

4. Торчигин В. П. Организация многопроцессорных вычислительных комплексов с переменным количеством процессорных элементов // Автометрия. — 1992. — № 1.
5. Ramanan S. V., Jordan H. F. A new serial array time domain permutation algorithm // Optical Computing. — Kobe, Japan, 1990.
6. Каляев А. В. Многопроцессорные системы с программируемой архитектурой. — М.: Радио и связь, 1984.

Поступила в редакцию 18 июля 1991 г.

УДК 681.323.535

В. П. Торчигин

(Москва)

ОРГАНИЗАЦИЯ МНОГОПРОЦЕССОРНЫХ ВЫЧИСЛИТЕЛЬНЫХ КОМПЛЕКСОВ С ПЕРЕМЕННЫМ КОЛИЧЕСТВОМ ПРОЦЕССОРНЫХ ЭЛЕМЕНТОВ

Рассматриваются вопросы реализации на основе конвейерных векторных ЭВМ модели вычислений, принятой в многопроцессорных вычислительных комплексах SIMD-архитектуры с массовым параллелизмом типа "Connection Machine". Проанализированы достоинства такого подхода и необходимые дополнительные аппаратные средства с традиционных векторных ЭВМ.

Введение. С 1986 г. в США фирмой "Thinking Machine" коммерчески поставляется суперЭВМ под названием "Connection Machine" (СМ), содержащая $64K$ процессорных элементов (ПЭ), где $K = 1024$. Используемый в этой ЭВМ подход к организации вычислений привлек внимание многих разработчиков суперЭВМ, поскольку он позволяет за счет массового распараллеливания вычислений резко повысить общую производительность вычислительной системы.

Появилось достаточно много публикаций [1—7], цель которых показать довольно широкую область применения СМ и преимущества ее архитектуры при программировании и решении различных сложных прикладных задач, требующих большого объема вычислений. На основе накопленного опыта распараллеливания алгоритмов, имеющих, на первый взгляд, сугубо последовательную структуру, авторы выдвинули предположение, что для многопроцессорных систем с архитектурой СМ возможно эффективное параллельное выполнение практически любых алгоритмов, для которых отношение объема обрабатываемых данных к длине программы достаточно велико. Все попытки найти опровергающий данное предположение контрпример пока не увенчались успехом. Многие рассматриваемые в указанных работах примеры первоначально привлекли к себе внимание как возможные контрпримеры, но затем перешли в разряд примеров, подтверждающих сделанное предположение.

Этот нетривиальный результат дает основание по-новому подойти к организации решения таких задач на ЭВМ, сходных с векторными и конвейерными, но имеющими дополнительные аппаратные средства для передачи сообщений между процессорными элементами. Данные, относящиеся к различным процессорным элементам, образуют вектор и обрабатываются последовательно исполнительным устройством в конвейерном режиме. При этом коммуникационные средства СМ могут быть реализованы дополнительными минимальными аппаратными средствами, выполняющими перетасовку данных в векторе. Особенно оправдан этот подход при использовании сверхбыстродействующих оптических элементов, для которых период поступления обрабатываемых оптических сигналов τ значительно меньше времени передачи сигналов T между элементами ($T/\tau = 10^3—10^5$) [8].

При таком подходе происходит как бы возврат от параллельных вычислительных систем к последовательным, что, на первый взгляд, противоречит общим тенденциям развития вычислительной техники. Однако последовательная конвейерная ЭВМ, моделирующая работу многопроцессорной системы типа СМ, имеет ряд серьезных преимуществ. 1. Более полно используются возможности аппаратуры в центральном процессоре (ЦП) и оперативной памяти (ОП). 2. Количество виртуальных процессорных элементов может устанавливаться в гораздо более широких пределах и с гораздо большей точностью. 3. Вместо дорогостоящей коммуникационной сети (КС), занимающей большой объем и требующей большое количество проводниковых линий связи, используются аппаратные средства на основе стандартного запоминающего устройства с произвольной выборкой (ЗУПВ). 4. На несколько порядков увеличивается сложность выполняемых в ПЭ команд при сохранении того же темпа обработки при незначительном увеличении объема оборудования за счет выполнения операций в конвейерном режиме.

