РОССИЙСКАЯ АКАДЕМИЯ НАУК
СИБИРСКОЕ ОТДЕЛЕНИЕ
А В Т О М Е Т Р И Я

№ 2 1992

ОРГАНИЗАЦИЯ ВЫЧИСЛИТЕЛЬНЫХ СИСТЕМ НА НОВЫХ ОПТИЧЕСКИХ ПРИНЦИПАХ

УДК 681.323 : 535

В. А. Кострюков, В. П. Торчигин

(Москва)

ОПТИЧЕСКИЙ МНОГОПРОЦЕССОРНЫЙ ВЫЧИСЛИТЕЛЬНЫЙ КОМПЛЕКС, УПРАВЛЯЕМЫЙ ПОТОКАМИ ДАННЫХ

Рассматриваются возможности реализации вычислительной системы, использующей концепцию управления потоками данных, на основе оптического многопроцессорного вычислительного комплекса, время работы которого разделяется между тысячами виртуальных процессорных элементов. Показано, что трудности современных вычислительных систем, связанные с относительно большим временем распространения сигналов между логическими элементами, могут быть преодолены при использовании рассмотренного подхода.

Название данной статьи может вызвать естественный вопрос: имеет ли смысл рассматривать особенности оптических многопроцессорных вычислительных комплексов (МВК), управляемых потоками данных, в настоящее время, когда еще не созданы обычные фон-неймановские однопроцессорные оптические компьютеры, в которых, вместо электрических сигналов, используются оптические. В работе [1] показано, что если пытаться использовать основное преимущество оптики перед электроникой — широкополосность оптических линий связи (полоса пропускания которых, по крайней мере, в 1000 раз больше, чем электрических) — и ориентироваться на оптические вентили, в полной мере использующие это преимущество, то довольно простым образом можно получить на основе собранного из таких вентилей компьютера N одинаковых виртуальных компьютеров. При этом N равно отношению T/τ, где T — задержка в распространении сигналов с выхода одного вентиля до выхода другого, непосредственно с ним соединенного; τ — период следования обрабатываемых оптических импульсов, совпадающий с периодом синхроимпульсов (СИ). Предполагается, что задержки при передаче сигналов между вентилями одинаковы во всем компьютере и значительно (в 10^2—10^5 раз) превосходят период следования оптических импульсов. Обычно T измеряется наносекундами, τ — пикосекундами.

В той же работе показано, что для достаточно широкого класса задач, требующих больших объемов вычислений, суммарная производительность N виртуальных компьютеров не зависит от величины задержки T. Это легко понять, если иметь в виду, что количество виртуальных компьютеров пропорционально T/τ, а производительность каждого из них пропорциональна $1/T$. Суммарная производительность, равная произведению производительности одного компьютера на их общее количество, при этом не зависит от T и пропорциональна $1/\tau$.

Этот вывод позволяет разрешить острейшую в вычислительной технике проблему, являющуюся основным препятствием для повышения скорости работы устройств вычислительной техники, заключающуюся в том, что время переключения современных сверхбыстродействующих электронных вентилей оказывается значительно меньше времени передачи сигналов между

вентилями, расположенными даже на одном кристалле СБИС. Поэтому дальнейшее уменьшение времени переключения вентилей не имеет смысла, так как скорость работы устройства определяется главным образом задержками в передаче сигналов между вентилями.

Важно подчеркнуть, что отмеченная выше независимость общей производительности МВК от величины задержек справедлива только при использовании оптических линий связи, где имеет место волновой характер распространения сигналов (в отличие от «диффузионного», «теплового» характера передачи электрических сигналов в обычных компьютерах). В первом случае имеется возможность посылать в линию связи очередной сигнал, не дожидаясь приема предыдущего. Во втором случае такая возможность отсутствует. При «диффузионном» характере распространения сигналов следующий сигнал может быть послан только после того, как принят предыдущий.

