

УДК 004.78

*И.И. Левин*

Южный научный центр РАН, г. Таганрог, Россия, levin@mvs.tsure.ru

*П.М. Коновальчик, А.И. Иванов, А.Д. Малеванчук*

В/ч № 26165, г. Москва, Россия, andrey\_iv@mail.ru

## Многопроцессорная система, адаптируемая под информационную структуру задач различных классов

Рассматриваются принципы построения многопроцессорных систем на основе ПЛИС-технологии, структура которых может адаптироваться к структуре решаемой задачи или ее фрагменту. Адаптированная многопроцессорная система строится из унифицированных базовых модулей, состоящих из множества макрообъектов, каждый из которых содержит объекты преобразования информации, коммутационные элементы и контроллеры распределенной памяти, а также локальное устройство управления. Состав базового модуля обеспечивает эффективную реализацию различных классов вычислительных задач.

Для решения расчетоемких задач в реальных постановках необходимо использовать параллельные вычислительные системы, поскольку однопроцессорные ЭВМ имеют недостаточно высокую производительность. В то же время реальная производительность многопроцессорных вычислительных систем (МВС), которые ориентированы на традиционные методы распараллеливания, зачастую не превышает 10 – 15 % от заявляемой пиковой производительности [1] вследствие необходимости реализации множества процедур межпроцессорного обмена, а также синхронизации последовательных процессов, выполняемых в процессорах системы. В этом случае накладное время, необходимое для организации параллельного вычислительного процесса, оказывается большим, чем собственно параллельные вычисления.

Известно, что максимальная скорость обработки информации может быть обеспечена в специализированной вычислительной системе, как правило, структурно реализующей решение задачи. Однако появление новых алгоритмов обработки информации требует разработки и создания различных проблемно-ориентированных вычислителей, что приводит к увеличению стоимости решения как отдельной прикладной задачи, так и всего комплекса решаемых задач.

В этой связи актуальной является разработка реконфигурируемых вычислительных систем, в рамках архитектуры которых пользователь мог бы синтезировать виртуальные параллельно-конвейерные вычислители, структура которых была бы адекватна информационной структуре задачи или ее фрагменту [2].

Синтез виртуальной проблемно-ориентированной архитектуры в рамках универсальной многопроцессорной системы осуществляется путем соответствующей настройки компонентов системы: процессоров – на выполнение определенных арифметико-логических операций, контроллеров распределенной памяти – на определенные процедуры обращений к каналам распределенной памяти, коммутаторов – на определенные коммутационные структуры, обеспечивающие пространственное соединение между собой множества процессоров и каналов распределенной памяти в соответствии с информационным графом задачи.

В то же время наиболее производительной будет система, архитектура которой будет адаптирована к структуре решаемой задачи на уровне логических элементов, однако такой подход потребует чрезвычайно сложного управления и может привести к резкому увеличению времени перестройки системы с одной вычислительной структуры на другую.

За последнее десятилетие сформировалось мнение, что комбинация микропроцессор/программные средства служит основой гибкости и функциональности и лишь в некоторых случаях обеспечивает решение проблемы быстродействия, в то время как аппаратные средства обеспечивают необходимое быстродействие при функциональной неизменности [3]. Согласно принципу дуальности аппаратно-программной реализации вычислений [4], [5], любая вычислительная компонента может быть выполнена как программно, так и аппаратно. Появившиеся в последние годы системы с реконфигурируемой структурой подтверждают данную гипотезу. В то же время появление программируемых логических интегральных схем (ПЛИС) не изменило ситуацию: устройства, построенные с применением микропроцессоров, называют универсальными, а устройства, использующие аппаратные ускорения, – специализированными, или проблемно-ориентированными. Даже комплексные системы, которые содержат как микропроцессоры, так и ПЛИС, не выделены в отдельный класс. Такие устройства называют гибридными, различая даже в одной архитектуре две независимые подсистемы.

Такое разделение связано с некорректной трактовкой, что программируемые логические устройства являются технологически удачной заменой специализированных устройств. Это справедливо, но только отчасти, ведь и микропроцессоры разрабатывались как замена специализированных устройств. Можно утверждать, что микропроцессоры и ПЛИС являются полуфабрикатами, оба класса устройств не могут функционировать без дополнительных информационных средств. Такую специфику микросхем принято называть программированием. Для макропроцессоров присуще программирование во времени, а для ПЛИС – программирование в пространстве [3].

