

Министерство образования и науки Украины  
Национальная академия наук Украины  
Институт кибернетики имени В. М. Глушкова  
Винницкий национальный технический университет

**Ю. С. Яковлев**

**ОДНОКРИСТАЛЬНЫЕ КОМПЬЮТЕРНЫЕ  
СИСТЕМЫ ВЫСОКОЙ  
ПРОИЗВОДИТЕЛЬНОСТИ**

**Особенности архитектурно-структурной  
организации и внутренних процессов**

**Монография**

Вінниця  
ВНТУ  
2009

УДК 004.27; 004.25; 004.382.2

Я47

Рецензенты:

**В. П. Боюн**, д. т. н., член-корреспондент НАНУ

**В. П. Клименко**, доктор физ.-мат. наук

Рекомендовано к изданию Ученым советом Института кибернетики имени В. М. Глушкова Национальной академии наук Украины (протокол № 6 от 28 апреля 2009 г.)

**Яковлев Ю. С.**

Я47 Однокристалльные компьютерные системы высокой производительности. Особенности архитектурно-структурной организации и внутренних процессов : Монография. – Вінниця : ВНТУ, 2009. – 293 с.

ISBN 978-966-641-310-2

Рассмотрены особенности архитектуры и процессов управления памятью однокристалльных компьютерных систем (КС) высокой производительности и составляющих их компонентов для КС с классической архитектурой и КС типа "процессор-в-памяти" (РІМ - системы) с жесткой и с реконфигурируемой архитектурой (РК-РІМ). Предложен способ оценки "скрытых потерь" производительности классических КС, особенности управления памятью РІМ-систем, классификация методов и средств распределения памяти, обобщенная модель КС и стратегия проектирования РК-РІМ с использованием онтологии. Предложен вариант РК-РІМ, который интегрирует положительные свойства РІМ-систем и исключает недостатки известных КС этого класса.

Для специалистов в области вычислительной техники, аспирантов и студентов.

УДК 004.27; 004.25; 004.382.2

ISBN 978-966-641-310-2

© Ю. Яковлев, 2009

## СОДЕРЖАНИЕ

Список сокращений.....	7
Введение.....	10
1 Однокристалльные компьютерные системы, использующие Память с классической архитектурой.....	25
1.1 Основы технологии реализации систем на одном кристалле.....	25
1.2 Однокристалльные многопроцессорные системы с класси- ческой архитектурой памяти и множественными потоками управления.....	29
1.3 Архитектура для вычислений кристаллической решетки.....	33
1.4 Клеточная архитектура.....	38
1.5 Архитектура КС типа “Память-в-процессоре” (“Memory-in-Processor”, МIP-системы).....	40
1.6 Архитектура типа “Циклоп”.....	46
1.7 “Скрытые потери” производительности компьютерных систем, основанных на применении памяти с классической архитектурой.....	49
1.7.1. Краткая характеристика “скрытых потерь”.....	49
1.7.2 Оценка времени доступа к памяти локальной ЭВМ.....	51
1.7.3 Оценка времени доступа к памяти КС, распреде- ленной в линейном адресном пространстве.....	56
1.7.4 Адресация к памяти КС, распределенной в многомерном пространстве.....	59
1.7.5 Приближенная оценка “скрытых потерь” производительности КС.....	64
1.8 Об особенностях и тенденциях развития систем на кристалле.....	65
2 Компьютерные системы на основе памяти, совмещающей функции хранения и обработки информации (PIM-системы).....	68
2.1 Основные принципы построения PIM-систем (“Processor-in - Memory”).....	68
2.2 Преимущества PIM-систем перед КС с классической архитектурой.....	74
2.3 Применение PIM-систем. Задачи, на решение которых ориентированы PIM-системы.....	77
2.3.1 Обработка изображений.....	78

2.3.2	Применение PIM-систем для улучшения характеристик сетевого интерфейса.....	79
2.3.3	Применение PIM-систем для обработки радарных сигналов .....	81
2.3.4	Применение идеологии построения “Процессор-в-памяти” для портативных и мобильных компьютерных систем .....	84
2.3.5	Применение PIM-систем для космических приложений .....	85
2.4	Примеры зарубежных PIM-систем и особенности их архитектурно-структурной организации.....	88
2.4.1	Пример архитектуры PIM-системы на базе чипа SRAM.....	88
2.4.2	PIM – массив рабочей станции Terasys .....	90
2.4.3	PIM-система FlexRAM .....	92
2.4.4	PIM-система Gilgamesh .....	93
2.4.5	Архитектуры PIM-системы на базе чипа DIVA .....	95
2.4.6	Система технического зрения в реальном масштабе времени (RVS-2) .....	97
2.4.7	Параметры наиболее распространенных PIM-систем.....	97
3	PIM-чипы как основа построения PIM-систем.....	99
3.1	Чип типа SRAM .....	99
3.2	Трехмерный чип SRAM .....	103
3.3	PIM-чип для рабочей станции Terasys .....	105
3.4	Чип FlexRAM.....	107
3.5	Чип MIND.....	110
3.6	Чип V-IRAM.....	114
3.7	Чип DIVA.....	117
3.8	Чип IMAP.....	119
3.9	Другие разновидности PIM-чипов .....	121
4	Основные компоненты чипа PIM-системы .....	124
4.1	Процессоры памяти.....	124
4.1.1	Исходные посылки построения и развития процессоров памяти PIM-чипа .....	124
4.1.2	Одноразрядные процессоры памяти PIM-чипа .....	125
4.1.3	Восьмиразрядные процессорные элементы PIM-систем .....	127
4.1.4	Операционная среда с настраиваемой разрядностью.....	128
4.2	Особенности организации массива памяти внутри чипа.....	132
4.2.1	Проблема быстродействия и потребляемой мощности .....	132