Что касается преимуществ СМ, связанных с массивным распараллеливанием вычислений, то в принятом подходе они полностью сохраняются. Конвейерные ЭВМ также могут быть объединены с помощью КС, как это делается в СМ для ПЭ, но при этом количество узлов в такой КС может быть значительно меньше при условии, что производительности сравниваемых систем одинаковы.

Внимание к векторным ЭВМ объясняется также тем обстоятельством, что при их реализации могут быть использованы оптические логические элементы, которые допускают очень высокий темп обработки поступающих на них логических сигналов, но имеют относительно большие времена задержки.

В работе сначала приводятся принципы построения и основные параметры СМ, а затем рассматриваются архитектура, система команд и схемотехника векторной ЭВМ, выполняющей те же функции, оценивается объем необходимого для реализации оборудования.

Архитектура и методы организации вычислений на СМ. СМ состоит из $64 K$ процессорных элементов, связанных коммуникационной сетью типа гиперкуб. Каждый ПЭ работает со своим комплектом данных под управлением одной и той же программы, т. е. СМ относится к классу SIMD (рис. 1). Здесь 0—3 — управляющие ЭВМ типа VAX, которые связаны через электронный коммутатор с четырьмя устройствами управления (УУ) 5—8, каждое из которых управляет $16 K$ процессорными элементами, выполняющими одну и ту же команду. Высокопроизводительная система ввода-вывода 9 соединяет ПЭ с внешними запоминающими устройствами (ВЗУ) 10 и с высокоскоростным графическим дисплеем 11.

Программное обеспечение основано на операционной системе (ОС) управляющей ЭВМ и имеет минимальные расширения. Программа на СМ выполняется обычным последовательным образом и не требует дополнительных средств для синхронизации. Пользователь может достаточно просто составлять программы, использующие всю мощь аппаратуры СМ.

Полная конфигурация СМ из $64 K$ ПЭ содержит $4 K$ кристаллов для ПЭ, $2 K$ кристаллов для интерфейса с $2 K$ кристаллами для выполнения операций

с плавающей запятой. Каждый ПЭ имеет $64 K$ оперативной памяти, адресуемой с точностью до бита. Общий объем оперативной памяти у $64 K$ ПЭ составляет $4 K^3$ бит. АЛУ у каждого ПЭ одноразрядное, состоит из 3-входного и 2-выходного логического элементов и регистра признаков. Один из двух результатов заносится в оперативную память ПЭ, другой — в регистр признаков. Выполнение операции зависит от значения признака. Если оно равно нулю, то результат не заносится в ОП.

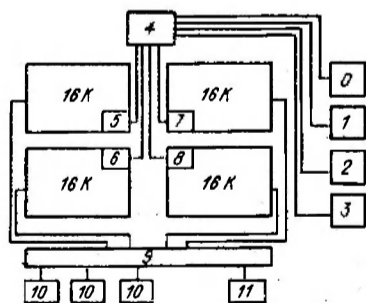


Рис. 1

Арифметические операции выполняются последовательным образом поочередно с каждым разрядом.

СМ имеет следующие возможности для передачи данных между различными ПЭ.

1. Одновременная рассылка одной и той же информации из управляющей ЭВМ по всем ПЭ (broadcasting).

2. Глобальное ИЛИ, позволяющее объединять по алгоритму «логическое ИЛИ» выходы всех ПЭ, благодаря чему появляется возможность быстро определять исключительные ситуации или условия окончания.

3. Связь через коммуникационную сеть типа гиперкуб, являющаяся основной для ОС и многочисленных параллельных примитивов. Аппаратно обеспечиваются параллельная передача сообщений в конвейерном режиме, выбор маршрута для каждого сообщения через свободные в данное время линии связи, буферизация данных в узлах гиперкуба при неравномерном поступлении информации, объединение сообщений, поступающих по одному адресу по алгоритмам «логическое ИЛИ», следующее сообщение забивает предыдущее, запись максимума между следующим и предыдущим сообщениями, запись минимума.

