

# Механизмы ускорения взаимодействия устройств в вычислительных структурах с сетевой организацией связей

Г.Г. Стецюра

Институт проблем управления РАН

Рассмотрены решения, ускоряющие реакцию систем управления и обработки данных на внутренние и внешние события за счет совмещения обмена данными с распределенной обработкой данных и применения одноканальной распределенной беспроводной коммутации. Показана целесообразность использования для этих целей оптических технических средств.

## 1. Введение

Так как современные вычислительные и управляющие системы содержат множество взаимосвязанных компонентов, то для эффективного решения прикладных задач естественно требуется быстрое и гибкое взаимодействие компонентов системы и быстрая реакция на внутренние и внешние события. Такие же и даже более жесткие требования выдвигают внутрисистемные задачи управления – задачи поддержки "жизнедеятельности" систем.

Ярким примером повышения требований к управлению состоянием системы служит выдвинутая IBM концепция "Autonomic Computing" (AC), возникшая как реакция на возрастающую сложность вычислительных и управляющих комплексов [1]. Концепция AC ориентирована на обеспечение без участия человека четырех свойств системы: самоконфигурирования при изменяющихся условиях функционирования, самовосстановления при любых нарушениях в работе системы, непрерывной самооптимизации и самозащиты от произвольного вида враждебных воздействий (4 «само»). Выступая с AC, фирма IBM предлагает создавать совершенно новый вид систем и фактически *инициирует междисциплинарные исследования в области управления поведением сложных полностью автоматических комплексов, действующих в реальном масштабе времени*. Эта инициатива была широко воспринята.

Для достижения объявленных IBM целей, видимо, целесообразно параллельно со средствами обработки данных иметь дополнительную систему средств, непрерывно наблюдающую за состоянием всех цифровых средств и обеспечивающую достижение указанных четырех "само". Такая система должна, возможно, в большей степени, чем основная система, обеспечивать высокую скорость реакции на возникающие события, автоматически прогнозировать появление нестандартных нежелательных ситуаций, их обнаруживать и устранять. Таким образом, задачи самоуправления функционированием систем выдвигают дополнительные требования к управлению взаимодействием устройств систем, и их также надо учитывать.

Для всех указанных применений в настоящем докладе предложена совокупность механизмов низкого уровня, ускоряющих и упрощающих реализацию трех видов процессов в цифровых системах – совмещения обмена данными с распределенными вычислениями, децентрализованного устранения конфликтов доступа к ресурсам, децентрализованной коммутации.

Рассматривается организация функционирования этих механизмов, и приводятся ссылки на публикации других авторов по конкретной физической реализации необходимых устройств с использованием, прежде всего, оптических средств. Оптические средства позволяют получить простые, быстродействующие и мало энергоемкие решения, и количество работ по созданию оптических устройств с требуемыми в этой работе свойствами непрерывно возрастает.

Так результаты, полученные в NRL [2–4], непосредственно применимы при реализации устройств из разделов 2.2, 2.5 – 2.8. Для разделов 2.1 и 2.3 важны исследования в IBM по оптическим переключателям [5, 6]. Электронные варианты устройств для этих разделов были созданы в ИПУ [7]. В разделе 2.4 устройства, переключающиеся медленнее микросекунды, реализуемы многими способами [8]; для более высоких скоростей имеются публикации только на уровне описания результатов физических экспериментов [9–12].

## 2. Механизмы ускорения взаимодействия узлов цифровых систем

В восьми подразделах данного раздела рассмотрены существенно различающиеся между собой механизмы, обеспечивающие ускорение взаимодействия устройств систем управления и обработки данных (далее "узлов"). Все изложенные результаты получены в Институте проблем управления РАН (ИПУ).

### 2.1 Групповые операции (совмещение обмена данными с распределенными вычислениями)

Обычно логические элементы вычислительных средств действуют следующим образом. Поступающий на вход элемента сигнал перестраивает структуру элемента так, что на его выходе появляется сигнал, создаваемый за счет энергии источника питания элемента. На перестройку элемента требуется время, и возникающая временная задержка сигнала определяет его быстроедействие. При последовательном соединении элементов сигнал с выхода одного элемента поступает на вход другого элемента, перестраивает его и т.д.: задержка накапливается.

Чтобы избежать накопления задержки и сделать время вычисления такого вида практически не зависящим от количества участвующих в нем узлов, был разработан метод вычислений, совмещающий процесс передачи данных с вычислением [7, 13–15]. Один из вариантов реализации метода иллюстрирует рис. 1.

