

# КОМП'ЮТЕРНІ ЗАСОБИ, МЕРЕЖІ ТА СИСТЕМИ

*E.V. Yelisyeyeva, Yu.S. Yakovlev*

## **THE CONCEPT AND PRINCIPLES OF CONSTRUCTION OF FORMATS AND SET OF COMMANDS OF HYPOTHETICAL PIM-SYSTEM.**

*The analysis of commands of some commercial PIM-systems is executed. Construction principles, base formats and the reduced set of commands of a hypothetical PIM-systems reflecting features of its architecturally-structural organization, defined by a level of development of electronic components and properties of realized algorithm are offered.*

*Виконаний аналіз команд деяких комерційних PIM-систем. Запропоновані принципи побудови, базові формати та скорочений набір команд гіпотетичної PIM-системи, які відображають особливості її архітектурно-структурної організації.*

*Выполнен анализ команд некоторых коммерческих PIM-систем. Предложены принципы построения, базовые форматы и сокращенный набор команд гипотетической PIM-системы, отражающие особенности её архитектурно-структурной организации, определяемые уровнем развития элементной базы и свойствами реализуемого алгоритма.*

© Е.В. Елисеєва, Ю.С. Яковлев,  
2008

УДК 681.324

Е.В. ЕЛИСЕЕВА, Ю.С. ЯКОВЛЕВ

## **КОНЦЕПЦИЯ И ПРИНЦИПЫ ПОСТРОЕНИЯ ФОРМАТОВ И НАБОРА КОМАНД ГИПОТЕТИЧЕСКОЙ PIM-СИСТЕМЫ**

**Введение.** PIM-системы по сравнению с компьютерными системами (КС), построенными по классическим принципам, имеют ряд преимуществ, особенно при реализации задач, требующих массового обращения к памяти и обладающих возможностями широкого распараллеливания алгоритмов их решения [1, 2]. Эти преимущества, в основном, определяются особенностями их архитектурно-структурной организации и выражаются в следующем:

– достижение высокой производительности системы в целом (более чем на порядок) за счет широкой полосы пропускания по каналу процессор-память и тем самым за счет оптимального согласования скорости работы процессора и памяти, которые размещены на одном кристалле;

– уменьшение потребляемой мощности, габаритов и веса PIM-системы при одинаковой производительности с КС и др.

Особенности PIM-систем с архитектурной точки зрения, прежде всего, представляются через систему команд и структуру их форматов, которые по мере развития интегральной технологии и возрастания степени интеграции БИС развиваются и усложняются, так как появляется возможность применять внутри БИС PIM-системы более сложные процессорные элементы с большей емкостью памяти и расширенными функциональными возможностями. Это достаточно убедительно подтверждается форматами команд коммерческих PIM-систем, созданных за рубежом в различные периоды времени.

Учитывая высокую актуальность проблемы создания и применения PIM-систем, в том числе в качестве приставки к ЭВМ и кластерам, а также возможность реализации такой приставки – ускорителя на базе ПЛИС, целесообразно разработать проекты форматов и систему команд, так называемой гипотетической PIM-системы (GPIM), особенности архитектурно-структурной организации которой определены современным состоянием микроэлектроники, а также расширенными функциональными возможностями, диктуемыми возрастающей сложностью пользовательских задач [3]. Естественно, что такую гипотетическую архитектуру можно использовать в качестве прототипа при разработке проектов PIM-систем различного назначения.

**Примеры форматов команд коммерческих PIM-систем**

*Форматы команд PIM-системы типа CRAM.* В первых PIM-системах типа CRAM (Computational-RAM) были использованы в качестве базовых процессоров памяти элементарные 1-битовые процессоры последовательного типа [4], что позволило создать БИС памяти с минимальными конструктивными и технологическими изменениями.

Все команды CRAM 32-разрядной ширины и имеют однородный RISC-подобный формат (рис. 1). 8-битовый OPCODE (код операции) представляет операцию (например, сложение), которая будет выполнена командой.

