

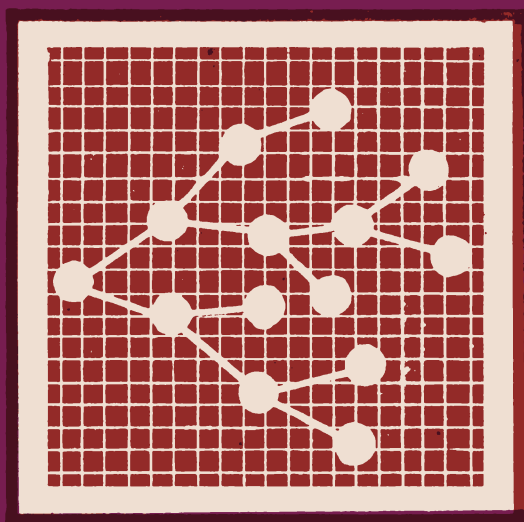
НОВОЕ  
ЖИЗНИ, НАУКЕ,  
ТЕХНИКЕ

ЗНАНИЕ

С. Я. Беркович

Ю. Я. Кочин

# АССОЦИАТИВНАЯ ПАМЯТЬ



8/1976

СЕРИЯ  
МАТЕМАТИКА,  
КИБЕРНЕТИКА

---

НОВОЕ  
В ЖИЗНИ, НАУКЕ,  
ТЕХНИКЕ

Серия «Математика, кибернетика»  
№ 8, 1976 г.  
Издается ежемесячно с 1967 г.

---

**С. Я. Беркович,**

кандидат физико-математических наук

**Ю. Я. Кочин,**

кандидат технических наук

## **АССОЦИАТИВНАЯ ПАМЯТЬ**

**ИЗДАТЕЛЬСТВО «ЗНАНИЕ»  
Москва 1976**

## СОДЕРЖАНИЕ

Предисловие . . . . .	3
1. Общие принципы функционирования . . . . .	5
2. Особенности обработки информации . . . . .	15
3. Вопросы построения . . . . .	32
4. Организация ассоциативной виртуальной памяти . . . . .	44
Заключение . . . . .	58
Литература . . . . .	63

**Беркович С. Я. и Кочин Ю. Я.**

**Б48 Ассоциативная память. М., «Знание», 1976.**

64 с. (Новое в жизни, науке, технике. Серия «Математика, кибернетика», 8. Издается ежемесячно с 1967 г.)

В брошюре рассматриваются вопросы организации процессов поиска и обработки информации на основе механизма ассоциативной памяти, который обеспечивает выборку данных по их содержанию. Ассоциативная память является одним из важнейших средств повышения эффективности работы современных и будущих ЭВМ. Затронутый круг вопросов примыкает ко многим областям теоретической и технической кибернетики.

Брошюра рассчитана на широкий круг читателей, интересующихся проблемами построения и применения вычислительных машин.

30500

© Издательство «Знание», 1976 г.

## ПРЕДИСЛОВИЕ

Решение задач по повышению эффективности управления народным хозяйством страны, поставленных партией и правительством на XXV съезде КПСС, требует дальнейшего совершенствования средств вычислительной техники. Одно из важнейших направлений в этом деле связано с созданием информационных систем, способных хранить и обрабатывать большие объемы данных.

Для эффективного выполнения операции поиска, сортировки, упорядочения и логического сравнения массивов данных, а также для групповой обработки массивов в перспективных управляющих вычислительных системах начинают применять ассоциативные запоминающие устройства (АЗУ), в которых запись и извлечение информации осуществляются не по адресу (номеру) ячейки, а по некоторым признакам, свойственным этой информации, представляющим собой часть искомого слова или отдельные его разряды, расположенные произвольно. Поэтому в АЗУ не нужно знать месторасположение информации, достаточно подать на вход некоторые признаки искомой информации, и тогда на выходе АЗУ можно получить всю информацию, обладающую заданными признаками. При этом операции просмотра и сравнения выполняются параллельно по всем словам, хранящимся в памяти.

Использование ассоциативного принципа поиска и обработки информации позволяет во многих случаях повысить производительность систем, обеспечить быстрый поиск информации в памяти большого объема, осуществить замену части программных средств аппаратурными, упростить программирование и сократить время, требуемое на решение задач.

Использование ассоциативных запоминающих устройств в вычислительных системах позволяет эффективно управ-

лать поиском информации в оперативных запоминающих устройствах, осуществлять динамическое распределение памяти аппаратными средствами, реализовать ассоциативную виртуальную память и ускорить обмен информации между различными уровнями памяти машин, осуществить аппаратную реализацию транслятора и части операционной системы и др. Ассоциативный принцип обмена информацией с внешними накопителями и поиск информации в больших массивах данных упрощают операционную систему и увеличивают ее быстродействие.

За последние десять лет по вопросам построения и способам реализации АЗУ опубликовано большое число статей в отечественной и иностранной периодической литературе. Однако до сих пор отсутствует обобщающая работа по организации ассоциативного доступа к большим информационным массивам и построению ассоциативной виртуальной памяти, по эффективности использования АЗУ для поисковых и вычислительных задач и возможности построения больших интегральных схем ассоциативной логической структуры.

Настоящая брошюра позволит в значительной степени восполнить пробел и окажется полезной для разработчиков перспективных ЭВМ и их математического обеспечения.

Профессор *И. В. Прангишвили*



# 1

## ОБЩИЕ ПРИНЦИПЫ ФУНКЦИОНИРОВАНИЯ

Память является важнейшей составной частью любой кибернетической системы.

Дальнейший прогресс в области вычислительной техники в значительной мере связан с необходимостью совершенствования устройств хранения информации.

При этом следует иметь в виду не только улучшение количественных характеристик запоминающих устройств (ЗУ), таких, как информационная емкость, быстродействие, стоимость, надежность и т. п., но и создание качественно более эффективных структур памяти, способствующих поиску требуемой информации.

В составе вычислительных машин содержатся различные по техническим характеристикам ЗУ, образующие память ЭВМ с иерархической структурой. Это определяется тем, что в одном типе ЗУ не удастся совместить противоречивые требования, связанные, например, с необходимостью обеспечить большую емкость и высокое быстродействие памяти. В иерархической системе ЗУ можно выделить внутреннюю (оперативную) и внешнюю память. Оперативная память предназначена для размещения информации, непосредственно используемой в вычислительном процессе. Внешняя память обычно имеет значительно большую емкость и меньшее быстродействие, и для использования хранящейся в ней информации необходимо предварительно переместить эту информацию в оперативную память.

В настоящее время почти все ЗУ строятся по адресному принципу, т. е. каждой ячейке присваивается определенный номер (адрес), по которому производится обращение к содержимому ячейки. С физической точки зрения адрес запоминающей ячейки, как правило, определяет ее положение. С логической точки зрения адрес представляет со-

бой не более чем добавку к хранимому в ячейке слову, необходимую для его нахождения.

Применение адресного принципа организации памяти диктуется особенностями используемых элементов вычислительной техники и, вообще говоря, не является ни наиболее естественным, ни наиболее удобным.

Первоначальные соображения по поводу построения ЗУ с неадресным обращением типа каталога появились в 1956 году в связи с начавшейся разработкой сверхпроводниковых элементов — криотронов. Логические и математические вопросы организации ЗУ, в которых за счет усложнения их структуры достигается возможность производить обращение к словам по содержанию некоторой их части, были разработаны в 1960 году Г. Г. Стецурой.

Такого рода ЗУ вне зависимости от их специфических особенностей называют ассоциативными запоминающими устройствами (АЗУ). Термин «ассоциативный», взятый из физиологии, характеризует известное свойство человеческой памяти, позволяющее устанавливать взаимосвязи между различными информационными объектами.

На исследования и разработки в области ассоциативных ЗУ были затрачены значительные усилия. Имеется много сотен работ, посвященных этой проблеме, и в настоящее время число публикаций по данному вопросу составляет несколько десятков в год.

Наиболее общим случаем устройства, допускающего обращение к информации по содержанию, является так называемое полностью ассоциативное устройство, в котором признак поиска может быть произвольно расположенным во всех разрядах хранящегося слова. Можно выделить также три наиболее характерных типа частично-ассоциативных ЗУ, в которых поиск проводится по некоторой ограниченной части слова:

- 1) память, обладающая свойствами полностью ассоциативного устройства, но в пределах ограниченной по длине части слова;

- 2) память с несколькими ассоциативными признаками, любая комбинация которых может участвовать в опросе;

- 3) память с одним признаком фиксированной длины, все разряды которого одновременно используются для опроса (при этом искомой информацией является остальная часть слова).

Эта классификация, будучи связанной с проблемами технической реализации ассоциативных ЗУ, отражает так-

же различные функциональные требования к ассоциативной памяти со стороны информационных систем.

Информационную структуру объектов можно представлять в виде многомерной точечной решетки, на которой задается функция какой-либо природы. Компонентой такой структуры  $n$ -мерного объекта является значение функции, снабженное  $n$  независимыми индексами.

Свойства памяти, касающиеся проблем информационного поиска, удобно трактовать в терминах адекватности отображения размерности информационных структур. Для случая биологических систем указанные структуры в пределе можно рассматривать и как непрерывные. Размерность, как известно, представляет собой топологический инвариант, и для того чтобы этот инвариант при отображении в памяти электронно-вычислительной машины или живого организма сохранялся, такое отображение должно быть «непрерывным» в том смысле, что «близкие» элементы объекта должны оставаться «близкими» в устройстве хранения информации. При этом элементами, «близкими» в устройстве хранения информации, следует считать такие, связь между которыми может быть осуществлена за достаточно малое время, так что в этом смысле близость оказывается характеристикой, вообще говоря, логического, а не физического отношения.

Малость промежутка времени, отводимого на установление связей, следует понимать в сопоставлении с некоторым характерным временем, затрачиваемым на определенные процедуры обработки информационного содержания объекта. Например, в случае рассмотрения технических устройств памяти в составе информационно-справочных систем имеет смысл говорить, что в них адекватно отображается размерность информационной структуры объекта, когда время перехода между соседними элементами структуры может быть сделано сопоставимым с требуемым временем реакции системы. В терминах уменьшения степени адекватности отображения размерности информационной структуры объектов можно трактовать, в частности, также такие известные факты, как замедление переработки информации при моделировании вычислительной среды на некоторой решетке посредством такой же среды на решетке меньшей размерности. В случае биологической памяти при рассмотрении вопроса об адекватном отображении размерности объекта следует сопоставлять время, в течение которого должно быть обеспечено установление связей меж-



ду элементами информационной структуры и минимальный разрешающий интервал между различными возбуждающими воздействиями.

Следует подчеркнуть, что рассмотрение ограничения времени для установления связей является важным принципиальным моментом, поскольку без учета временного фактора любые структурные взаимоотношения элементов независимо от их расположения могут быть установлены посредством медленной операции последовательного просмотра содержимого памяти. В случае биологической памяти время установления связей между элементами объекта не должно сколько-нибудь заметно превышать минимальный разрешающий интервал между возбуждающими воздействиями, ибо тогда восприятие этих связей, а следовательно, восприятие размерности образа будет нарушено. Таким образом, ограничения на величину размерности адекватно отображаемых объектов представляются следствием ограничения на скорость процессов обработки информации.