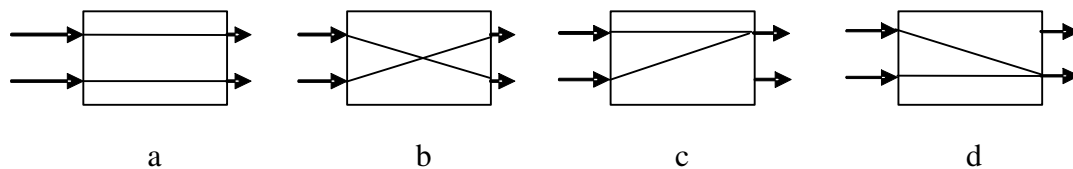


Рис. 1. Логический элемент

Здесь a, b, c, d – четыре состояния логического элемента, представленного прямоугольником. В элемент входят две линии и из него выходят также две линии. Элемент выполняет указанные на рис.1 соединения между входными и выходными линиями. Сигнал, подаваемый на элемент, поступает только на одну из входных линий. Будем считать, что наличие сигнала на верхней входной линии интерпретируется как двоичная единица (1), сигнал на нижней линии интерпретируется как логический ноль (0). Элемент в состоянии "a" передает сигнал на выход без изменения, в состоянии "b" инвертирует значение входного сигнала, в состоянии "c" превращает любое значение входа в "1" на выходе, в состоянии "d" превращает любое значение входа в "0" на выходе. Переключение состояний выполняет внешнее по отношению к элементу устройство.

Указанных состояний элемента достаточно для выполнения логических операций, нахождения *max* и *min*, выполнения арифметических операций сложения, вычитания и умножения.

Рассмотренный логический элемент размещается в системе совмещающей вычисления с передачей данных способом, показанным на рис.2.

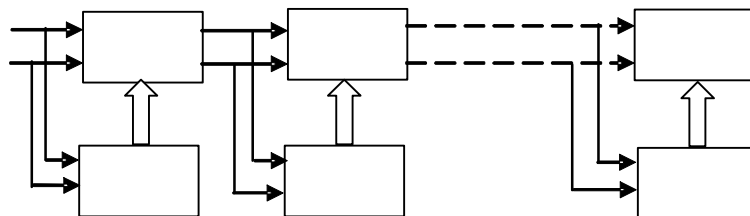


Рис.2. Система с совмещением вычислений и передачи данных.

Здесь каждый прямоугольник из верхнего ряда обозначает рассмотренный выше логический элемент, каждый прямоугольник из нижнего ряда – устройство, управляющее состоянием соответствующего элемента из верхнего ряда. Фигурная стрелка обозначает канал управления

состоянием элемента. Показанные на рис.2 ответвления входов в устройства предназначены для приема входящих данных. Совокупность элемента с управляющим им устройством составляет узел системы. По каналу, соединяющему элементы, данные передаются последовательно поразрядно.

Распределенное вычисление выполняется над операндами, каждый из которых размещен в соответствующем узле рис. 2. Так можно, например, выполнить распределенное суммирование чисел. При этом состояние элемента из рис. 1 определяет соответствующий разряд операнда в узле, и *перестройка связей в элементе выполняется до прихода по каналу разряда, на значение которого эта перестройка должна повлиять.*

Отсюда следует: *все узлы одновременно устанавливают входящие в них логические элементы в требуемое состояние, после этого сигнал проходит через все элементы без задержки на установку элементов.* Операция выполняется за время переноса сигнала через элементы (один такт) и практически не зависит от числа последовательно соединенных элементов.

Отметим, что все элементы, включая первый, не создают новый сигнал, они лишь транслируют поступивший сигнал. Сигнал создает внешнее устройство – генератор сигналов, посылающий сигналы на верхний вход первого элемента в цепочке. Отсюда следует второй важный результат: *в системе не создаются мощные сигналы, требуемые для взаимодействия между узлами системы.* Это уменьшает энергопотребление и рассеивание энергии в системе, и позволяет создавать распределенные системы, потребляющие малую мощность, например, системы интеллектуальных датчиков.

Выше рассмотрен способ совмещения, использующий изменение пути прохождения сигнала, но можно также изменять свойства сигналов. В публикации [7] показано использование для этого поляризации оптического сигнала, в [13] использована замена без задержки передаваемого по линии двоичного сигнала на противоположное значение. Было бы полезно выявить и другие способы изменения сигналов, передаваемых без их задержки.

Операции, выполняющие рассмотренные распределенные вычисления названы групповыми операциями.

## 2.2 Децентрализованное приоритетное управление доступом к ресурсу (ДПУ)

Способ доступа ДПУ был разработан для решения следующей задачи. Имеется группа узлов, которым требуется доступ к некоторому ресурсу, в частности к каналу связи. Эта потребность возникает в произвольные не коррелированные для разных узлов моменты времени. Для решения задачи в ИПУ был разработан ряд способов, объединенных под общим названием "ДПУ". Наиболее полно варианты ДПУ изложены в [14]. Здесь приведем один из вариантов ДПУ – децентрализованное кодовое управление (ДКУ).

При ДКУ узлам присваиваются различающиеся коды приоритетов, определяющие право источников на занятие канала.

Источник ожидает наступления момента времени, когда можно начинать борьбу. Пусть, например, этот момент определяется наступлением паузы – отсутствием сигнала в связывающем источниками канале в течение времени  $T_{пз}$ .