Независимость общей производительности МВК от величины задержки в распространении сигналов между вентилями позволяет использовать оптические вентили, в которых имеет место длительное взаимодействие входных оптических сигналов в процессе их совместного распространения по нелинейному световоду. Противоречивые, на первый взгляд, свойства — длительные (~10 нс) взаимодействия коротких импульсов (10 пс) — позволят значительно снизить требования к свойствам используемых нелинейных материалов, постепенно накапливая эффекты при длительном взаимодействии.

Таким образом, рассмотренный в [1] подход к построению оптического компьютера использует еще одно отличие оптических систем от электрических, которое не отмечается при перечислении преимуществ оптики перед электроникой. Неизбежные задержки при передаче сигналов между вентилями и задержки в самих вентилях, являющиеся, безусловно, отрицательным фактором в электронике, могут быть нейтрализованы при оптической обработке сигналов путем использования этих задержек для хранения результатов логических преобразований в процессе их распространения между вентилями и в самих вентилях.

Из вышеизложенного следует, что указанный в названии статьи оптический МВК является не композицией многих отдельных компьютеров, полученной методом мультиплицирования оборудования, как это принято в электронике, а возникает естественным образом как результат использования достоинств оптики в целях повышения производительности вычислительных систем.

Кроме многих процессорных элементов (ПЭ), в МВК принципиальную роль играет коммуникационная сеть (КС), обеспечивающая обмен данными между различными ПЭ. Обычно электронная КС представляет собой набор так называемых связных процессоров (СП), объединенных между собой линиями связи в некоторую регулярную структуру (2-мерная, 3-мерная решетки, n-мерный гиперкуб и т. п.). В работе [2] показано, что при реализации одного СП на рассматриваемых оптических вентилях также может быть получено N виртуальных СП, объединенных между собою КС, топология которой может легко перестраиваться.

Таким образом, при реализации на оптических вентилях одного ПЭ и одного СП получим полноценный МВК, построенный таким образом, что задержки в оптических вентилях и линиях связи используются для хранения результатов работы вентиля в предыдущих $N - 1$ тактах, а каждый оптический вентиль обслуживает при этом N одинаковых виртуальных ПЭ или СП.

На таком МВК могут решаться те же задачи и теми же методами, что и на известных электронных МВК, состоящих из многих одинаковых ПЭ, объединенных некоторой регулярной КС. Однако оптический МВК обладает рядом преимуществ перед своим электронным аналогом.

В работе [3] рассмотрена конструкция оптического МВК и показано, что он может быть реализован на основе программно-перестраиваемой регулярной оптической среды, что позволяет оперативно приспосабливать возможности МВК к специфике конкретного применения практически без дополнительных накладных расходов.

В работе [4] рассмотрена реализация на основе оптического МВК вычислительных систем с массовым параллелизмом SIMD-архитектуры. Наиболее известным представителем таких систем является Connection Machine (СМ) фирмы "Thinking Machine". Показано, что реализация СМ в виде векторной ЭВМ имеет ряд преимуществ, основным из которых является возможность иметь различное количество ПЭ при решении различных задач при полном использовании всего имеющегося оборудования.

Такая векторная ЭВМ естественным образом реализуется с использованием электронных и оптических средств. Оперативная память может быть построена на основе традиционных полупроводниковых кристаллов запоминающих устройств с произвольной выборкой (ЗУПВ), которым в настоящее время нет конкурентов в оптике. СП и исполнительные устройства (ИУ) в ПЭ реализуются оптическими средствами на основе рассмотренных в [1—3] подходов.

У многих специалистов сложилось мнение, что эти идеи применимы только для организации векторной ЭВМ. Цель настоящей работы — показать, что аналогичный подход может быть использован для организации МВК MIMD-архитектуры. В частности, на основе такого подхода может быть реализована вычислительная система, управляемая потоками данных.

