

КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

E.V. Yelisyeyeva

STRUCTURE AND FUNCTIONS OF A CONTROL PACKET OF INTELLECTUAL SYSTEM OF MEMORY

Structure and functions of a control packet of intellectual system of memory and its component fields are offered.

Запропоновані структура і функції керуючого пакету інтелектуальної системи пам'яті (ІСП) і його складових полів.

Предложены структура и функции управляющего пакета интеллектуальной системы памяти (ИСП) и его составляющих полей.

© Е.В. Елисеєва, 2009

УДК 681.324

Е.В. ЕЛИСЕЕВА

СТРУКТУРА И ФУНКЦИИ УПРАВЛЯЮЩЕГО ПАКЕТА ИНТЕЛЛЕКТУАЛЬНОЙ СИСТЕМЫ ПАМЯТИ

Введение. Интеллектуальная система памяти (ИСП) компьютера – это система памяти, совмещающая функции хранения и обработки информации за счет перенесения от основного процессора части вычислений непосредственно в память, где размещено множество элементов обработки информации. ИСП часто определяют как “процессор-в-памяти” (“Processor-in-memory”) или PIM-системы по основному признаку – наличие в БИС памяти процессорных элементов. Благодаря особенностям архитектурно-структурной организации ИСП, построенные на их основе компьютерные системы (КС), имеют ряд преимуществ по сравнению с КС с классической архитектурой, особенно, когда речь идет о решении задач в Petaflops-ном диапазоне при массовом обращении к памяти за данными и широком распараллеливании вычислительного процесса [1, 2].

За рубежом создано несколько архитектур КС на базе интеллектуальной памяти. Среди них наиболее известны [1–3]: Active Pages (“активная страница”) Data Intensive Architecture – DIVA (интенсивная архитектура данных), архитектура IRAM (интеллектуальная оперативная память), VIRAM (векторная IRAM), архитектура с потоком памяти пользовательского уровня (User-Level Memory Thread – ULMT) и др. Далее приведены особенности отдельных ИСП и их эффективность использования согласно [1 – 3].

DIVA – уникальна для поддержки нерегулярных приложений, включая умножение разреженной матрицы и отслеживание указа-

теля. Моделирования показали, что такие типы нерегулярных приложений обеспечивают ускорение от 1.6 до 30 раз, когда они выполнены на архитектуре DIVA. Подход к интегрированию элементарных процессоров с системой обычной памяти при сохранении ведущего процессора делает архитектуру DIVA близко связанной с проектом Active Pages. Однако, Active Pages интегрирует большое число элементарных процессоров в системе памяти, в то время как DIVA содержит меньше, но более сложных процессоров. Еще одно отличие – Active Pages ограничивает совместное использование связи между элементарными процессорами в системе памяти, тогда как DIVA содержит специализированную сеть, чтобы поддержать коммуникацию между чипами PIM.

Интеллектуальная оперативная память (IRAM) состоит из процессора и нескольких банков динамической оперативной памяти. Миграция динамической оперативной памяти на процессорную подложку значительно увеличивает полосу пропускания памяти и понижает время ожидания памяти процессором. Моделирование показывает, что архитектура VIRAM в состоянии вычислить переходное выражение ориентированного графа в плотном представлении более чем в два раза быстрее, чем на рабочей станции Intel P4 1.5 ГГц.

VIRAM – это архитектура IRAM с применением векторного процессора. Он в состоянии выполнить идентичные операции на всех элементах вектора или массива параллельно. Поскольку векторные вычисления чрезвычайно ценны для многих мультимедийных приложений, система VIRAM главным образом предназначена для потоковых или мультимедийных приложений в реальном времени. Пиковая полоса пропускания памяти архитектуры VIRAM – 6.4 ГВ/с, что в 5–10 раз выше, чем у большинства КС с классической архитектурой. Как IRAM, так и VIRAM архитектуры предложены как возможные системные архитектуры интеллектуальной памяти для поддержки мультимедийных и встраиваемых приложений.

