

## АРХИТЕКТУРНО-СТРУКТУРНАЯ ОРГАНИЗАЦИЯ КОМПЬЮТЕРНЫХ СРЕДСТВ КЛАССА “ПРОЦЕССОР-В-ПАМЯТИ”

---

**Abstract:** The new type of architecture of “processor-in-memory” is offered, possessing a high performance due to the run-time tuning of resources and effective use of processors. It is achieved by the choice of leading processors of the system, amounts of the processors up-diffused on the array of memory, the optimum processed data and etc.

**Key words:** “processor-in-memory”, system on a single crystal, tuning resources.

**Анотація:** Запропоновано новий тип архітектури класу “процесор-в-пам’яті”, що має високу продуктивність за рахунок динамічного настроювання ресурсів і ефективного використання процесорів. Це досягається шляхом вибору провідних процесорів системи; кількості процесорів, розподілених по масиві пам’яті; оптимальної розрядності оброблюваних даних та ін.

**Ключові слова:** “процесор-в-пам’яті”, система на одному кристалі, настроювання ресурсів.

**Аннотация:** Предложен новый тип архитектуры класса “процессор-в-памяти”, обладающий высокой производительностью за счет динамической настройки ресурсов и эффективного использования процессоров. Это достигается путем выбора ведущих процессоров системы, количества процессоров, распределенных по массиву памяти, оптимальной разрядности обрабатываемых данных и др.

**Ключевые слова:** “процессор-в-памяти”, система на одном кристалле, настройка ресурсов..

### 1. Введение

Одной из главных проблем при построении компьютерных систем является проблема установления оптимального баланса между скоростью работы памяти и тактовой частотой процессора. Развитие интегральной технологии изготовления БИС, что позволило создавать более совершенные архитектуры памяти, интегрирующие на одном кристалле (чипе) средства хранения достаточно большой емкости и множество средств обработки. Такие изделия образуют новый перспективный класс компьютерных средств, обозначенные по исходной идеологии построения как класс PIM-устройств (Processor-in-memory) [1, 2]. Этот класс устройств содержит несколько разновидностей, которые отличаются типом размещенных в области памяти процессоров, архитектурно-структурной организацией самого массива памяти, организацией интерфейса процессор-память, организацией коммуникационной среды внутри кристалла (чипа) и т.д. Среди средств класса “процессор-в-памяти” можно выделить следующие наиболее яркие типы, при построении которых используются известные стратегии их архитектурно-структурной организации: “процессор-в-памяти” (PIM) как простейший вариант исходной стратегии реализации такого класса устройств, вычисления в архитектуре памяти (Computing-in-Memory Architecture) – CIMA, векторные DRAM (V-DRAM), интеллектуальные RAM (IRAM), интеллектуальные векторные (V-IRAM) архитектуры, мультипроцессор на кристалле (Multiprocessor-on-a-Chip), интегрированный CPU/FPGA/DRAM, архитектура типа MIND (Memory, Intelligence and Networking Device – память, интеллект и сетевое устройство) и др. Все типы этих устройств обладают общими признаками, характеризующими их как класс. К общим признакам этих устройств можно отнести разбиение внутри кристалла массива памяти на подмассивы с возможностью одновременного чтения данных из каждого подмассива, так что на выходе распределенной памяти может образоваться строка длиной в несколько десятков (и даже сотен) полноразрядных слов; размещение на том же кристалле в каждом подмассиве устройства обработки (процессора), непосредственно

подключенного по входам и выходам каждый к своему подмассиву памяти, что обеспечивает возможность одновременной обработки всей строки; резкое повышение (по сравнению с классическими вариантами построения систем) производительности, снижение потребляемой мощности системы за счет размещения на одном кристалле памяти вместе с процессором и соединения их посредством коротких линий связи, не требующих умощнения и буферизации передаваемых сигналов; возможность использования БИС памяти такого типа как в качестве средств хранения информации, так и в качестве средств обработки самостоятельно либо в виде приставки – ускорителя вычислений, и другие положительные свойства, отмеченные в [2]. Учитывая перспективность систем такого типа (по оценкам зарубежных специалистов), представляется целесообразным рассмотреть особенности архитектурно-структурной организации основных разновидностей изделий этого класса, чтобы выявить имеющиеся недостатки и предложить эффективные решения для их устранения. В результате выполненного анализа разновидностей PIM-архитектур предложен вариант архитектуры, свободный от замеченных в них недостатков и способный стать основой развития ЭВМ и систем данного архитектурного направления.

## 2. Особенности построения архитектур PIM-устройств различных типов

Один из первых проектов, реализованных с применением основной идеи простейшей PIM-архитектуры – это Terasys PIM-матрица [3, 4], в которой большое количество простых вычислительных элементов (как правило, упрощенных ядер процессоров) интегрируются с DRAM-устройствами памяти.