Реализация оптического МВК с MIMD-архитектурой. Основное различие между оптическими МВК с SIMD- и MIMD-архитектурами состоит в структуре ПЭ. В первом случае ПЭ представляет собой некоторый фиксированный набор исполнительных устройств либо некоторую регулярную среду, которая программным образом настраивается перед выполнением очередной операции над поступающими в ПЭ операндами. Код текущей операции и потоки операндов определяются управляющей ЭВМ, поскольку все виртуальные ПЭ выполняют одну и ту же операцию над различными данными.

Во втором случае каждый ПЭ работает по своей собственной программе. Поэтому, кроме потока операндов в MIMD-архитектуре, в ПЭ должен поступать поток команд. При этом ИУ должны выполнять различные операции над различными элементами потоков, а полученные результаты должны быть сформированы в непрерывный поток результатов. Это может быть реализовано путем выравнивания длины конвейеров в каждом ИУ, постоянно настроенном на выполнение некоторой определенной операции.

Кроме того, в ПЭ должен быть аналог устройства управления (УУ), подготавливающего операнды для операции и записывающего результаты операции в ОЗУ. Для вызова очередной команды это устройство должно иметь счетчик команд. Иными словами, УУ каждого виртуального ПЭ выполняет такие же функции, которые выполняет одна управляющая ЭВМ для всех ПЭ SIMD-архитектуры.

С учетом вышеизложенного одна из простейших структур оптического МВК MIMD-архитектуры для трехадресной системы команд показана на рис. 1. Слева приведена электронная часть, справа — оптическая. На их границе имеются гибридные элементы: преобразователи электрических сигналов в оптические (ЭО) и оптических сигналов в электрические (ОЭ). Поскольку темп передаваемых данных по оптическим каналам значительно выше, чем по электрическим, эти устройства, кроме того, обеспечивают соответственно мультиплексирование выходных потоков из ОЗУ и демультиплексирование входных потоков в ОЗУ. На рис. 1 это обстоятельство отражено тем, что количество выходных каналов у ЭО меньше, чем входных, а у ОЭ наоборот.

Так как темп поступления данных из кристаллов ЗУПВ (несколько десятков наносекунд на одно обращение) значительно ниже предельных скоростных возможностей электроники (около 1 нс), выходные потоки из ОЗУ проходят через электронные мультиплексоры (МЛ), которые объединяют в один быстрый канал с темпом передачи около 1 бит/нс несколько потоков данных, поступающих из ОЗУ по более медленным каналам. Аналогичным образом оптические входные в ОЗУ потоки после демультиплексирования в ОЭ до темпа передачи в одном канале 1 бит/нс преобразуются в электрические сигналы, поступающие на дополнительные электронные демультиплексоры

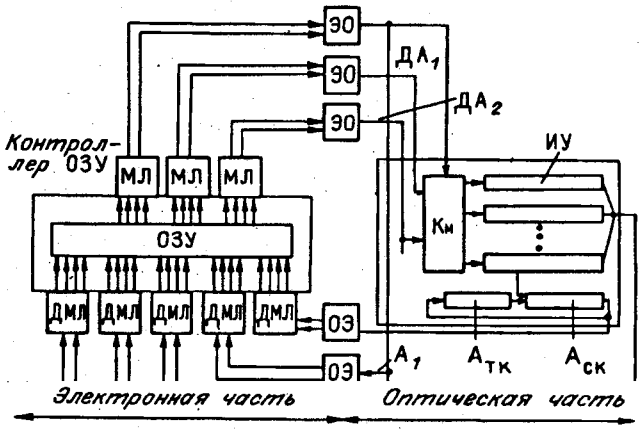

Рис. 1

(ДМЛ), на выходе которых темп передачи данных понижается до значения, при котором может осуществляться запись в кристаллы ЗУПВ.

Что касается контроллера ЗУПВ, то он выполняет функции формирования двух потоков считываемых операндов по двум потокам адресов и записи одного потока полученных результатов по одному потоку адресов записи.

Таким образом, как и в векторной SIMD ЭВМ, на исполнительные устройства поступают два вектора, однако их элементы считаны не по последовательным адресам, а в соответствии с потоком адресов. Кроме того, параллельно потокам операндов должен быть организован поток кодов операций.