Архитектура с потоком памяти пользовательского уровня (User-Level Memory Thread – ULMT) встраивает дополнительную вычислительную мощность в контроллер памяти, исключая объединение динамической оперативной памяти (DRAM) и логики обработки на одной подложке. Это позволяет использовать коммерческую БИС память. ULMT выполняет предвыборку корреляции в поддержку приложения, работающего на основном процессоре. Предвыборка корреляции использует предыдущие последовательности обращений к памяти, чтобы предсказать будущие доступы к памяти и предварительную выборку этих адресов в кэш процессора. Чипы динамической оперативной памяти остаются неизменными. Моделирование показало, что архитектура ULMT может обеспечить для некоторых приложений до 58 % ускорения по сравнению с реализацией этих приложений на классических КС.

Image-архитектура поддерживает приложения, показывающие высокий параллелизм данных. Image включает несколько арифметических блоков, размещенных в отдельный чип от чипов памяти. Такое разделение позволяет транзакции чтения и записи, инициализированные ведущим процессором, эффективно

перекрывать с вычислением в системе памяти. Image может значительно превзойти по быстродействию VIRAM для тех приложений, которые оцениваются существенно большим значением отношения количества операций на количество доступов к памяти.

Другая интеллектуальная архитектура системы памяти, названная “Модуль арифметики, памяти и интерфейса” (Memory Arithmetic Unit and Interface – MAUI) объединила в себе некоторые особенности архитектур Active Pages, DIVA и архитектуры ULMT [3]. Подобно ULMT (в отличие от Active Pages и DIVA) архитектура MAUI содержит логику обработки на том чипе, где размещен контроллер памяти.

Моделирование MAUI показало, что приложения могут быть реализованы на этой архитектуре приблизительно на 80 % быстрее, чем при использовании классической КС.

Основой потокового вычислительного процесса ИСП-архитектуры, является управляемое сообщением вычисление. Его выполняют соответствующие механизмы интерпретации “на модуле”, быстро опознающие последовательность операций, которые должны быть выполнены, и строки данных, над которыми выполняется операция. Поддержка управляемого сообщением вычисления опирается на концепцию пакетов, предложенных применительно к ИСП в проекте DIVA. Структура и функциональное назначение пакета и его отдельных компонентов (полей) для каждой PIM-системы являются специфическими, так как проектируемые PIM-системы ориентировались на конкретную задачу (в лучшем случае – на класс задач), и поэтому такого рода информация о пакетах отсутствует в отечественных и зарубежных источниках. Однако роль управляющего пакета в организации вычислительного процесса ИСП чрезвычайно велика: пакет, вырабатываемый хост-машиной, на системном уровне фактически запускает и управляет процессами как внутри чипа, так и за его пределами при взаимодействии с другими аналогичными чипами, дополнительной памятью и внешними устройствами. Поэтому в данной работе рассматривается структура и функции управляющего пакета, который в своей основе может быть использован в качестве исходного варианта при проектировании ИСП.

Структура и функции управляющего пакета ИСП. Как показал зарубежный опыт, создание интеллектуальных систем памяти является реальным при современном уровне развития интегральной технологии, а их применение вполне оправданным. Так как построенные на их основе компьютерные системы при решении определенного класса задач обеспечивают выигрыш в производительности от нескольких процентов до нескольких тысяч раз по сравнению с КС с классической архитектурой. Этот выигрыш становится особенно ощутимым, если речь идет о задачах, поддающихся широкому распараллеливанию при массовом обращении к памяти за данными и при требованиях к скорости обработки информации в Petaflops-ном диапазоне. Однако эти положительные качества интеллектуальных систем памяти, объясняемые, прежде всего, особенностями их архитектурно-структурной организации, снижаются следующими факторами,

которые в основном продиктованы интегральной технологией создания БИС памяти. При этом в качестве очевидных факторов выступают:

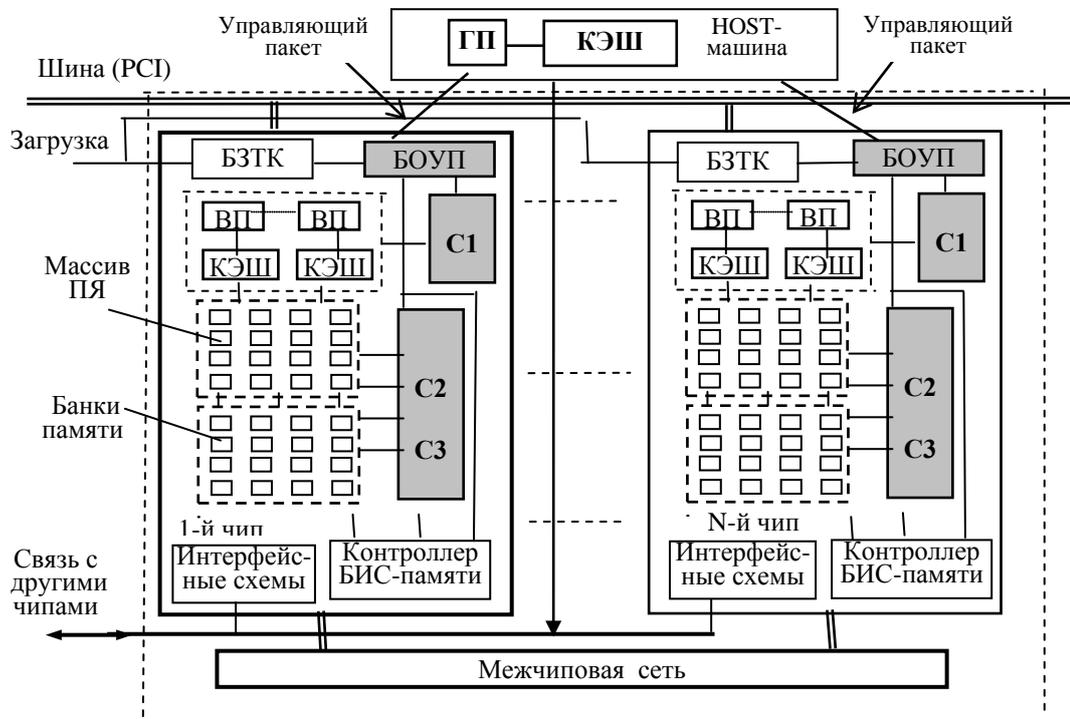
– фиксированное количество элементов памяти и логических элементов на кристалле, ориентированных на решение конкретной задачи (в лучшем случае класса задач), что препятствует оптимизации интеллектуальной памяти по емкости памяти и соответственно по стоимости;

– необходимость доработки существующих системного и прикладного программного обеспечения, что продиктовано особенностями их архитектурно-структурной организации и характером решаемых задач [4].

Эти факторы в основном препятствовали широкому распространению и использованию ИСП в различных сферах человеческой деятельности. Появление программируемых логических схем (ПЛИС) существенно не изменило создавшееся положение, поскольку они, во-первых, не исключили второй фактор. Во-вторых, обеспечивая возможность перестройки компонентов архитектуры ИСП, они при этом из-за особенностей технологии их создания снижают производительность системы в целом. С этой точки зрения наиболее перспективным является подход к оптимизации архитектуры БИС памяти путем выбора для решения конкретной задачи находящихся на кристалле заведомо введенных избыточных ресурсов [5]. Такой подход фактически снимает влияние первого фактора на серийноспособность и широкое использование ИСП. На рис. 1 показана укрупненная структурная схема ИСП такого типа.

В соответствие с этим в данной работе приводится описание структуры управляющего пакета и функций его полей с ориентацией на реконфигурируемую ПИМ-систему широкого назначения, подробно рассмотренную в [5].