В интервале времени, когда можно проводить борьбу, источники передают последовательно разряды своих кодов приоритетов, начиная со старшего разряда. Сигналы, представляющие двоичные 1 и 0 кода приоритета, должны быть такими, чтобы можно было обнаружить наличие сигнала 1 в канале при наличии в нем сигнала 0. Источник передает сигналы с интервалом времени  $T_p$ . Сигнал передается течение времени  $T_p Q$  ( $0 < Q < 1$ ). В течение времени  $T_p(1 - Q)$  источник проверяет состояние линии. Источник, передавший 0 и обнаруживший 1 в канале, прекращает борьбу за канал. Источник, передавший все разряды кода приоритета, занимает канал. Изложенные действия – это операция, определяющая в канале максимальное значение в группе чисел. Для корректности работы алгоритма достаточно, чтобы выполнялось соотношение  $T_p \geq 2\tau / (1 - Q)$ . Здесь  $\tau$  – время прохождения сигнала по каналу [14].

ДКУ с помощью регулирования значений приоритетов позволяет достигнуть очень гибкого управления работой распределенных систем, и обеспечивает гарантированный доступ узлов к ресурсу. Приоритеты узлов могут изменяться в динамике, учитывая текущее состояние узла.

ДКУ эффективно работает при достаточно коротких расстояниях между узлами системы, так как интервал времени между передачей разрядов кода приоритета учитывает время прохождения сигнала по каналу. ДКУ асинхронно.

Позднее фирма Philips опубликовала интерфейс I<sup>2</sup>S, использующий близкий способ разрешения конфликтов, но работающий синхронно и более ограниченно использующий приоритеты доступа [16]. Интерфейс I<sup>2</sup>S применяется во многих микроконтроллерах.

### 2.3 Групповые команды, групповые программы

Групповые операции, приведенные в разделе 2.1, используются в групповых командах и групповых программах. Групповая команда представляет собой сообщение, передаваемое между узлами, предварительно объединенными каналом так, как это сделано в разделе 2.1.

Групповая команда содержит операнд и перемещается через цепочку узлов (см. раздел 2.1). Вторые операнды находятся в каждом из узлов. При прохождении сообщения через узел *без задержки сообщения* выполняется требуемая операция над двумя указанными операндами. *Ключевое свойство* совмещенных операций – результат вычисления помещается на место, которое в сообщении занимает операнд. После прохождения сообщения через цепочку узлов в нем вместо первого операнда будет находиться результат действия всех узлов.

Чтобы указать узлу, как он должен обрабатывать операнд, в сообщении, перед операндом помещается код групповой операции, понимаемой всей группой узлов. Последовательность из кода групповой операции и операнда составляет простейшую групповую команду. В общем случае групповая команда имеет переменную длину и содержит группу операндов. Код групповой команды может содержать сложные указания по обработке содержимого сообщения, определяющие различные способы обработки разных операндов в групповой команде. Так код групповой операции информирует узлы о структуре сообщения.

Действия групповой команды могут распространяться не только на сообщение, содержащее эту команду, но и на определенную в команде группу последующих сообщений. Групповая команда может также изменять состояние конкретных узлов системы, в которые она поступает, тем самым в динамике реконфигурируя систему, включая изменение прав отдельных узлов. Например, так можно задавать ветвление вычислений, передачу от одного узла другому прав лидера – узла, имеющего право посылать групповые команды.

Следующий шаг в построении групповых взаимодействий – определение групповой программы. Групповая программа – это сообщение, в разных частях тела которого размещаются определенные выше групповые команды. Чтобы структуру групповой программы можно было формировать произвольным образом, будем код каждой групповой операции отмечать специальной меткой. На рис. 3 показан пример сообщения – групповой программы.

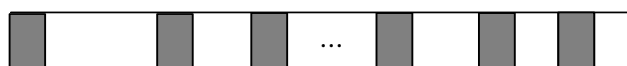


Рис. 3. Сообщение – групповая программа

Здесь серым цветом обозначены коды групповых команд, без окраски – данные групповых команд.

Основное назначение групповых команд и программ – формирование данных, необходимых для управления сложной системой. Так, например, в работах [17, 18] даны способы получения взаимно согласованных решений при дублировании вычислений и регулировании доступа к ресурсам. Однако групповые команды можно использовать для проведения распределенных вычислений в прикладных задачах при соответствующем изменении вычислительных алгоритмов.

В таблице 1 показано полученное ускорение вычислений для ряда задач по сравнению с другими известными методами [19]. Здесь  $n$  – количество компонентов в задаче (размерность матрицы, количество членов уравнения, полинома и т.д.). В строках таблицы для заданного количества процессоров показано время, требуемое для выполнения вычисления. Время выражено в условных единицах  $T$ . Сравниваются наиболее быстрый из известных методов с методом совмещения передачи данных и распределенного вычисления.

В системах, использующих параллельные многозарядные высокоскоростные каналы связи между узлами, синхронная передача разрядов данных усложняется, и во многих случаях последовательная поразрядная передача становится конкурентоспособной. Способ обработки данных, примененный в групповых командах, естественно сочетается с таким последовательным способом передачи.