При передаче сообщения через гиперкуб адрес, по которому передается сообщение, берется из оперативной памяти ПЭ, откуда передается сообщение. В процессорных элементах он хранится в виде указателя (номер ПЭ, которому сообщение предназначено). Сообщение представляет собой целое количество смежных битов в оперативной памяти ПЭ и определяется адресом в оперативной памяти начального бита и количеством последующих битов.

Время передачи сообщений зависит от степени соответствия топологии коммуникационной сети и топологии связей между ПЭ, участвующими в обмене сообщениями. В том случае, если связи между ПЭ можно представить в виде регулярной многомерной решетки, в узлах которой находятся ПЭ, время передачи составляет 3 мкс на бит. Тридцатидвухразрядное слово передается за 96 мкс. С учетом одновременной передачи по всему множеству связей темп передачи составляет 20 Гбит/с (20×10^9 бит/с).

В том случае, если связи между ПЭ произвольны, время передачи 32-разрядного слова составляет 600 мкс (темп — 3 Гбит/с). Пиковая полоса пропускания всей коммуникационной сети превосходит 50 Гбит/с.

Организация и принципы работы векторной ЭВМ. Нетривиальный вывод, сделанный разработчиками СМ на основе опыта программирования задач различных классов, о том, что СМ может быть использована для решения любых задач, для которых объем обрабатываемых данных существенно превосходит объем обрабатываемой программы, заставляет по-новому подойти к организации вычислительного процесса для аналогичных задач на векторной ЭВМ.

Рассмотрим, как будет выполняться в векторной ЭВМ с одним конвейерным исполнительным устройством операция, которая в СМ выполняется многими ПЭ в одно и то же время, и какие при этом могут быть проигнорированы требования, учитываемые в ПЭ для СМ и в традиционных суперЭВМ. Можно выделить следующие особенности работы рассматриваемой векторной ЭВМ.

1. Адресуемой порцией данных является не машинное слово, а порция в K раз больше, где K — количество ПЭ в СМ.

2. В исполнительное устройство порция данных поступает не вся сразу, а последовательно: сначала поступают данные, относящиеся к 0-му ПЭ, затем — к первому ПЭ и т. д., последняя порция данных относится к $(K - 1)$ -му ПЭ.

3. Результаты выполнения операции над данными, относящимися к первым ПЭ, получатся раньше, чем будет произведена операция над той частью порции данных, которые относятся к последним ПЭ. Таким образом, результат получается не сразу, а постепенно и представлен также в виде потока данных.

4. Признаки выполнения операций, которые в СМ хранятся в регистре каждого ПЭ, в рассматриваемой векторной ЭВМ хранятся в той же памяти, что

и данные. При выполнении команды значения указанных в ней признаков поступают в виде однобитовых потоков по отдельным каналам синхронно с операндами. При этом имеются два потока считываемых признаков и один поток записываемых. Таким образом, оперативная память рассматриваемой векторной ЭВМ имеет восемь портов: два порта $K1, K2$ для считывания массивов операндов, участвующих в операции, один порт $K5$ для записи получаемого потока результатов, два порта $K3, K4$ для считывания признаков, влияющих на выполнение операции, один порт $K6$ для записи получаемого потока признаков и два порта $K7, K8$ для обмена с внешними запоминающими устройствами (рис. 2, а).

Обращение из каждого порта в ЗУПВ производится поочередно в режиме разделения времени с использованием широко известного приема, называемого интерливингом. Аналогичные методы используются в традиционных векторных ЭВМ. При выполнении операции над элементами вектора организуются четыре потока считываемых операндов по адресам $A1 - A4$ и с некоторой задержкой два потока записываемых операндов по адресам $A5, A6$ (рис. 2, б).

5. Размер обрабатываемой порции данных может равняться не только одному биту, как в ПЭ, но и быть практически произвольным.

6. Набор команд, выполняемых исполнительным устройством, может быть гораздо богаче, чем в ПЭ для СМ, где набор полезных операций над одноразрядными операндами весьма ограничен. Над многоразрядными операндами могут выполняться все операции, используемые в традиционных ЭВМ, в том числе и в операции над числами с плавающей запятой.