В общем случае пакет может быть представлен в виде кортежа $W = (A, B, D)$, где A – микросервер адресата, B – команда, D – список параметров. Микросервер адресата представляется виртуальной адресацией, которая отображается аппаратными средствами, чтобы адресовать узел, которому нужно поставить пакет. Компонент команды пакета, вместе с его параметрами, определяет действие, которое будет выполнено по прибытию пакета на объект адресата. Это может быть любая спецификация микро метода или явная последовательность кода, которая будет выполнена в микросервере адресата. Список параметров пакета содержит параметры любого вида, поддержанного системой. Интерпретация параметров может зависеть от приложения, команды пакета, его объекта адресата; параметр может использоваться, чтобы определить "продолжение" действия, инициализированного в микросервере адресата, такого как возвращение значения. Пакеты позволяют осуществлять разбиение или расщепление операций, сокращая тем самым общее время задержки информации для всей системы. Их эффективность существенно повышается при аппаратной поддержке реализуемых ими функций. На рис. 2 показан пример структуры пакета, содержащего поля, согласно вышеприведенному представлению пакета.



Обозначения: ГП – главный процессор (процессор хост-машины); ВП– ведущий процессор; ПЯ – процессорные ядра (процессоры памяти); БОУП – блок обработки управляющего пакета; БЗТК – блок загрузки и тестового контроля; КЭШ – память типа КЭШ; С1, С2, С3 – селекторы выбора ВП, ПЯ и банков памяти соответственно.

РИС. 1. Схема ИСП, иллюстрирующая способ реконфигурации архитектуры путем выбора с помощью селекторов требуемых ресурсов из избыточных



РИС. 2. Вариант структуры управляющего пакета ИСП

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 собственного кристалла или дополнительных ресурсов за счет других чипов, подключенных через соответствующие интерфейсы;

КЗ_{сп} – коды запуска сервисных программ для выполнения функций: разбиения задачи на параллельные фрагменты, разбиения алгоритма на гранулы верхнего уровня, формирования нового управляющего пакета, библиотеки стандартных подпрограмм, загрузки данных и тестирования чипа и др.;

КК_{чип} – код для управления коммутацией внутри чипа;

КК_{м/чип} – код для управления коммутацией межчиповой сети;

ЧТ_{в.адр} – поле адреса для чтения отдаленного, виртуального адресного обращения;

ПД_{м.чип} – поле адреса для передачи блоков данных между отдаленными чипами;

П_{доп} – дополнительные (вспомогательные) поля, необходимые для надежной транспортировки, обнаружения ошибок, маршрутизации и управления контекста;

ПД_{зак} – заключительный элемент пакета – поле, определяющее последующие действия после окончания выполнения данного пакета, например, могут быть созданы один или более потоков и посланы отдаленным (вне чипа) узлам. Иногда действие пакета приводит к созданию одного или более дочерних пакетов.

Применительно к предлагаемой системе памяти (рис. 1) управляющий пакет, помимо вышеуказанных полей, содержит поля и признаки, отражающие характер и последовательность действий, обеспечивающих настройку архитектурно-структурного образа системы (П_{настр}) перед запуском её на решение конкретной задачи, а также при необходимости перестройку системы в процессе работы, путем подключения соответствующего ведущего процессора (ВП), выбора типа и количества соответствующих банков памяти, а также выбора типа и количества процессорных ядер (ПЯ).

Все пакеты направляются к объекту – получателю или адресату, все адреса пакета – виртуальные адреса, которые могут идентифицировать индивидуальные переменные, блоки данных, структур, объектов или потоков, а также потоков ввода-вывода. Пакет может изменяться по длине, таким образом, обеспечивая эффективную обработку простых операций с маленькими пакетами и эффек-

тивным использованием полосы пропускания для перемещения больших блоков данных. Интерфейсы пакета (последовательно-параллельные) представлены каналами для активного управляемого сообщением вычисления между чипами.

Пакеты также могут использоваться для выполнения операций типа отдаленной загрузки или хранения, а также для вызова методов обработки на другой аналогичный чип системы памяти, перемещая тем самым с помощью пакетов методы обработки ближе к данным за меньшее время, чем данные (как обычно) передаются к ним. Элементы чипов (например, процессорные ядра), принимая переданные методы обработки информации, исполняют указанные действия, приводящие к локальной модификации их состояния, к порождению и отправке новых пакетов к другим чипам. Пакеты позволяют осуществлять разбиение или расщепление операций, обеспечивая тем самым допустимое время ожидания для всей системы. Их эффективность существенно повышается при аппаратной поддержке реализуемых ими функций.