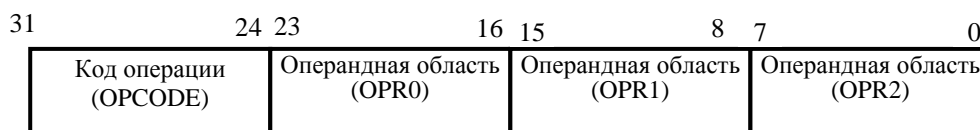


РИС. 1. Формат команды PIM-системы типа CRAM

Эта область (поле) указывает на адрес начала операции микрокоманды в памяти управления. 8-битовая ширина означает, что память управления может быть до 256 слов, хотя используется адресное картографирование.

В формате команды есть три 8-битовых операндных области, которые могут быть или адресами операндов в операции CRAM, или непосредственными значениями в командах для загрузки регистров CRAM-контроллера. Чтобы адресоваться к строке CRAM-памяти за пределами 256, операндные адреса расширены с помощью регистров расширения адреса.

*Форматы команд PIM-систем типа PIMLite* [5]. Коммерческая PIMLite имеет 16-разрядные команды: 6-битовый код операции, 5 адресных битов и 5 битов регистров источников данных. Так как используется 6 битов, чтобы закодировать код операции, то PIMLite может содержать 64 команды.

*Форматы команд PIM-системы IMAP.* Данная PIM-система типа IMAP (Integrated Memory Array Processor) ориентирована на обработку графических изображений [6]. Каждый чип содержит 64 процессорных элемента, интегрированных с 2 Мбайтами статической памяти типа SRAM. При этом полоса про-

пускания по каналу память-процессор составляет 1,28 Гбайт/с; используется 8-разрядный процессорный элемент (PE), так как пиксель, в основном, представляется и обрабатывается 8-разрядами данными. Формат команды IMAP показан на рис. 2.

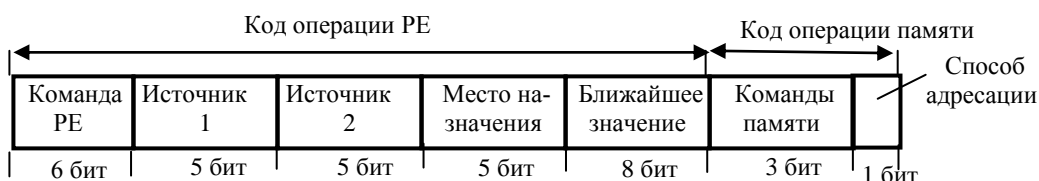


РИС. 2. Коды команды IMAP

Формат команды, представленный на рис. 2, содержит поля кода операции PE и поля кода операции доступа к памяти. Поэтому как PE, так и память могут выполнять свои операции одновременно.

Набор команд IMAP состоит из 37 инструкций, в том числе: 11 команд АЛУ, 8 команд поиска по таблице, 7 команд сдвига, 2 флаговые команды, 4 команды передачи внутри PE, 3 команды маски установки/сброса и 2 команды выдачи данных.

Код операции памяти образован из 3-байтового поля команды операции памяти и 1-битного поля режима (метода адресации). Набор команд обращения к памяти содержит две рабочие команды и две команды хранения (запоминания). При выполнении операции доступа к памяти старшая или младшая половина слова передается между блоком памяти и регистром данных памяти. Один бит способа адресации указывает на прямой или косвенный способы адресации.

**Базовые форматы команд чипа DIVA.** Чип DIVA (Data Intensive Architecture) – устройство памяти с вычислительными возможностями и аппаратными средствами коммуникации состоит из памяти емкостью несколько мегабайт и процессора [7 – 9]. Узлы на PIM-чипе совместно используют host-интерфейс и компоненту маршрутизации одного PIM (PIM Routing Component – PiRC), которая реализует маршрутизацию пакетов из чипа через PIM-to-PIM линию связи и определяет направление перемещения пакетов внутри чипа.