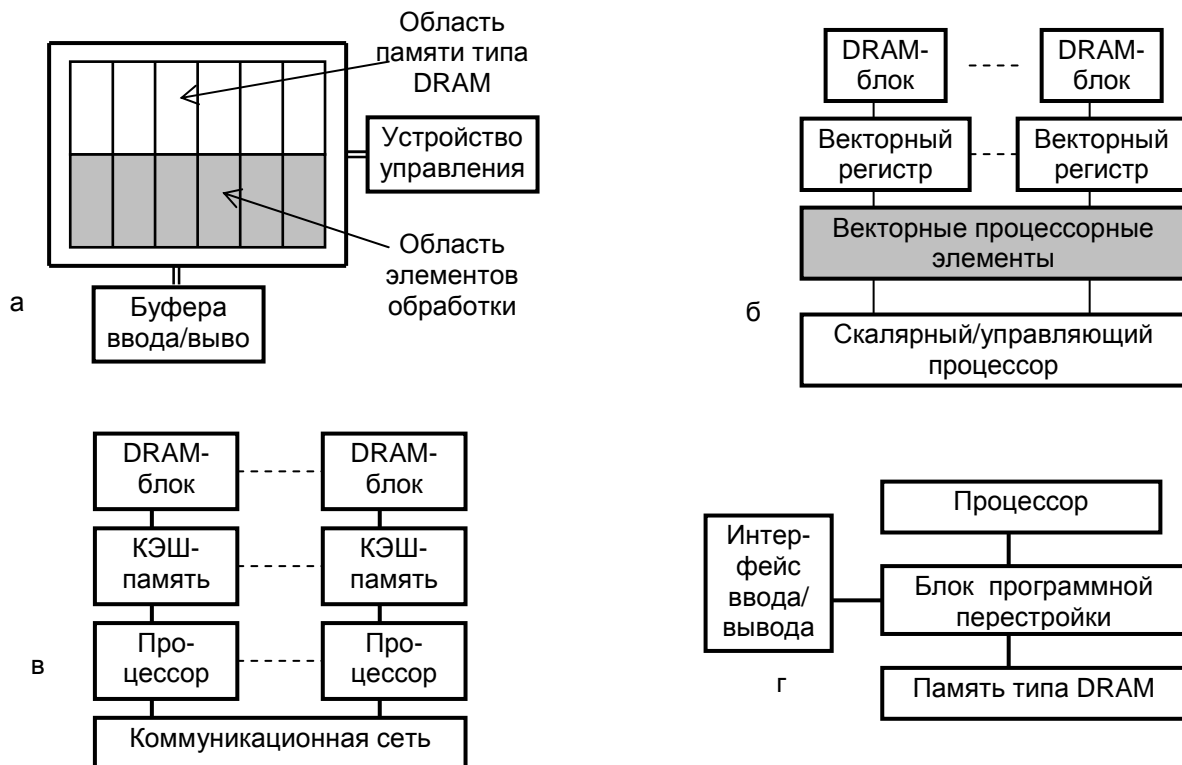


Рис. 1. Отображение особенностей организации различных PIM-систем

Эти вычислительные элементы встроены в область памяти, а результаты их совместной работы интегрируются на выходе чувствительных усилителей кристалла под управлением одного устройства управления (как в SIMD-процессоре). Тем самым обеспечивается большая ширина полосы пропускания DRAM. Идея организации архитектуры такого типа показана на рис. 1а.

*Особенности построения векторных DRAM* состоят в том, что вместо упрощенных ядер процессоров применены векторные процессоры, которые соединены с DRAM-памятью на одном DRAM-чипе. При этом, если пространственные ограничения на кристалле не настолько критичны, то могут быть использованы достаточно мощные элементы обработки (рис. 1б). Архитектура векторной DRAM может, в принципе, обеспечить наибольший эффект в повышении производительности, поскольку она балансирует величину нагрузки с использованием достаточно совершенной модели программирования.

*Multiprocessor-on-a-Chip* содержит несколько относительно простых, но полностью независимых кэшированных RISC-процессоров, каждый с определенным количеством локальной памяти (например, 8 МБ-DRAM), образуя элемент обработки или PPRAM-узел. Такие узлы связаны широкополосным интерфейсом и могут быть запрограммированными, используя стандартную разделяемую память или параллельные алгоритмы передачи сообщений (рис. 1в). Архитектура Multiprocessor-on-a-chip может иметь хороший эффект при реализации в традиционном сервере или рабочей станции. Однако в некоторых приложениях (например, обработка изображений) эта архитектура не обеспечивает достаточного параллелизма.

*Интегрированный чип типа CPU/FPGA/DRAM* объединяет центральный процессор и динамическую оперативную память с программно перестраиваемой логикой (FPGA), чтобы создать реконфигурируемые системы, которые могут приспосабливаться к изменению потребностей обработки (рис. 1г). Реконфигурируемость позволяет обеспечивать ресурсы, которые необходимы для текущего приложения, при этом соответствующая программная библиотека может быть загружена в FPGA по мере необходимости для каждого индивидуального приложения.

Если использовать современную технологию изготовления БИС и интегрировать большую динамическую оперативную память, ядро центрального процессора и достаточно большой массив FPGA, то это приведет к необоснованно большой площади кристалла. Однако в перспективе при доступных технологических нормах  $\leq 0,18\text{мк}$  требуемая площадь кристалла становится достаточно приемлемой. Таким образом, архитектура CPU/FPGA/DRAM обладает наибольшим потенциалом для улучшения параметров системы при дальнейшем развитии технологического процесса создания БИС.

*IRAM-архитектура* ориентирована на значительное повышение скорости вычислений при так называемых “память-интенсивных” применениях.

*V-IRAM-архитектуры* соединяют некоторые достоинства векторных и IRAM-архитектур. Упрощенная архитектура чипа V-IRAM показана на рис. 2 [5].

Чип V-IRAM, используемый совместно с сопроцессором обработки изображений в системе обработки радарных сигналов [5], содержит одно векторно-процессорное устройство, память типа DRAM емкостью 8МГбайт и скалярный процессор. В свою очередь, векторно-процессорное устройство содержит два арифметических устройства, два флаговых процессора и два устройства

согласования нагрузка/память. Эти устройства – конвейеризованы. DRAM-память разделена на две части, каждая из которых имеет четыре банка. Между памятью типа DRAM и векторным процессором применен перекрестный коммутатор. Возможно и другое разделение DRAM-памяти внутри кристалла в зависимости от заданной точности вычислений, например, на 4 модуля (вместо 2), для реализации 64-разрядных операций или на 8 модулей для 32-разрядных операций. Применен 512-разрядный тракт данных между процессорами и памятью типа DRAM.

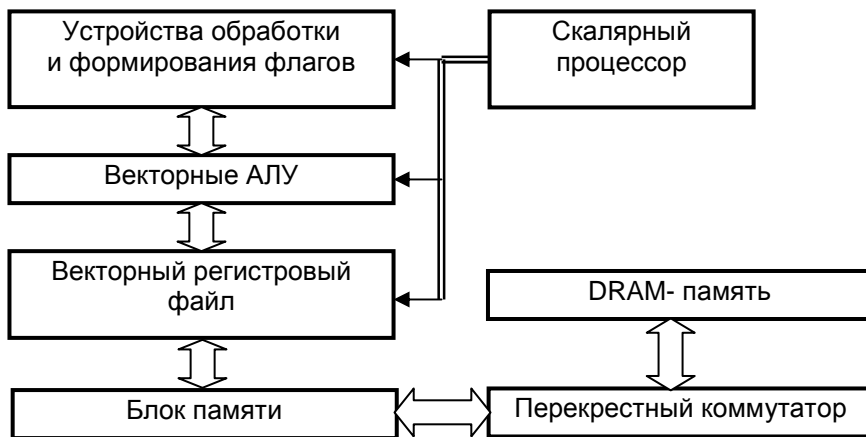


Рис. 2. Упрощенная архитектура чипа V-IRAM

Архитектура типа MIND расширяет PIM-возможности, включая более совершенные механизмы для виртуализации задач и данных и адаптивное управление ресурсами для балансирования загрузки и времени ожидания. Упрощенная архитектура чипа MIND изображена на рис. 3 [6, 7].