Исходя из изложенной трактовки, можно показать, каким образом на основе голографической модели функционирования биологической памяти можно развить известную гипотезу Пуанкаре о физиологической обусловленности свойства трехмерности пространства восприятия.

Обычно адресное запоминающее устройство по своей природе приспособлено для отображения одномерных объектов, элементы которых размещаются в соответствии с упорядочением по одному параметру. Для установления связей между элементами при отображении многомерных объектов нужно применять специальную программную организацию на основе так называемых списковых структур.

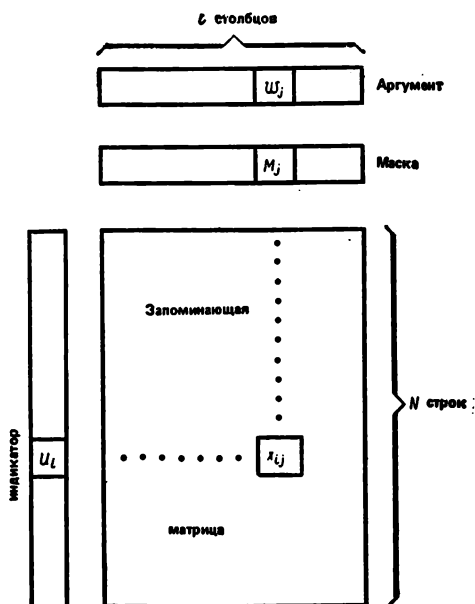
В ассоциативной памяти использование значений индексов в качестве признака поиска позволяет удобно представить объект произвольной размерности, конечно, если отвлечься от технических деталей, касающихся ограниченной разрядности машинного слова. В этом случае при одном обращении к памяти можно сразу выделить подмножество элементов объекта в произвольном разрезе; извлечение же отдельных элементов осуществляется весьма быстро при использовании эффективных процедур так называемого разделения многозначного ответа.

На рис. 1 представлена абстрактная логическая схема АЗУ. Запоминающая среда организована в виде матрицы  $X$ , состоящей из  $N$  строк, в которых размещаются

$r$ -разрядные слова. Признак поиска формируется с помощью векторов аргумента  $W$  и маски  $M$ , размер которых  $r$ ; результат поиска фиксируется индикатором  $U$ , имеющим размер  $N$ . Отличие частично-ассоциативных ЗУ от полностью ассоциативных ЗУ заключается в том, что в частично-ассоциативных ЗУ накладываются соответствующие ограничения на значения элементов вектора  $M$ .

Вектор маски определяет, из каких разрядов аргумента следует сформировать признак поиска. Содержимое разрядов, по которым производится поиск, сравнивается одновременно со всеми элементами матрицы  $X$ , находящимися в одноименных столбцах. Если в какой-либо строке все значения элементов матрицы совпадают с признаком поиска, то этот результат фиксируется соответствующей компонентой индикатора.

Формально логику работы АЗУ можно описать следующим образом. Будем считать, что значение разряда маски  $M_j$ , равное «1», означает разрешение операции сравнения по разряду аргумента  $W_j$ ; соответственно «0» означает запрещение. Если содержимое  $W_j$ , по которому произво-



Р и с . 1;

дится сравнение, совпадает с каким-либо значением  $x_{ij}$ , то в данном элементе матрицы  $X$  вырабатывается сигнал совпадения  $u_{ij}$ , равный «1», при несовпадении — вырабатывается сигнал «0». Случай запрещения операции сравнения фиксируется как совпадение. Сигналы совпадения и несовпадения в индикаторе обозначаются также — «1» и «0».

Выпишем функциональную таблицу для операций опроса содержимого ассоциативной памяти.

$M_j$	$w_j$	$x_{ij}$	$u_{ij}$
0	0	0	1
0	0	1	1
0	1	0	1
0	1	1	1
1	0	0	1
1	0	1	0
1	1	0	0
1	1	1	1

Нетрудно видеть, что сигнал несовпадения в каждом элементе матрицы  $x$  представляет собой логическое произведение маски и отрицания равнозначности (сложение по модулю 2) аргумента и содержимого памяти:

$$\bar{u}_{ij} = M_j \wedge (w_j \oplus x_{ij}).$$

Значение сигнала совпадения в компоненте индикатора представляет собой логическое произведение всех сигналов совпадения в данной строке:

$$u_j = \bigwedge_{i=1}^r u_{ij}.$$

Вектор  $U$  фиксирует результат основной операции — выделения подмножества слов, обладающих заданным признаком.

Таким образом, если при адресной организации ЗУ выбор слов по номеру является однозначным, то ассоциативный принцип построения памяти приводит к возможности неоднозначного выбора. Возникает специфическая для АЗУ проблема разделения многозначного ответа — последовательная выборка слов из выделенного подмножества.

Схема на рис. 1, представляющая операцию поиска в однородной запоминающей среде, не может описывать операцию разделения многозначного ответа, поскольку для того, чтобы выбрать какое-то слово из выделенного подмножества, необходимо, чтобы это слово хоть чем-нибудь отличалось от остальных; для этого в структуру ассоциативной па-

мяти требуется внести извне некоторую неоднородность, например, с помощью адресного дешифратора.

Интересно отметить связь данного фактора с вопросами, касающимися так называемой аксиомы выбора Цермело. В основе многих математических рассуждений, как впервые заметил Цермело в 1904 году, лежит неявное предположение о существовании «выборочной» функции, позволяющей на заданном множестве непустых множеств каждому из этих множеств поставить в соответствие один его элемент, т. е. указать для каждого множества его представителя. Это предположение было названо аксиомой выбора. Важнейшим следствием аксиомы выбора является доказанная Цермело теорема о существовании полного порядка на любом множестве.

В ассоциативной памяти для того, чтобы производить обращения к отдельным словам вообще и при разделении многозначного ответа в частности, необходимо предусмотреть дополнительно специальные средства управления, которые позволяли бы выбирать из любого подмножества слов хотя бы одно какое-нибудь слово, иначе выборка слов из ассоциативной памяти будет невозможной. Сформальной точки зрения можно сказать, что для множества всех подмножеств слов в ассоциативной памяти рассматриваемые специальные средства представляют собой механизм, обеспечивающий выполнение аксиомы выбора. Отсюда с необходимостью следует, что указанные средства, которыми надо снабдить ассоциативную память, должны быть таковы, чтобы задавать на множестве ее числовых линеек (строк) отношение упорядочения. Это можно сделать либо путем введения цепочки очередности, связывающей слова ассоциативной памяти, либо путем использования адресного дешифратора, который осуществляет отображение множества числовых линеек на некоторый отрезок натурального ряда. Следует обратить внимание на то, что возможны варианты ассоциативной памяти с полностью однородной структурой, как в некоторых ЗУ голографического типа, однако в подобных случаях не предусматривается возможность выбора информации из произвольного подмножества слов.

Недостатками цепочки очередности являются малая скорость работы и большое количество элементов. В связи с этим в АЗУ вводят адресную систему, наличие которой является одним из важнейших свойств быстродействующих АЗУ большой емкости. Это придает АЗУ все свойства обыч-

ной адресной памяти и может быть эффективно использовано не только при разделении многозначного ответа, но и в целом ряде других ассоциативных операций.

Если конструкция дешифратора в адресных ЗУ обычно допускает обращение только к одному слову в одно и то же время, то в случае АЗУ дешифратор должен обеспечить возможность одновременного обращения к группе слов; при этом адрес можно рассматривать как переменный ассоциативный признак, отдельные разряды которого могут быть замаскированы. АЗУ с таким дешифратором является эквивалентным полностью ассоциативному ЗУ без дешифратора, имеющему для каждого слова увеличенное число разрядов, в которые внесен номер данного слова.

Таким образом, проблема разделения многозначного ответа сводится к проблеме упорядоченной выборки слов по содержанию разрядов, хранящих номера этих слов. Хотя обычно выборка рассматривается в порядке возрастания или убывания номеров, ввиду взаимной независимости разрядов, ее можно производить в любом лексикографическом порядке.

Существует целый ряд алгоритмов, позволяющих осуществлять упорядоченную выборку из АЗУ. Подробное описание этих алгоритмов и их сравнительный анализ имеются в книге Краймер Л. П. и др. «Ассоциативные запоминающие устройства», выпущенной издательством «Энергия» в 1967 году.

На основе методов упорядоченной выборки слов в ассоциативной памяти можно производить поиск информации по критериям более сложным, чем простое совпадение; например, поиск максимального (минимального) числа, поиск ближайшего большего (меньшего) в заданных пределах и т. п.

Ассоциативное свойство памяти фактически помогает эффективно реализовать всего две процедуры: выделение подмножества слов, отвечающих заданному критерию опроса, и одновременное одинаковое изменение содержимого определенных разрядов выделенных слов.

Полезность ассоциативного принципа организации памяти представляется довольно очевидной. Перечисление задач, для решения которых желательно использование АЗУ, занимает, по Э. Хенлону, около трех страниц и относится практически ко всем как экономическим, так и военным, а равно и научным применениям ЭВМ. Такого рода оптимистические утверждения основываются на том, что

во многих задачах в равной степени встречаются операции, идеально приспособленные для АЗУ. Однако ввиду относительной сложности создания ассоциативных ЗУ при современном уровне развития технологии и их более высокой стоимости целесообразность применения АЗУ требует подробного анализа. Интересно привести по этому поводу цитату из книги М. Минского, С. Пейперта «Перцептроны», выпущенной издательством «Мир» в 1971 году. «У нас нет теории тех ситуаций, — пишут авторы, — в которых «ассоциативная» память оправдывает свою более высокую по сравнению с адресной стоимость. Бытует множество легенд относительно такого рода противопоставлений, однако большая часть их основана просто на предубеждениях. В тех случаях, которые мы тщательно исследовали, широко распространенные ходячие представления оказались не только «недоказанными» — они зачастую были попросту неверными».

Применение АЗУ наиболее эффективно для задач информационного поиска. При использовании АЗУ в качестве оперативной памяти для вычислительных задач следует учитывать, что существенное ускорение решения может быть достигнуто лишь тогда, когда удельный вес операций ассоциативного типа достаточно высок, ибо в противном случае выигрыш в скорости может быть потерян на фоне большого количества действий другого типа или обменов с внешними устройствами. При этом достижение эффективности при использовании АЗУ не получается просто само по себе, а требует некоторых усилий в деле совершенствования организации вычислительного процесса.