Чип DIVA содержит два информационных канала: 32-разрядный скалярный информационный канал, который выполняет операции подобно стандартным 32-разрядным целочисленным модулям, и 256-битовый информационный канал широких слов (WideWord), который выполняет параллельные операции с 8, 16, или 32-разрядными операндами. Система команд спроектирована таким образом, что оба информационных канала могут использовать те же самые коды операции и коды условия (состояния), генерируя большое функциональное перекрытие. Скалярный информационный канал представлен стандартной RISC архитектурой, дополненной несколькими специфическими функциями DIVA для координации с широкословным информационным каналом. Каждый информационный канал имеет свой собственный независимый универсальный регист-

ровый файл с 32 регистрами. Специальные команды обеспечивают прямые передачи между регистровыми файлами, минуя память.

Большинство команд скаляра DIVA используют формат с тремя полями, чтобы специфицировать два регистра источников и регистр места назначения (рис. 3). Для этих типов команд код операции вообще обозначает класс операций, таких как арифметика, а функция обозначает специфическую операцию, такую как сложение.



РИС. 3. Форматы арифметико-логических команд скалярного DIVA

Бит C указывает, выполнялась ли операция командой обновления кодов условия. Вместо второго регистра источника, 16-разрядное непосредственное значение может быть определено. Набор скалярной машинной команды включает типичные арифметические функции типа сложения, вычитания, умножения и деления; логические функции И, ИЛИ, НЕ, неэквивалентность, а также логические операции / операции арифметического сдвига.

Большинство команд DIVA WideWord подобно скалярным командам используют трехоперандный формат, чтобы установить два 256-битовых регистра источника и 256-битовый регистр адреса (рис. 4).

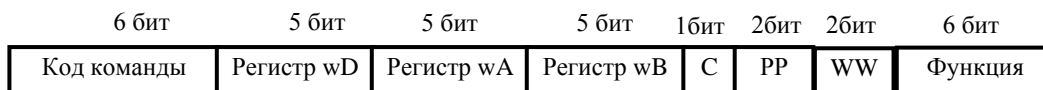


РИС. 4. Формат команд WideWord канала DIVA

Код операции, так же как и для скалярного канала, определяет класс операций, таких как арифметика, а функция обозначает специфическую операцию типа сложения или вычитания. Бит C указывает, выполнялась ли операция путем выполнения машинной команды обновления кодов условий. Поле WW указывает ширину операнда, позволяя данным WideWord, быть обработанными как упакованный массив объектов размером из восьми, шестнадцати или тридцати двух битов. Эта характеристика означает, что арифметико-логическое устройство WideWord может быть представлено как переменное (по ширине) количество параллельных АЛУ.

Поле PP указывает режим участия, вид выборочного выполнения подполя, которое зависит от состояния локальных и соседних кодов условия. Подполя, которые участвуют в условном выполнении данной машинной команды, формируются из кодов условия или регистра маски плюс 2-битовое поле участия машинной команды.

Состав набора команд WideWord включает (примерно) 30 команд, осуществляющих типичные команды арифметики, такие как сложение, вычитание и умножение, логические функции И, ИЛИ, НЕ, неэквивалентность и логические операции арифметического сдвига. Кроме того, имеются команды загрузки/хранения и команды передачи управления, которые используются для массово насыщенных взаимодействий между скалярным и WideWord информационными каналами. Некоторые специальные команды ИС включают перестановку, слияние и упаковку/распаковку.

Есть также несколько команд, которые являются особенно полезными для обеспечения эффективности операций поиска данных, например: ELO – закодировать крайний левый, CLO – очистить крайний левый, BA – переход по всем, BN – переход ни по одному.

#### **Форматы и набор команд гипотетической PIM-системы**

При построении системы команд целесообразно принять за основу следующие концептуальные положения, вытекающие из особенностей архитектурно-структурной организации PIM-систем:

1. Работа PIM-системы в следующих режимах: в режиме оперативной (основной) памяти процессора, реализующей классические функции (чтение, запись, хранение данных), в режиме дополнительной памяти, расширяющей емкость основной памяти, и в режиме среды обработки информации. Возможность работы с операндами большой разрядности (64 и более бит)

3. Одновременная обработка большого количества операндов, входящих в длинное слово, разрядность которого соответствует полной разрядности строки, принадлежащей банкам памяти, размещенным на чипе, которая считывается за одно обращение к памяти чипа.