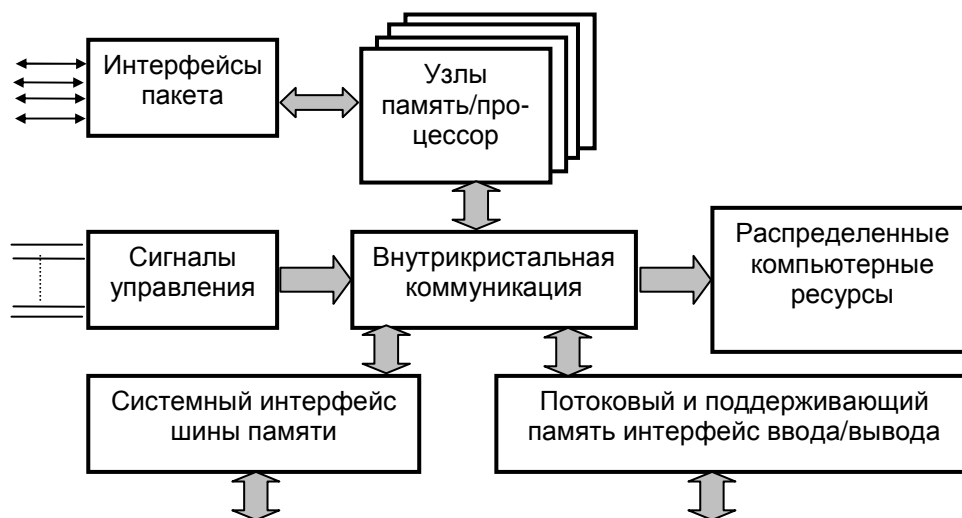


Рис. 3. Упрощенная архитектура типа MIND

Подобно другим PIM-системам, архитектура MIND использует возможности широкой полосы пропускания памяти, обращаясь к полной строке памяти (типично – 2048 бит на обычном

чипе DRAM) одновременно при разделении общей памяти чипа на множество независимо доступных блоков. Это обеспечивает пиковую полосу пропускания памяти на чипе почти на два порядка выше величины полосы пропускания обычных систем памяти сопоставимой емкости [6, 7]. Чтобы эффективно использовать эту полосу, “широкое” арифметико-логическое устройство (АЛУ) интегрировано с буфером строки каждого банка памяти. Плотная связь на кристалле “широкого” АЛУ с буфером строки позволяет получить низкое время ожидания доступа к данным. В дополнение к сказанному, архитектура MIND обладает следующими полезными свойствами:

- управляет виртуальным пространством имён, которое разделено среди множественных узлов MIND, при этом трансляция виртуального адреса на физический реализуется как встроенная функция каждого узла, поддерживая полное виртуальное адресное пространство, общедоступное для всех чипов MIND;

- использует в качестве протоколов (среди чипов MIND и их узлов) пакеты, поддерживающие вычисления, управляемые сообщением, которые включают значения параметра и спецификаторы действия, целенаправленные определяемому адресату;

- содержит встроенные для каждого узла на чипе MIND механизмы контекстного переключения многопоточности, управляющие местными ресурсами и упрощающие логику процессора;

- содержит механизмы обнаружения и изоляции ошибок на границах и в пределах индивидуального узла;

- включает параллельный интерфейс для соединения с шинами обычной памяти при интеграции с системами рабочих станций и серверов; это обеспечивает высокую степень применимости чипа MIND.

Внешний системный интерфейс шины памяти (параллельный интерфейс) обеспечивает средства для подключения чипов MIND к шинам памяти системной платы сервера (или обычной рабочей станции).

Интерфейсы пакета (последовательно-параллельные) поддержаны каналами для активного вычисления, управляемого сообщением, между чипами MIND, обеспечивая выполнение операций над данными полностью в пределах системы памяти.

Потоковый интерфейс ввода/вывода данных (параллельный) используется для быстрого передвижения данных в чип или из чипа. Это необходимо при применении высокоскоростных вторичных устройств хранения данных или при доступе к отдаленным устройствам типа видеокамер в реальном времени. Имеются некоторые дополнительные линии передачи сигналов, используемые для аппаратного сброса и обращения к операционной системе. Чип MIND также включает конвейерные арифметические модули с плавающей запятой, которые разделены среди узлов посредством коммуникационных связей.

### **3. Организация архитектуры PIM-систем на базе PIM-устройств**

Классу архитектур типа PIM-систем свойственна иерархическая гибридная многопроцессорная среда, которая включает ведущие процессоры и процессоры памяти. Ведущие процессоры по сравнению с процессорами в памяти являются более мощными, с глубокими иерархиями КЭШ и

более высокой задержкой доступа к памяти. В качестве процессоров памяти обычно используют менее мощные процессоры (так называемые ядра процессоров), но с гораздо меньшей задержкой доступа к памяти. Один из вариантов такой архитектуры типа PIM показан на рис. 4 [8].

Схема содержит N чипов памяти (ЧП), на каждом из которых размещены ведущий процессор (ВП), массив устройств обработки (процессорных ядер – ПЯ), массив динамической памяти (например, типа DRAM), набор регистров памяти, выполняющих функции КЭШ и размещенных на кристалле между ВП и массивом ПЯ. При этом все ЧП объединены межчиповой сетью, а каждый из них соединен по входам и выходам с общей шиной (например, PCI), к которой подключена HOST – машина (главный процессор), содержащая, как правило, один или два уровня памяти типа КЭШ. Пример структуры одного ЧП приведен подробнее на рис. 5 [9].

Каждый из ПЯ связан с банком DRAM. Чтобы работать при скоростной несогласованности между процессорами и памятью, эти чипы организованы подобно нетривиальной иерархии памяти для каждого банка DRAM (рис. 5а).

На рис. 5б изображена архитектура иерархической памяти для одного "банка", содержащая память типа КЭШ с аппаратной реализацией последовательной выборки и с упреждением одной строки. Непосредственно банк DRAM разделен на суббанки и имеет буфера данных (БД) и буфера строки (БС).