Проведенные исследования показали, что для большинства обычных алгоритмов задач линейной алгебры и линейного программирования использование АЗУ не приводит к значительному увеличению скорости решения, поскольку эти алгоритмы в большинстве случаев используют поэлементные преобразования информационных массивов; АЗУ в данном случае может обеспечить небольшой выигрыш в скорости, практически мало зависящий от размеров задачи. Рассмотрен вопрос об эффективности использования АЗУ для ряда экстремальных комбинаторных задач. В задачах этого типа наиболее часто повторяющиеся вычислительные процедуры связаны с затратой большого количества действий на различные довольно сложные операции поиска при относительно малом объеме арифметических операций. Были проанализированы венгерский

метод, метод потенциалов, алгоритм Джонсона в теории расписаний и метод ветвей и границ. Оказывается, что в данных задачах во всех основных процедурах выигрыш в скорости при использовании АЗУ памятью составляет  $\sim n/r$ , где  $n$  — размерность задачи, а  $r$  — количество разрядов в машинном представлении чисел, так что для рассматриваемого класса задач эта величина является характерным показателем алгоритмической эффективности АЗУ.

Использование ассоциативных принципов обработки информации открывает различные пути совершенствования средств вычислительной техники.

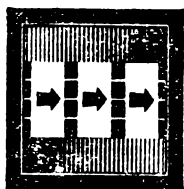
На основе АЗУ с развитой логической структурой можно строить специализированные вычислительные устройства для эффективного выполнения групповых операций — ассоциативные параллельные процессоры. Некоторые особенности ассоциативных принципов обработки информации будут рассмотрены в этой брошюре далее.

Использование ассоциативной памяти наряду с повышением производительности дает возможность улучшить ряд характеристик вычислительных систем, в частности, позволяет изменить структуру математического обеспечения в сторону значительного упрощения.

Следует отметить, что к настоящему времени АЗУ не получили еще достаточно широкого распространения. Это объясняется тем, что создание АЗУ наталкивается на определенные технические трудности. Предполагаемое широкое использование АЗУ в будущем связывают с успехами технологии производства больших интегральных схем (БИС).

Создание 3У ассоциативного типа возможно на основе самых разнообразных физических принципов. С этой точки зрения были изучены практически все типы магнитных запоминающих элементов, полупроводниковые схемы, сверхпроводниковые устройства, оптические системы, различные линии задержки и т. д. Двадцатилетний опыт исследований и разработок в области микроэлектроники в нашей стране и за рубежом показал, что для построения оперативных быстродействующих АЗУ большой емкости (объемом  $\geq 10^7$  бит и с временем ассоциативного обращения  $\leq 1$  мксек) наиболее благоприятная перспектива — использование сверхпроводниковых элементов. (Особенности сверхпроводниковых АЗУ рассмотрены в третьем разделе).

В настоящее время чрезвычайно актуальной является проблема организации ассоциативного доступа к большим информационным массивам на основе использования существующей многоуровневой системы ЗУ. Сложность этой проблемы определяется двумя основными трудностями: во-первых, это ограниченность объема оперативной памяти, во-вторых — наличие противоречия между одномерным характером адресной памяти и многомерной природой пространства признаков. В четвертом разделе описан подход к решению указанной проблемы, позволяющий организовать одноуровневую виртуальную память ассоциативного типа.



2

## ОСОБЕННОСТИ ОБРАБОТКИ ИНФОРМАЦИИ

**Л о г и к а э л е м е н т а р н ы х о п е р а ц и й.** Для выполнения в ассоциативной памяти арифметической и логической обработки информации необходимо, чтобы реализуемая в запоминающей среде система булевых функций обладала полнотой.

Можно считать, что накопитель ассоциативной памяти (см. матрицу  $X$  на рис. 1) представляет собой совмещение матрицы, собственно, запоминающих элементов с некоторой логической сетью, имеющей также вид матрицы, вдоль столбцов и строк которой определены разные логические операции. По характеру операции при опросе следует различать две разновидности ассоциативной памяти:

— память с логикой совпадения, в которой вдоль столбцов выполняется операция равнозначности с последующей конъюнкцией для всех разрядов каждой строки;

— память с логикой несовпадения, в которой выполняется операция неравнозначности с последующей дизъюнкцией.

Указанное обстоятельство приводит к тому, что набор булевых функций, выполняемых между двумя двоичными логическими переменными, зависит от их расположения относительно логической сети.

Возможны следующие три случая;



- обе переменные располагаются в одной строке матрицы памяти;
- переменные располагаются в разных строках;
- одна переменная находится в матрице памяти, а другая — вне поля памяти.

В первом случае реализуется полная система булевых функций. В самом деле, логика, связанная с регистрами опроса и маски, обеспечивает возможность выполнения следующих операций над содержимым произвольного элемента памяти  $x_{ij}$ : «опрос по 1» (при  $M_j=1$ ,  $w_j=1$ ), «опрос по 0» ( $M_j=1$ ,  $w_j=0$ ), «нет опроса» ( $M_j=0$ ). Нетрудно видеть, что в результате воздействия указанных операций реализуются логические функции, соответственно, «тождество», «отрицание» и «константа 1» для ассоциативной памяти с логикой совпадения или «отрицание», «тождество» и «константа 0» для ассоциативной памяти с логикой несовпадения. Кроме того, как было сказано выше, в каждой строке матрицы памяти в зависимости от разновидности выполняется функция конъюнкции или дизъюнкции. Как известно, каждая из систем функций — отрицание и конъюнкция или отрицание и дизъюнкция — является полной и образует минимальный базис в классе функций алгебры логики. Следовательно, любую булеву функцию можно представить суперпозицией функций в любом из указанных базисов.

Второй случай расположения аргументов можно свести к первому, если имеются цепи сдвига в регистре результата ассоциативного опроса для помещения обоих аргументов в одну строку.

В третьем случае можно выполнить любую булеву функцию, если предположить, что аргумент, расположенный вне поля памяти, известен. Иными словами, необходимо дополнительно иметь логические схемы для анализа состояния указанного аргумента. При этом логическая функция от двух переменных сводится к функции от одной переменной.

Проведенный анализ позволяет при построении на основе АЗУ устройства обработки информации выбрать минимально необходимый набор микроопераций.

Таким образом, в каждой строке запоминающей матрицы АЗУ можно производить вычисления логических и арифметических выражений любой сложности, если обеспечена суперпозиция базисных функций. Последнее достига-

ется записью в матрицу памяти содержимого регистра результата ассоциативного опроса. При этом обработка информации осуществляется чередованием операций опроса матрицы памяти и записи в память.

Представляет интерес рассмотреть булевы функции реализуемые при обращении к АЗУ не только для опроса, но также и для записи в память. С логической точки зрения можно считать, что для выбора строк при записи в память производится ассоциативный опрос регистра результата — индикатора  $U$ . Нетрудно видеть, что роль этого регистра может выполнять любой столбец матрицы памяти. Поэтому обе операции — «опрос матрицы» и «запись в память» — можно считать двумя фазами одной операции опроса — записи.

Пусть мы опрашиваем  $j_1$ -й столбец матрицы  $X$  с записью результата опроса в  $j_2$ -й столбец. Это соответствует последовательно-поразрядному обращению к ассоциативной памяти. Можно показать, что в зависимости от значений кодов опроса и записи между одноименными элементами векторов  $X_{j_1}$  и  $X_{j_2}$  могут выполняться четыре разные логические функции: дизъюнкция, функция запрета, импликация и конъюнкция. Этот результат справедлив для обеих разновидностей ассоциативной памяти. Полученные зависимости определяют микрооперации, выполняемые при ассоциативном обращении к памяти. Ими можно пользоваться при составлении микропрограмм групповой обработки информации в ассоциативной памяти.

**Представление алгоритмов.** В общем случае опросу или записи может одновременно подвергаться некоторая совокупность разрядов, что соответствует параллельно-поразрядному обращению к памяти. Простейшим примером такой операции является запись в числовые линейки одинаковой информации, в частности очистка всей памяти или ее части. Для описания групповой обработки информации удобно задавать алгоритмы в виде граф-схем Л. А. Калужнина. Граф-схема Калужнина представляет алгоритм как взаимосвязанную совокупность распознавателей, определяющих наличие или отсутствия у слова определенного свойства, и преобразователей, выполняющих те или иные изменения в слове. В рассматриваемом случае распознавателями  $\Phi$  являются операторы опроса, а преобразователями  $A$  — операторы записи; они следуют друг за другом, образуя граф-схему в виде цепочки, как показано на рис. 2.

Такая граф-схема алгоритма  $\Gamma$  может быть задана в виде набора признаков опроса и кодов записи

$$\Gamma = (\Phi^1, A^1), (\Phi^2, A^2), \dots, (\Phi^v, A^v), \dots, \quad (1)$$

где  $\Phi^v = (\varphi_1^v, \varphi_2^v, \dots, \varphi_r^v)$ ,  $A^v = (\alpha_1^v, \alpha_2^v, \dots, \alpha_r^v)$ ;

$r$  — количество разрядов в преобразуемом слове;  $\{\varphi^v\}$  и  $\{\alpha^v\}$  — наборы троичных переменных, принимающих значения 0, 1 и  $M$ . Эти значения задаются с помощью двух регистров аргумента и маски (см. рис. 1) и соответствуют элементарным операциям в памяти (в случае опроса: 0 — «опрос по нулю», 1 — «опрос по единице»,  $M$  — «запрещение опроса»; в случае записи 0 — «запись нуля», 1 — «запись единицы»,  $M$  — «запрещение записи»).

При реализации алгоритма все слова, представляемые в виде совокупности двоичных переменных  $(x_1, x_2, \dots, x_r)$ , одновременно поступают на вход граф-схемы, и каждое из них проходит ее по своему пути, определяемому значениями булевых функций  $f^v$ , вырабатываемых распознавателями  $\Phi^v$  в виде элементарных конъюнкций:

$$f = \bigwedge_{j=1}^r x_j^{\varphi_j}, \text{ где } x_j^{\varphi_j} = \begin{cases} x_j & \text{при } \varphi_j = 1 \\ \bar{x}_j & \text{при } \varphi_j = 0 \\ 1 & \text{при } \varphi_j = M \end{cases} \quad (2)$$

(С целью упрощения записи все переменные приводятся без индекса  $v$ , указывающего на их отношение к соответствующему звену граф-схемы.) Для ассоциативной памяти значение  $f = 1$  означает наличие запрашиваемого признака, а значение  $f = 0$  — его отсутствие.

Слова, для которых  $f^v = 0$ , передаются на следующий распознаватель  $\Phi^{v+1}$ , а слова, для которых  $f^v = 1$ , — на преобразователь  $A^v$ , который в обычной ассоциативной памяти ставит в соответствие входной совокупности  $\{x_i^v\}$  новую совокупность  $\{x_i^{v+1}\}$  по правилам

$$x_j^{v+1} = \begin{cases} x_j^v \vee f^v & \text{при } \alpha_j = 1 \\ x_j^v \wedge \bar{f}^v & \text{при } \alpha_j = 0 \\ x_j^v & \text{при } \alpha_j = M \end{cases} \quad (3)$$

При достаточном количестве звеньев в граф-схеме можно реализовать любую булеву функцию от  $r - 1$  перемен-

ного, для чего, например, представив эту функцию в дизъюнктивной нормальной форме, можно каждой ее элементарной конъюнкции сопоставить распознаватель, в котором для каждого разряда  $j (1 \leq j \leq r-1)$  наличию переменной  $x_j$  соответствует 1,  $\bar{x}_j$  — 0, а отсутствию переменной —  $M$ . Действие преобразователей должно состоять в записи единицы в оставшийся свободным  $r$ -й разряд, в котором в результате логического суммирования и будет получено значение функции. Однако возможности преобразований с помощью ассоциативной памяти являются гораздо более широкими, поскольку они позволяют реализовать вычисление системы булевых функций, т. е. осуществить отображение множества полноразрядных слов, в себя.