4. Передача любого операнда длинного слова, считанного из памяти чипа, к любому процессорному элементу (процессорному ядру – ПЯ) за счет аппаратно-программных средств, обеспечивающих их коммутацию.

5. Обработка скалярных, векторных и векторно-скалярных данных.

6. Возможность использования в качестве ведущих процессоров (ВП) и ПЯ функционально насыщенных, достаточно сложных процессоров, в том числе специализированных.

7. Наличие помимо классических команд арифметики и управления дополнительных специфических команд, отражающих особенности организации вычислительного процесса PIM-системы, в том числе – согласно пп. 1–5.

8. Исходная настройка (выбор) и возможность динамической перестройки ресурсов: ВП и ПЯ, банков памяти и иерархической системы памяти в целом (как в реконфигурируемых системах). Распределение загрузки ВП и ПЯ при реализации алгоритма.

9. Управление вычислительным процессом с помощью управляющих пакетов, сформированных либо хост-компьютером, либо отдельным блоком, размещенном на одном кристалле с процессорами и памятью.

10. Масштабирование (наращивание мощности) системы.

В общем случае РИМ-система, как и любая другая компьютерная система, должна содержать стандартный набор команд, а также команды, отражающие специфику архитектурно-структурной организации и применения систем такого класса, которые должны быть сформированы в соответствии с вышеизложенными принципами.

Стандартный набор команд содержит команды арифметики, сдвига, условного/безусловного переходов, анализа и поиска по таблицам (например, виртуального адреса), сброса/установки маски, пересылок, ввода-вывода и др.

Стандартные арифметические команды могут иметь скалярный, векторный, или векторно-скалярные форматы.

Скалярный формат работает с двумя  $m$ -разрядными скалярами аналогично типичному RISC-микропроцессору. Операнды определяются маскированием  $m$ -разрядных слов в пределах регистров длинного  $N$ -разрядного слова ( $N = k \times m$ ;  $N \gg m$ ). Полученное в результате выполнения операции длинное слово записывается полностью в КЭШ. В соответствии с этим предлагается базовый 32-разрядный формат скалярных команд гипотетической РИМ-системы (рис. 5).

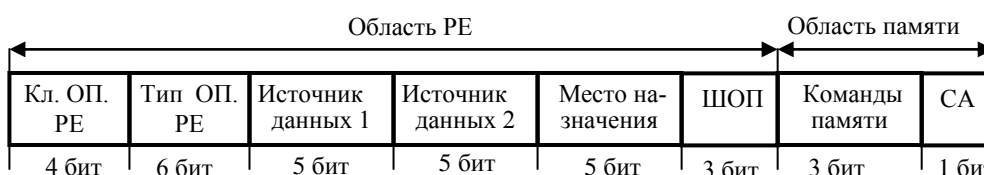


РИС. 5. Базовый формат скалярных команд гипотетической РИМ-системы

При этом приняты обозначения: Кл.ОП.РЕ – класс операций (арифметические, логические, пересылки, сдвиги и т. п. – всего 16 классов); Тип ОП.РЕ – тип операций (сложение, умножение, вычитание и т. п. – всего 64 типа); ШОП – ширина операнда (например, 8 бит, 16 бит, 32 бит и т. д. – всего 8 разновидностей); СА – способ адресации (прямая, косвенная).