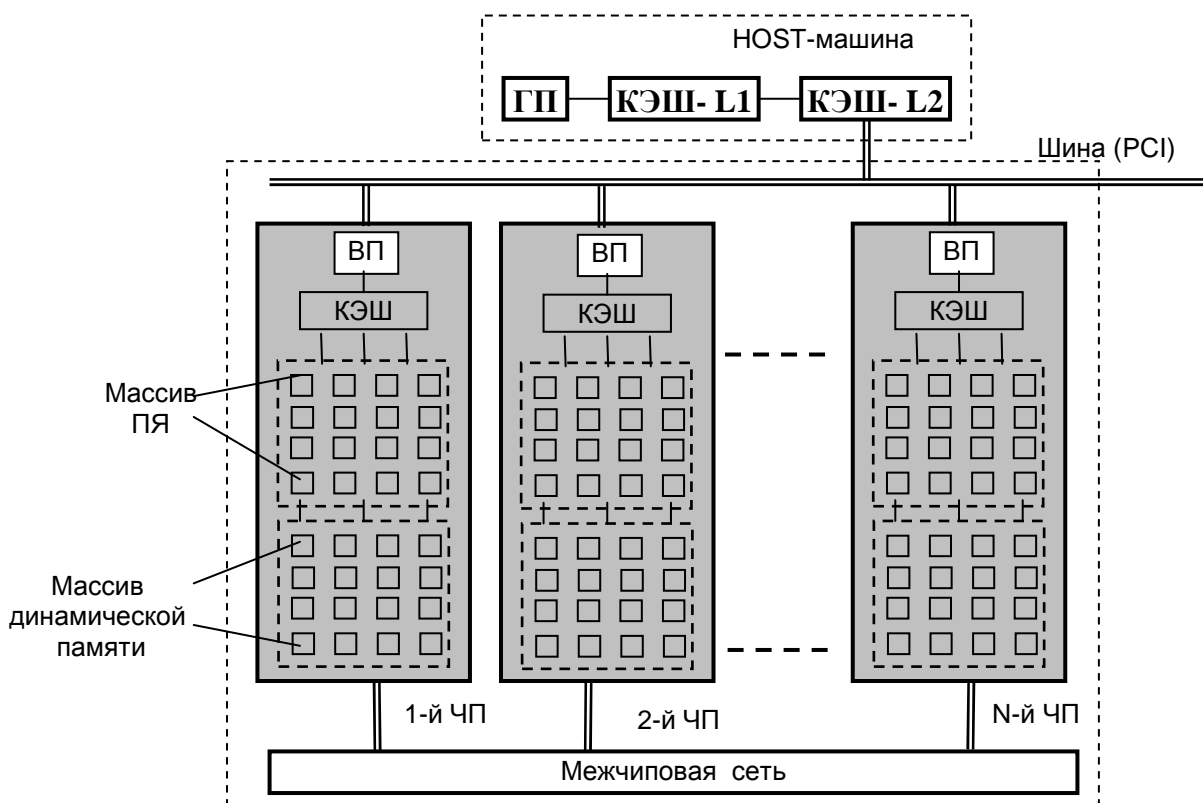


Рис. 4. Упрощенная схема системы с архитектурой "процессор -в -памяти"

Аппаратные средства, отражающие особенности PIM-системы, поддерживают глобальное виртуальное общедоступное адресное пространство, управляемое сообщением вычисление и параллельные процессы. Подобно простым блокам памяти, PIM получает внешние запросы, чтобы обращаться и манипулировать содержанием ячеек памяти. PIM-устройство может также

выполнять сложные последовательности операций в памяти, определенные приложением пользователя или сервисными программами супервизора. На рис. 6 представлена архитектура типа “процессор-в-памяти”, в которой в качестве ведомого процессора используется векторный процессор [10], а на рис. 7 – одна из разновидностей архитектур, построенная на базе MIND-чипов. Это многопоточковая, масштабируемая, гомогенная Gilgamesh (Giga Logic Gate Assemblies with Mesh Integration) архитектура, построенная на базе единственного типа устройства – чипа MIND для высокоэффективных вычислений петафлопсного диапазона [6, 7].