Изложенная организация групповых программ ориентирована на самые быстрые взаимодействия узлов. Теперь рассмотрим возможность использования групповых программ в смешанных системах, использующих помимо локальных также удаленные связи, например, соединяющие кластеры через интернет при проведении облачных вычислений или в grid computing.

**Таблица 1.** Ускорение распределенных вычислений

Вид операции	время вычисления		число процессоров
	метод совмещения	обычно: $\geq$	
Распределенное умножение матриц	$Tn$	$Tn \log_2 n$	$n^2$
	$T$	$T(\lceil \log_2 n \rceil + 1)$	$n^3$
Алгебраические уравнения, метод Гаусса-Жордана, нахождение главных элементов	$Tn$	$Tn \log_2 n$	$n(n + 1)$
Вычисление значения полинома	$T$	$T \log_2 n$	$n$
Вычисление значения произведения полиномов	$2T$	$2T \log_2 n$	$n$
Свертка	$T$	$T \log_2 n$	$n$
Дискретное преобразование Фурье	$Tn$	$Tn \log_2 n$	$n^2$
Сортировка	$T(\lceil n/2 \rceil + 1)$	$Tn(\log_2 n)$	$n$

Удаленную передачу необходимо проводить с помощью сообщений, удовлетворяющих международным стандартам, например, протоколу TCP/IP. В это сообщение поместим групповую команду или групповую программу, адресованную удаленным узлам. Формирование сообщения в формате TCP/IP выполняет узел-передатчик, обеспечивающий удаленную связь. Передатчик сохраняет у себя полученную групповую программу, создает сообщение TCP/IP, упаковывает в него в качестве данных групповую программу, адресованную удаленным узлам, и направляет это сообщение удаленному приемнику сообщений через сеть общего пользования. Приемник извлекает из принятого сообщения групповую программу и направляет ее узлам удаленной системы.

Отметим, что в последние годы проводится много работ по внедрению активных сетей, сообщения которых содержат программы, влияющие на конфигурацию сети и внутреннее поведение узлов сети [20, 21]. В этом отношении программы активных сетей подобны групповым программам, но ориентированы на значительно более медленные процессы.

Поэтому там, где требуется быстрое время реакции на события, например, в управляющих системах, целесообразно сообщения оформлять в виде групповых команд или программ.

Кроме того, команды в активных системах в отличие от групповых программ не позволяют организовать совмещенные с обменом данными распределенные вычисления.

## 2.4 Распределенная оптическая коммутация: парные соединения

Цель раздела – показать возможность уменьшения сложности распределенных коммутаторов и ускорения процесса коммутации за счет использования оптических средств. Показано, что для коммутации  $n$  узлов возможно построение одноканальных коммутаторов, содержащих  $n \log_2 n$  элементов коммутации, каждый из которых принимает одно из двух возможных состояний. Ускорение и уменьшение сложности коммутации достигается за счет установления непосредственных связей между источниками и приемниками сигналов.

Рассматривается только коммутация макроустройств, например, процессоров или компьютеров в сосредоточенном многомашинном комплексе или электронных модулей на печатной плате.

Уменьшение сложности достигается за счет того, что элемент коммутации, имеющий только два состояния, может одновременно коммутировать много каналов и его сложность не увеличивается при увеличении количества каналов.

При этом каждый узел содержит все необходимые ему оптические средства коммутации.

Чтобы указать такому локальному средству коммутации на конкретный из  $n$  узлов обработки данных, с которым требуется установить соединение, узлу требуется задать не менее  $\log_2 n$  битов информации, что и определяет минимальное количество коммутационного оборудования. В докладе достигнута именно эта величина. В практически используемых средствах коммутации оно больше, и может достигать значения  $\sim n^2$ .

Рассмотрим распределенный коммутатор, соединяющий  $n$  узлов и содержащий в каждом из таких узлов локальный демультиплексор, соединяющий этот узел с остальными узлами. В разделе приведены примеры возможной реализации такого коммутатора, в которых для построения демультиплексора требуется  $\log_2 n$  элементов коммутации, принимающих два состояния. Использованы известные решения, в которые внесен ряд дополнений.

Вначале рассмотрим коммутатор, выполненный как микро-электромеханическая система (MEMS) [8], содержащая электромеханически перемещаемые зеркала. Пусть информация между узлами передается оптическими сигналами, а локальный демультиплексор имеет структуру, пример которой показан на рис. 4.

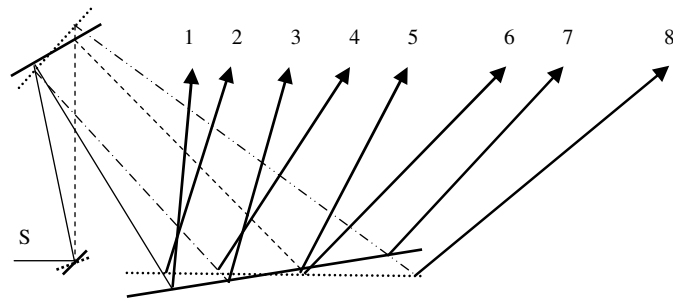


Рис. 4. Коммутатор с перемещаемыми зеркалаи

Здесь показаны три поворачивающихся зеркала, на первое из которых направляет луч света лазерный источник сигналов S. Отразившись от всех зеркал, луч займет одно из восьми указанных положений, определяемых положением зеркал.