В процессе работы процессор последовательно получает команды и в соответствии с каждой из них осуществляет преобразование информации. Усложнение алгоритма приводит к увеличению типа команд и времени обработки. Для ПЛИС алгоритм выполняется в единой структуре (максимально параллельно). Усложнение алгоритма приводит к увеличению числа задействованных логических элементов и площади кристалла.

Микропроцессоры и ПЛИС можно рассматривать в качестве граничных элементов, реализующих алгоритмы в различных физических пространствах (временные модели). Промежуточным решением, которое позволило бы объединить преимущества обоих методов обработки информации, являются динамически перестраиваемые программируемые логические элементы, у которых должна существовать возможность динамически в процессе решения задачи заменять всю структуру устройства или ее часть.

В ряде современных ПЛИС уже встраиваются макропроцессоры.

В основе предлагаемой компанией Altera микросхемы Excalibur может лежать как ядро ARM922, так и ядро MIPS32 4Кс. Названная системой на программируемом кристалле (system on a programmable chip), она содержит несколько фиксированных периферийных узлов типа универсальных асинхронных приемопередатчиков и таймеров. Обмен данными между процессором и имеющейся на кристалле памятью и периферийными узлами, а также между процессором и программируемой логической матрицей осуществляется по двухуровневой 32-разрядной шине конвейерного типа

АМВА с тактовой частотой до 200 МГц. Микросхема Excalibur поддерживает несколько вариантов конфигурирования. В варианте PLD first загрузка статической аппаратной конфигурации в микросхему осуществляется из предварительно запрограммированной ПМЛ (программируемая матричная логика PLD array).

Производитель программируемых логических устройств – компания Xilinx предлагает свой вариант подобных устройств, названный им Platform FPGA. Микросхема Platform FPGA является комбинацией программируемой логической архитектуры Virtex-II от Xilinx и процессорного ядра IBM PowerPC 405. Для реализации на базе FPGA за пределами процессора других аппаратных IP-блоков используются две технологии: IP Immersion, обеспечивающая размещение аппаратных IP-ядер в любом месте матрицы, и Active Interconnect, на базе которой создаются каналы передачи данных между аппаратными и программируемыми элементами независимо от их местоположения в матрице. Используя Platform FPGA, отдельные элементы можно разрабатывать на базе программируемой логики. Затем, из соображений объема производства, необходимости снижения размеров кристалла и/или стоимости, их можно реализовать в виде жёсткой логики. Элементы схемного решения, обеспечивающие гибкость дизайна, можно держать в виде программируемой логики, одновременно снижая себестоимость и размеры уже зашитых элементов. Поскольку программируемая матрица это своего рода ОЗУ, ее можно перепрограммировать, используя как внутрикристалльный процессор, так и внешние источники.

Компания «Xilinx» предлагает также технологию ExtremeDSP, в основе которой лежит архитектура Virtex-II и такие базовые элементы, как таблицы поиска и регистры, из которых могут строиться различные функциональные устройства типа умножителей-аккумуляторов. Пользуясь этой технологией, алгоритмы цифровой обработки сигналов можно создавать в программируемой матрице в виде конвейеров из подобных устройств, которые могут обеспечивать производительность до 160 миллионов выборок в секунду. Модификация алгоритма при этом может осуществляться путем динамической загрузки в FPGA новой конфигурации.

Примером еще одной из систем на кристалле с высокой степенью конфигурируемости является реконфигурируемый коммуникационный процессор CS2000 компании Chameleon Systems. В состав CS2000 входит RISC-процессор и программируемая матрица (RFP reconfigurable processing fabric). Однако это не обычная программируемая логическая матрица, а более специализированная, ориентированная исключительно на беспроводные телекоммуникационные цифровые системы. Интересной особенностью является и то, что сам процессор (ядро ARC компании Advanced RISC Cores) тоже конфигурируемый. Кроме процессора и реконфигурируемой вычислительной структуры, в состав CS2000 входят 32-разрядный PCI-контроллер, 64-разрядный контроллер памяти и 16-канальный контроллер прямого доступа в память. Внутренний обмен данными между различными подсистемами осуществляется по шине RoadRunner, представляющей собой 128-разрядную шину с разделением транзакций и пропускной способностью 2 Гбайт/с (в обменах между подсистемами на одном кристалле). Кроме того, имеется 160 выводов для подключения программируемых устройств ввода/вывода.