В АЗУ параллельным образом можно выполнять арифметические и логические операции над словами; если в одной числовой линейке располагать по нескольку слов, то также параллельным образом можно осуществлять операции между массивами.

Для иллюстрации рассмотрим простую, но весьма характерную задачу: будем прибавлять одно и то же слагаемое к числам, находящимся в памяти. Представим все числа в форме с фиксированной запятой в дополнительном коде, тогда операция сложения будет выполняться с учетом знака, а результат будем записывать в АЗУ на место исходных чисел.

Содержимое разрядов чисел в памяти обозначим через  $x_j$  ( $j = 1, 2, \dots, n-1$ ),  $n$ -й разряд выделим для записи переноса. Для обозначения соответствующих разрядов слагаемого используем переменную  $b_j$  ( $j = 1, 2, \dots, n-1$ ). Тогда алгоритм поразрядного сложения  $\{x_j\}$  и  $\{b_j\}$  задается в виде следующей таблицы ( $\tilde{x}_j$ ,  $x_n$  и  $\tilde{x}_n$  — соответственно значения  $j$ -го разряда суммы, переноса в  $j$ -й разряд и переноса из  $j$ -го разряда):

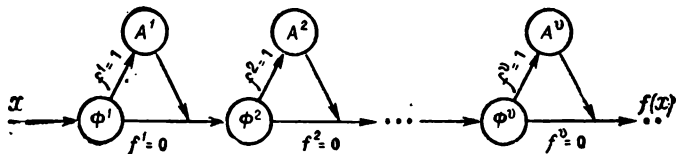


Таблица 1

Номер строки	Ф			А	
	$x_j$	$x_n$	$b_j$	$\tilde{x}_j$	$\tilde{x}_n$
0	0	0	0	0	0
1	0	0	1	1	0
2	0	1	0	1	0
3	0	1	1	0	1
4	1	0	0	1	0
5	1	0	1	0	1
6	1	1	0	0	1
7	1	1	1	1	1

Первые три столбца таблицы характеризуют распознаватели, последние два — преобразователи алгоритма. Поскольку в строках 0, 3, 4, 7  $x_j = \tilde{x}_j$  и  $x_n = \tilde{x}_n$ , то соответствующие действия для вычисления каждого разряда суммы можно представить с помощью двух звеньев графсхемы, определяемых значением  $j$ -го разряда слагаемого. В соответствии с табл. 1 коды опроса и записи для этого участка алгоритма принимают вид, описываемый табл. 2.

Перед выполнением операции в разряд переноса  $n$  записывается 0, далее вычислительный процесс начинается с младшего разряда и развивается последовательно по разрядам, что приводит к параллельному вычислению в числовых линейках алгебраической суммы первоначально содержащихся чисел с заданным слагаемым.

*Однородная вычислительная среда на основе ассоциативных преобразований.* Методы преобразования слов в ассоциативной памяти можно перенести на структурную организацию некоторой однородной вычислительной среды;

Таблица 2

Номер строки таблицы	Ф	А
2 или 5	$n, n-1, \dots, j, \dots, 1$ ( $\bar{b}_j M \dots b_j \dots M$ )	$n, n-1, \dots, j, \dots, 1$ ( $b_j M \dots \bar{b}_j \dots M$ )
6 или 1	( $\bar{b}_j M \dots \bar{b}_j \dots M$ )	( $MM \dots b_j \dots M$ )

такого типа устройства предназначаются для реализации универсальных алгоритмов за счет функциональной перестройки своих элементов. Предлагаемый вариант однородной вычислительной среды строится в соответствии со структурой рассмотренных граф-схем алгоритмов. Оказывается, что получающаяся при этом вычислительная среда не превосходит по сложности другие аналогичные устройства, однако она имеет преимущество из-за простоты настройки благодаря программной совместимости с ассоциативным процессором. На рис. 3 изображена схема ячейки вычислительной среды, соответствующая одному звену граф-схемы алгоритма. Элементы  $\phi$ , реализующие на своих выходах промежуточные значения переменных  $\{x^\phi\}$ , и схема совпадения, объединяющая эти выходы, соответствуют распознавателю  $\Phi$ , элементы  $\alpha$ , управляемые сигналами входных переменных  $\{x\}$  и выходом схемы совпадения  $f = \bigwedge_{i=1}^r x_i^{\phi_i}$ , соответствуют преобразователю  $A$ , ко-

торый вырабатывает выходные сигналы  $\{\tilde{x}\}$ , подаваемые на вход следующей ячейки. Элементы  $\phi$  и  $\alpha$  настраиваются с помощью трюичных кодов, описывающих выполняемый алгоритм  $\Gamma$  (1), а логика их работы определяется формулами (2) и (3). Таким образом, устанавливается соответствие между алгоритмами для ассоциативной памяти и вычислительными средами.

Данная однородная структура представляет собой перестраиваемую комбинационную схему, которая как бы разворачивает в пространстве параллельные преобразования, выполняемые в ассоциативной памяти последовательно во времени. Она удобно стыкуется с ассоциативным процессором в составе одной вычислительной системы — программы для их работы, задаваемые в виде граф-схем  $\Gamma$  (1), одинаковы.

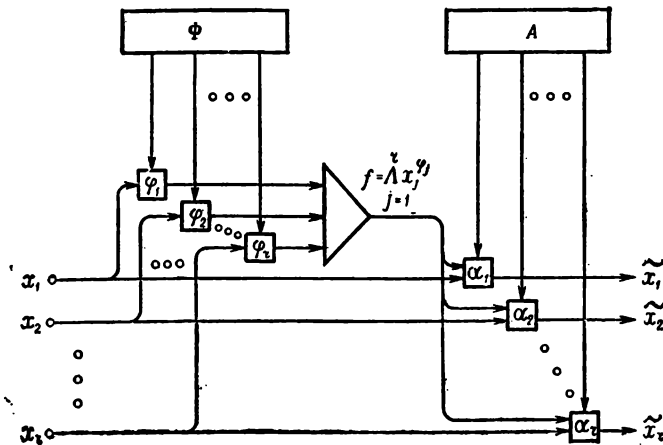
Рассмотрим метод организации ассоциативных преобразований на примере счетчика единиц. Прибавление единицы можно осуществить с помощью общего правила сложения числа с массивом, описанного выше, однако в данном конкретном случае это можно сделать проще. Чтобы прибавить единицу, надо значение самого младшего из разрядов, в котором содержится 0, изменить на 1, а значение всех более младших разрядов, в которых содержится 1, изменить на 0. Для случая трехразрядных чисел такая процедура реализуется следующим алгоритмом, представленным табл. 3.

Таблица 3

Номер звена	Ф				А			
	$\varphi_4$	$\varphi_3$	$\varphi_2$	$\varphi_1$	$\alpha_4$	$\alpha_3$	$\alpha_2$	$\alpha_1$
1	(М	М	М	М)	(0	М	М	М)
2	(М	1	1	1)	(1	0	0	0)
3	(М	0	1	1)	(1	1	0	0)
4	(М	М	0	1)	(1	М	1	0)
5	(0	М	М	0)	(М	М	М	1)

Четвертый разряд играет вспомогательную роль и служит для того, чтобы слова, преобразованные 2, 3 и 4 звеньями и получившие в первом разряде 0, могли бы быть отличены от исходных слов, имеющих в первом разряде 0 и подлежащих преобразованию 5-м звеном. Кодовая комбинация 111 переходит в 000, так что счет единиц производится по модулю 8.

На рис. 4 представлена схема, которая получается как результат настройки в соответствии с указанным только что алгоритмом из универсальной вычислительной среды, построенной на основе соединений элементов, изображенных на рис. 3. Первое звено алгоритма, записывающее 0 в 4-й разряд всех чисел, опущено и заменено константой 0 на входе  $x_4$ .



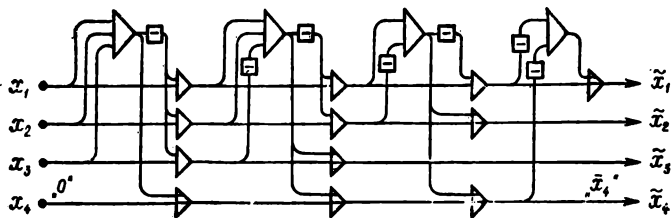
Однородное вычислительное устройство на основе ассоциативных преобразований является весьма экономным по количеству элементов и по информационной избыточности. При параллельной организации алгоритма в АЗУ необходимо задействовать все операторы, входящие в состав граф-схемы этого алгоритма. Поэтому в отличие от последовательной обработки слов, когда наиболее эффективными являются алгоритмы с минимальной средней длиной от входа до выхода граф-схемы, в случае групповой обработки слов в ассоциативной памяти более эффективными оказываются алгоритмы с минимальным общим количеством звеньев. Это же относится, разумеется, и к количеству ячеек в однородном функциональном устройстве, соответствующем данному алгоритму вычисления.

Рассмотрение ряда конкретных алгоритмов показывает, что для простых операций, таких, как сдвиг, логические преобразования, различные варианты алгебраического сложения, количество звеньев в граф-схеме пропорционально  $n$ ; для более сложных операций, как, например, для умножения, деления, преобразования кодов из десятичной системы в двоичную и обратно, количество звеньев пропорционально  $n^2$ .

Функциональные возможности ассоциативной памяти могут быть расширены путем использования трехзначных запоминающих элементов.

*Сравнение ассоциативной памяти на двузначных и трехзначных элементах.* В рассмотренной структуре АЗУ (см. рис. 1) накопитель состоит из двузначных элементов, а коды, определяющие обращения к памяти для опроса и записи, являются троичными.

Было показано, что при выполнении групповых операций в АЗУ на двузначных элементах в каждую строку матрицы памяти помещается некоторый набор исходных двоичных переменных, представляющий, например, два опе-



Р и с. 4



ранда. Обработка информации заключается в последовательном обращении к памяти поочередно для опроса и записи. Можно считать, что каждая пара таких обращений реализует некоторую микрооперацию, которая задается в регистрах аргумента и маски (или в одном входном регистре, состоящем из трехзначных элементов) и выполняется одновременно над содержимым всех строк памяти. Результаты вычислений формируются непосредственно в строках АЗУ, так что содержимое накопителя в процессе вычислений изменяется.

Представим себе обратную картину: в строки памяти помещаются троичные коды, а на вход устройства последовательно подаются двоичные данные. Мы приходим к концепции АЗУ на трехзначных элементах.