Таким образом, достаточно иметь  $\log_2 n$  управляемых зеркал, чтобы одновременно, за один такт, выбрать одно из  $n$  положений луча. Чтобы направить лучи 1 – 8 произвольно расположенным приемникам сигналов, возможно на пути лучей придется поместить дополнительную систему неподвижных зеркал, направляющую лучи в заданном направлении. Требуется  $n^2$  таких неподвижных зеркал для обслуживания связи со всеми  $n$  узлами. Неподвижные зеркала всех демультиплексоров располагаются вне узлов. Эти зеркала составляют часть среды распространения лучей, могут быть выполнены в виде монолитной пластины с зеркалами и их сложность существенно меньше сложности остальных компонентов рассматриваемой системы коммутации. Зеркала ориентируются так, чтобы связать места, где могут быть размещены коммутируемые узлы, а система наращивается размещением узлов в отведенные для них места без какой-либо настройки демультиплексоров.

Последнее из зеркал рис. 4 должно иметь размер, достаточный для отражения от него всех  $n$  лучей. Поэтому при больших  $n$  желательно использовать всю площадь зеркала:  $n^{1/2}$  лучей размещается в плоскости рисунка и  $n^{1/2}$  – в перпендикулярной плоскости. При этом для перемещения луча между источником S и рассматриваемым демультиплексором требуется расположить дополнительно демультиплексор на  $n^{1/2}$  выходов. Он отличается от основного в следующем. Любой выходящий из него луч должен поступать в основной демультиплексор парал-

тельно лучу от источника S, но со сдвигом. Это достигается установкой на выходе демультиплексора  $n^{1/2}$  неподвижных зеркал, создающих требуемое изменение направления лучей.

Выделим особенности описанного узла.

1. Количество перемещаемых зеркал и их индивидуальных средств управления положением равно количеству двоичных разрядов числа, представляющего коммутируемые узлы. Положения, в которое перемещаются зеркала, не зависят от уровня сигнала, управляющего положением зеркала.

2. Каждый из двоичных элементов коммутации, переходя в одно из двух состояний, позволяет переключать одновременно группу виртуальных каналов (лучей); размеры группы варьируются от 2 до  $n$  для разных элементов демультиплексора. Для управления любой группой требуется однобитовый сигнал. В каждый момент времени реализуется только один из виртуальных каналов.

3. Каждое зеркало способно коммутировать передаваемые параллельно многоразрядные слова.

Отмеченные в п.п. 2 и 3 особенности и обеспечивают уменьшение сложности коммутатора.

4. Источники света S могут быть вынесены не только за пределы коммутаторов, но и за пределы всей системы. В элементах коммутации не создаются новые сигналы. Все это уменьшает энергопотребление в системе.

На переключение положения зеркал требуется более микросекунды, и электромеханические переключатели можно использовать длительно используемой системы соединений. Однако использование физических сред с изменяемыми оптическими свойствами позволяет создавать быстродействующие коммутаторы [9–12, 22]. Так, например, диоксид ванадия ( $\text{VO}_2$ ) за время порядка 12 фмтсек переходит из прозрачного состояния в зеркально отражающее свет [9–11]. Рис. 5 показывает, как может выглядеть демультиплексор, использующий это свойство.

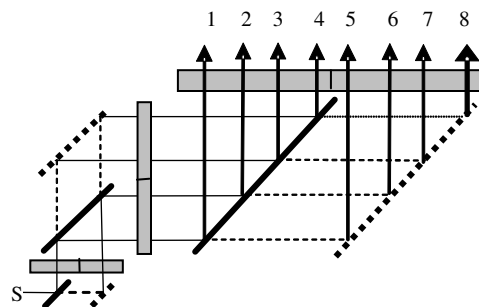


Рис. 5. Коммутатор, использующий  $\text{VO}_2$

Здесь сплошными наклонными линиями показаны три управляемых зеркала из  $\text{VO}_2$ , пунктирными линиями – обычные зеркала или призмы. Источник S направляет луч света на управляемое зеркало, которое или отразит свет на вторую группу зеркал или пропустит его на неуправляемое зеркало. Так действуют все пары зеркал. В результате луч света на выходе примет одно из восьми положений. Все 8 направлений взаимно параллельны, и, чтобы направить их на произвольно расположенные приемники, требуется вне узлов установить указанные выше зеркала.

Пропускающее луч зеркало в незначительной степени и отражает его, что создает помеху. Для избавления от помех введем указанные на рис. 5 прямоугольниками управляемые пары фильтров, которые блокируют лучи от одного зеркала из пары зеркал, нежелательные при данной комбинации управляющих сигналов. Это могут быть фильтры, изменяющие пропускную способность при подаче управляющего сигнала, либо управляемые зеркала из  $\text{VO}_2$ , отводящие нежелательный сигнал в сторону. Для управления фильтрами не требуются дополнительные сигналы, и сложность по управлению сохраняется.