4.2.2	О согласовании скорости работы памяти и скорости работы процессоров .....	134
4.2.3	О разделении массива памяти на банки .....	134
4.2.4	Еще один способ оптимизации параметров памяти .....	136
4.2.5	Устранение холостых пробогов данных .....	137
4.2.6	Вопросы построения контроллера памяти.....	137
4.3	Особенности построения контроллера управления памятью PIM-системы .....	138
4.3.1	Основные принципы построения контроллера .....	138
4.3.2	Архитектурно-структурная организация контроллера PIM-системы типа SRAM .....	139
4.3.3	Особенности построения интеллектуального контроллера управления памятью PIM-системы .....	144
5	Особенности управления памятью в PIM-системах .....	147
5.1	Общие вопросы управления памятью в PIM-системах .....	147
5.2	Управление формированием адреса .....	150
5.3	Распределение памяти .....	153
5.4	Программы распределения памяти и их сравнение .....	158
5.5	Особенности размещения данных в памяти PIM-систем.....	170
5.5.1	О проблеме размещения данных.....	170
5.5.2	Методы размещения данных, основанные на окнах выполнения.....	171
5.5.3	Методы размещения данных, основанные на применении коллекций и макросерверов.....	174
5.5.4	Метод распределения данных с помощью специальных программных систем.....	179
6	Реконфигурируемые PIM - системы: методология построения, примеры моделей.....	182
6.1	Общие вопросы построения, основные определения реконфигурируемых PIM-систем и их компонентов.....	182
6.2	Особенности архитектуры РК-PIM, использующей FPGA.....	185
6.3	Особенности операционной системы РК-PIM с использованием FPGA.....	188
6.4	Особенности алгоритмов распределения ресурсов для РК-PIM.....	190
6.5	Особенности алгоритмов распределения области FPGA.....	193
6.6	Особенности алгоритмов разделения приложений.....	195
6.7	Примеры моделей реконфигурируемых PIM-систем.....	197
6.7.1	Модель реконфигурируемой PIM-системы типа ReConfigME на FPGA.....	197
6.7.2	Реконфигурируемая память с реализацией функций хранения и обработки информации.....	206

7	Вопросы методологии проектирования распределенных КС на базе однокристалльных систем высокой производительности .....	214
7.1	Обобщенная модель распределенной компьютерной системы, выполненной на базе однокристалльных КС.....	214
7.2	Вопросы методологии проектирования РК-РІМ.....	221
7.3	Онтология предметной области “Реконфигурированные РІМ-системы” – эффективное инструментальное средство проектирования .....	226
8	Предложение по созданию модифицированной РК-РІМ с улучшенными пользовательскими характеристиками.....	234
8.1	Структурная схема модифицированной РК-РІМ и её состав .....	234
8.2	Описание основных компонентов системы.....	236
8.3	Краткое описание режимов работы РК-РІМ.....	250
8.4	Работа предлагаемой системы РК-РІМ.....	262
	8.4.1 Загрузка и настройка системы.....	262
	8.4.2 Обработка информации в режиме СО.....	268
	Выводы.....	274
	Литература.....	285

## СПИСОК СОКРАЩЕНИЙ

- ASIC – специализированная интегральная схема  
реконфигурируемой логики
- CIMA – Computing-in-Memory Architecture (вычисления в  
архитектуре памяти)
- CLB – Configurable Logical Bloc (конфигурируемый логический блок)
- COP – Cod Operation (код управления операциями)
- CPU (ЦПУ) – центральное процессорное устройство
- CPU/FPGA/DRAM – процессор/программируемая среда/память
- CRAM – Computational RAM (вычислительная оперативная память)
- CSLC – Coherent side-lobe canceller (компенсация побочного  
лепестка)
- DES – Data Encryption Algorithm (алгоритм шифрования данных)
- DIP – Digital Image Processing (Цифровая обработка изображения)
- DIVA – Data Intensive Architecture (PIM-чип для построения  
системы)
- DRAM – Dynamic Random Access Memory (динамическая  
оперативная память)
- EL – Elaboration Language (язык доработок и уточнений)
- FIFO – First in – First out (память типа “первый пришел – первый  
вышел”)
- FPGA – Field Programmable Gate Arrays (программируемая  
логическая матрица)
- Gilgamesh – Giga Logic Gate Assemblies with Mesh Integration  
(многопоточная архитектура для эффективного вычисления)
- HLF – Higher Layer Function (блоки обработки с высокими функ-  
циональными возможностями)
- HOST – базовая (основная) ЭВМ
- IDEF – Integrated Definition (метод описания интеграции)
- IEIB – Internal Element Interconnect Bus (внутриэлементная шина)
- IMAP – Integrated Memory Array Processor (интегрированная область  
памяти и процессора)
- IRAM – интеллектуальная RAM
- ISA – instruction set architecture (набор команд)
- ISP – In-System Programming (внутрисистемное программирование)
- LSI – большая интегральная схема
- MAYI – Memory Arithmetic Unit and Interface (модуль арифметики,  
памяти и интерфейса”)
- MIN – Multi-stage Interconnect Network (многоступенчатая  
коммутирующая сеть)

MIND – Memory, Intelligence and Networking Device (память, интеллект и сетевое устройство)

MIC – Memory Interface Controller (контроллер разделяемой памяти)

MIP – “Memory-in-processor” (“Память-в-процессоре”)

NIC – Network Information Center (центр управления сетевой информацией)

PCI – Peripheral Component Interface (шина связи с периферией)

PE – Processor Element (процессорный элемент)

PIM - системы – “Processor-in-Memory” (компьютерные системы типа “процессор-в - памяти”)

POR – Partition OR network (разделенная ИЛИ-сеть)

PPN – Parallel Prefix Network (параллельная префиксная сеть)

RAM – Static Random Access Memory (статическая оперативная память)