При использовании АЗУ на трехзначных элементах в строках матрицы памяти размещаются коды ассоциативных микроопераций, причем для каждой микрооперации выделяется отдельная строка. Для этого каждая ячейка памяти, кроме основных состояний «0» и «1», должна иметь еще третье состояние «М», в котором вырабатывается сигнал соответствия при сравнении как с нулем, так и с единицей. На вход АЗУ поочередно подаются входные данные (операнды). При параллельном опросе накопителя выделяются микрооперации, предназначенные для обработки только тех данных, которые содержатся во входном регистре; информация, связанная с отмеченными строками памяти, определяет выходной результат. Очевидно, за одно обращение к памяти можно выполнить произвольный набор булевых функций; при этом содержимое АЗУ не изменяется; запись в память осуществляется только при настройке АЗУ на выполнение новой операции.

Таким образом, с помощью АЗУ на двузначных элементах вычислительный процесс может выполняться параллельно по операндам и последовательно по микрооперациям, а с помощью АЗУ на трехзначных элементах — последовательно по операндам и параллельно по микрооперациям. Следует заметить, что при перенесении обработки информации из одного типа АЗУ в другой для реализации одной и той же операции в общем случае может потребоваться другой набор микроопераций.

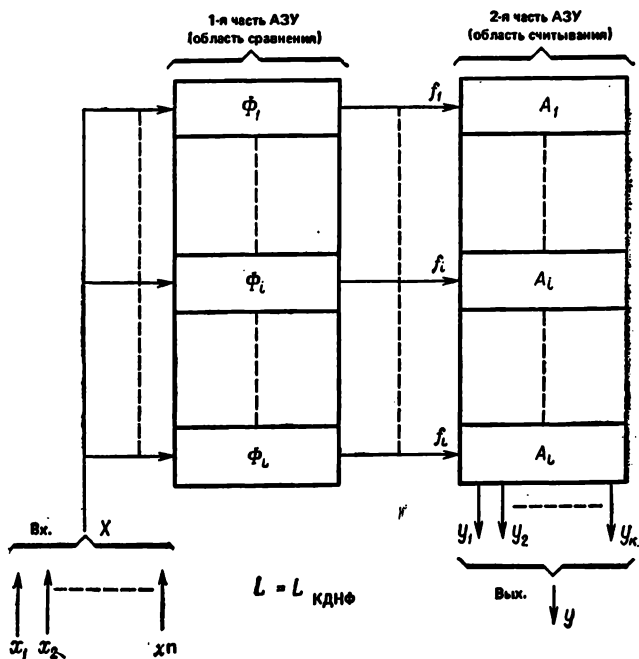
Очевидно, функции АЗУ на трехзначных элементах могут быть выполнены в АЗУ на двузначных элементах с удвоенным количеством разрядов. Можно, например, представить три состояния следующим образом: «0» — 01,

«1» — 10, третье состояние — 11 и производить сравнение по критериям  $M1$  для «0» и  $1M$  для «1». Этот способ очень удобен, хотя и требует повышенного расхода памяти.

Для АЗУ на трехзначных элементах информация о выполняемом преобразовании записывается в память устройства, и сложность преобразования целесообразно оценивать количеством требуемых строк памяти. Можно выделить две структуры АЗУ на трехзначных элементах: последовательную и параллельную.

Последовательная структура представляет собой аппаратную реализацию преобразований, определяемых кодами опроса — записи. Такую структуру имеет рассмотренное выше вычислительное устройство на основе ассоциативных преобразований.

В параллельной структуре АЗУ на трехзначных элементах можно выделить две части: первая часть служит для хранения распознавателей, вторая — преобразователей, которые в данном случае реализуют функцию типа



Р и с. 5

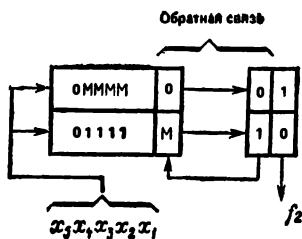
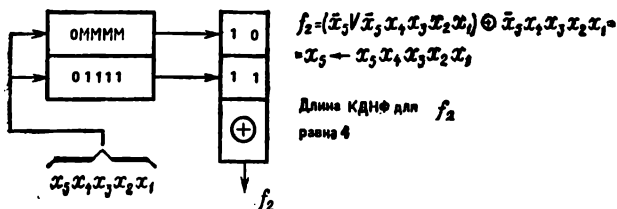
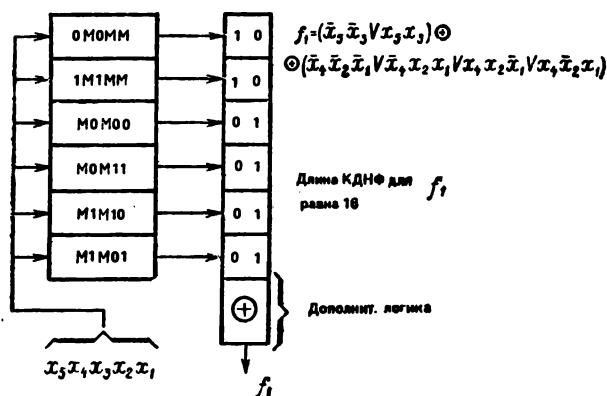
константы. В таком АЗУ каждой строке, выделяемой при опросе, сопоставляется частичный результат, задаваемый двоичным кодом. Результат обращения к АЗУ образуется как дизъюнкция частичных результатов, выделенных при опросе (рис. 5).

Для реализации одной булевой функции достаточно иметь один разряд — столбец во второй части АЗУ. В такой структуре наименьшее количество строк памяти, требуемое для вычисления одной булевой функции, равно длине кратчайшей ДНФ, представляющей эту функцию. Во многих случаях расход памяти удастся уменьшить, если для каждой реализуемой функции выделить более одного столбца второй части АЗУ; при этом заданная функция должна быть представлена в виде некоторой функции от подфункций, вычисляемых в каждом столбце. Примером могут служить такие формы представления функций, в которых элементарные конъюнкции связываются операциями не только дизъюнкции, но и сложения по модулю 2 или запрета. Эти формы по числу элементарных конъюнкций могут оказаться короче, чем КДНФ. Например, известно, что среди булевых функций от  $n$  переменных имеются две функции, для каждой из которых КДНФ совпадает с совершенной ДНФ и длина КДНФ равна  $2^{n-1}$ . На рис. 6, а дана реализация одной из таких функций (для  $n=5$ ); количество требуемых строк памяти равно 6 (вместо 16). На рис. 6, б показана реализация другой функции, в представлении которой целесообразно использовать операцию запрета. В такой же форме можно представлять функцию, если ввести обратную связь, как показано на рис. 6, в.

Особенности АЗУ на трехзначных элементах удобно использовать при реализации вычислительных операций структурными методами. В третьем разделе даны примеры выполнения булевых функций в настраиваемой ассоциативной матрице на трехзначных элементах. Ниже рассмотрен способ быстрого поиска чисел, ближайших к заданному, и описаны некоторые его применения.

*Нахождение чисел, ближайших к заданному.* Одной из часто встречающихся операций при решении задач обработки информации является поиск в массиве числа, которое находится к заданному числу в отношении: ближайшее меньшее или ближайшее большее. Во многих задачах, связанных с сортировкой потоков данных в ограниченное время, эта операция должна выполняться достаточно быст-

ро. Важным примером такой задачи может служить организация ассоциативной виртуальной памяти, которая рассмотрена в последнем разделе.



Р и с. 6

Количество обращений к ЗУ, требуемое для осуществления операций указанного типа на основе применения традиционных методов, пропорционально логарифму количества слов в массиве в случае использования адресной памяти или пропорционально разрядности чисел в случае использования ассоциативной памяти.

Применение АЗУ на трехзначных элементах позволяет реализовать рассматриваемые операции за одно обращение к устройству. Пусть  $n$  — разрядность, а  $k$  — количество исходных двоичных чисел, среди которых должен осуществляться поиск. В соответствии с набором этих чисел множество всевозможных  $n$ -разрядных двоичных кодов разбивается на непересекающиеся классы, для каждого из которых любой элемент в качестве своего ближайшего меньшего или большего имеет одно и то же число. Если бы мы имели ЗУ достаточно большой емкости, в котором каждому из  $2^n$  двоичных кодов соответствовала бы отдельная ячейка памяти, то требуемую процедуру поиска можно было бы выполнить за одно обращение к ЗУ.

Очевидно, что непосредственная реализация этой идеи потребовала бы чрезмерного расхода памяти. В случае использования АЗУ на трехзначных элементах расход этот удастся снизить до допустимых пределов благодаря тому, что двоичные числа каждого класса могут быть представлены сравнительно небольшим количеством кодовых комбинаций, состоящих из символов «0», «1», и «М».

Кодовая комбинация, содержащая  $i$  символов  $M$ , определяет  $i$ -мерный интервал на булевом пространстве. Если рассматривать отрезок числовой оси, образуемый соответствующими двоичными числами каждого класса, как область единичных значений некоторой булевой функции, то можно сказать, что задача представления данных отрезков числовой оси при помощи троичных кодовых комбинаций сводится к задаче минимизации системы таких отрезковых булевых функций, области единичных значений которых не пересекаются.

Рассмотрим следующий иллюстративный пример (рис. 7). Пусть исходные двоичные числа, среди которых должен проводиться поиск ближайшего меньшего, суть: 0000, 0100 и 1001 (отмечены крестиками). Тогда множество всевозможных двоичных кодов разбивается на три непересекающихся класса. Количество интервалов кратчайшего покрытия каждого класса может быть существенно меньше количества элементов в соответствующем классе; при этом само

покрытие, как, например, для последнего класса может быть неоднозначным.

Возникают вопросы: каким образом найти интервалы покрытия и какова величина памяти, требуемой для их размещения?

Оказывается, что для минимизации отрезковой булевой функции (или для нахождения покрытия соответствующего отрезка интервалами) имеется достаточно простой алгоритм. В этом алгоритме используются  $i$ -мерные интервалы ( $i = 0, 1, \dots, n$ ), все  $2^i$  элементов каждого из которых занимают только один непрерывный участок числовой оси; трюичные кодовые комбинации, представляющие такие интервалы, имеют в самых младших разрядах  $i$  символов  $M$ . Алгоритм заключается в том, что последовательно находятся непересекающиеся интервалы указанного типа максимально возможной размерности до тех пор, пока они не покроют весь отрезок.

Интересно отметить, что формирование совокупности кратчайших покрытий для всех отрезков числовой оси

Двоичные коды	Ближайшее меньшее (равное)	Интервалы покрытия		
0 0 0 0	0 0 0 0			
x 0 0 0 1	0 0 0 0	0 0 M M		
0 0 1 0	0 0 0 0			
0 0 1 1	0 0 0 0			
0 1 0 0	0 1 0 0			
0 1 0 1	0 1 0 0			
x 0 1 1 0	0 1 0 0	0 1 M M		
0 1 1 1	0 1 0 0	1 0 0 0		
1 0 0 0	0 1 0 0			
1 0 0 1	1 0 0 1			
1 0 1 0	1 0 0 1			
1 0 1 1	1 0 0 1	1 M 0 1	1 M M 1	1 0 0 1
x 1 1 0 0	1 0 0 1	1 M 1 M или	1 M 1 M или	1 0 1 M
1 1 0 1	1 0 0 1	1 1 0 0	1 1 M M	1 1 M M
1 1 1 0	1 0 0 1			
1 1 1 1	1 0 0 1			

Рис. 7

является хорошим примером вычислительного процесса, допускающего при реализации его в ассоциативной памяти эффективное распараллеливание по операндам.

