с.
5. Pat., 6,253,292, US, 711/146; 709/218; 711/148; 711/E12.025, EU, G06F 12/08; G06F 012/00. Distributed shared memory multiprocessor system based on a unidirectional ring bus using a snooping scheme. / Jhang; Seong Tae, Jhon; Chu Shik, Kim; Hyung Ho. // 2001. – 22 с.

Окончание на стр. 88

6. *Pat.*, 6,658,509, US, 710/100; 370/223; 710/300, EU G06F 13/42; G06F 013/00; G01R 031/08. Multi-tier point-to-point ring memory interface / Bonella; Randy M., Halbert; John B. // 2003. – 19 с.
7. *Пат.* 57629 А Україна, МПК (2006) G06F 15/16, G06F 13/42. Інтелектуальна розподілена система пам'яті з кільцевою шиною / Палагін О.В., Яковлев Ю.С., Тихонов Б.М., Єлісеєва О.В.; ІК ім. В.М. Глушкова НАН; Заявл. 16.07.2010; Опубл. 10.03.2011, Бюл. № 5/2011. – 41 с.
8. *Елісеєва Е.В.* Структура и функции управляющего пакета интеллектуальной системы памяти // Комп'ютерні засоби, мережі та системи. – 2009. – № 8. – С. 130–137.
9. *Яковлев Ю.С., Елісеєва Е.В.* Оптимальное распределение ресурсов сложной компьютерной системы как важный фактор для оптимизации вычислительного процесса // Пр. міжнар. симп. «Питання оптимізації обчислень (ПОО–ХХХV)». – К.: ІК ім. В.М. Глушкова НАН України, 2009. – Т. 2. – С. 440–445.
10. *Яковлев Ю.С., Тихонов Б.М.* Об оптимизации размещения данных в PIM-системе // Математичні машини і системи. – 2006. – № 3. – С. 24–35.
11. *Яковлев Ю.С., Елісеєва Е.В.* Математическая модель и стратегия распределения приложений для интеллектуальной памяти распределенных компьютерных систем // Математичні машини і системи. – 2009. – № 4. – С. 3–17.
12. *Яковлев Ю.С., Елісеєва Е.В.* Основные принципы и методика распределения приложений в сложных компьютерных системах типа «Процессор–в–памяти» // УСиМ – 2009. – № 6. – С. 56–63.
13. *Елісеєва Е.В.* Метод распределения приложений для оптимизации вычислений в компьютерной системе типа «Процессор–в–памяти» // Пр. міжнар. симп. «Питання оптимізації обчислень (ПОО–ХХХV)». – К.: ІК ім. В.М. Глушкова НАН України, 2009. – Т. 1. – С. 227–231.
14. *Елісеєва Е.В.* Реализация служебной функции средств поддержки вычислительного процесса интеллектуальной памяти компьютерных систем // Інформаційні технології та комп'ютерна інженерія. – 2009. – № 3. – С. 43–47.

Поступила 24.12.2010

Тел. для справок: (044) 526-3207 (Київ)

E-mail: jakus@bigmir.net, evo55555@ukr.net

© Ю.С. Яковлев, Б.М. Тихонов, Е.В. Елісеєва, 2011