RCA – аппаратные ресурсы системы

РСП – программные ресурсы системы

RVS-2 – система технического зрения в реальном масштабе времени

RVSC – большая интегральная схема контроллера

SAGE – Statement-Analysis-Grouping-Evaluation (программа распределения приложения по процессорам)

SBT – выделенное бинарное дерево

SCDS – Single-Center Data Scheduling (метод единственного центра размещения данных)

SoC – System on Chip (система на кристалле”)

SL – Schematic Language (язык описания схем)

SqF – последовательный подходящий алгоритм

TCK – Test Clock (тестовая синхронизация)

TDI – Test Data In (вход тестовых данных)

TDO – Test Data Out (выход тестовых данных)

TLU – таблица просмотра

TMS – Test Mode Select (выбор тестового режима)

ТТОР – код операции таблицы истинности

V-DRAM – векторная DRAM

V-IRAM – интеллектуальная векторная RAM

WPG – Weight Partition Dependence Graph (диаграмма зависимости веса разделения приложения)

БП – блок памяти

БПП – библиотеки прикладных программ

ВП – ведущий процессор

Гопс – количество гигаопераций с фиксированной точкой в секунду

Гфлопс – количество гигаопераций с плавающей точкой в секунду



ИМП – интеллектуальный менеджер памяти  
ИС – интегральная схема  
КМЯ – компилятор языка высокого уровня  
КС – компьютерная система  
КЭШ – память типа КЭШ  
МКС – макросерверы  
МОК – модуль очереди команды  
МП – микропроцессор  
МЧПС – межчиповая сеть  
ПДД – поток двоичных данных  
ПК – персональный компьютер  
ПКК – подсистема контроля и координации  
ПЛИС – программируемая логическая интегральная схема  
ПО – программное обеспечение  
ПЭВМ – персональная ЭВМ  
ПЯ – процессорное ядро  
ЯД ОС – ядро операционной системы

## ВВЕДЕНИЕ

Основные направления развития средств вычислительной техники на каждом этапе продиктованы, главным образом, существенным возрастанием сложности задач, решаемых пользователем, а также развитием интегральной технологии в целом и элементной базой для компьютерных средств – в частности. При этом возрастающая во времени сложность пользовательских задач существенно опережает увеличение возможностей существующих средств вычислительной техники. Так ряд задач в таких областях, как управление ядерными процессами, предсказание погоды на планете, задачи геномной инженерии, медицины и др., отличающиеся большим объемом вычислений и массовым использованием данных, плохо поддаются решению (или вообще не подлежат решению) на вычислительных комплексах с классической архитектурой. Широко используемые кластерные структуры в таких системах кардинально не решают проблему обеспечения требуемой производительности для этих задач, поскольку линейное наращивание количества элементов кластера и количества кластеров не приводит к адекватному увеличению производительности из-за существенных физических задержек передач информации через коммутационную среду между удаленными компонентами и обработки достаточно длинных алгоритмических цепочек анализа и управления такими передачами. Более того, следование классическим архитектурно-структурным традициям при перенесении даже всей системы на один кристалл не дает возможности использовать в максимальной степени достижения современной интегральной технологии, оставляя большую часть кристалла не заполненным, увеличивая тем самым количество используемых при создании системы интегральных микросхем и повышая потребляемую мощность за счет необходимости усиления сигналов при передаче между микросхемами.

В связи с этим возникла необходимость в пересмотре основных положений концепции организации вычислительного процесса в высокопроизводительных компьютерных системах, используя в максимальной степени достижения интегральной технологии. При этом там, где это возможно и целесообразно, не исключается применение традиционных кластерных систем. С этой точки зрения можно выделить следующие основные направления развития архитектурных линий ЭВМ, компьютерных систем и их компонентов, реализованных на твердотельной элементной базе:

- 1). Линия классических архитектур ЭВМ и традиционных кластерных систем, основанных на использовании новой элементной базы, в первую очередь – микропроцессоров нового поколения, средств

их функционального расширения и ускорения сложных операций, выполненных на интегральных микросхемах со сверхбольшой степенью интеграции.

2). Линия метакомпьютеров и GRID-систем, основанная на использовании огромного количества как однородных, так и разнородных распределенных в пространстве компьютеров, соединенных между собой скоростными сетями связи.

3). Линия архитектур, основанная на использовании новой машинной алгебры (например, алгебры матриц, рядов Фурье и т.д.) вместо традиционной, широко используемой в настоящее время алгебры действительных чисел. К такому направлению развития архитектурных линий может быть отнесена ЭВМ, использующая, например, алгебру матриц (матрично-алгебраическая ЭВМ).

4). Линия архитектур, основанная на расширении функциональных возможностей и повышении эффективности использования устройств и систем памяти. К такому направлению развития архитектурных линий отнесены компьютерные системы, построенные по принципам “Процессор-в-памяти”(“Processor-in-Memory”(PIM)), “Память-в-процессоре”(“Memory-in-processor” (MIP)), “Псевдоинтеллектуальной памяти” и др.

Четвертая линия развития архитектур, по мнению зарубежных ученых, является весьма перспективной и поэтому уже более десяти лет успешно реализуется за рубежом (системы Terasys, FlexRAM, Gilgamesh и др.).

Однако архитектура и структура каждой такой системы при её создании оптимизировалась под конкретный класс задач (или под одну задачу), при этом, как правило, создавалась специфическая элементная база, что существенно увеличивало время и стоимость разработки. Это, несомненно, явилось отрицательным фактором при множестве достоинств подобного рода систем, что существенно ограничивало области их применения. Поэтому выделилась следующая – пятая линия развития архитектур:

5). Линия развития, основанная на использовании статической и динамической (в процессе работы) реконфигурации архитектуры и структуры системы с целью их оптимизации на конкретные задачи в различных предметных областях (реконфигурируемый компьютеринг). При этом реконфигурация осуществляется тремя возможными способами:

– применение программируемой логической схемы (ПЛИС), на кристалле которой при современном уровне интегральной технологии можно разместить достаточно сложные функциональные схемы такие,

как процессоры, блоки памяти, средства коммутации и интерфейсные схемы;

– применение избыточных аппаратно-программных средств (средства обработки, хранения и коммутации) в составе системы и подключения их к работе в требуемом количестве с соответствующими связями для оптимизации процесса решения конкретной задачи или класса задач;

– применение обоих подходов к реконфигурации в соответствующей их взаимной пропорции.