Для построения демультиплексора можно сочетать пластины с двойным лучепреломлением с электрооптическими поляризаторами Погкельса [12].

Известны и многие другие способы реализации оптического коммутатора, но в значительной их части величина отклонения луча зависит от величины управляющего сигнала, что создает дополнительные трудности. В приведенных примерах этот недостаток отсутствует. Более

подробно материалы раздела 2.4 изложены в [22].

Приведенные парные взаимодействия можно использовать также для соединения модулей на уровне электронной платы, расширяя возможности известного решения [23]

Особенности 1 – 4, приведенные выше для MEMS-демультиплексора, необходимо обеспечивать и при других способах построения демультиплексоров, выполняющих рассматриваемые парные соединения.

## 2.5 Распределенная оптическая коммутация: связь узла с группой узлов

Задача следующая: узел должен одновременно передать одинаковые данные группе связанных с ним узлов. Рассылка выполняется по инициативе передающего узла.

Для таких рассылок данных, помимо сложного передающего оборудования, требуются большие затраты энергии. Ниже приведено решение, устраняющее указанные проблемы. В нем использованы некоторые результаты работ [2–4].

Узлы группы посылают узлу, имеющему ретрорефлектор, немодулированные лучи света, которые этот узел должен модулировать двоичными сигналами и вернуть узлам группы в виде общих для всех узлов данных.

Рассмотрим структуру, показанную на рис. 6. Здесь два луча от двух источников группы (сплошная и пунктирная линии) падают на ретрорефлектор (на рис. 6 это уголкового отражателя – катафот).

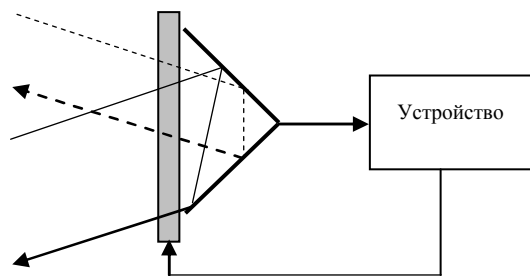


Рис. 6. Применение ретрорефлектора

Отражаясь от катафота, каждый луч возвращается к пославшему его источнику. Перед катафотом расположен управляемый сигналами приемника фильтр, который модулирует значение возвращаемых источникам сигналов. Модулировать сигнал можно, изменяя в фильтре любые свойства отраженного сигнала или изменяя прозрачность граней уголкового отражателя. Можно также расположить перед катафотом управляемое зеркало, которое будет прерывать возврат сигнала адресату.

В отличие от [2] ретрорефлектор здесь используется для связи не между парой устройств, а с группой устройств. В [3, 4] каждый приемник имеет индивидуальные средства для связи с каждым источником.

## 2.6 Распределенная оптическая коммутация: устранение конфликтов доступа

Пусть имеется узел (центр), с которым требуется установить связь группе узлов, но это должно быть разрешено одновременно только одному из них. Центр имеет устройство связи, показанное на рис. 6. Частично свет, падающий на катафот центра, проходит в приемное устройство, которое обнаруживает поступление сигнала, соответствующего логической единице в поступившем в центр сообщении.

Возникающий конфликт будем устранять с помощью алгоритма ДКУ.

Конфликтующим узлам группы приписаны различающиеся коды приоритетов доступа. Узлы группы посылают центру лучи, модулированные двоичными сигналами, представляющими коды приоритета узлов, и центр при поступлении к нему двоичной единицы хотя бы от одного из узлов группы сообщает об этом всем узлам группы.

Такое оповещение можно выполнить многими способами, например, прерывая возврат све-



того сигнала в его источники.

Будем считать, что поступившие в центр двоичные сигналы центр интерпретирует так: наложение сигналов "1" и "0" воспринимается как сигнал "1", множество одновременно поступающих сигналов "1" ("0") воспринимается как "1" ("0").

В соответствии с алгоритмом ДКУ узлы, пославшие в центр сигнал "0" и получившие в ответ сообщение от центра о приходе к нему сигнала "1", перестают участвовать в устранении конфликта. После передачи всех разрядов приведенная процедура нахождения максимума выделит единственный источник с наибольшим значением кода приоритета.

Возможны различные модификации приведенной процедуры с изменением как логики процесса [9], так и его физической реализации, например при использовании сигналов разной длины волны для передачи данных и управления.

## **2.7 Распределенная оптическая коммутация: вызов узлом группы узлов**

Пусть узлу системы требуется установить соединение с группой узлов и затем передать этим узлам данные. Пусть все узлы имеют ретрорефлектор и модулятор, показанные на рис. 6.

Выделим в системе фиксированный узел вызова. Этот узел имеет фиксированное положение, и для посылки на его катафот луча узлам не требуется привлекать коммутатор. Все узлы следят за узлом вызова, и с его помощью, действуя в соответствии с разделом 2.6, любые узлы могут провести борьбу за право доступа к узлу вызова. Узел, получивший это право, модулируя лучи устройств с помощью узла вызова, сообщит требуемым ему узлам о необходимости взаимодействия.