Для размещения одной трюичной кодовой комбинации требуется одна числовая линейка АЗУ. Поэтому для оценки расхода памяти необходимо знать общее количество интервалов  $Q$ , покрывающих все  $k$  отрезков, на которые разбивается множество всевозможных  $n$ -разрядных двоичных чисел.

Проведенный анализ показал, что среднее значение величины  $Q$  хорошо аппроксимируется следующим выражением:

$$Q \approx (n - \log k - 1)(k - 1) + 1. \quad (4)$$

Следует заметить, что максимальное значение количества интервалов, соответствующее «наихудшему» расположению отрезков на числовой оси, незначительно превосходит среднюю величину.

Таким образом, избыточность памяти для требуемого представления массива чисел может быть вполне приемлемой. В ряде случаев указанный избыточный расход памяти удастся уменьшить и даже вообще избежать, если длины отрезков равны целым степеням двойки.

*Статистическое моделирование.* Использование АЗУ оказывается весьма эффективным при статистическом моделировании, а именно организации случайных блужданий, т. е. переходов с заданными вероятностями на определенном множестве состояний. Это — основа вычислительной процедуры при моделировании марковских цепей. Действия, которые надо выполнить при этом, состоят в следующем: если в текущий момент система находится в некотором состоянии  $i$ , то нужно выбрать состояние  $j$ , в которое система перейдет в следующий момент. Для организации этой процедуры надо решить две задачи: 1) размещение информации, описывающей связи между состояниями системы (в больших системах матрица вероятностей перехода содержит, как правило, значительное число нулевых элементов), и 2) определение состояния, куда должна попасть система в соответствии с совокупностью из  $k$  вероятностей перехода, связанных с состоянием  $i$ .

При использовании ассоциативной памяти решение первой задачи является очень удобным: индексы  $i$  можно взять в качестве ассоциативных признаков, так что при обращении

нии по такому признаку сразу будет выделен набор  $k$  состояний, в которые может перейти система.

Вторая задача может решаться за одно обращение к ассоциативному ЗУ, если использовать рассмотренный выше способ быстрого поиска чисел, ближайших к заданному. Для этого надо для каждого состояния  $i$  предварительно произвести разбиение множества всевозможных  $n$ -разрядных двоичных кодов на непересекающиеся подмножества, количество которых равно числу  $k$  состояний, а мощности пропорциональны значениям вероятностей перехода; тогда определение принадлежности случайного входного числа некоторому подмножеству эквивалентно выбору соответствующего состояния с заданной вероятностью.

Указанные непересекающиеся подмножества представляются троичными кодовыми комбинациями. В данном случае эти подмножества не обязательно должны быть непрерывными участками (отрезками) числовой оси. Благодаря этому, как показали оценки, количество интервалов для их представления может составлять в среднем до 50% от значения, получаемого согласно выражению (4).

Если величины вероятностей разлагаются в сумму не большого числа отрицательных степеней двойки, то необходимая избыточность памяти резко снижается. Существует класс задач, использующих модели конечных цепей Маркова, для решения которых важно лишь сохранить структуру переходов между состояниями, а сами значения переходных вероятностей, вообще говоря, несущественны. К таким задачам можно отнести ряд задач для систем с отрицательными матрицами; примером могут служить некоторые задачи математической экономики. Для таких задач применение ассоциативной памяти с трехзначными запоминающими элементами получается весьма эффективным: для состояний, между которыми имеются связи, задаются некоторые переходные вероятности с использованием достаточно малой избыточности числовых линеек, при этом благодаря возможности перехода из одного состояния в другое за один такт обращения к памяти может быть обеспечен выигрыш в скорости решения в несколько десятков раз(1)





### 3

## ВОПРОСЫ ПОСТРОЕНИЯ

К запоминающим и управляющим элементам ассоциативной памяти предъявляется ряд специфических требований. В отличие от элемента обычного адресного ЗУ элемент АЗУ должен обладать способностью проводить операцию сравнения, для чего нужно выдавать сигнал соответствия или несоответствия хранящегося и опрашиваемого кодов.

Для формирования переменных ассоциативных признаков необходимо иметь возможность маскировать определенные разряды, т. е. не производить в них операцию сравнения, при этом отсутствие сравнения в данном элементе должно восприниматься так же, как случай совпадения.

Ассоциативные запоминающие устройства могут быть реализованы с помощью различных физических принципов. Применение же тех или иных элементов в значительной мере определяет структуру всего АЗУ в целом, его технические и экономические показатели, успешность реализации.

Прототипом электронных устройств ассоциативной памяти являются средства механизации поисковых работ с помощью перфокарт, например карт с краевой перфорацией. Системы таких перфокарт используются для постоянного хранения и ассоциативного поиска информации. Принцип выборки данных в них такой же, как и в электронных АЗУ.

На каждую перфокарту записывается информация о некотором документе. На краях таких карт имеются отверстия, пробитые в один или несколько рядов. Признаки информации наносятся с помощью вырезов к этим отверстиям. Распределение отверстий под ассоциативные признаки, а также способ кодировки букв и цифр с помощью вырезов устанавливаются предварительно и в дальнейшем не меняются.

Для поиска информации, обладающей заданными признаками (рис. 8), в соответствующие отверстия перфокарт 1 вводят спицы 2. Перфокарты, у которых вырезы прихо-

дятся на места спиц, выпадают из пачки. Очевидно, эти карты содержат искомую информацию.

Каждая перфокарта соответствует одной строке матрицы ассоциативной памяти, отверстия на картах сопоставляются разрядам (см. рис. 1). Введение спиц в некоторые отверстия означает проведение ассоциативного опроса в соответствующих разрядах; роль маскированных разрядов играют отверстия, в которые спицы не вводятся. Выпадение перфокарт из пачки указывает на совпадение признаков опроса с признаками хранимой информации.

Просмотр выпавших перфокарт, соответствующий процедуре разделения многозначного ответа, производится извне человеком.

Заметим, что для рассматриваемой системы не имеется выборочной функции, позволяющей указать представителя для подмножества выделенных перфокарт, т. е. не выполняются условия аксиомы выбора. Действительно, в рассматриваемой системе на множестве всех перфокарт не задано отношение порядка.

В качестве элементов АЗУ были изучены практически все типы магнитных запоминающих элементов. Несомненные их достоинства, определяющие широкое распространение в адресных ЗУ, в случае ассоциативных не могут обеспечить достаточно приемлемые технико-экономические показатели даже для устройств сравнительно небольшой емкости. Недостатки АЗУ на магнитных элементах обуславливаются в основном громоздкостью и высокой стоимостью устройств управления. Определенные перспективы создания АЗУ большой емкости открываются при

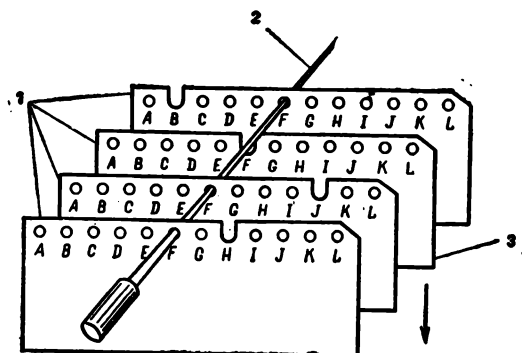


Рис. 8

использовании устройств на основе подвижных магнитных доменов; однако в этом случае не обеспечивается необходимое быстроедействие, дающее возможность применять такие АЗУ в качестве оперативной памяти ЭВМ.

Интересные возможности построения АЗУ открывает применение оптических явлений, в особенности голографии. Привлекательность голографического принципа связана с тем, что он не требует создания микрогеометрических конструкций в запоминающей среде и при большой плотности записи информации обеспечивает высокую надежность хранения — благодаря тому, что фиксируемая информация не локализуется в пространстве. Главный недостаток голографических ЗУ состоит в том, что в них нельзя производить выборочное стирание информации; большие затруднения возникают и при построении запоминающей среды с оперативной фиксацией информации. Основное применение голографические ассоциативные ЗУ найдут себе, по-видимому, при создании больших постоянных информационных архивов либо в составе крупных вычислительных систем, либо в качестве самостоятельных поисковых устройств для опознавания образов по различным фрагментам, в частности для дешифровки сложных изображений.

В настоящее время для построения быстродействующих устройств ассоциативной памяти начинают широко использоваться полупроводниковые интегральные схемы. В качестве компонентов интегральных схем используются как биполярные, так и МДП-транзисторы. Схемы на МДП-транзисторах допускают больший масштаб интеграции, менее чувствительны к помехам, более технологичны, а также обладают рядом других достоинств, хотя несколько уступают биполярным интегральным схемам в быстродействии. Основным препятствием к созданию больших полупроводниковых устройств ассоциативного типа являются эффекты, связанные с чрезмерным рассеиванием мощности; это определяется спецификой функционирования АЗУ, которая требует одновременной работы всех запоминающих элементов устройства. По-видимому, на основе полупроводниковых интегральных схем целесообразно строить ассоциативную память емкостью примерно до  $10^6$  бит.

Существующее представление о перспективности сверхпроводниковых элементов — криотронов для построения АЗУ связано с тем, что они удовлетворяют всему комплек-

су физико-технологических требований, предъявляемых к элементам, на которых должны строиться большие системы дискретного действия. Достоинства криотронов определяются конструктивной простотой, высоким уровнем эксплуатационной надежности, отсутствием рассеяния мощности при хранении информации и весьма малой величиной энергии, выделяемой в процессе переключения; с помощью криотронов без каких-либо дополнительных элементов можно строить схемы для выполнения логических и запоминающих функций.

Криотрон представляет собой сверхпроводниковый элемент, содержащий две электрически изолированные цепи — вентильную и управляющую; ток в управляющей цепи своим магнитным полем может переключать состояние вентильной цепи из сверхпроводящего в нормальное. Наибольший интерес вызывают криотроны, использующие так называемый эффект Джозефсона. Преимущества этих криотронов перед другими определяют следующие факторы: возможность значительной микроминиатюризации, высокое быстродействие, слабая зависимость параметров от изменений рабочей температуры.

Внедрение сверхпроводниковых устройств осложняется тем, что в отличие от внедрения устройств на различных других элементах оно не может осуществляться постепенным наращиванием степени интеграции; в то же время по целому ряду технических и экономических соображений использование криотронных схем с невысоким уровнем интеграции практически нецелесообразно. Можно надеяться, что имеющиеся трудности в создании криотронных больших интегральных схем по мере развития технологии будут преодолены и многолетняя мечта разработчиков вычислительных машин о быстродействующем ЗУ большой емкости с ассоциативным доступом будет, таким образом, осуществлена.

*АЗУ на криотронах.* Рассмотрим принципиальную схему АЗУ на криотронных элементах и проиллюстрируем этапы прохождения основной операции — обращение по признаку.