Каждый из этих способов реконфигурации существенно расширяет область применения компьютерных систем, повышая соответственно их серийность и уменьшая сроки и стоимость разработки.

Исследования в области Computer-science показывают, что основные архитектурно-структурные решения, появляющиеся с развитием интегральной технологии, первоначально апробируются при построении суперЭВМ, и в дальнейшем, при появлении элементной базы следующего поколения, переносятся в качестве базовых решений при создании персональных ЭВМ (ПЭВМ), которые (благодаря совершенствованию интегральной технологии) по сравнению с суперЭВМ являются более дешевыми и более доступными для пользователя. Поэтому не удивительно, что современные персональные ЭВМ по своим основным техническим характеристикам не хуже, чем суперЭВМ предыдущего поколения, “накрывая” по своим параметрам области применения серверов и майнфреймов. Учитывая бурно развивающийся в настоящее время подкласс мобильных персональных ЭВМ в виде ноутбуков и карманных компьютеров, можно выделить следующую линию развития архитектур средств вычислительной техники:

б). Линия развития архитектур персональных ЭВМ, основанная на новых достижениях интегральной технологии для создания элементной базы с малой потребляемой мощностью. Развитие этой линии предполагает появление высокоэффективных мобильных компьютеров типа ноутбук и карманных персональных компьютеров.

Расширение области применения ПЭВМ реализуется, в основном, за счет создания и использования специализированных приставок-расширителей функций и улучшения технических характеристик процессоров и памяти ПЭВМ. Это ускорители используются для выполнения центральным процессором сложных операций, средств ввода – вывода цифровой информации и аналоговых сигналов от датчиков и т.д. Наиболее распространенными расширителями функций и одновременно ускорителями сложных операций ПЭВМ являются графические ускорители, выпускаемые в виде блоков, встроенных в сис-

темную (материнскую) плату ПЭВМ, а также в виде самостоятельных модулей – плат, которые вставляются в каркас системного блока. Таким образом, определилась следующая линия развития архитектур.

7). Линия развития архитектур специализированных средств обработки информации, используемых как самостоятельно, так и в качестве расширителя функций и/или в качестве ускорителя отдельных операций, реализуемых центральным процессором для создания на основе ПЭВМ проблемно-ориентированных АРМ.

Успехи в развитии интегральной технологии обеспечили возможность реализации на одном кристалле большого количества узлов и блоков, представляющие в совокупности сложные функционально законченные системы, которые могут использоваться как в виде компонентов более сложных систем (например, в виде узлов распределенной компьютерной системы), так и самостоятельно при специализации её архитектуры и структуры на решение конкретной задачи (например, однокристалльные микроконтроллеры, Фурье-преобразователи, устройства ввода-вывода информации и др.). Таким образом, основа любой компьютерной системы – элементная база приобрела новые качественные и количественные признаки, такие как функциональная завершенность для реализации фрагментов алгоритма или даже целого алгоритма (вместо выполнения отдельной команды), существенное повышение скорости передачи информации между компонентами системы, поскольку они размещены на одном кристалле, снижение потребляемой мощности, габаритов, веса, так как для построения более сложной системы теперь используется меньшее количество интегральных схем, и т.д. Другими словами – выделилась еще одна специфическая линия развития.

8). Линия развития качественно новой элементной базы как аппаратного базиса для реализации указанных выше архитектурных линий средств компьютерной техники.

В монографии представлена информация об архитектурно-структурных особенностях двух архитектурных линий: линии однокристалльных высокопроизводительных компьютерных систем с классической архитектурой памяти и линии КС типа “процессор-в-памяти” с отображением предложенных новых архитектурных решений. При этом, используя методологию системной интеграции процессов, процедур и объектов, на первый план выносятся особенности построения КС, определяемые уровнем развития твердотельной интегральной технологии и высокими требованиями со стороны пользовательских задач. В соответствие с этим монография содержит 8 разделов.

В первом разделе рассмотрены основы твердотельной технологии, обеспечивающие реализацию распределенной многопроцессорной системы на одном кристалле. Понятие "система на кристалле" (System on Chip – SoC) предполагает функционально завершенную сложную систему, содержащую, как правило, на одном кристалле микропроцессорное ядро, память и другие компоненты системы, необходимые для её функционирования. Такие системы, обладающие жесткой архитектурно-структурной конфигурацией, как правило, ориентированы при их создании на определенный класс задач, где они могут обеспечить максимальную эффективность.

Рассмотрены однокристалльные многопроцессорные системы с классической архитектурой и множественными потоками управления. К ним отнесены системы для вычислений кристаллической решетки, обработки графических изображений и радарных сигналов, системы типа "Память-в-процессоре" (MIP-системы), системы типа "Циклоп" и другие.

Архитектура для вычислений кристаллической решетки использует трансляцию и изменение отображения пространственной решетки алгоритма на гранулированную структуру памяти (слова, строки и т.д.). Так как модули обработки – группы узлов распространены равномерно через всю кристаллическую решетку, при такой организации достаточно просто выполнить многосеточные вычисления. Так как все местоположения кристаллической решетки обрабатываются идентично, это – идеальная структура для SIMD-систем.

В данном разделе рассмотрены также основные положения концепции построения так называемой клеточной архитектуры. Положительные свойства такой архитектуры проявились в реализованном проекте "Cell", главным "потребительским" показателем которого является производительность с пиковым воздействием выполнения операций с плавающей точкой на уровне 256Гфлопс (миллиардов операций в секунду). Архитектура Cell, в которой присутствует управляющий (ведущий) универсальный процессор и ряд соединенных с ним "клеток", по сути, является аналогом распределенных вычислителей (например, кластеров).