После этого выбранные узлы направят лучи на катафот запрашивающего узла, и начнется связь этого узла с группой узлов.

Узел вызова используется только для управления обменом данными, который далее выполняется по индивидуальным связям.

## **2.8 Распределенная оптическая коммутация: связь с подвижными узлами**

В системе могут быть подвижные узлы, ими могут быть как периферийные устройства, так и устройства обработки данных. Необходимо рассмотреть взаимодействие и с такими узлами. Ниже рассмотрим оптические связи, позволяющие получить новые полезные свойства, но вначале укажем на использование для таких связей ДПУ, использующее радиосвязь. Установление связи источника с приемником состоит из двух шагов. На первом шаге при наличии нескольких источников – претендентов на передачу данных алгоритм ДКУ выделяет одного из них. На втором шаге происходит собственно обмен данными. На обоих шагах достаточно иметь единственную частоту радиосигналов, общую для всех узлов. Однако при этом узлы должны иметь источники радиосигналов, что не всегда желательно. Применение оптических средств позволяет разместить источник сигналов в единственном узле.

Рассмотрим следующую систему. В поле видимости узлов находится ретранслятор сигналов, который принимает оптические сигналы на одной частоте  $f_0$  и ретранслирует их на другой частоте  $f_1$ . Кроме этого ретранслятор имеет источник непрерывного оптического излучения на частоте  $f_0$ , который также видят все узлы системы. Каждый узел имеет два приемника оптических сигналов – для сигналов частоты  $f_0$  и частоты  $f_1$ .

Пусть выделен узел-источник данных, который должен разослать эти данные заданным узлам-приемникам данных. Взаимодействие узлов осуществляется следующим образом. Все узлы получают непрерывный сигнал частоты  $f_0$  от ретранслятора, и только источник данных возвращает сигналы, несущие свои данные в ретранслятор на частоте  $f_0$ . Для этого источник имеет ретрорефлектор сигналов и расположенный перед ним модулятор оптических сигналов, аналогичные показанным на рис. 6.

При передаче данных источник модулирует с помощью модулятора поступающий на ретрорефлектор свет и возвращает его в ретранслятор. Источник модулирует сигнал (разрешает возврат сигнала в ретранслятор), если ему требуется передать единицу. Предполагается синхронность посылок, позволяющая определить "передачу" нуля. Усложнив схемотехнику, можно перейти к несинхронной передаче. Ретранслятор полученные модулированные сигналы

ретранслирует всем узлам, используя оптические сигналы на частоте  $f_1$ . Сообщение, содержащее данные, имеет адресную часть, идентифицирующую приемники данных.

Наличие ретранслятора позволяет узлам передавать данные, не имея собственных источников оптических сигналов – используется только источник в ретрансляторе. Изложенное решение позволяет узлам перемещаться относительно ретранслятора. Сложность в узлах устройств, управляющих описанным взаимодействием узлов, не зависит от количества последних.

Теперь рассмотрим вопрос выделения единственного источника данных из числа многих узлов, желающих стать источниками данных. Для этого используем ДКУ в следующей редакции. Все узлы получают непрерывный сигнал  $f_0$  от ретранслятора, и этот сигнал модулируют только те узлы, старший разряд кода приоритета которых равен единице. Модулированный сигнал возвращается в ретранслятор, который ретранслирует его всем узлам на частоте  $f_1$ . Если узлы в старшем разряде кода приоритета имели "0" и получили сигнал  $f_1$ , то они прекращают борьбу за право быть источником. Оставшиеся узлы выполняют указанные действия со следующим разрядом кода приоритета и т.д. В результате будет выделен единственный узел, который и становится источником.

Ретранслятор может быть размещен среди узлов. Для этого должна быть использована отражающая свет среда, на которую посылает свет ретранслятор, и отраженный от среды свет поступает к узлам и от узлов к ретранслятору. При необходимости такое расположение позволяет разместить средства ретрансляции в каждом узле.

### 3. Заключение

Полученные решения направлены на ускорение важных процессов в системах обработки данных и управления – устранения конфликтов при доступе к ресурсам, распределенной обработки данных, распределенной коммутации. С этой целью разработаны механизмы со следующими возможностями:

- децентрализованное приоритетное управление (ДПУ) позволяет узлам системы разрешать без выделенного центра конфликты доступа к ресурсам;
- групповые команды и программы, совмещающие передачу данных с распределенными вычислениями, позволяют быстро оценивать состояние системы и воздействовать на ее структуру и функционирование;
- распределенная коммутация непосредственно, без привлечения промежуточных активных устройств, связывает бесконфликтно источники данных с приемниками. Конфликт возможен только на входе в приемник, но он разрешается с привлечением ДПУ.

Значительная часть приведенных в докладе результатов возможна только с привлечением оптических средств, интенсивно разрабатываемых в настоящее время во многих исследовательских центрах. Следует отметить, что в России вопросам применения оптики в вычислительных системах значительное внимание уделял В.С. Бурцев [24].