Контроллер ОЗУ осуществляет также считывание потока команд по потоку значений счетчика команд. УУ выделяет из первого потока три потока по считыванию: поток кодов операций КОП и два потока адресов операндов A_1, A_2. С задержкой на длину конвейера УУ также выделяет один поток адресов A_3 для записи потока получившихся результатов $ДА_3$.

Таким образом, одновременно контроллер ОЗУ обрабатывает четыре потока: три потока операндов $ДА_1$, $ДА_2$, $ДА_3$ и поток команд (КМД). Каждый из этих потоков сопровождается соответствующим потоком адресов. Потоки адресов для $ДА_1$, $ДА_2$, $ДА_3$ формируются из потока КМД.

УУ должно формировать из потока значений счетчиков текущих команд поток значений счетчиков для следующих команд. Элементы последнего потока получаются из элементов первого прибавлением к их значению 1. Для команд условного перехода соответствующее значение счетчика может быть скорректировано, если выполняется условие перехода. Для команд безусловного перехода такая коррекция производится всегда. Эти действия выполняются с помощью счетчика команд $A_{СК}$ (см. рис. 1). Адрес очередной команды для каждого виртуального ПЭ получается путем прибавления единицы к адресу текущей команды, находящемуся в регистре сдвига $A_{ТК}$. Исключением являются операции условных и безусловных переходов, при выполнении которых адрес следующей команды берется из потока команд.

Получающийся поток счетчиков следующих команд может быть направлен для запоминания в ОЗУ. Однако при этом ОЗУ должно иметь возможность оперировать двумя потоками адресов команд: считывать поток адресов текущих команд и записывать поток следующих команд. Альтернативным решением является сохранение потока следующих команд в регистре сдвига

$A_{тк}$, реализованном на таких же оптических вентилях, что и ИУ. Это решение приведено на рис. 1.

На рис. 2 показаны временные диаграммы существования различных потоков при выполнении двух последовательных команд.

На первый взгляд может показаться, что при считывании из ОЗУ по произвольному потоку адресов не может быть организован непрерывный поток операндов, поскольку могут быть конфликты при обращении подряд к одному и тому же кристаллу ЗУПВ. Однако при более тщательном рассмотрении можно убедиться, что такой ситуации легко избежать.

Действительно, рассмотрим частный случай, когда один кристалл ЗУПВ соответствует памяти одного виртуального ПЭ. В этом случае элементы потока адресов относятся к различным ПЭ и, следовательно, к различным кристаллам. Произвольность адресов проявляется лишь в том, что в каждом кристалле производится обращение по произвольному адресу, указанному в потоке адресов.

Рассмотренный частный случай легко обобщается на случай, когда количество кристаллов в целое число раз K меньше количества виртуальных ПЭ. При этом память каждого кристалла делится на K равных частей. Первая часть i-го кристалла соответствует ПЭ с номером i, вторая часть — ПЭ с номером $i + N/K$ и т. д.

Аналогичным образом можно убедиться, что конфликты при обращении к одному и тому же кристаллу будут отсутствовать и при обработке потока адресов команд. Для этого достаточно предположить, что программы для каждого ПЭ хранятся в той же памяти, что и его данные.

При приведенном подходе СП можно рассматривать как специализированное ИУ, на вход которого в некоторые моменты времени поступают в качестве одного операнда данные, адрес которых указывается в поле A_1 команды пересылки, и номер ПЭ-приемника, который указывается в поле A_2. Эти данные СП перемещает в соответствующий временной интервал и пытается записать по адресу, указанному в поле A_3 команды пересылки. Может оказаться, что этот временной интервал в потоке результатов занят результатом команды, выполняемой от имени ПЭ-приемника, либо данными, передаваемыми из некоторого другого ПЭ-передатчика. В этом случае рассматриваемая команда пересылки данных оказывается невыполненной. В СП имеются средства для приостановки выполнения очередных команд пересылки, которые необходимы для разрешения конфликтов при транзитных передачах данных (см. структуру СП типа гиперкуб в [2]).