Векторные команды подобны скалярным командам: они также используют трехоперандный формат, идентифицирующий два  $N$ -разрядных регистра источника и  $N$ -разрядный регистр адреса. Векторный формат позволяет все длинные слова данных обрабатывать одновременно. Например, при выполнении команды “векторного сложения” происходит суммирование двух векторов (например,  $N$ -разрядных) и сохранение полученного вектора той же разрядности. Наконец, команды вектор-скаляр оперируют с одним скалярным и одним векторным операндами. Например, команда “векторно-скалярного сложения” одновременно суммирует  $m$ -разрядный скаляр к каждому из  $k$   $m$ -разрядных слов вектора длиной  $N$  разрядов. Передача скаляра к каждому  $m$ -разрядному компоненту инфор-

мационного канала реализуется параллельно. Кроме того, может быть использована специальная сеть перестановки, которая позволяет длинным словам данных быть перестроенными.

Состав набора векторных команд так же, как и скалярных команд, включает типичные команды арифметики (например, сложение, вычитание и умножение), логические функции И, ИЛИ, НЕ, неэквивалентность и логические операции/операции арифметического сдвига. Кроме того, имеются команды загрузки/хранения и команды передачи управления, которые используются для массиво насыщенных взаимодействий между скалярным и векторным информационными каналами. Некоторые специальные команды включают перестановку, слияние и упаковку/распаковку. Перестановка сети поддерживает быстрое совмещение и преобразование данных в широких регистрах. Перестановочная сеть допускает любому  $m$ -разрядному полю данных регистра источника быть перемещенным в любое 8-битовое поле данных регистра адреса. Перестановка определяется вектором перестановки, который содержит индексы, соответствующие полям адреса векторного регистра.

Машинная команда слияния позволяет адресату широкого слова быть созданным из смешивания подполей от двух операндов источника, где источник для каждого подполя адресата, выбранный условием, определяется в машинной команде. Команда слияния осуществляет эффективную классификацию (сортировку). Команды упаковки/распаковки позволяют усечение/увеличение типов данных. Набор стандартных команд управления включает также команды *перехода, ветвления и объединения*. В свою очередь, переходы имеют разновидности: переход всегда, переход по нулю и переход по признаку – меньше чем. Целевой адрес перехода может быть относительным числом РС или вычисленный с использованием базового регистра, комбинированного со смещением. Чтобы поддерживать функциональные запросы, формат команды ветвления включает  $q$  бит для определения соединения, т. е. возвращает адрес команды, который должен быть сохранен. Формат перехода также включает 3-битовое поле условия (ПУС) для определения одного из восьми условий перехода: всегда, равный, не равный, меньше чем, меньше чем или равный, больше чем, больше чем или равный, или переполнение.

На рис. 6 показаны базовые форматы команд ветвления гипотетической РИМ-системы.



РИС. 6. Базовые форматы команд ветвления гипотетической РИМ-системы

В дополнение к стандартным командам, которые обеспечивают доступ в пределах данной области (кадра) памяти, в наборе команд могут быть команды, которые обеспечивают манипуляцию с глобальной памятью, системными запросами и другими командами (в частности, команды *обращения к системной таблице*, которая отслеживает виртуальные адреса, команда перемещения данных между различными кадрами в локальной памяти и др.). К таким командам также можно отнести команды запроса к исходным данным и к дочерним кадрам, также команды *обращения к глобальной памяти* (например, команды – загрузить и сохранить).

Чтобы в полной мере использовать преимущества PIM, набор команд должен обеспечивать возможности работать с множеством слов данных (например, *k m*-разрядных слов) одновременно. В этом случае набор команд помимо стандартных команд арифметики и команд управления должен содержать так называемую *команду перестановки*, чтобы соответствующие данные длинного слова можно было связать с конкретным микропроцессором [3].

Как было отмечено ранее, применение PIM-систем наиболее эффективно при решении задач, требующих массового обращения к памяти за данными. Мощные потоки данных пересылаются между узлами и блоками, размещенными на кристалле БИС, часто оказывая существенное влияние на производительность системы в целом. Поэтому проблема организации пересылок потоков информации внутри PIM-систем является весьма актуальной. Учитывая, что процедурам пересылок могут быть подвергнуты не только потоки данных, но и потоки команд, формат команд пересылок предлагается представить в виде рис. 7.