Структура RFP делится на четыре полосы (slice), каждая из которых представляет отдельный независимо конфигурируемый базовый компонент. Каждая полоса состоит из трех пластин (tile), которые, в свою очередь, состоят из семи 32-разрядных узлов передачи данных DPU (data path unit), двух одноктактных устройств умножения 16-разрядного числа на 24-разрядное, четырех внутренних 32-разрядных буферов памяти на 128 элементов и блока управления (CLU). Друг с другом все модули соединяются динамическими линиями связи.

Не исключено, что в ближайшее время определяющее значение будет иметь динамическая реконфигурируемая система, поэтому разработка комплекса аппаратно-программных средств, учитывающих программирование в пространстве (конфигурации структур) и во времени (последовательности команд), перспективна и актуальна, если учесть, что удельная производительность ПЛИС на 2 – 3 порядка превышает данный показатель для микропроцессоров.

Реконфигурируемая многопроцессорная система должна строиться из модулей, связанных между собой пространственной коммутационной структурой. В общем случае структура модулей системы может быть различной, хотя для повышения технологичности целесообразно, чтобы реконфигурируемая система строилась бы из унифицированных базовых модулей.

Базовый модуль состоит из множества макрообъектов, каждый из которых представляет собой ПЛИС высокой степени интеграции, содержащей несколько миллионов эквивалентных вентилей.

Если для МВС со структурно-процедурными вычислениями можно обозначить реализуемые функциональные блоки (макропроцессоры, реализующие обработку информации, макропамять, обеспечивающую хранение информационных массивов, а также скоростной параллельный доступ и макрокоммутатор, реализующие соединение компонентов системы, в том числе и динамически), то для адаптируемых многопроцессорных систем даже функции объекта могут изменяться в процессе решения задачи. Устройство обработки информации может в следующий момент времени быть преобразовано в память, а коммутатор стать функциональным устройством. Разумеется, кроме ПЛИС, базовый модуль должен содержать стандартные микросхемы памяти, которые фиксировано связаны с одной или несколькими ПЛИС. Архитектура базового модуля и системы в целом может произвольным образом изменяться.

Каждый макрообъект, входящий в состав базового модуля, может содержать множество разнородных объектов, объединенных между собой некоторой внутренней коммутационной структурой, которая позволит соединить объекты в функционально законченный блок, а также обеспечивает транзит информации между различными макрообъектами. Кроме того, каждый макрообъект обязательно содержит интерфейсную схему, которая обеспечивает корректное взаимодействие между объектами, принадлежащими различным макрообъектам.

Объектами могут быть: элементарный процессор (ЭП), контроллер распределенной памяти (КРП), преобразователь информации (П) или динамический коммутатор (ДК).

Элементарный процессор осуществляет обработку пары операндов в соответствии с кодом операции, а также обеспечивает синхронизацию потоков операндов и реализацию структур операций (операции с константой, с обратной связью, с запаздыванием). Преобразователь информации обеспечивает реализацию фиксированной функции, которая не может быть изменена при данной конфигурации ПЛИС. Следует отметить, что функциональное преобразование может быть выполнено над группой операндов, например, скалярное произведение векторов, численное интегрирование, умножение матриц, решение системы логических уравнений, при этом в преобразователе информации синхронизация информационных потоков, как правило, не обеспечивается.

Совокупность элементарных процессоров, настроенных на выполнение определенных арифметико-логических операций, и преобразователей информации, входы которых соединены внутренней пространственной коммутационной структурой в соответствии с информационным графом решаемой задачи, образуют вычислительную структуру, реализующую обработку информационных массивов в

соответствии с алгоритмом решения задачи (фрагмента решения задачи). В вычислительную структуру могут быть включены дополнительные узлы и блоки, например буферы данных для синхронизации информационных потоков или коммутаторы-переключатели.

Информационные потоки в вычислительной структуре образуются с помощью контроллеров распределенной памяти, которые организуют скоростные параллельные обращения к сегментированной внутренней и внешней памяти макрообъекта. Динамические коммутаторы обеспечивают переключение информационных потоков между источниками и приемниками, причем изменение адреса источника (приемника) может производиться произвольно с каждым данным.