Из этих особенностей вытекают следующие следствия, которые существенным образом влияют на архитектуру векторной ЭВМ. 1. Важно не абсолютное время выполнения операции над данными, относящимися к одному ПЭ, а темп поступления данных в исполнительное устройство. 2. Производительность векторной ЭВМ при достаточно больших K определяется в основном временем выполнения операции над данными для всех ПЭ (макрооперации), которое в этом случае не зависит от типа выполняемой операции и пропорционально количеству моделируемых ПЭ.

Отсюда следует, что для получения максимальной производительности векторной ЭВМ необходимо обеспечить максимальный темп поступления операндов, т. е. максимально возможную частоту тактирования. При использовании 2-тактной системы синхронизации и синхронной схемотехники максимальная частота тактирования равна $1/2\tau \dots 1/4\tau$, где τ — время переключения используемых элементов [9]. Таким образом, в конвейерном режиме возможен темп поступления операндов в исполнительное устройство, равный $1/2\tau \dots 1/4\tau$, и, следовательно, производительность устройства равна $1/2\tau \dots 1/4\tau$ операций любой сложности в секунду.

Что касается оперативной памяти, то требуемый темп считывания и записи операндов может быть обеспечен известными средствами с использованием интерливинга. Например, если используются кристаллы статического запоминающего устройства с произвольной выборкой (СЗУПВ), с временем выборки 32 нс, шириной шины данных 8 разрядов и емкостью 256 Кбит [1], то применение $2K$ таких кристаллов позволяет организовать оперативную

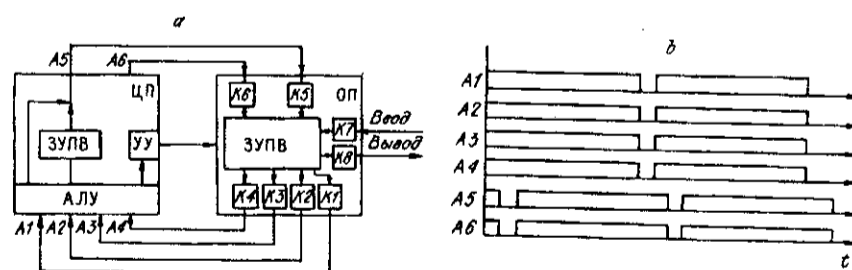


Рис. 2

память емкостью 256 Мбит с восьмью портами, имеющими шину данных шириной 64 разряда и темп чтения/записи 64 бит/нс, что находится в полном соответствии с требованиями СЗУПВ со стороны рассматриваемого АЛУ.

При необходимости общий объем используемой оперативной памяти может быть увеличен за счет мультиплицирования оборудования в любое целое число раз.

Отметим следующие архитектурные особенности векторной ЭВМ при принятом методе организации вычислений. 1. Совершенно оказываются ненужными средства, имеющиеся практически во всех ЭВМ, которые предназначены для сокращения количества обращений к ОП и повышения за счет этого общей производительности ЭВМ (регистры общего назначения, ассоциативное запоминающее устройство чисел, КЭШ, аппаратно реализованная верхушка стека и т. д.). Как уже отмечалось, в рассматриваемой векторной ЭВМ производительность определяется только темпом поступления операндов в АЛУ, который зависит только от частоты тактирования используемых элементов. 2. Поскольку информация из ОП передается большими порциями в виде потока данных, а макрокоманды поступают из УУ, совершенно из других соображений выбирается оптимальное количество адресов в команде. В традиционных ЭВМ при таком выборе в первую очередь принимается во внимание соотношение между величиной машинного слова и объемом ОП, а также то обстоятельство, что одноадресная (а лучше безадресная) система команд позволяет минимизировать объем передаваемых данных между ОП и АЛУ.

В рассматриваемой векторной ЭВМ наиболее естественна 6-адресная система команд (рис. 3). Первые два адреса задают параметры двух потоков данных, над которыми выполняется операция, третий адрес задает место в ОП, с которого начинает записываться поток получающихся результатов. Четвертый и пятый адреса задают параметры считываемых потоков признаков, шестой адрес — параметры записываемого потока признаков.