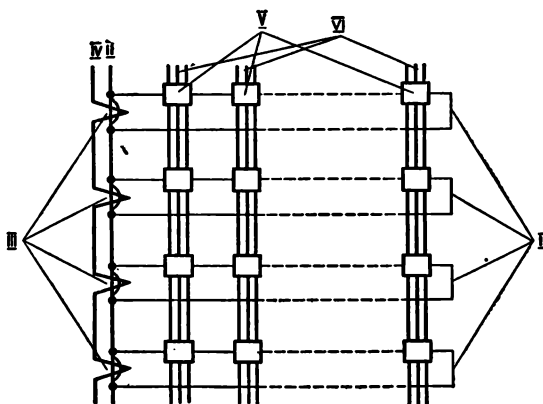
В криотронном АЗУ (рис. 9) числовые линейки I присоединяются к общей шине II, проходящей через все устройство. Эта шина содержит криотроны III, которые управляют включением тока в контуры числовых линеек. Управляющие цепи указанных криотронов соединены с дешифрирующей системой и позволяют одновременно

переключать ток во всех числовых линейках; на рисунке эти управляющие цепи условно соединены последовательно и объединены в шину IV.

На числовых линейках располагаются ассоциативные запоминающие ячейки V, которые при проведении операции сравнения вырабатывают сигнал в следующем виде: если сравниваемый код совпадает с хранящимся, то путь тока через ячейку остается сверхпроводящим, если нет, то путь тока через ячейку становится резистивным. (Отсутствие операции сравнения сохраняет сверхпроводящим путь через ячейку и, следовательно, воспринимается как совпадение.)

Через ячейки, расположенные в одноименных разрядах каждой числовой линейки, последовательно проходят управляющие разрядные шины VI, с помощью которых можно производить операции записи, сравнения и считывания. В отличие от других схем АЗУ в описываемой удалось обеспечить выполнение указанных операций, используя только один контур в составе каждой числовой линейки.

Принцип осуществления операции поиска заключается в следующем. Перед операцией поиска ток по шине II проходит так, как показано на рис. 10, а. Для начала операции подается импульс тока в шину IV, который и перекрывает сверхпроводящий путь через криотроны III, в результате чего ток будет проходить через контуры числовых линеек I (рис. 10, б). Если во время проведения



Р и с. 9

операции сравнения в какой-либо запоминающей ячейке произошло несовпадение хранящегося кода с кодом опроса, подаваемого по шинам VI, то в этой ячейке появится сопротивление, и сверхпроводящий путь для тока через нее будет закрыт. Следовательно, ток, проходящий через числовую линейку, содержащую хотя бы одну ячейку, в которой произошло несовпадение, переключится обратно в шунтирующий данную числовую линейку криотрон III, вентиль которого после окончания управляющего импульса в шине IV будет снова находиться в сверхпроводящем состоянии. Таким образом, ток сохранится в тех и только тех числовых линейках, у которых во всех запоминающих ячейках произойдет совпадение хранящегося и искомого кодов.

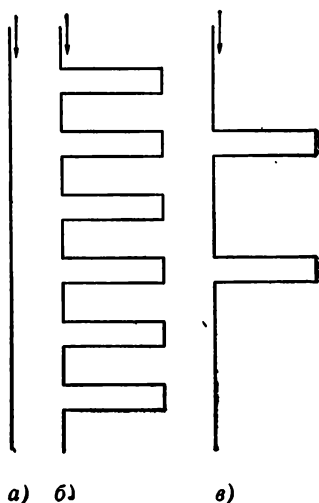


Рис. 10

Результат операции поиска фиксируется в виде пути тока, проходящего только через найденные числовые линейки (рис. 10, в). Дальнейшая обработка выделенных слов может состоять в последовательном обращении к ним путём разделения многозначного ответа или в проведении некоторой групповой операции над полученным подмножеством слов.

Сложность криотронного АЗУ в целом в значительной степени определяется схемой отдельной запоминающей ячейки. Известно несколько десятков различных типов таких ячеек. Хотя сложные ячейки, казалось бы, могут расширять круг логических операций в АЗУ, это является далеко не достаточной компенсацией за уменьшение плотности расположения ячеек и снижение скорости выполнения основных операций.

Описанная ниже криотронная ассоциативная запоминающая ячейка является наиболее простой и эффективной. Она позволяет выполнять все необходимые операции при наличии всего 4-х криотронов и одного контура в числовой линейке, при этом обеспечиваются неразруша-

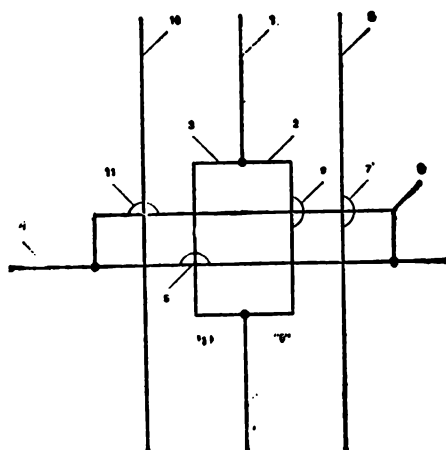


Рис. 11

ющее считывание, селективная запись, подтверждение информации при обращении и широкие допуски на разброс параметров криотронных элементов и управляющих импульсов.

Схема ячейки изображена на рис. 11. Через ячейку проходят три разрядные шины и одна числовая. По разрядной шине 1 идет постоянный ток  $I$ . Если в ячейке хранится

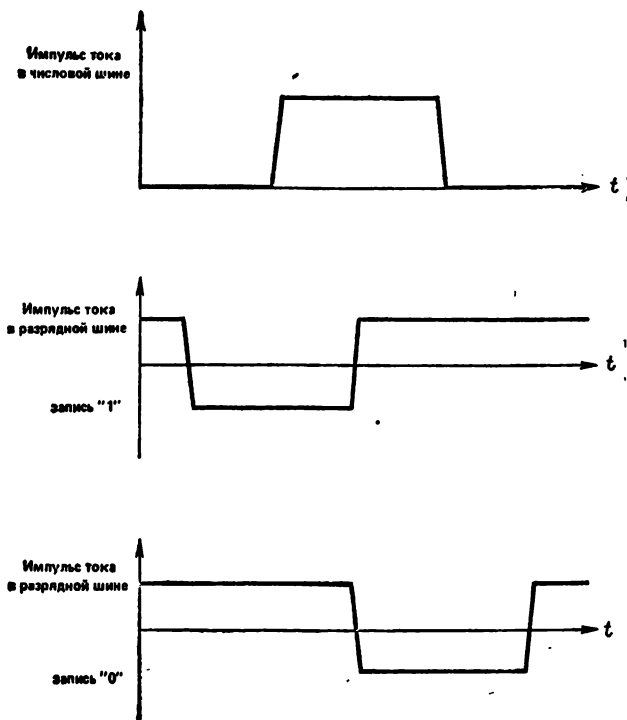
«0», то ток идет по ветви 2; если «1» — по ветви 3. Индуктивности ветвей 2 и 3 примерно одинаковы.

При считывании ток, подаваемый в числовую шину 4, в зависимости от состояния ячейки имеет один из двух путей: путь с малой индуктивностью через вентиль криотрона 5 (в случае «0»), или путь с большей индуктивностью через ветвь 6 (в случае «1»). В последнем случае запирается криотрон 7, и на шине считывания 8 появляется сигнал.

Запись в ячейку осуществляется с помощью криотрона 9 подачей импульсов тока в числовую шину 4 и разрядную шину 1 (рис. 12). Импульс в шине 4 такой же, как и при считывании; импульс в шине 1 имеет амплитуду —  $2I$ , т. е. он меняет полярность тока в шине 1. Кодирование информации при записи осуществляется временным сдвигом разрядного импульса. Изменение полярности тока в разрядной шине 1 при отсутствии тока в числовой шине 4 переводит ток в ячейке из одной ветви в другую; та же операция при наличии тока в шине 4 устанавливает ток в ветви 3 независимо от исходного состояния ячейки. Поэтому если разрядный импульс начинается раньше числового, а заканчивается во время его действия, то в ячейке запишется «1»; если разрядный импульс начинается во время действия числового, а заканчивается после него, то — «0».

Операция сравнения осуществляется следующим образом. По шине 1 подается код сравнения: если производится сравнение по «1», то на время выполнения операции меняется полярность тока  $I$ , если по «0», то не меняется. После этого (см. рис. 11) подается импульс в шину сравнения 10, который запирает криотрон 11. В случае совпадения кода сравнения с кодом, хранящимся в ячейке, ток в ячейке оказывается в ветви 2, при этом имеется сверхпроводящий путь по шине 4 через вентиль криотрона 5; в случае несовпадения криотрон 5 заперт и сверхпроводящий путь числовому току через ячейку закрыт. Когда операция сравнения в ячейке не производится, импульс в шину 10 не подается, и независимо от хранящегося кода имеется сверхпроводящий путь по шине 4.

В АЗУ, построенном на таких ячейках, осуществляет универсальная операция обращения, при которой мож-



Р и с. 12



но производить независимо и одновременно считывание и запись в любых разрядах выбранной числовой линейки. Следует заметить, что операция записи обладает свойством селективности, т. е. при ее выполнении в одной части разрядов числовой линейки состояние других разрядов не затрагивается. Это свойство в сочетании с возможностью одновременной записи информации в некоторое выделенное подмножество слов является необходимым для проведения групповой обработки информации в АЗУ.

Отметим еще одну особенность АЗУ. Свойства запоминающей ячейки таковы, что если в ней хранится «1», но какая-то часть тока идет по ветви 2, то при подаче тока в шину 4 криотрон 9 запретится, и эта часть вытеснится в ветвь 3, т. е. произойдет «подтверждение» состояния «1». Таким образом, на каждом этапе поиска ток, заходя в числовые линейки, подтверждает состояния «1» во всей памяти. При измененной полярности тока в разряде точно так же подтверждаются состояния «0». Эта особенность снижает вероятность сбоев и обеспечивает устойчивость информации во всем АЗУ.

*О структуре БИС на основе ассоциативных принципов.* Использование принципов функционирования АЗУ позволяет разработать настраиваемые структуры больших интегральных схем, с помощью которых можно создавать как оперативные, так и комбинационные устройства ассоциативной памяти. Возможность настройки является основой для широкого применения однотипных БИС.

Во втором разделе была рассмотрена одна из настраиваемых схем такого рода — однородное вычислительное устройство на основе ассоциативных преобразований. Такое устройство позволяет организовать конвейерную обработку информации. Это является необходимым при использовании элементов с высоким быстродействием (например, криотронов на эффекте Джозефсона), поскольку при этом исключаются излишние пересылки информации, приводящие к снижению производительности вычислительных устройств.