На рис.1 представлена структурная схема макрообъекта.

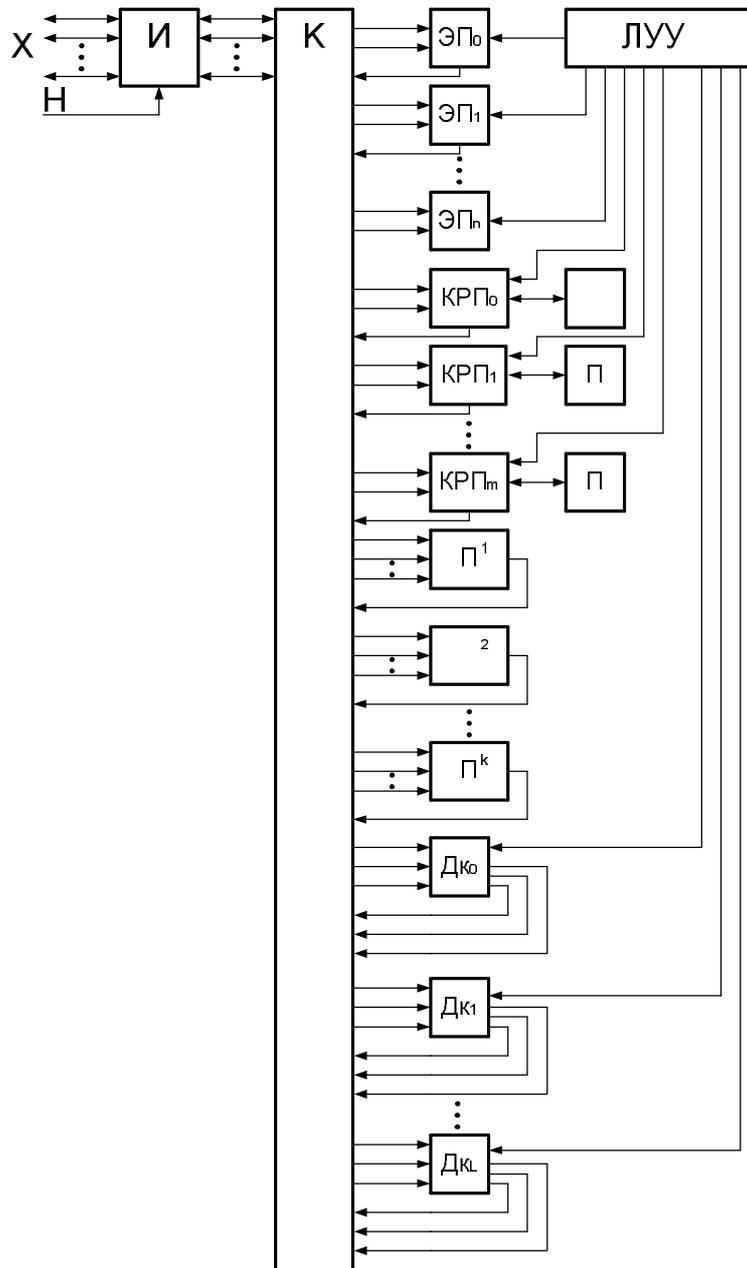


Рисунок 1

В макрообъект может быть введено логическое устройство управления (ЛУУ), которое дополняет архитектуру перестраиваемого параллельного конвейера и делает систему практически универсальной.

Логическое устройство управления обслуживает параллельный конвейер макрообъекта, а также реализует обработку нештатных ситуаций. Необходимость введения ЛУУ в состав макрообъекта объясняется тем, что наличие только одной host-машины приведет к проблеме «узкого горла». Загрузка данных и программ в макрообъекты, а также управление и диспетчеризация множеством конвейеров от единственного устройства управления может привести к значительному снижению производительности.

Введение ЛУУ в каждый макрообъект снимает указанные проблемы. Управление процессом вычислений будет реализовано по иерархическому принципу: host-машина передает указания ЛУУ, которые непосредственно управляют вычислительными процессами в макрообъектах.

Информационная связь между базовыми модулями осуществляется с помощью коммутационной пространственной системы, которая реализована внутри базового модуля. Коммутационная система базового модуля реализует не только информационные связи между компонентами базового модуля, но и информационные связи между БМ. Подобная организация коммутационной структуры обеспечивает однородность вычислительной системы (система состоит только из базовых модулей) и высокую технологичность организации вычислительных процессов.