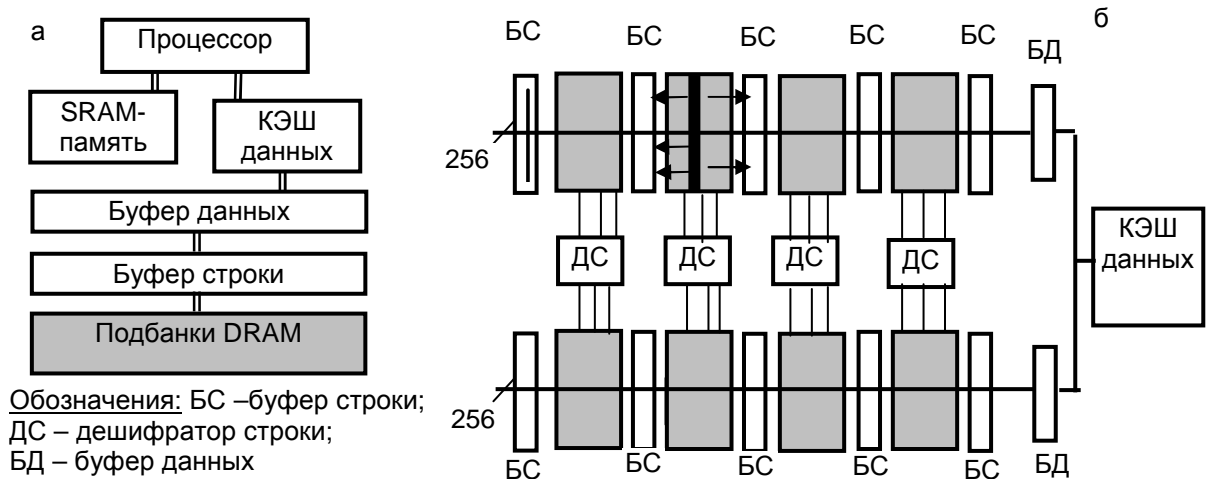


Рис. 5. Пример структуры одного PIM-чипа памяти

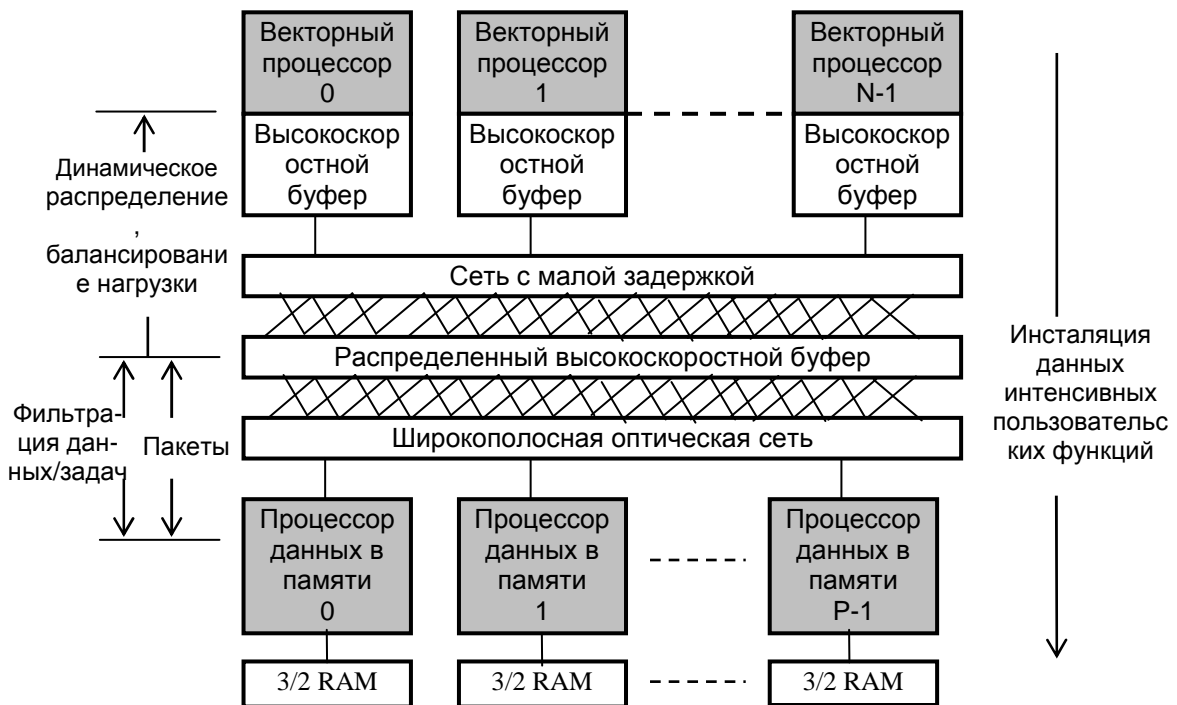


Рис. 6. Архитектура типа “процессор-в-памяти”, в которой в качестве ведомого процессора используется векторный процессор

Gilgamesh – это интеграция (возможно, очень большого количества) чипов MIND, связанных системной локальной сетью. Каждый чип MIND содержит множественные банки памяти DRAM, обрабатывающую логику, внешние интерфейсы ввода - вывода и межчиповые каналы связи. Все чипы MIND связываются друг с другом через пакеты специального класса активных сообщений для управления вычислением. Пакеты назначают виртуальные адреса и могут использоваться для того, чтобы выполнять обычные операций типа отдаленной загрузки или хранения, а также направлять методы организации вычислений на отдаленный чип MIND. Другими словами, элементы чипов MIND воспринимают задание через пакеты и исполняют указанные действия по модификации состояния в локальном масштабе, порождения и отправки новых пакетов к другим частям архитектуры.



Рис. 7. Одна из разновидностей архитектур, построенная на базе MIND-чипов

Узел MIND (рис. 8) интегрирует блоки DRAM, логику и управление, необходимые для доступа к хранимым данным и выполнения операций над ними.

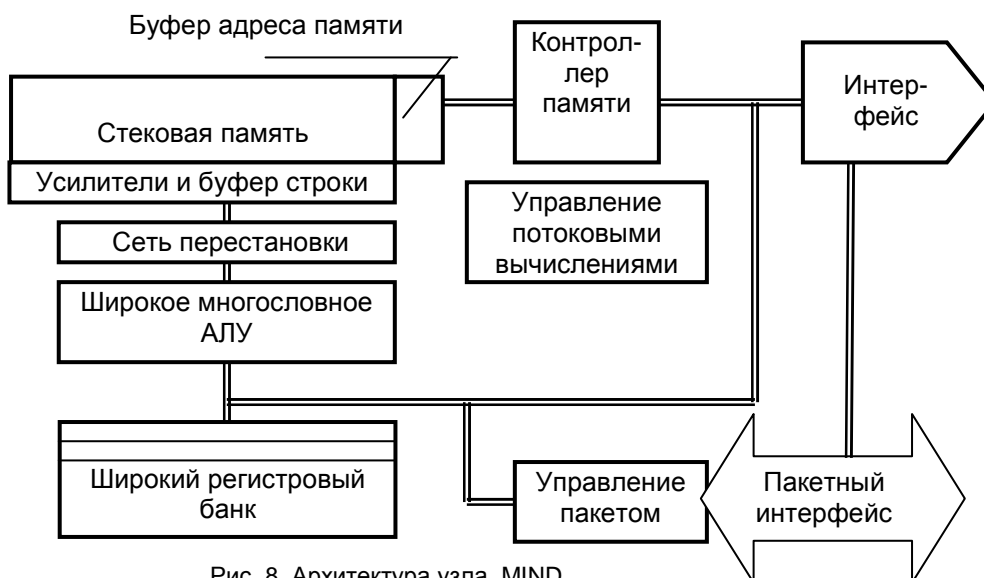


Рис. 8. Архитектура узла MIND

Локальные “широкие” регистры отображают множество функций, включая состояние потока, КЭШ команды, векторного регистра, буфера хранения трансляции, буфера пакета и временного



буфера строки памяти. Обработывающая логика MIND отличается от типичного ядра процессора RISC более простым управлением и очень “широкой” логикой АЛУ и регистров. При этом АЛУ допускает двойные (или соединенные) операции на полях значений, связанных с ключом или тэгом, и может выполнять операции с плавающей точкой за небольшое количество тактов на множественных парах данных одновременно. Сеть перестановки, связанная с АЛУ, позволяет осуществлять перестановку байтов в полной (“широкой”) строке. Такие перестановки могут быть выполнены в одном цикле, включая выравнивание и распределение “один ко многим”.

Внутренняя структура устройства MIND построена так, чтобы использовать на один – три порядка выше полосу пропускания памяти к логике АЛУ, чем достигнуто в обычной многопроцессорной архитектуре. Архитектура узла MIND содержит простой, но эффективный многопоточный механизм обработки команды. Банк широких регистров может использоваться в различных случаях, включая состояние потока. Узел содержит также средства контроля состояния каждого потока, представленного широким регистром.

Ввод - вывод в Gilgamesh управляется индивидуальными чипами MIND, которые поддерживают основной протокол интерфейса шины памяти, высокую полосу пропускания потокового интерфейса и могут использоваться независимо или вместе для управления внешней памятью и интерфейсами пользователя. Маршрутизация через глобальные межсоединения Gilgamesh архитектуры также выполнена с помощью чипов MIND.