Кл. Пер.	Тип слова	Источник данных	Место назначения	ШОП	Код пересылки	Код коммутации
1 бит	4 бит	5 бит	5 бит	3 бит	6 бит	8 бит

РИС. 7. Базовый формат команды пересылок гипотетической PIM-системы

В представленном формате указаны поля: Кл.Пер. – класс пересылки определяет пересылку данных либо команд; Тип слова определяет скаляр, вектор, цепочку слов, матрицу и т. д. – всего 16 разновидностей; ШОП определяет ширину операнда (например, 8 бит, 16 бит, 32 бит и т. д. – всего 8 разновидностей); Код пересылки определяет разновидности путей пересылки внутри чипа между источниками и местами назначения, между различными чипами, между чипом и внешней памятью, между чипом и внешними устройствами и т. д. – всего 128 вариантов; Код коммутации управляет коммутаторами (селекторами) внутри чипа и определяет направления коммутации – всего 512 направлений.

К сожалению, в ограниченном объеме статьи не могут быть представлены все разновидности форматов команд PIM-системы. Авторы также допускают определенные изменения состава полей форматов и их размеров, что является естественным при разработке конкретного проекта. Тем не менее, предложен-



ные базовые форматы и принципы их построения могут быть использованы при проектировании архитектуры и структуры PIM-систем подобного типа.

1. 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.
2. Системы памяти с интеграцией функций хранения и обработки информации (PIM-системы) / А.В. Палагин, Ю.С. Яковлев, Б.М. Тихонов – Киев: 2006. – 33 с. – (Препр. / НАН Украины, Ин-т кибернетики им. В.М. Глушкова; 2006-3).
3. Сергієнко І.В., Кривонос Ю.Г., Палагін О.В., Коваль В.М., Яковлев Ю.С., Тихонов Б.М. Система пам’яті з інтеграцією функцій зберігання та обробки інформації на одному кристалі. – Деклараційний патент на корисну модель. № 6259. G06F13/00, G06F12/00. 15.04.2005. Бюл. № 4. – 14 с.
4. System Design for a Computational-RAM. Logic-In-Memory Parallel-Processing Machine. – [http://129.215.96.3:1234/~chrb/papers/cram\\_thesis.pdf](http://129.215.96.3:1234/~chrb/papers/cram_thesis.pdf)
5. Kyung-Hoon Kang Edward. Design and implementation of a multithreaded, wide word operation, processing in memory architecture / A thesis for the degree of master of science. – 2003. – P. 92. – [http://etd.nd.edu/etd\\_data/theses/available/etd-12032003-113051/unrestricted/KangEK12112004.pdf](http://etd.nd.edu/etd_data/theses/available/etd-12032003-113051/unrestricted/KangEK12112004.pdf)
6. Yamashita N., Kimura T., Fujita Y., Aimoto Y., Manabe T., Okazaki S., Nakamura K. and Yamashina M. A 3,84 GIPS Integrated Memory Array Processor with 64 Processing Elements and 2-Mb SRAM // IEEE J. Of Solid – State Circuits. – 1994. – 29, N 11. – P. 1336–1342.
7. The Architecture of the DIVA Processing-In-Memory Chip / Jeff Draper, Jacqueline Chame, Mary Hall et al. – <http://www.isi.edu/~mhall/diva-ics02.ps>
8. DIVA Chip Breaks Through Memory Wall. – <http://www.isi.edu/stories/31.html>
9. Draper Jeff., J. Tim Barrett, Jeff Sondeen, Sumit Mediratta, Chang Woo Kang, Ihn Kim And Gokhan Daglikoca. A Prototype Processing-In-Memory (PIM) Chip for the Data-Intensive Architecture (DIVA) System // J. of VLSI Signal Processing. – 2005. – 40. – P. 73–84. – [http://www.isi.edu/~jtb/papers/jvlsi\\_05.pdf](http://www.isi.edu/~jtb/papers/jvlsi_05.pdf)

Получено 25.03.2007