Обмен информацией между объектами, принадлежащими разным базовым модулям, организуется точно так же, как и обмен между объектами внутри базового модуля.

Адаптированная система может реализовать самые разнообразные структуры даже при условии, что связи между макрообъектами останутся неизменными, что соответствует соединению выводов ПЛИС на плате базового модуля. В более общем случае это не так, поскольку коммутационная структура макрообъектов может обеспечивать не только соединение объектов макрообъекта в единую вычислительную систему, но и коммутацию внешних входов/выходов для обеспечения транзита информации.

Рассмотрим базовый модуль адаптивной МВС, содержащей  $n \times m$  макрообъектов, образующих цилиндрическое соединение (рис. 2), вертикальные связи первой и последней строк макрообъектов используются для наращивания.

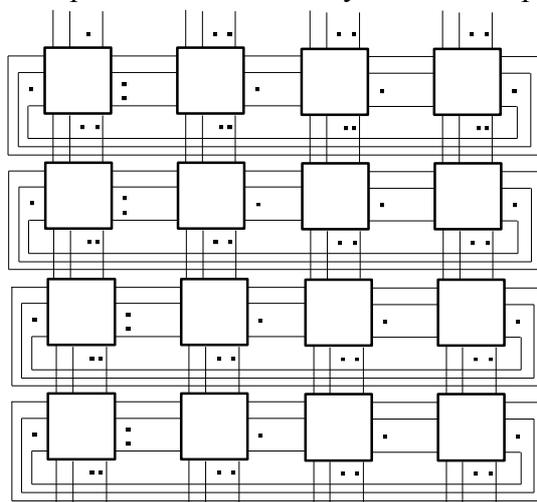


Рисунок 2

Перестраивая макрообъекты на функции обработки информации и функции информационных потоков, можно получить различные конфигурации базового модуля, представленные на рис. 3. Здесь черный макрообъект соответствует элементу распределенной памяти, а белый – функции обработки информации.

При перераспределении функциональных блоков можно получить различные проблемно-ориентированные архитектуры, эффективно реализующие определенный класс задач.

Конвейерная обработка информации на базовом модуле адаптивной многопроцессорной системы приведена на рис. 3а. Здесь имеется единственный источник информационного массива, который проходит через последовательность функциональных блоков. Результат преобразования информации записывается в приемник.

Более сложная схема параллельного конвейера обработки информационных структур показана на рис. 3б. В данном примере реализовано два независимых конвейера. Подобная организация вычислений характерна для решения задач символьной обработки информации, цифровой обработки изображений и сигналов. Следует отметить, что между функциональными макрообъектами, образующими вычислительный конвейер, может быть сложная взаимосвязь.

На рис. 3в приведена вычислительная структура базового модуля, настроенного на реализацию четырех независимых конвейеров, а на рис. 3г – четырех взаимосвязанных конвейеров. Такая организация вычислений характерна для задач математической физики.

На основании принципов адаптации структуры многопроцессорной системы можно получить векторную архитектуру (рис. 3д) или матричную (рис. 3е). Принципиальным отличием от традиционных архитектур является то, что элемент обработки информации представляет собой не процессор, последовательно осуществляющий преобразование информации, а сложное функциональное устройство, состоящее из множества взаимосвязанных пространственной коммутационной структурой объектов. Обработка информации осуществляется структурно.