Среди других систем с классической архитектурой выделена архитектура "Память-в-процессоре" ("Memory-in-Processor) или MIP-система, которая обеспечивает логическую и физическую интеграцию памяти в функциональных блоках микропроцессора.

MIP - системы содержат блоки обработки HLF (Higher Layer Function) с высокими функциональными возможностями, но зато блоки памяти незначительной емкости. В качестве HLF могут быть использованы: матричные множительные устройства, матричные ин-

верторы, сортирующие устройства, решающие устройства алгоритма графики и др. MIP спроектирована на одном кристалле с “компилятором-на-кремнии”, генерирующим требуемые команды для модулей HLF узла. Архитектура узла MIP обеспечивает возможности построения кластера, таким образом, образуя “суперкомпьютер на кристалле”, параметры которого поддерживаются компилятором на кремнии. Архитектура кластера MIP может быть представлена в виде гибридной кластерной пирамиды, структура которой соответствует иерархическому процессу отображения алгоритма решаемой задачи.

Другой тип архитектуры, которая рассмотрена в первом разделе, это архитектура типа “Циклоп”, представляющая собой архитектуру для высокоэффективных параллельных компьютеров. Основная концепция построения архитектуры “Циклопа” состоит в том, чтобы заполнить чип большим количеством простых (например, последовательных) независимых друг от друга устройств обработки, связанных в каждом чипе регулярным образом через коммуникационную среду. В качестве таких устройств обработки могут быть использованы и специализированные процессоры. Дорогие ресурсы такие, как устройства с плавающей запятой и КЭШ, подсоединены к группам простых устройств. Архитектура типа “Циклоп” предназначена для решения задач, имеющих две важные особенности: они должны обеспечить массовый параллелизм для загрузки порядка миллиона процессоров, кроме того, они должны интенсивно обрабатываться.

Существующие инженерно-технические решения в направлении обеспечения высокой производительности (в том числе при построении однокристалльных систем), как правило, рассматривались без тесной взаимосвязи с системным программным обеспечением, функции которого по управлению памятью фактически с момента зарождения операционных систем претерпели незначительные изменения: остались страничный, сегментный и сегментно-страничный способы распределения и адресации памяти, не решена проблема внутренней и внешней фрагментаций памяти и т.п. Все это приводит к потерям производительности системы, которые для разработчика структуры и программистов являются как бы “скрытыми”, поскольку они порождаются особенностями работы операционной системы. Поэтому в данном разделе предложено формальное описание этих потерь и выполнена их оценка. Показано, что, если требуемая информация не содержится в КЭШ-памяти, а размещена в ОЗУ, то доля времени, затрачиваемая на формирование адреса для обращения к ОЗУ по отношению к полному времени обращения к этой памяти составляет не менее 70% независимо от цели и способа доступа, а доля “скрытых потерь” времени при выполнении наиболее распространенных операций та-

ких, как сложение и умножение двух операндов, составляет не менее 90%.

На основе анализа рассмотренных выше архитектур сделан вывод, что интеграция компонентов обработки и компонентов памяти, при наращивании этих компонентов – вынужденная парадигма любой заказной микропроцессорной системы или кластера. Тем не менее, зарубежный опыт с такими системами, которые обеспечивают пиковую производительность терафлопсного диапазона, показывает, что этот подход строго ограничен в его эффективности для многих важных типов приложений и кроме того стоимость, мощность, и пространственные требования делают их развертывание весьма дорогим. Поэтому они, вряд ли, обеспечат основу для архитектуры будущего поколения, обеспечивая рентабельную и эффективную длительную работу терафлопсном диапазоне в течение следующих десятилетий. Появляющаяся альтернатива, которая может способствовать решению важной проблемы – повышение производительности компьютерных систем при одновременном улучшении других пользовательских характеристик – это технология, относящаяся к категории SoC, которая называется технологией “процессор-в-памяти” (“processor-in-memory”) или PIM-технология, способная обеспечить основу для создания нового класса высокоэффективных компьютерных систем.

*Второй раздел* посвящен анализу особенностей архитектурно-структурной организации компьютерных систем, построенных на основе памяти, совмещающей функции хранения и обработки информации (PIM-системы). PIM-система представляет собой распределенную вычислительную среду на одном кристалле БИС памяти, содержащую массив памяти, разделенный на банки памяти с устройствами обработки, при этом часть вычислительных функций центрального процессора перемещена в банки памяти с целью обеспечения высоких значений пользовательских характеристик системы.

Рассмотрены принципы построения PIM-систем, основные области их эффективного использования и задачи, на решение которых ориентированы системы этого класса. Как правило, это задачи, которые плохо поддаются решению (или их решение вообще невозможно) на КС с классической архитектурой. Характерными признаками таких задач является: массовое обращение к памяти за данными, широкое распараллеливание алгоритмов и фрагментов алгоритмов пользовательской задачи, высокая интенсивность обработки в области петафлопсного диапазона и др. Это задачи аэродинамики, космических приложений, обработки радарных сигналов, сетевого интерфейса, моделирования мозга и имитации в вычислительных неврологиях и биологических науках, задачи молекулярной динамики, физического



и логического моделирования, имитации виртуальной реальности на основе моделирования и др. При этом большинство из этих задач связаны с решением уравнений линейной алгебры огромной размерности, с вычислениями пространственной решетки, рекурсивных и высокопараллельных вычислений изображения и представления двумерной и трехмерной графики и др.