Принципиальным отличием рассматриваемой векторной ЭВМ от традиционных является введение дополнительных аппаратных средств, выполняющих те же функции относительно обрабатываемых данных, что и коммуникационная среда в СМ. Как неоднократно указывалось авторами СМ, именно наличие мощной и достаточно универсальной КС позволяет использовать СМ для решения самых различных классов задач.

В рассматриваемой векторной ЭВМ передача сообщения от одного ПЭ к другому сводится к перемещению элемента вектора, соответствующего ПЭ-передатчику, на место элемента вектора, соответствующего ПЭ-приемнику. Изменение местоположения элементов вектора в векторе может быть реализовано при помощи обычного ЗУПВ, если в него записывать поток последовательных элементов вектора по произвольным адресам, поступающим во втором потоке, а затем считывать последовательным образом.

Таким образом, в рассматриваемой векторной ЭВМ КС сводится к организации в ЦП дополнительной оперативной памяти с произвольным доступом, объем которой в словах равен максимально возможному количеству виртуальных ПЭ, например 10^6 слов. В эту ОП должны быть записаны значения из первого потока по произвольным адресам, указанным во втором потоке, причем значения и их адреса поступают каждый такт.

При сделанных предположениях объем этой ОП должен быть равен 64 Мбит (8 Мбайт), т. е. объем этой памяти в 64 раза меньше, чем объем всей ОП в СМ, и в 4 раза меньше, чем объем устройства, показанного на рис. 2. Разумеется, при реализации этой памяти могут быть использованы кристаллы меньшей емкости и поэтому более быстродействующие.

Методы, используемые в векторной ЭВМ при организации ОП, обеспечивающие считывание/запись информации в виде непрерывных потоков в смежные области ОП, оказываются здесь неприменимы, поскольку из-за непредсказуемого характера поступления значений адресов не удастся организовать гарантированную запись всех поступающих данных. В этом случае в том или ином виде должен быть использован буфер записываемых данных.

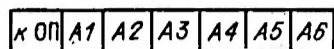


Рис. 3

На рис. 4 показана схема одного из возможных устройств, которое решает поставленную задачу. Это устройство состоит из двух частей, каждая из которых является ОП объемом 10^6 слов. При этом размер слова в левой части равен 64, а в правой — $64 + A$, где A — количество разрядов, отведенных для задания номера виртуального ПЭ (20 разрядов в рассматриваемом случае).

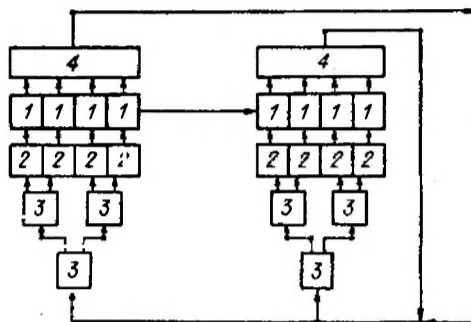


Рис. 4

Алгоритм работы этого устройства следующий. Поступающие на вход устройства потоки значений и указателей разветвляются по двум направлениям и конвейерным образом распространяются к СЗУПВ 1, в которые значения могут быть записаны по соответствующим адресам.

При прохождении по левой части младшие разряды соответствующего указателя определяют путь от одного узла 3 к другому (например, 0 соответствует передаче информации налево, а 1 — направо).

Если информация достигла некоторого СЗУПВ 1, приемный регистр 2 которого в этот момент оказался свободным, то производится запись значения в это СЗУПВ по адресу, определяемому оставшимися старшими разрядами указателя. При этом информация, поступающая в правую часть, никуда не записывается и бесследно исчезает.