Следует отметить, что состав полей управляющего пакета и разрядность каждого поля определяется типом и функциональными возможностями ИСП, а также особенностями решаемых с помощью ИСП задач. При этом длина (количество разрядов) управляющего пакета может быть равна длине строки хранимых в памяти данных (например, 256 бит), и поэтому пакеты могут быть сохранены в строке широких регистров и обработаны непосредственно широким арифметико-логическим устройством (АЛУ).

Управляющий пакет формируется хост-машиной и поступает на блок обработки управляющего пакета (БОУП), который может работать непосредственно с локальным контроллером памяти и механизмом переадресации, не участвуя в управлении потоком и в модулях выполнения (рис. 1).

БОУП анализирует все поля поступающего пакета и вырабатывает сигналы синхронизации и управляющие сигналы для селектора С1 выбора ВП, второго селектора С2 выбора банка памяти, третьего селектора С3 выбора ПЯ, а также сигналы синхронизации и управляющие сигналы для обеспечения работоспособности интерфейса внешней памяти, интерфейса ввода-вывода, контроллера памяти, а также других схем и узлов системы памяти.

БОУП может использовать пакет как основу для установления потока, который продолжит предназначенное действие. После завершения действия, один или более потоков могут быть созданы и посланы отдаленным (вне чипа) узлам. БОУП принимает спецификацию пакета возвращения от менеджера потока и запускает его через межчиповую сеть. Однако, для маленького подмножества простых операций пакета БОУП способен сам создавать результирующий пакет из первоначального пакета.

Выводы. Создание и применение интеллектуальных систем памяти является актуальным, так как построенные на их основе компьютерные системы обеспечивают выигрыш в производительности от нескольких процентов до нескольких тысяч раз по сравнению с КС с классической архитектурой при решении определенного класса задач, характерными особенностями которых являются:

массовое обращение к памяти, возможность широкого распараллеливания при требованиях Petaflops-ного диапазона производительности системы.

Определение структуры и функций управляющего пакета является чрезвычайно важным для проектирования интеллектуальной системы памяти и КС на её основе, так как управляющий пакет является основным инструментом на системном уровне организации потоковой обработки информации, управляя основными процессами внутри чипа и за его пределами при взаимосвязи между чипами, внешней памятью и различными устройствами ввода-вывода.

Предложенные структура и функции управляющего пакета могут быть использованы в качестве исходных при проектировании ИПС и компьютерных систем на их основе с потоковой обработкой информации, в частности при создании предложенной ранее в Институте кибернетики имени В.М. Глушкова НАН Украины интеллектуальной системы памяти с реконфигурируемой структурой.

1. Палагин А.В., Яковлев Ю.С., Тихонов Б.М. Основные принципы построения вычислительных систем с архитектурой “Процессор-в-памяти”(Processor-in-Memory) // Управляющие системы и машины. – 2004. – № 5. – С. 30 – 37.
2. Палагин А.В., Яковлев Ю.С., Тихонов Б.М., Перико И.М. Архитектурно-структурная организация компьютерных средств класса “Процессор-в-памяти” // Математичні машини і системи. – 2005. – № 3. – С. 3 – 16.
3. Teller Justin Stevenson. Performance characteristics of an intelligent memory system. – <http://www.ece.osu.edu/~tellerj/ms-thesis.pdf>.
4. Елисеєва Е.В., Яковлев Ю.С. О концепции построения программной среды PIM-систем // Управляющие системы и машины. – 2008. – № 4. – С. 58 – 67
5. Сергієнко І.В., Кривонос Ю.Г., Палагін О.В., Коваль В.М., Яковлев Ю.С., Тихонов Б.М. Інститут кібернетики імені В.М. Глушкова НАН України. Система пам'яті з інтеграцією функцій зберегання та обробки інформації на одному кристалі. Деклараційний патент на корисну модель. № 6259. G06F13/00, G06F12/00. 15.04.2005. Бюл. № 4. – 24 с.

Получено 05.03.2009