Решение этих задач с помощью PIM-систем реализуется за счет особенностей их архитектурно-структурной организации, обеспечивающих (по сравнению с КС с классической архитектурой) более высокие пользовательские характеристики такие, как: более широкая полоса пропускания по каналам процессор-память, возможность одновременной обработки в петафлопсном диапазоне огромного количества данных (до нескольких тысяч слов), возможность обработки слов большой разрядности и др. К таким системам отнесены: PIM-системы на базе чипа CRAM; PIM - массив рабочей станции Terasys; PIM-система FlexRAM; PIM-система Gilgamesh; PIM-система на базе чипа DIVA; система технического зрения в реальном масштабе времени (RVS-2) и др. Приведены параметры наиболее распространенных зарубежных компьютерных систем такого класса.

Исследования показали, что применение PIM-систем позволяет решить одну и ту же задачу в несколько раз быстрее, чем на КС с классической архитектурой. Любая архитектура, построенная по принципу “процессор-в-памяти”, реализует указанные приложения с пиковым временем в среднем на порядок меньше пикового времени традиционной RISC-архитектуры. При этом следует отметить, что доля используемой площади кристалла БИС памяти под логические схемы за последние несколько лет существенно возросла (от 8 % для первых PIM-систем до 30% для современных), что является явным показателем расширения функциональных возможностей БИС динамической памяти.

*Третий раздел* посвящен архитектурно-структурной организации PIM-чипов как базового ядра построения рассмотренных в предыдущем разделе PIM-систем. Характерной особенностью таких систем является то, что практически для каждой PIM-системы, ориентированной на определенный класс решаемых задач, разрабатывалась своя микросхема как основа построения системы в целом (исключение лишь составляют реконфигурируемые PIM-системы). Поэтому с некоторыми отступлениями можно сделать заключение, что создано столько разновидностей PIM-чипов, сколько существует работающих типов PIM-систем. В данном разделе монографии рассмотрены лишь наиболее яркие в функциональном отношении и архитектурно-структурной организации экземпляры PIM-чипов: чип типа CRAM;

PIM-чип для рабочей станции Terasys; чипы FlexRAM, MIND, V-IRAM, DIVA, IMAP и др.

Анализ параметров большинства из рассмотренных в монографии чипов показал, что при построении PIM-систем в зависимости от их назначения использовались как одnorазрядные процессорные ядра (системы Terasys и CRAM), 8-разрядные (системы IMAP, PIP-RAM), так и 16-и 32-разрядные (системы V-IRAM и DIVA) с количеством процессоров на одном кристалле от 64 до 256 штук при емкости памяти на кристалле от 16Кбит до 64 Мбайт. Естественно, что с таким разбросом ресурсов эти системы на различных пользовательских задачах обеспечивали различную производительность.

*В четвертом разделе* рассмотрены более подробно основные компоненты чипа PIM-системы: процессорные ядра (процессоры памяти), массив памяти внутри чипа и контроллеры управления памятью. На начальном этапе построения PIM-систем одной из главных проблем стала проблема совместимости на одном кристалле различных технологий – технологии создания элементов памяти и технологии создания логических элементов. Поэтому в первых PIM-системах типа CRAM были использованы в качестве базовых процессоров памяти элементарные 1-битовые процессоры последовательного типа, один из вариантов архитектуры которого приведен в данном разделе монографии. Однако при их большом количестве для вполне определенных задач PIM-системы с 1-битовыми процессорами по сравнению с классическими КС обеспечивали существенно большую производительность при небольшой потребляемой мощности. Далее появились 4-разрядные процессорные элементы, 8-разрядные, шестнадцати - и 32-разрядные и т.п., чему способствовало появление новых интегральных технологий (например, технологии BiCMOS, пригодных для создания, как элементов памяти, так и процессорных (логических) элементов. Для большинства систем обработки изображений, исходя из требований к обработке каждого пиксела, наиболее подходящими в SIMD-системах являются 8-разрядный процессорный элемент, структурная схема которого также рассмотрена в данном разделе.

Часто при реализации алгоритмов требуется выполнять как целочисленные вычисления, так и вычисления с плавающей запятой, при этом средство, предназначенное для реализации такого алгоритма, в наибольшей степени отвечает концепции построения PIM-систем, обеспечивающих широкую полосу пропускания по каналу процессор-память. В данном разделе приводится структурная схема процессорного элемента с настраиваемой разрядностью, который эффективно поддерживает оба эти вычисления, выполняя за счет реконфигурации канала обработки параллельные арифметические операции

над 8, 16, или 32-разрядными целочисленными данными или над 32-разрядным данными с плавающей запятой.

Требование параллельной обработки множества данных в PIM-системе приводит к необходимости активизации ячеек памяти, пронизывающих весь массив памяти огромной разрядности, что сопряжено с повышенной потребляемой мощностью, которая, в свою очередь, отрицательно влияет на производительность системы в целом. Поэтому важной проблемой при архитектурно-структурной организации массива памяти становится проблема обеспечения сверхвысокой производительности при малых величинах потребляемой мощности. Именно этим и продиктованы особенности построения массива памяти внутри чипа PIM-системы, основные положения которых освещены в данном разделе монографии. При этом в качестве основного способа снижения потребляемой мощности при обращении к памяти приводится способ разделения всего запоминающего массива на банки и подбанки памяти, что существенно уменьшает паразитную емкость цепочки опросных ячеек памяти и следовательно – уменьшает потребляемую мощность.

Рассмотрены и другие способы уменьшения потребляемой мощности и повышения быстродействия памяти PIM-системы: совмещение чтения данных из разных подбанков памяти, подключенных к одной и той же шине (конвейерное чтение и соответственно конвейерная обработка), устранение “холостых пробегов данных”, т.е. чтение только нужных для последующей операции данных, а не всей цепочки, состоящей из множества слов, с возвратом не используемых данных обратно в память.

Рассмотрены особенности, основные принципы построения, а также архитектурно-структурная организация контроллера управления памятью на примере PIM-системы типа SRAM.