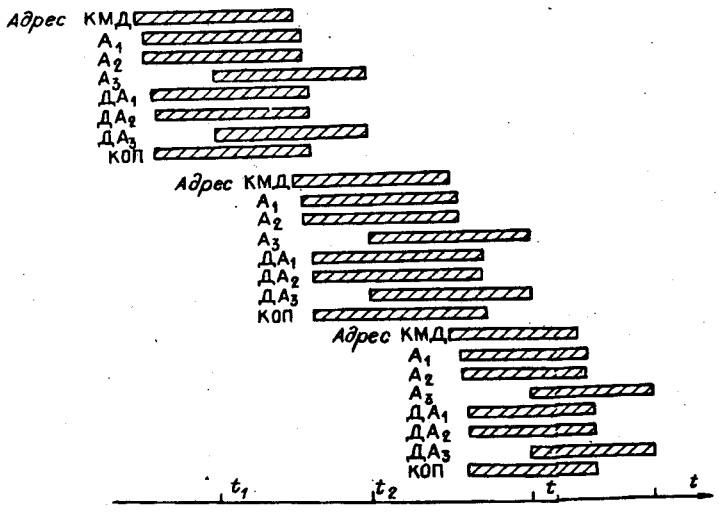

Рис. 2

Организация архитектур, основанных на концепции потоков данных. Поиск путей к использованию в полной степени возможностей, заложенных в MIMD-архитектуре, приводит к рассмотрению организации вычислительного процесса, при которой каждый ПЭ выполняет над поступающими на него данными определенные действия, указанные в программе, по которой он работает (остальные ПЭ могут в это время работать каждый по своей программе), и передает полученные результаты некоторым другим ПЭ. Почти такая же постановка задачи имеет место в концепции ЭВМ, управляемой потоком данных, при которой не указывается строгая очередность выполняемых операций, а допускается одновременное выполнение всех операций, для которых готовы операнды.

Вопросам организации вычислений на основе концепции управления потоком данных в многопроцессорных вычислительных системах посвящено достаточно много работ. В частности, реализация потока данных на МВК с КС типа гиперкуб и связанные с этим проблемы оптимального разбиения и загрузки графа программы рассматриваются в [5].

Можно организовать процесс вычислений в предлагаемом оптическом МВК на основе чистой концепции управления потоком данных, т. е. разбить граф программы на отдельные, относительно замкнутые (с минимальным количеством внешних связей) фрагменты, загрузить каждый из этих фрагментов на свой ПЭ и выполнять все команды по мере готовности операндов (готовность операндов можно определять либо аппаратным, либо даже программным способом). Полезно рассмотреть реализацию так называемого гибридного метода, при котором каждый из фрагментов графа, загруженный на отдельный ПЭ, выполняется последовательным образом, за исключением команд, требующих операнды со стороны других ПЭ, которые определяются на этапе трансляции и должны организовываться на основе концепции управления потоком данных.

В этом подходе понятие команды над операндами обобщено до понятия программы, обрабатывающей входные данные и выдающей один или несколько результатов. ПЭ имеет непосредственный доступ к большинству из обрабатываемых им данных. Поэтому в то время, когда ПЭ не занят выполнением программы, он может использоваться для выявления нового комплекта данных, готового для очередной обработки.

Описанный подход промоделирован на МВК, представляющем собой 3-мерную решетку, в которой ПЭ расположены в целочисленных координатах x, y, z ($x, y, z = 0, ..., 7$) [6]. Все ПЭ с одинаковыми координатами x, y и различной координатой z связаны через общие шины. Аналогичным образом связаны общими шинами ПЭ, имеющие общие координаты x, z и различные координаты y, а также ПЭ, имеющие общие координаты y, z и различные координаты x. Так же как и в СМ, для каждого ПЭ имеется свой СП, который осуществляет прием, буферизацию и передачу сообщений. Алгоритм работы СП таков, что порядок передачи сообщений вдоль координат x, y, z не имеет значения и сообщение передается вдоль той координаты, где в данное время свободна шина.