Важная проблема, которую необходимо решить при построении и использовании всех перечисленных выше систем, состоит в том, чтобы разделить первоначальную программу решения задачи на несколько частей (подзадач), оптимально их распределить по различным процессорам (часто отличающимся по вычислительной мощности и другим параметрам) с целью снижения времени их простоя. Существуют различные подходы для решения такой проблемы. Например, в [8] предлагается система SAGE (*Statement-Analysis-Grouping-Evaluation*), которая может анализировать и разделять исходную программу на отдельные блоки, определять “вес” каждого блока и затем отправлять соответствующие задания процессорам памяти и главному компьютеру. Применение этой программы может обеспечить существенное ускорение решения задачи (до 3,87 раза для PIM-архитектуры, описанной в [8]).

#### **4. Особенности программного обеспечения PIM-систем**

Разработка программной инфраструктуры для PIM-систем содержит три главных проблемы, которые должны быть решены:

- 1) распределение данных и балансирование загрузки всех узлов системы;*
- 2) обеспечение эффективного использования параллелизма узла;*
- 3) проблема отказоустойчивости системы.*

В [6, 7] рассматривается подход к решению первой проблемы за счет применения уровня микропрограммных средств, который назван уровнем макросерверов. Макросерверы – это прикладные программы, работающие в своем собственном виртуальном адресном пространстве и выполняемые как асинхронные сети динамических объектов. Макросерверы формируют переменные, методы, потоки и внешние интерфейсы с расширенными функциональными

возможностями для реализации вычислений на PIM-системах, предлагая мощный механизм поддержки не только адаптивных и динамических цифровых кодов, но также и интеллектуальных исполнительных устройств. Макросерверы обеспечивают богатую семантику для распределения групп данных через узлы архитектуры, включая правильные и нерегулярные отображения, динамическое перераспределение и средство для определяемых пользователем специализированных классов, поддерживающих специфические методы доступа и стратегии трансляции. Распределенная структура данных, содержащаяся в макросервере, обычно привязывается к набору серверов вершин (узлов), поддерживающих локальное управление связанных дистрибутивных сегментов.

Приложения выполняются как асинхронные сети макросерверов. Сложное приложение может отобразить параллелизм на многих уровнях абстракции.

Клиентами функциональных возможностей макросервера, главным образом, являются компиляторы языка высокого уровня и системы спецификаций: пользователи могут программировать на более высоком уровне абстракции, возможно, в директивно-основанном расширении ФОРТРАНа 95 или C/C ++, или полностью на новом языке.

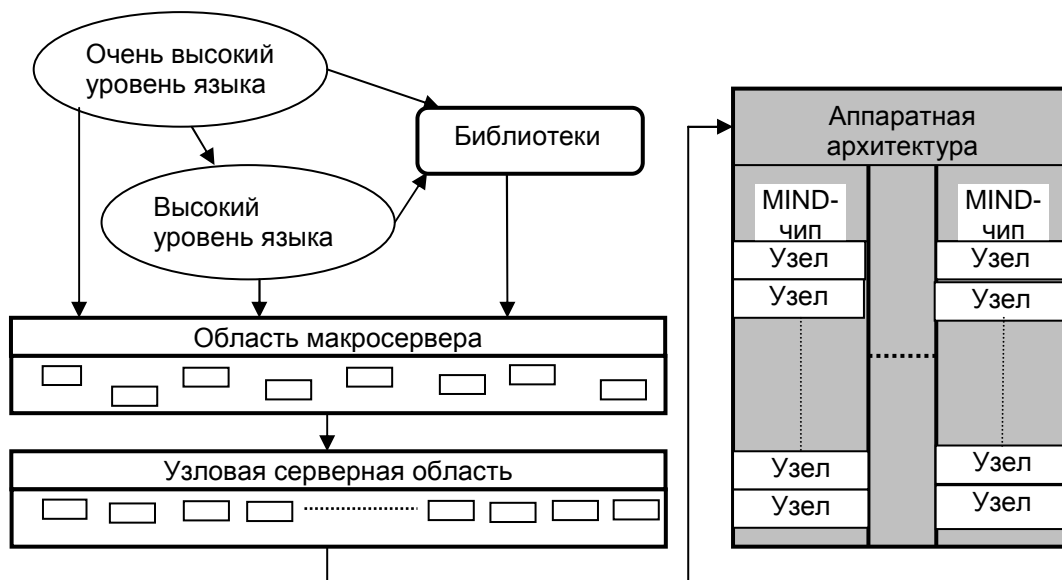


Рис. 9. Связь макросерверов системы Gilgamesh

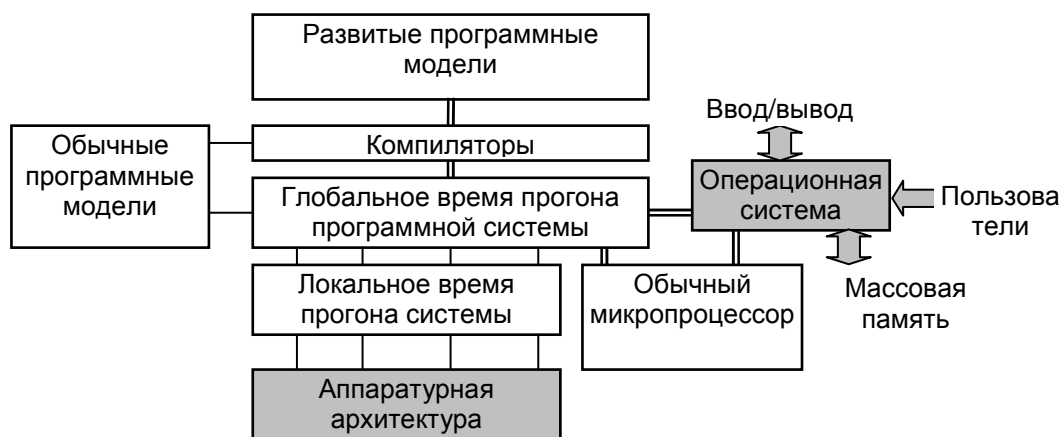


Рис. 10. Вариант набора компонентов программного обеспечения PIM – системы, показанной на рис. 6

Рис. 9 иллюстрирует связь макросерверов в программной архитектуре системы Gilgamesh между языками высокого уровня и системами спецификации очень высокого уровня, с одной стороны, и программно-аппаратной инфраструктурой нижнего уровня, с другой стороны [6].

Функциональные возможности макросервера могут быть реализованы с использованием серверов узла, и их работа с сетями реализуется с помощью пакетов.

Общий вид набора компонентов архитектуры программного обеспечения PIM – системы типа, изображенного на рис. 6, приведен на рис. 10 [10].