Контроллер памяти размещается на одном кристалле с массивом памяти и процессорами, что по сравнению с контроллером, размещенном на отдельном кристалле, обеспечивает более высокую стабильность работы, повышенное быстродействие и более низкую потребляемую мощность, так как исключается необходимость усиления сигналов управления при их передаче от кристалла контроллера к кристаллу памяти. При этом по сравнению с контроллерами памяти с классической архитектурой, на данный контроллер помимо обычных функций по управлению классическими режимами памяти (запись, чтение информации и т.п.) возлагается существенный набор дополнительных функций, связанных с реализацией операций в памяти, передачей информации между памятью других аналогичных чипов и внешней (дополнительной) памятью, контролем информации и т.д.

Приведена интеллектуальная система памяти (ИСП), которая встраивается в чип динамической оперативной памяти и берет на себя функции управления памятью, выполняемые обычно центральным процессором. Показано, что перемещение функций управления памятью от центрального процессора к интеллектуальной памяти устраняет в среднем 60 % неудачных обращений в КЭШ по сравнению с обычным методом, где эти же функции выполняются центральным процессором. В качестве такого контроллера может быть использован специальный вариант PIM-компьютера.

*Пятый раздел* освещает особенности управления памятью в PIM-системах, рассматривая при этом вопросы общей стратегии управления памятью, управление формированием адреса, способы и средства распределения памяти с представлением процесса управления распределением памяти, а также особенности размещения данных в распределенных областях памяти PIM-системы, включая методы размещения данных, основанные на окнах выполнения, на применении коллекций и макросерверов, а также с помощью специальных программных систем типа SAGE. Некоторые из этих задач рассмотрены на примере их решения в коммерческих зарубежных системах, в частности – Gilgamesh.

Особенностью управления памятью PIM-систем состоит в том, что функции управления памятью в системе разделены между хост-процессором и процессорными ядрами банков памяти. На хост-процессоре стандартная операционная система расширена функциональными возможностями, чтобы поддержать работу PIM. На каждом PIM - процессоре во время выполнения имеется своя небольшая программа управления (стержень ОС), которая всегда является резидентом. Все это накладывает определенный отпечаток на специфику решения задач управления памятью, представленных в монографии. К таким задачам, прежде всего, относятся:

- формирование и передача адресной информации к любой ячейке массива памяти с учетом разбиения массива памяти на банки и подбанки;
- оптимизация разделения (распределения) всего массива памяти в зависимости от разделения решаемой задачи между хост-процессором и ядрами процессоров, приписанных к банкам и подбанкам памяти, емкости памяти всего массива, размещенного на кристалле, количества узлов, типа реализуемых алгоритмов и других параметров КС (проблема разделения памяти);
- оптимизация размещения данных в виртуальных и физических пространствах памяти по разделенным областям;
- динамическая корректировка информации (замещение стра-

ниц, подкачка информации в нужную область памяти и др.).

Оценивая рассмотренные в данном разделе средства и способы управления памятью PIM-систем, сделан вывод, что при решении проблемы распределения памяти и оптимального размещения данных, целесообразно прежде всего, основываться на положительных особенностях систем этого типа (по сравнению с классическими системами на одном кристалле), используя в ряде случаев следующие их возможности:

- возможность чтения и одновременной обработки большого количества полноразрядных слов (до нескольких тысяч), а также возможность изменения их разрядности и режимов обработки (с плавающей или фиксированной запятой);

- возможность направления команд (работы) и ресурсов обработки информации разного уровня к соответствующим массивам данных, расположенным в банках и подбанках памяти различных узлов распределенной PIM-системы, вместо массовых пересылок данных к средствам обработки и динамического перераспределения памяти в процессе реализации приложений

В *шестом разделе* рассмотрены вопросы построения реконфигурируемых PIM-систем (PK-PIM), включая особенности архитектурно-структурной организации системы в целом и их компонентов, а также особенности функций операционной системы, алгоритмов распределения ресурсов и разделения приложений для PK-PIM. При этом выделены три основных подхода к решению проблемы реконфигурации: реконфигурация с использованием FPGA; реконфигурация без использования FPGA путем выбора и распределения необходимых ресурсов под реализуемые приложения из заведомо заложенной в архитектуру PIM-систем аппаратной и программной избыточностей; комбинированный способ реконфигурации, использующий как заведомо внесенную избыточность ресурсов и выбор их под реализуемое приложение, так и применение FPGA. Рассмотрены две типовые модели реконфигурируемых PIM-систем: модель реконфигурируемого PIM-компьютера типа ReConfigME и реконфигурируемая система памяти.

Проведенный анализ способов и средств реконфигурации в области вычислительной техники позволил сделать выводы, что реконфигурация – это перспективное направление создания и применения компьютерных систем и комплексов, обеспечивающее перенастройку их архитектуры на оптимальное решение пользовательских задач. Перенесение принципов реконфигурации на PIM-архитектуру при построении компьютерных систем позволяет сделать существенный скачок в улучшении их параметров. Поэтому проблема реконфигура-

ции PIM-систем (в частности – с использованием FPGA) в настоящее время приобретает высокую актуальность, особенно для стран, где отсутствует собственная микроэлектронная промышленность, поскольку применение FPGA является единственно возможным путем апробировать новые идеи и инженерно-технические решения в области вычислительного машиностроения и даже в некоторых случаях создать промышленные образцы изделий на базе РК-PIM.

*Седьмой раздел* посвящен вопросам методологии проектирования распределенных КС высокой производительности, выполненных на одном кристалле. Отмечено, что при проектировании таких КС осуществляется поиск оптимального соответствия реализуемого алгоритма решаемой задачи на её архитектурно-структурную организацию. Только при проектировании КС на заказных БИС такое соответствие достигается во время разработки и создания непосредственно самих БИС, а при проектировании РК-PIM соответствие достигается либо за счет использования программируемой среды (например, ПЛИС) либо за счет применения селекторов для выбора требуемых компонентов системы из имеющегося избыточного набора компонентов на кристалле.