Если же приемный регистр 2 СЗУПВ в левой части оказался занятым, то рассматриваемая информация в левой части никуда не записывается и бесследно исчезает. Эта же информация в правой части, которая представляет, по сути дела, буфер, в этом случае должна быть записана в очередную свободную ячейку. Поскольку запись в правой части производится по последовательным адресам, все рассмотренные выше методы организации интерливинга оказываются здесь применимыми. Отличие состоит лишь в том, что информация в правую часть поступает не сплошным потоком, а потоком с разрывами, которые получают в тех случаях, когда удается записать информацию в левую часть. Поэтому правая часть управляется не периодически следующими синхрипульсами (СИ), а тактовыми импульсами, получающимися из сигналов, соответствующих занятости СЗУПВ в левой части.

Возникает естественный вопрос: чем рассматриваемая организация векторной ЭВМ эффективнее той, которая используется в СМ и в традиционных ЭВМ? При ответе на этот вопрос следует отметить следующие моменты.

1. По сравнению с обычными векторными ЭВМ совершенно изменяется стиль программирования и обработки информации. Все подготовительные операции над элементами векторов, необходимое упорядочивание их элементов для последующей обработки в конвейерном режиме в рассматриваемой векторной ЭВМ выполняются при помощи дополнительных аппаратных средств в виде рассмотренного ЗУПВ в составе ЦП.

2. Экономия оборудования. В рассматриваемой векторной ЭВМ нет необходимости минимизировать абсолютное время выполнения операции над данными, соответствующими одному ПЭ. Это позволяет выбрать предельно высокую частоту тактирования, отличающуюся от оптимальной для традиционных ЭВМ.

Известно, что в традиционных ЭВМ время между смежными сериями синхрипульсов в 9—10 раз превосходит время переключения используемых элементов (между смежными регистровыми станциями имеются комбинационные схемы, в которых максимальная цепочка последовательных переключений логических элементов достигает 9—10). При этом каждый эле-

мент за время между смежными сериями СИ производит только одно полезное логическое срабатывание, хотя по своим скоростным возможностям мог бы переключаться 9—10 раз.

Это обстоятельство позволяет на векторных ЭВМ получать производительность в несколько десятков раз выше, чем на скалярных, использующих ту же элементную базу. В СМ такая возможность не используется, хотя одновременное выполнение на различных ПЭ многих независимых по данным операций дает для этого все основания.

3. Параметризуя размер участвующих в операции порций данных, мож- операций несколько замедляется. Вообще говоря, в СМ предусмотрены средства для организации виртуальных ПЭ, количество которых больше количества физических ПЭ. Однако предельное количество виртуальных ПЭ ограничено снизу достаточно большим числом в 64 К. Кроме того, в принятом в СМ методе имитации работы нескольких виртуальных ПЭ на одном физическом никак не учитывается тот факт, что данные, относящиеся к различным виртуальным ПЭ, независимы и могут обрабатываться конвейерным образом (т. е. в 10 раз быстрее).

4. Значительно сокращаются аппаратные средства, необходимые для реализации различных имеющихся в СМ коммуникационных цепей. Объясняется это тем, что в СМ процессорные элементы пространственно разделены между собой, и поэтому для организации связи между ними требуются соответствующие физические линии связи и аппаратура, обеспечивающая их функционирование. В рассматриваемой векторной ЭВМ данные, которые должны передаваться между ПЭ, оказываются в различные моменты времени в одном и том же месте пространства. В этом случае необходимые связи между ПЭ организуются гораздо проще.

Действительно, рассылка одной и той же информации из управляющей ЭВМ или УУ по всем ПЭ производится просто циклической передачей этой информации в поток результатов. Коммуникация между ПЭ типа «глобальное ИЛИ» реализуется в АЛУ путем выполнения этой операции над операндами, поступающими в одном из входных потоков.

Несколько сложнее реализуется общий случай коммуникации, для реализации которого в СМ используется коммуникационная сеть типа гиперкуб, включающая в себя, кроме линий связи, 4 К связанных процессоров. Использование для этих целей ЗУПВ в ЦП рассматриваемой векторной ЭВМ приводит к существенно меньшим затратам оборудования, особенно линий связи.

5. Каждый виртуальный ПЭ векторной ЭВМ представляет собой не примитивнейший одноразрядный ПЭ, какой имеется в СМ, а полноразрядный ПЭ с богатой системой команд. Время выполнения любой команды практически не зависит от ее сложности. Это обстоятельство позволяет за счет повышения уровня выполняемых команд повысить общую производительность векторной ЭВМ (концепция CISC — Complex Instruction Set Computer, противоположная концепции RISC — Reduce Instruction Set Computer).