## **5. Предложения по совершенствованию компьютерных средств класса “процессор-в-памяти”**

Анализ приведенных выше архитектурных решений PIM-устройств позволил выделить следующие их недостатки: специализация структуры кристалла БИС и соответствующего программного обеспечения под конкретный класс задач, что приводит к мелкосерийному производству и, следовательно, к высокой стоимости изделия; неэффективное использование распределенных по массиву памяти процессоров из-за наличия жестко сформированных при изготовлении БИС связей между процессорами и массивом памяти, а также из-за их использования при решении вспомогательных задач, связанных с чтением, записью и передачей широкоформатного слова по линиям связи; трудность модификации и изменения схмотехники кристаллов PIM, на котором размещена DRAM-память; трудность верификации изделия из-за наличия на кристалле отличающихся по структуре и технологии изготовления памяти и процессора и др.

Совершенствование компьютерных средств класса “процессор-в-памяти” предлагается путем расширения функциональных возможностей системы в целом и её компонентов за счет динамической настройки архитектуры системы на класс решаемых задач, а также за счет оптимальной загрузки всех процессорных элементов полезной (вычислительной) работой и использования в составе БИС памяти новых схмотехнических решений, обеспечивающих возможность верификации и отладки как системы в целом, так и при необходимости особо ответственных узлов [11]. Тем самым обеспечивается возможность гибкой настройки архитектуры системы памяти, в частности, 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. При этом передачи могут осуществляться не только полной  $N$ -разрядной строкой, но и произвольными кратными  $R$ -разрядными группами (например, частями строки разрядностью  $R, 2R, \dots, \alpha R; \alpha R \leq N$ ) как со своими соответствующими контрольными разрядами, так и без них. Для управления процессом настройки и вычислительным процессом используется управляющий пакет, который, помимо управляющих полей общего назначения, содержит поля и признаки, отражающие последовательность действий PIM-систем, обеспечивающих:

1) настройку архитектурно-структурного образа системы перед запуском её на решение конкретной задачи, а также при необходимости перестройку системы в процессе работы путем подключения соответствующего ведущего процессора, выбора типа и количества соответствующих

банков памяти, на которые разделен весь массив памяти, а также выбора типа и количества соответствующих подбанков памяти в составе выбранных банков;

2) различные режимы работы системы памяти, в том числе:

- в качестве обычной памяти с использованием емкости памяти, размещенной только на кристалле (чипе);
- в качестве обычной памяти с использованием емкости памяти на кристалле и дополнительной внешней памяти, подключенной к чипу через интерфейс внешней памяти;
- в качестве “процессора-в-памяти” с использованием ресурсов обработки информации только собственного кристалла;
- в качестве “процессора-в-памяти” с использованием собственных ресурсов обработки информации и дополнительных ресурсов за счет других чипов, подключенных через интерфейс ввода/вывода и интерфейс внешней памяти ;

3) применение либо отключение контрольных кодов обнаружения и исправления ошибок;

4) формирование управляющих сигналов и сигналов синхронизации для обеспечения работы всех узлов системы памяти, в том числе:

- управление работой КЭШ-памяти с учетом реализации механизмов когерентности;
- реализация режимов “Чтение/Запись” как при обращении к памяти внутри чипа, так и за его пределами через интерфейс внешней памяти и др.

Для реализации этих функций в состав чипа памяти введены (рис.11) устройство анализа входного управляющего пакета и выработки всей совокупности управляющих сигналов, необходимых для функционирования БИС; набор проблемно-ориентированных ведущих процессоров (ВП); набор селекторов для выбора и активации ведущих процессоров, а также соответствующих банков (БП) и подбанков (ПБП); коммутатор (К) и регистр ( $R_{\text{ПБП}}$ ) в каждый ПБП; интерфейс связи с внешней памятью, интерфейс связи с аналогичными чипами или с HOST-машиной, интерфейс для загрузки программ и данных в соответствующие массивы памяти. При этом каждый ПБП так же, как и в рассмотренных выше PIM-архитектурах, содержит процессор (процессорное ядро – ПЯ), которому принадлежит массив памяти (М). Учитывая, что предлагаемая система может быть рекомендована к массовому производству, так как фактически она является универсальной за счет настройки её архитектуры, введено устройство интерфейса диагностики и отладки, потому что помимо регулярного массива памяти в него встроены нерегулярные логические схемы, которые затрудняют верификацию и отладку БИС. Эта встроенная избыточность, обеспечивающая специальную поддержку внешнего испытательного оборудования, позволяет снизить затраты на тестирования и отладку БИС.

## 6. Выводы

Учитывая, что в настоящее время в мире насчитывается несколько десятков высокопроизводительных систем, построенных на принципах PIM-архитектуры, а также мнения зарубежных специалистов, что буквально через 5 – 7 лет такие системы станут доминирующими, исследования в области архитектурно-структурной организации систем такого класса являются весьма актуальными. При этом предложения, направленные на устранение существенных

недостатков систем такого класса, открывают реальную возможность эффективного их использования как самостоятельно с HOST-машиной, так и в качестве приставки для супер-ЭВМ и кластеров при решении сложных задач, подлежащих распараллеливанию.