Так же как и в вычислительных системах, управляемых потоками данных (ВСУПД), программа представляется в виде направленного графа. В каждый ПЭ загружаются программы, соответствующие различным фрагментам этого графа. Разбиение графа на фрагменты производится на стадии трансляции. Алгоритм выполнения программы в каждом узле подчиняется дисциплине, принятой в ВСУПД. Программа начинает выполняться только после того, как в ПЭ поступили все входные параметры. Полученные при выполнении программы результаты передаются при помощи СП в те ПЭ, которые соответствуют фрагментам графа, связанным с рассматриваемым. В каждом ПЭ принимаемые данные накапливаются до тех пор, пока не соберется комплект данных, необходимый для запуска программы по их обработке.

В качестве еще одного примера эффективного использования вычислительной машины данной архитектуры можно рассмотреть процесс решения на такой вычислительной машине одной из тех задач, которые успеш-

но решаются на Connection Machine (например, задача обтекания некоторого профиля жидкостью).

Выполняющаяся на каждом виртуальном ПЭ программа должна описывать поведение жидкости в некоторой элементарной области пространства. Сама программа разделена на отдельные участки, каждый из которых описывает поведение жидкости при различных ее параметрах. В зависимости от значения этих параметров поведение жидкости может описываться различными алгоритмами. Например, если скорость жидкости больше некоторой величины, то движение жидкости из ламинарного переходит в турбулентное; если температура жидкости выше точки кипения, то она может закипеть и т. п. О подобных явлениях говорят как об особых случаях.

Connection Machine критикуется (и отчасти справедливо) за то, что она плохо приспособлена к обработке особых случаев. Действительно, в то время, когда управляющая ЭВМ будет выдавать команды, относящиеся к поочередной обработке различных особых случаев, многие ПЭ будут простаивать.

На МВК рассматриваемой архитектуры программа загружается в память каждого из ПЭ. Все они работают асинхронно, причем каждый из них выполняет только ту часть программы, которая соответствует значениям имеющихся у него параметров. Синхронизация происходит только при обмене данными после окончания очередной итерации. Это позволит одновременно обсчитывать состояния жидкости с любыми параметрами в любых точках пространства.

Еще одним плюсом (по сравнению с традиционной Connection Machine) является то, что в этом случае появляется реальная возможность введения режима мультипрограммирования. Это намечает пути к решению еще одной проблемы, возникающей при использовании Connection Machine, — проблемы недозагрузки процессорных элементов в случае, когда размерность задачи не совпадает с числом виртуальных процессоров Connection Machine.

Рассматриваемый подход реализации концепции вычислительного процесса, управляемого потоком данных, обладает следующими положительными свойствами.

1. Богатые возможности по структурированию вычислительного процесса. Каждому ПЭ, например, может быть назначен произвольный фрагмент графа либо несколько таких фрагментов, которые в общем случае между собой не связаны. Объем фрагмента может изменяться в значительных пределах и выбираться в соответствии со спецификой решаемой задачи.

2. Естественная локализация процесса вычислений. Известно, что все реальные вычислительные процессы обладают свойством локальности, заключающимся в том, что вероятность обращения к данным, которые только что участвовали в вычислениях, гораздо больше, чем ко всем другим. Для использования этого свойства практически во всех ЭВМ имеется так называемая сверхоперативная память, которая организована либо в виде регистров общего назначения, либо в виде ассоциативной буферной памяти, либо в виде КЕШ-памяти. До 90 % обращений к оперативной памяти относится к обращениям к такой сверхоперативной памяти. В традиционных ВСУПД данные, относящиеся к различным фрагментам графа, обезличены и хранятся в общей памяти. Это приводит к неоправданно длинному пути по передаче таких локальных данных от ПЭ в эту память и обратно. При рассматриваемом подходе исполнительные устройства в виде ПЭ находятся рядом с теми локальными данными, которые они обрабатывают, что уменьшает интенсивность потока данных по коммуникационной сети.