Полученные решения также полезны при реализации четырех "само" инициативы "Autonomic Computing" фирмы IBM [1], требующих для управления функционированием систем обработки данных децентрализованного управления взаимодействием узлов и особо быстрой реакции на события.

### Литература

1. An architectural blueprint for autonomic computing. Autonomic Computing White Paper, Third Edition. IBM. 2005.
2. Gilbreath G.C. et al. Large-aperture multiple quantum well modulating retroreflector for free-space optical data transfer on unmanned aerial vehicles// Opt. Eng. 2001. no.7. P. 1348-1356.
3. Rabinovich W.S. et al. Free space optical data links using cat's eye modulating retroreflectors. [http://esto.nasa.gov/conferences/estc2003/papers/A6P3\(Rabinovich\).pdf](http://esto.nasa.gov/conferences/estc2003/papers/A6P3(Rabinovich).pdf)
4. Rabinovich W.S. et al. 45-Mbit/s cat's-eye modulating retroreflectors// Optical Engineering. 2007. Vol. 46. no. 10. P. 1-8.

5. Green W.M., Rooks M.J., Sekaric L., Vlasov Y.A. Ultra-compact, low RF power, 10 Gb/s silicon Mach-Zehnder modulator // Optics Express. 2007. Vol. 15. Issue 25. P. 17106-17113.
6. Made in IBM Labs: Breakthrough Chip Technology Lights the Path to Exascale Computing  
<http://www-03.ibm.com/press/us/en/pressrelease/33115.wss>
7. Стецюра Г.Г. Методы совмещения вычислений и передачи данных в мультипроцессорных системах и локальных сетях. М.: Ин-т пробл. упр., 2005. 86 С.  
<http://www.ipu.ru/labs/lab31kom/ggs.zip>
8. Kaajakari V. Practical MEMS. Small Gear Publ. 2009. 496 P.
9. Suh J. Y. et al. Modulated optical transmission of subwavelength hole arrays in metal-VO<sub>2</sub> films// Applied Physics Letters. 2006. Vol. 88. P. 133115-1 – 133115-3.
10. Huber R. et al. THz Slow Motion of an Ultrafast Insulator-Metal Transition in VO<sub>2</sub>: Coherent Structural Dynamics and Electronic Correlations. Ultrafast Phenomena XVI. Springer Series in Chemical Physics. 2009. Vol. 92. Part 3. P. 179-181.  
<http://www.springerlink.com/content/t5737480x7393g27/>
11. Виноградова О.П. и др. Синтез и свойства нанокристаллов диоксида ванадия в силикатных пористых стеклах// Физика твердого тела. 2008. том 50. вып. 4. С. 734-740.  
<http://journals.ioffe.ru/ftt/2008/04/p734-740.pdf>
12. Kulcke W. et al. Fast, Digital-Indexed Light Deflector // IBM Journal. 1964. Jan. P. 64-67.
13. Прангишвили И.В., Стецюра Г.Г. Микропроцессорные системы. М.: Наука. 1980. 237 С.
14. Прангишвили И.В., Подлазов В.С., Стецюра Г.Г. Локальные микропроцессорные вычислительные сети. М.: Наука. 1984. 176 С.
15. Подлазов В.С., Стецюра Г.Г. О потребности в новых логических элементах//Журнал радиотехники и электроники РАН (электронный журнал). 2009. №6.  
<http://jre.cplire.ru/jre/journal.html>
16. The I<sup>2</sup>C-Bus Specification Version 2.1. Philips. 2000. 46 P.
17. Стецюра Г.Г. Быстрые распределенные алгоритмы получения взаимно согласованных решений в системах жесткого реального времени// Автоматика и телемеханика. 2008. №10. С. 162-167.
18. Стецюра Г.Г. Быстрые децентрализованные алгоритмы устранения конфликтов и тупиков при доступе к ресурсам в системах обработки данных и управления//Автоматика и телемеханика. 2010. №4. С. 181-190.
19. Стецюра Г.Г. Совмещение вычислений и передачи данных в системах с коммутаторами. Автоматика и телемеханика. 2008. № 5. С. 170-179.
20. Sixto Ortiz Jr. Active Networks: The Programmable Pipeline//Computer. 1998. Vol. 31. no. 8. P. 19-21.
21. Hussain S. A. Active and programmable networks for adaptive architectures and services. Auerbach Publ. 2007. P. 329.
22. Стецюра Г.Г. Уменьшение сложности распределенного коммутатора для параллельных систем обработки данных // Автоматика и телемеханика. 2010. №5. С. 147-154.
23. Method and apparatus for free-space optical interconnects between arbitrary locations in a field using lenses, steering elements and a curved reflecting surface//Patent Application Publication Jun. 21, 2007 US 2007/0139749 A1
24. Бурцев В.С. Параллелизм вычислительных процессов и развитие архитектуры суперэвм (сборник статей) М.: Торус пресс. 2006. 414 С.