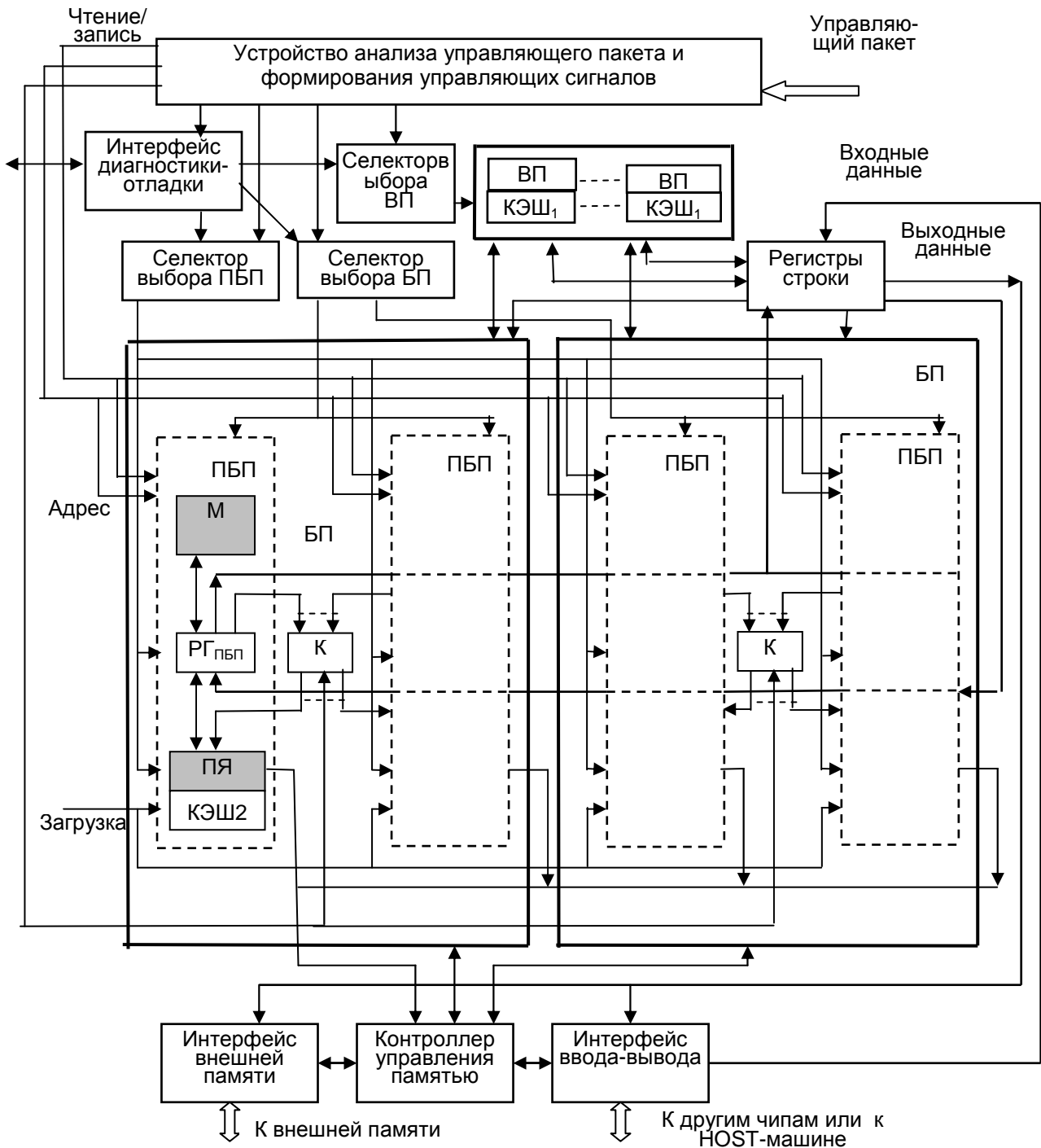


Рис. 11. Упрощенная структурная схема PIM-системы на чипе с настройкой архитектуры и динамическим распределением ресурсов

Реализация систем такого типа требует достаточно большой площади кристалла. Ожидается, что к 2009 г. чип памяти типа DRAM будет иметь ёмкость 2Гбайта. При этом существенно увеличится

степень интеграции БИС. В связи с этим можно утверждать, что приведенные выше предложения по совершенствованию PIM-архитектур за счет введения на кристалл дополнительных функциональных узлов [11] не противоречат общей тенденции развития интегральной технологии, обеспечивая при этом (по сравнению с классическими подходами к построению PIM-систем) повышение производительности и расширение функциональных возможностей за счет оптимальной загрузки распределенных по массиву памяти процессоров, динамической настройки архитектуры системы и перераспределения ресурсов при решении различных классов задач. Тем самым БИС с архитектурой "процессор-в-памяти" из специализированных становятся БИС широкого назначения, что, естественно, приводит к высокой серийности изделия и снижению их стоимости.

Для создания таких БИС необходимо выполнить комплекс исследований по организации их динамической настройки, распределению данных внутри среды хранения информации, управлению вычислительным процессом и созданию принципов построения соответствующего программного обеспечения. Большое значение приобретают при этом работы по созданию новой интегральной технологии изготовления на кристалле элементов логических схем, которая должна быть совместимой с технологией изготовления на этом же кристалле элементов памяти. На настоящий же момент времени применение ПЛИС с аппаратной поддержкой процедур, характерных для PIM-устройств, позволит практически реализовать и проверить эффективность предлагаемых решений, сократив сроки и стоимость разработки системы в целом.

## СПИСОК ЛИТЕРАТУРЫ

1. Carlos Carvalho. The Gap between Processor and Memory Speeds. – <http://www.lpdps.eece.unm.edu/1998/paper/074.pdf>.
2. 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.
3. Landis David, Hulina Paul and Deno Scott, Roth Luke and Coraor Lee. Evaluation of computing in memory architectures for digital image processing applications. – <http://csdl.computer.org/comp/proceeding/iccd/1999/0406/00/0406014abs.htm>.
4. Roth Luke, Coraor Lee, Landis David, Hulina Paul and Deno Scott. Computing in Memory Architectures for Digital Image Processing. – <http://www.csdl.computer.org/comp/proceedings/mtdt/1999/0259/00/02590008abs.htm>.
5. Suh Jinwoo and Crago Stephen P. PIM- и потоковый процессор для применений обработки радарных сигналов. – <http://www.pdcl.eng.wayne.edu/msp01/paper8.pdf>.
6. Sterling Thomas L. and Zima Hans P. Gilgamesh: A Multithreaded Processor-In-Memory Architecture for Petaflops Computing. – <http://www.sc-2002.org/paperdfs/pap.pap105.pdf>.
7. Zimaa Hans Pand, Sterlingb Thomas L. The Gilgamesh Processor\_in\_Memory Architecture and Its Execution Model. – <http://www.icsa.informatics.ed.ac.uk/cpc2001/proceedings/zima.ps>.
8. Huang Tsung-Chuan, Chu Slo-Li .A Parallelizing framework for intelligent memory architectures. – <http://www.parallel.iis.sinica.edu.tw/cthpc/7thpc/7th/03C THPC2001-Hardcopy.pdf>.
9. Huang M., Renau J., Yoo S-M. and Torrellas J. Energy/performance design of memory hierarchies for processor-in-memory chips. – <http://www.chooyu.cs.uiuc.edu/~renau/docs/verlag00.pdf>.
10. Sterling T. Keeping programming in MIND for the MTV petaflops generation. – [http://www.ecse.monash.edu.au/conferences/ics/2002/slides/panell\\_sterling.pdf](http://www.ecse.monash.edu.au/conferences/ics/2002/slides/panell_sterling.pdf).
11. 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 збереження та обробки інформації на одному кристалі. Деклараційний патент на корисну модель. №6259. G06F13/00, G06F12/00. 15.04.2005. Бюл. №4. Заявлено 15.02.2005.