Предлагается обобщенная модель многоуровневой стратегии оптимизации КС, выполненной на одном кристалле. При этом, по сравнению с известными подходами к оптимизации классической архитектуры КС, в предлагаемой модели появились новые параметры, определяемые новыми возможностями интегральной технологии создания БИС, новые целевые функции при проектировании узлов и блоков на одном кристалле. Рассмотрены вопросы методологии проектирования реконфигурируемых PIM-систем как наиболее перспективного направления развития средств компьютерной техники. Предлагается в качестве эффективного инструментального средства проектирования использовать онтологию. В данном разделе монографии приведен пример разработанной автором онтологии предметной области “Реконфигурируемые PIM-системы”.

Сделан вывод, что, несмотря на указанные в предыдущих разделах явные преимущества PIM-систем перед КС с классической архитектурой, они имеют ряд недостатков, так как не все имеющиеся резервы по повышению их производительности и потребляемой мощности в коммерческих зарубежных системах использованы. Исследования показали, что можно построить PIM-систему, которая, с одной стороны, содержала бы все преимущества этого класса машин по сравнению с КС с классической архитектурой, а с другой – не имела бы выявленных недостатков существующих PIM-систем.

*Восьмой раздел* достаточно подробно отражает описание архитектуры, структуры и организации вычислительного процесса предложенного модифицированного варианта РК-РІМ и её компонентов. В отличие от известных систем такого класса предложенная реконфигурируемая РІМ-система обладает явными преимуществами, сущность которых, прежде всего, состоит в том, что реализована возможность повышения производительности системы (от 2,5 до 10 раз) за счет исходной настройки архитектуры на класс решаемых задач, а также её перестройки в процессе работы (в динамике). Это достигается путем использования оригинальных схемотехнических решений и соответствующих режимов работы БИС памяти, обеспечивающих:

- выбор оптимального ведущего процессора (проблемно-ориентированного или специализированного);
- выбор необходимого массива памяти для решения конкретной задачи с возможностью подключения дополнительной памяти, находящейся за пределами чипа;
- выбор оптимальной разрядности данных и количества обрабатываемых слов в пределах всей строки массива памяти, ориентируясь на соответствующие методы обработки и имеющиеся ресурсы для распараллеливания алгоритма решаемой задачи.

Кроме того существенно снижена мощность, потребляемая одним чипом за счет:

- исключения “холостых пробегов” процессорных ядер массива памяти, так как они загружены только полезной вычислительной работой;
- сокращения количества одновременно передаваемых бит, так как передачи по линиям связи не всегда реализуются полной  $N$ -разрядной строкой, а при необходимости произвольными кратными  $R$ -разрядными группами (например, частями строки разрядностью  $R, 2R, \dots, \alpha R; \alpha R \leq N$ );
- Допускается более глубокое распараллеливание процесса обработки информации при тех же исходных ресурсах за счет повышения эффективности использования процессоров.
- Увеличивается серийность РІМ-чипа и соответственно уменьшается стоимость, поскольку он из специализированного преобразован в чип широкого назначения за счет настройки его аппаратно-программных ресурсов.

Таким образом, в связи с особенностями архитектур РІМ-систем возникает необходимость в создании специфической методологии их проектирования. Предлагаемые подходы к решению этой проблемы основываются на создании аппаратно-программных средств управления памятью, включая решение проблемы адресации, распределение

памяти, размещение данных и в ряде случаев целесообразность передачи команд и необходимых для их выполнения атрибутов к местам размещения нужных массивов данных вместо пересылки этих массивов к средствам обработки.

Одним из наиболее вероятных путей практической реализации и использования РІМ-систем в настоящее время является их ориентация на применение в качестве приставки-ускорителя ЭВМ, что дает возможность выполнить их физически в виде реконфигурируемой РІМ-системы.

В заключение автор выражает благодарность за ценные советы своим коллегам и оказание помощи в подготовке материалов для рукописи и её оформлению, в частности, м.н.с., к.ф-м.н. И.М.Першко, н.с. Б.М.Тихонову, аспирантке Е.В.Елисейевой и другим, а также своей семье за помощь и поддержку.



*Наукове видання*

**Юрій Сергійович Яковлев**

# **ОДНОКРИСТАЛЬНІ КОМП'ЮТЕРНІ СИСТЕМИ ВИСОКОЇ ПРОДУКТИВНОСТІ**

## **Особливості архітектурно-структурної організації і внутрішніх процесів**

(російською мовою)

**Монографія**

Оригінал-макет підготовлено Ю. С. Яковлевим

Видання здійснене в авторській редакції

Підписано до друку 27.07.2009 р.  
Формат 29,7×42 ¼. Папір офсетний.  
Гарнітура Times New Roman.  
Друк різнографічний. Ум. друк. арк. 16,91.  
Наклад 100 прим. Зам. № 2009-142.

Вінницький національний технічний університет,  
видавництво «УНІВЕРСУМ-Вінниця»  
21021, м. Вінниця, Хмельницьке шосе, 95  
ВНТУ, ГНК, к. 114  
Тел. (0432) 59-85-32  
Свідоцтво суб'єкта видавничої справи  
серія ДК № 3516 від 01.07.2009 р.

Віддруковано у Вінницькому національному технічному університеті,  
комп'ютерному інформаційно-видавничому центрі  
21021, м. Вінниця, Хмельницьке шосе, 95  
ВНТУ, ГНК, к. 114  
Тел. (0432) 59-81-59  
Свідоцтво суб'єкта видавничої справи  
серія ДК № 3516 від 01.07.2009 р.

---

Замовити цю книгу <https://press.vntu.edu.ua/index.php/vntu/catalog/book/479>

Видавництво Вінницького національного технічного університету

<https://press.vntu.edu.ua/index.php/vntu/catalog>