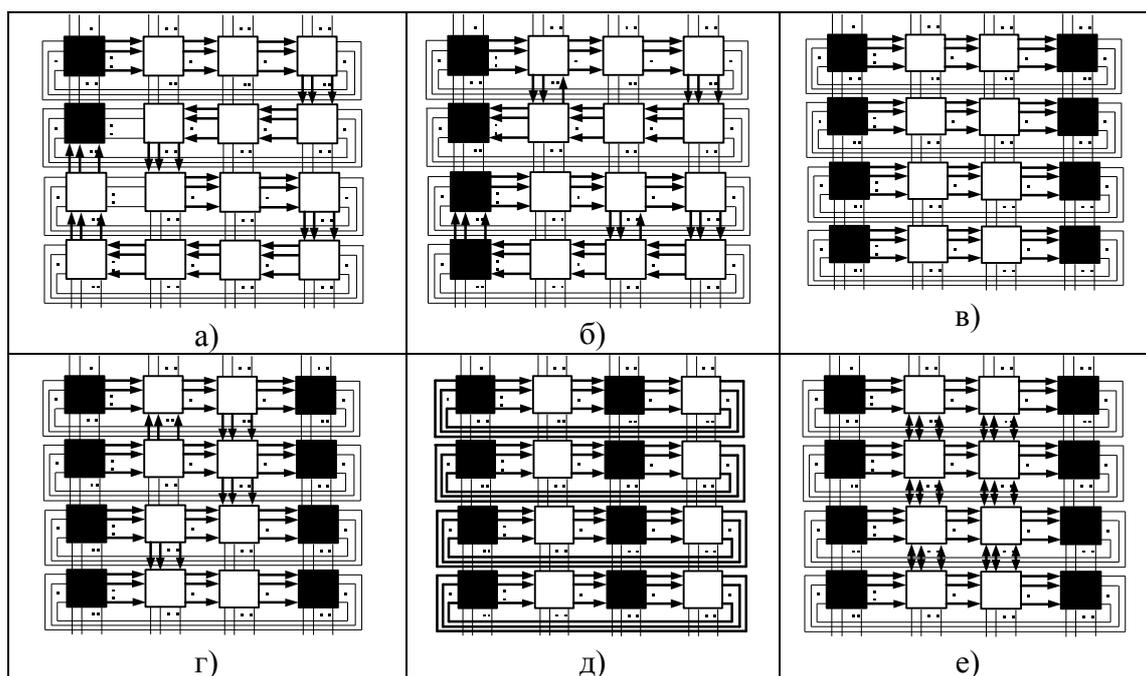


Рисунок 3

Для задач математической физики функциональный макрообъект осуществляет обработку информации в узле сеточной области, при этом элементарный процессор настроен на выполнение определенной операции, выполняемой в узле сетки. Динамические коммутаторы и контроллеры распределенной памяти организуют подачу данных, соответствующих значению информационных массивов в узлах сеточной области. При этом возможно, что макрообъект реализует несколько узловых процессоров.

Для задачи цифровой обработки сигналов функциональный макрообъект осуществляет структурную реализацию базовых операций, например процедуры быстрого преобразования Фурье, процедуры Уолша, косинусного преобразования. Размерность базового преобразования определяется вычислительным ресурсом макрообъекта. При решении задач цифровой обработки сигналов целесообразно использовать не только элементарные процессоры, способные динамически переключаться на выполнение различных команд, но и проблемно-ориентированные преобразователи информации с неизменной структурой, применение которых позволяет увеличить скорость обработки информации от двух до десяти раз.

Для задач символьной обработки информации использовать специализированные преобразователи информации является обязательным. Это обеспечивает увеличение производительности на два-три порядка.

Матричная и векторная организация связей эффективны не только для задач линейной алгебры, но и для решения систем логических уравнений и функционального моделирования. Кроме того, матричная структура может быть эффективно использована для реализации нейросетевых алгоритмов.

Если учесть, что топология межмакрообъектных связей может быть более сложной, чем цилиндрическая, например гиперкуб, то тогда число возможных проблемно-ориентированных архитектур, потенциально реализуемых в базовом модуле МВС, многократно возрастает.

Топология связей гиперкуба приведена на рис. 4.

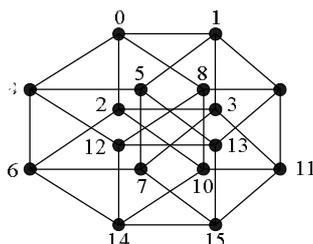


Рисунок 4

Следует отметить, что для базового модуля, содержащего 16 макрообъектов (большее количество вряд ли удастся реализовать на одной плате), топология гиперкуба реализуется в тороидальной структуре, представленной на рис. 5. Номера вершин гиперкуба (рис. 4) и макрообъектов (рис. 5) совпадают.

Каждый макрообъект связан по четырем направлениям с макрообъектами, номера которых записаны в двоичном коде, отличаются инверсией отдельного бита. Макрообъект, имеющий номер  $\langle x_0, x_1, x_2, x_3 \rangle$ , связан с объектами  $\{ \langle \bar{x}_0, x_1, x_2, x_3 \rangle, \langle x_0, \bar{x}_1, x_2, x_3 \rangle, \langle x_0, x_1, \bar{x}_2, x_3 \rangle, \langle x_0, x_1, x_2, \bar{x}_3 \rangle \}$ .