Что касается состава выполняемых ЦП команд, то он традиционен. Это арифметические операции над целыми и вещественными числами (разрядность чисел может быть параметризована, т. е. устанавливается в ЦП специальной командой), логические операции, операции отношения, выделение полей из значений (значения указываются в А1, параметры полей — в А2, адрес результатов — в А3). К перечисленным операциям следует добавить уже упоминавшиеся операции рассылки одной и той же информации из управляющей ЭВМ по всем ПЭ (рассылаемое значение указывается непосредственно в команде на месте А1 и А2, адрес в ОП, куда будет направлен поток

одинаковых значений, — в А3); операцию «глобальное ИЛИ» (в А1 указывается адрес начала потока значений, объединяемых по ИЛИ, в А2 кодируются другие возможные алгоритмы объединения, результат объединения остается в ЦП в виде признака); операцию, устанавливающую в ЦП и ОП регистры, определяющие размеры обрабатываемых порций данных; операцию, устанавливающую в ЦП разрядность обрабатываемых слов.

Коммуникационные средства, реализованные в СМ при помощи коммуникационной сети типа гиперкуб, в рассматриваемой векторной ЭВМ могут быть реализованы при помощи операции рассылки значений по указателям. В этой операции в А1 указывается адрес начального потока значений, в А2 — адрес начала потока указателей, в А3 — адрес записи результирующего потока значений, полученного путем перетасовки первого потока по А1 в соответствии с пачкой указателей по А2. Указатель представляет собой целое, определяющее номер виртуального ПЭ, на который он указывает.

Заключение. Рассмотрена архитектура многопроцессорных вычислительных комплексов, которая позволяет полностью использовать потенциальную пропускную способность кристаллов оперативной памяти и скоростные возможности применяемых логических элементов при решении широкого класса задач, требующих больших объемов вычислений. При этом имеется возможность программно устанавливать количество необходимых виртуальных процессорных элементов, приспособивая таким образом архитектуру МК к специфике конкретного применения.

Автор выражает благодарность И. К. Хайлову за ценные замечания и советы.

СПИСОК ЛИТЕРАТУРЫ

1. Turker L. W., Robertson G. G. Architecture and application of the connection machine // Computer.—1988.—N 8.—P. 26.
2. Fahman S. E. Connectionis architecture for artificial intelligence // Computer.—1987.—N 1.—P. 100.
3. Stanfil C., Kacle B. Parallel free text search of connection machine system // Commun. ACM.—1986.—N 12.
4. Thinking Machine corporation. Introduction to data level parallelism // Ibid.—N 4.
5. Hilis D., Steele S. Data parallel algorithms // Ibid.—N 12.
6. Stanfill C., Walts D. Toward memory-based reasoning // Ibid.
7. Walts D. Application of the connection machine // Computer.—1987.—N 1.—P. 85.
8. Торчигин В. П. Использование оптических средств для передачи и обработки информации в многопроцессорных вычислительных комплексах // Автоматика.—1992.—№ 1.
9. Филиппов А. Г., Белкин О. С. Проектирование логических узлов ЭВМ.—М.: Сов. радио, 1974.

Поступила в редакцию 22 апреля 1991 г.

УДК 681.325.65 : 681.7

А. А. Вербовецкий

(Москва)

СТРУКТУРНЫЕ И СХЕМОТЕХНИЧЕСКИЕ РЕШЕНИЯ МНОГОКАНАЛЬНЫХ СВЕТОВОДНЫХ ЛОГИЧЕСКИХ СИСТЕМ ДЛЯ СУПЕРЭВМ

Предложены структурные и схемотехнические варианты построения многоканальных световодных логических устройств. Оценка производительности таких устройств показывает, что при логической обработке 50-разрядных парафазных слов она может достигать 10^{15} опер./с. Рассмотрены применения метода и схем для выполнения операций параллельного сложения и вычитания двоичных переменных.