3. Возможность настройки ВСУПД на специфику конкретной задачи. Поскольку время передачи данных между различными ПЭ различно, то можно существенно повысить общий темп обмена данными, загружая фрагменты графа на ПЭ таким образом, чтобы те фрагменты графа, между которыми имеет место интенсивный обмен данными, оказались в соседних ПЭ. Подобной возможности настройки на специфику конкретной задачи в традиционных ВСУПД не имеется.

4. Более равномерное (по сравнению с SIMD-системами) распределение потоков данных в КС во времени. В отличие от SIMD-систем, и в частности от СМ, где обмен данными между различными ПЭ производится одновременно между всеми ПЭ по соответствующей команде, в рассматриваемой системе обмен данными осуществляется постепенно по мере возникновения такой потребности и никакими командами извне не управляется. Это, в свою очередь, ведет к сокращению времени передачи сообщений между ПЭ, поскольку КС оказывается более свободной и вероятность конфликтов в линиях связи и СП уменьшается.

5. В рассматриваемой архитектуре возможна организация распределенной системы управления вычислительным процессом, наиболее рационально распределяющей вычислительные ресурсы. Например, если некоторый ПЭ, на который загружен фрагмент графа, относящийся к часто используемой процедуре, не справляется с поступающим в него потоком заявок на выполнение этой процедуры с различными входными данными и тормозит тем самым весь ход вычислительного процесса, то для выполнения этого фрагмента в процессе трансляции можно назначить сразу несколько ПЭ. Однако часто исполняемые фрагменты графа не всегда можно определить на этапе трансляции, более того, это может довольно сильно зависеть от динамики выполнения программы (от начальных данных и т. п.).

Более оправданным представляется подход, при котором в некоторый ПЭ загружается некоторая программа-диспетчер, которая в динамике управляет загрузкой, размножением и исполнением отдельных фрагментов графа программы.

Из вышеизложенного следует, что на оптических элементах можно сделать МВК, работающий на основе управления потоком данных, который будет иметь не только высокую общую производительность, но и по степени универсальности, по крайней мере, не уступит известным проектам компьютеров на основе управления потоком данных.

В заключение обратим внимание на следующее обстоятельство. Замечено, что по мере развития сложные электронные системы становятся более регулярными. В качестве примера можно сослаться на программируемые логические матрицы (ПЛМ), на основе которых в последнее время создаются различные нерегулярные логические схемы. Например, в микропроцессоре М60020 используется более десятка различных ПЛМ. По мере роста количества ПЭ также видна тенденция к повышению степени регулярности вычислительной системы. Концепция управления потоками данных, предполагающая наличие многих исполнительных устройств и многих готовых к выполнению действий, довольно естественным образом может быть реализована на основе такой системы.

СПИСОК ЛИТЕРАТУРЫ

1. **Торчигин В. П.** Реализация чисто оптических многопроцессорных вычислительных комплексов // Вычислительные машины с нетрадиционной архитектурой суперВМ.—М.: Наука, 1990.
2. **Торчигин В. П.** Организация чисто оптических коммуникационных сред в многопроцессорных вычислительных комплексах // Автометрия.—1992.—№ 2.
3. **Торчигин В. П.** Использование оптических средств для передачи и обработки информации в многопроцессорных вычислительных комплексах // Автометрия.—1992.—№ 1.
4. **Торчигин В. П.** Организация многопроцессорных вычислительных комплексов с переменным количеством процессорных элементов // Там же.
5. **Hong Yang-Chang, Payne T. H.** Efficient computation of dataflow graphs in a hypercube architecture // Comput. Syst. Sci. and Eng.—1987.—2, N 1.—P. 29.
6. **Gaudiot.** A distributed VLSI architecture for efficient signal and data processing // IEEE Trans. on Comput.—1985.—34, N 12.—P. 1072.

Поступила в редакцию 22 апреля 1991 г.