Таким образом, многопроцессорная система адаптирована к структуре задачи и, с одной стороны, обеспечивает высокую скорость обработки информации за счет высокой удельной производительности и конвейерной обработки данных, а с другой стороны, реализует фон-неймановский детерминизм вычислений и может быть основой для создания сверхвысокопроизводительных вычислительных систем.

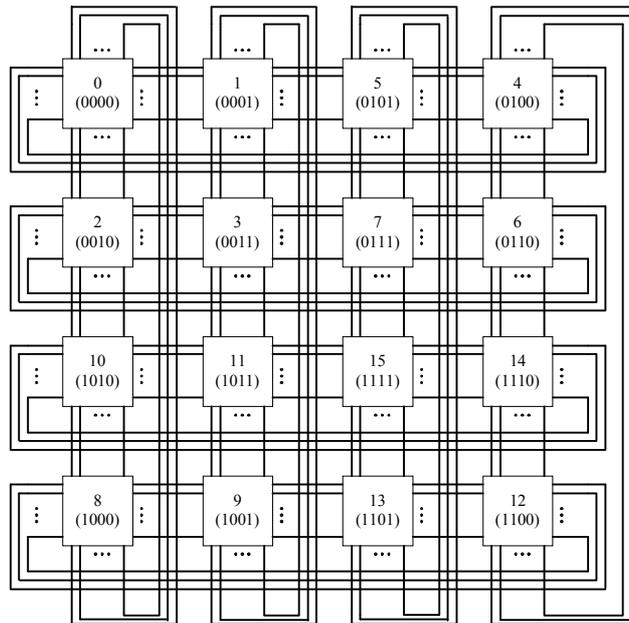


Рисунок 5

## Литература

1. Аладышев О.С., Дикарев Н.И., Овсянников А.П., Телегин П.Н., Шабанов Б.М. СуперЭВМ: области применения и требования к производительности // Известия ВУЗов. Электроника. – 2004. – № 1. – С. 13-17.
2. Каляев А.В. Программирование виртуальных архитектур в суперкомпьютерах с массовым параллелизмом // Информационные технологии и вычислительные системы. – 2000. – № 2.
3. Makimoto T. The Rising Wave of Field Programmability // Proc. of Tenth International Conf. on Field-Programmable Logic and Applications FLP-2000. – Villach (Austria). – 2000. – Springer Lecture Notes in Computer Science 1996. – P. 1-6.
4. Ясинявичус Р. Параллельные пространственно-временные вычислительные структуры. – Вильнюс: Моклас, 1988. – 183 с.
5. Каляев А.В. Многопроцессорные вычислительные системы с программируемой архитектурой. – М.: Радио и Связь, 1984. – 240 с.

***И.И. Левин, П.М. Коновальчик, О.И. Иванов, О.Д. Малеванчук***

### **Багатопроцесорна система, адаптована під інформаційну структуру задач різних класів**

Розглядаються принципи побудови багатопроцесорних систем на основі ПЛІС-технології, структура яких може адаптуватися до структури розв'язуваної задачі або її фрагмента. Адаптована багатопроцесорна система будується з уніфікованих базових модулів, що складаються з багатьох макрооб'єктів, кожний з яких містить об'єкти перетворення інформації, комутаційні елементи й контролери розподіленої пам'яті, а також локальний пристрій керування. Склад базового модуля забезпечує ефективну реалізацію різних класів обчислювальних задач.

***I.I. Levin, P.V. Konovalchuk, O.I. Ivanov, O.D. Malevanchuk***

### **Multiprocessor System Adapted in Informative Structure of Tasks of Various Classes**

In the article are viewed principals of multiprocessor systems design on the base of FPGA-technology. The structure of these systems is able to adapt to the structure of the solving task or its fragment. Specialized multiprocessor system is designed of unified basic modules which consist of number of macroobjects. Each macroobject includes objects of data transformation, switch elements and distributed memory controllers, and also local control device. The basic module structure provides effective realization of various classes of tasks.

*Статья поступила в редакцию 05.07.2004.*