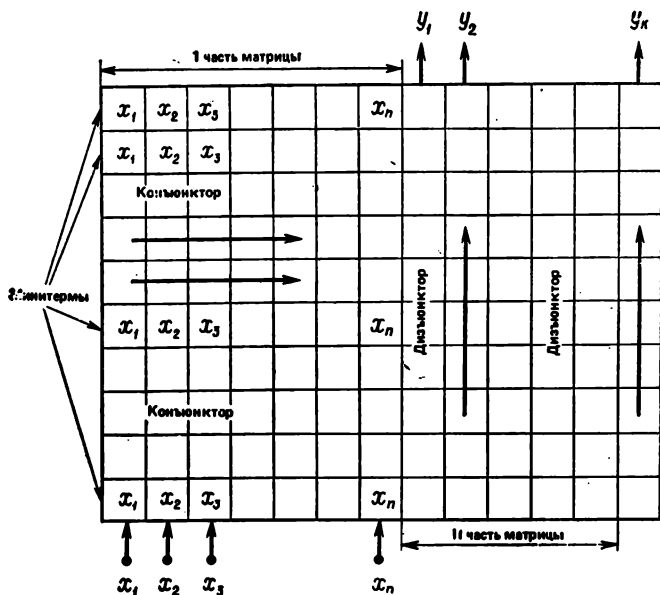
Другая настраиваемая схема представляет собой однородную матрицу, соответствующую параллельной структуре АЗУ на трехзначных элементах.

Ассоциативную матрицу можно условно разбить на две части (рис. 13). Каждая строка первой части является конъюнктом и предназначена для записи одной эле-

ментарной конъюнкции (или минитерма), образованной входными переменными  $x_1, x_2, \dots, x_n$ . Во второй части матрицы записываются значения выходных функций  $y_1, y_2, \dots, y_k$ , определенные на наборах переменных, заданных минитермами. Каждый столбец в этой части матрицы образует дизъюнктор и соответствует одной выходной функции. Число реализуемых выходных функций определяется числом столбцов второй части матрицы.

Первая часть матрицы подвергается ассоциативному опросу кодом, образуемым входным двоичным  $n$ -разрядным набором значений переменных. При этом выделяются все строки матрицы, содержимое которых совпадает со значением входного набора. Во второй части матрицы происходит считывание по вертикали выделенных строк, дизъюнкция которых образует значения выходных функций.

Очевидно, для сокращения расхода памяти желательно иметь кратчайшую систему дизъюнктивных нормальных форм для выходных функций. За один такт обращения к ассоциативной матрице реализуется любая комбинационная функция или система булевых функций, записанных



в матрице, благодаря чему обеспечивается высокая скорость выполнения операций.

Каждый разряд матрицы может работать в составе как конъюнктора, так и дизъюнктора. Количество и расположение столбцов матрицы, выполняющих функции дизъюнктора, не фиксируются и задаются путем предварительной настройки. Таким образом, ассоциативная матрица имеет регулярную структуру, что позволяет удовлетворить одному из важных требований технологии изготовления интегральных схем.

Рассмотрим реализацию системы из двух булевых функций в ассоциативной матрице. (Сами функции, заданные в ДНФ, и их табличная запись в ассоциативной матрице представлены на рис. 14 а). В конъюнкторе переменной без отрицания соответствует «1», с отрицанием — «0», отсутствию переменной — «М» (маска). При входном наборе ( $x_1 = 0, x_2 = 1, x_3 = 0$ ) происходит совпадение входного кода с кодами, записанными во второй и четвертой строках (отмечено стрелками), вследствие чего на шине совпадения этих строк появляется единичный сигнал, который вызывает считывание содержимого строк второй части матрицы.

В результате дизъюнкции содержимого второй и чет-

$$S \bar{z}, x, x_2 \vee z, \bar{x}, \bar{x}_2 \vee \bar{z}, x, \bar{x}_2 \vee \bar{z}, \bar{x}, x_2$$

$$z_2 \bar{z}, x_1 \vee z, x_2 \vee x, x_2$$

$S=0 \quad z_2=1$

↑      ↑

1	1	1	1	1
1	0	0	1	0
0	1	0	1	0
0	0	1	1	0
1	1	М	0	1
1	М	1	0	1
М	1	1	0	1

↑      ↑      ↑

$z_1=1 \quad x_1=1 \quad x_2=0$

$$y_1 = x, \bar{x}_2 \vee \bar{x}, x_2$$

$$y_2 \bar{x}, x_2 \bar{x}_3 \vee \bar{x}, x_2 \bar{x}_3 \vee x, \bar{x}_2 \bar{x}_3$$

$y_1=1 \quad y_2=1$

↑      ↑

1	0	М	1	0
0	1	М	1	0
1	1	1	0	1
0	1	0	1	1
1	0	0	1	1

↑      ↑      ↑

$x_1=0 \quad x_2=1 \quad x_3=0$

Рис. 14

вертой строк образуются значения выходных функций:  $y_1 = 1$  и  $y_2 = 1$ . Очевидно, при входном наборе ( $x_1 = 1$ ,  $x_2 = 1$ ,  $x_3 = 1$ ) совпадение реализуется в третьей строке матрицы, откуда получаем:  $y_1 = 0$  и  $y_2 = 1$ .

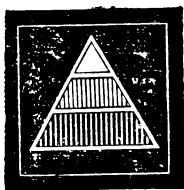
Для реализации указанной системы булевых функций требуется 5 строк (по общему числу минитермов) и 5 столбцов (по числу входных и выходных переменных) ассоциативной матрицы.

Реализация функции полного комбинационного сумматора в ассоциативной матрице приведена на рис. 14, б, где  $x_1$  и  $x_2$  — одноразрядные операнды;  $z_1$  — перенос с предыдущего разряда;  $S$  — сумма, а  $z_2$  — перенос в последующий разряд. Логические функции для сумм и переноса изображены в ДНФ. При подаче входного набора ( $z_1 = 1$ ,  $x_1 = 1$ ,  $x_2 = 0$ ) совпадение происходит с кодом, записанным в пятой строке (11М). В результате на выходе получаем значения  $S = 0$  и  $z_2 = 1$ , предварительно записанные в последних двух разрядах пятой строки.

Для реализации полного сумматора в матрице требуется 7 строк и 5 столбцов.

Очевидно, реализация сложных комбинационных функций потребует объединения некоторого числа матриц. При этом возникает каскадное соединение матриц, что увеличивает глубину логики; соответственно возрастают и задержки. Сложные логические функции от большого числа переменных следует разбивать на подфункции от меньшего числа переменных, используя известные методы декомпозиции функций. При этом вначале вычисляются промежуточные функции от числа переменных, допустимых для стандартной матрицы, что несколько увеличивает общее время, затрачиваемое на вычисление заданных функций.

С точки зрения технологии изготовления рассматриваемой ассоциативной матрицы наибольший интерес представляют биполярные и МДП-интегральные схемы. Размеры матрицы, выполненной в виде одной БИС, будут ограничиваться достигнутым уровнем интеграции компонентов и возможным числом электрических выводов.



#### 4

## ОРГАНИЗАЦИЯ АССОЦИАТИВНОЙ ВИРТУАЛЬНОЙ ПАМЯТИ

Разработка концепции виртуальной памяти направлена на создание программно-аппаратурных средств, которые позволяли бы организовать взаимодействие внешнего и оперативного ЗУ таким образом, чтобы в распоряжение пользователя предоставлялась как бы одноуровневая память большой емкости.

Идея организации виртуальной памяти, выдвинутая впервые при построении машин «Атлас» и «Стретч», получила в дальнейшем широкое распространение. Основные принципы организации виртуальной памяти заключаются в следующем. Внешнее ЗУ, объем которого существенно превосходит объем оперативного, разделяется на страницы определенного размера. В оперативном ЗУ имеется буферная зона, вмещающая относительно небольшое количество страниц. Информация о наличии и местоположении страниц в оперативном ЗУ фиксируется в справочной таблице. Обращение к памяти осуществляется посредством так называемого математического адреса, который состоит из двух частей: номера страницы и адреса слова внутри ее. В справочной таблице страниц по первой части математического адреса определяется, есть ли требуемая страница в оперативном ЗУ. Если есть, то происходит непосредственное обращение к этой странице; в противном случае предварительно производится замещение одной из страниц оперативного ЗУ новой из внешней памяти. Существуют различные стратегии замещения страниц, применение которых определяется условиями работы виртуальной памяти.

В случае эффективной организации виртуальной памяти большинство запросов будут адресоваться к страницам, находящимся в оперативном ЗУ, и для удовлетворения запросов не потребуется часто производить медленные операции обмена с внешней памятью. Чтобы не терять получаемый благодаря этому выигрыш в скорости, необходимо обеспечить достаточно быстрый поиск местоположения требуемой страницы в оперативном ЗУ. До-

стижение этой цели возможно только на основе реализации справочной таблицы с помощью ассоциативного ЗУ, причем емкость его определяется количеством страниц в буферной зоне и может быть относительно невелика.

Однако указанные принципы позволяют непосредственно строить виртуальную память только адресного типа.

Ниже описаны разработанные принципы организации ассоциативной виртуальной памяти, которая непосредственно обеспечивает доступ к информации по значению признаков. В основу разработки положен динамический метод построения массивов, предложенный С. Я. Берковичем. Благодаря этому в рамках обычной страничной организации получается простой единообразный механизм управления физическими ресурсами памяти и логическими структурами данных. Достоинство предлагаемой виртуальной памяти состоит в том, что она может эффективно функционировать в различных режимах работы вычислительных систем; ее использование обеспечивает экономный расход емкости запоминающих устройств и высокую скорость обработки различных видов запросов.

Рассмотрены две схемы виртуальной ассоциативной памяти.

Одна из схем обеспечивает ассоциативный доступ к данным по простому ключу и по своим основным техническим характеристикам в соответствии с классификацией, приведенной в первом разделе брошюры, может быть отнесена к третьему типу ассоциативной памяти — АЗУ с одним признаком фиксированной длины. Однако программные методы, используемые для реализации этой памяти, позволяют сделать переменной как длину признака опроса, так и искомой информации.

Другая схема служит для организации процедур доступа к информации по составному ключу и может быть отнесена к ассоциативной памяти первого и второго типов в рассмотренной классификации.

**Динамический метод.** К организации информационных массивов предъявляются, вообще говоря, различные, часто противоречивые, требования одно из наиболее важных требований заключается в том, что массив должен быть динамическим, т. е. его структура должна обеспечивать удобство наращивания системы, добавления, исправления и исключения данных. Естественно, при этом необходимо также обеспечить программную независимость

организации массива от объема и формата данных, минимальность времени проведения различных операций поиска и экономичность расхода памяти.

Динамический метод построения информационных массивов дает возможность в полной мере удовлетворять необходимой совокупности требований.

Общая схема построения массива согласно предлагаемому методу следующая (рис. 15). Пусть на множестве единиц информации, из которых составляется информационный массив, задано отношение упорядочения. Единицы информации входят в состав блоков, которые размещаются во внешней памяти. В оперативном же ЗУ сохраняется справочная таблица, позволяющая установить возможность принадлежности каждой единицы информации к тому или иному блоку; тем самым на множестве единиц информации устанавливается некоторое отношение эквивалентности. Динамика роста массива основана на принципе деления содержимого блока при его переполнении. Когда единиц информации мало, они размещаются в одном единственном блоке. Как только этот блок переполнится, то он разделяется на два таких, что любая единица информации в одном из блоков младше любой еди-

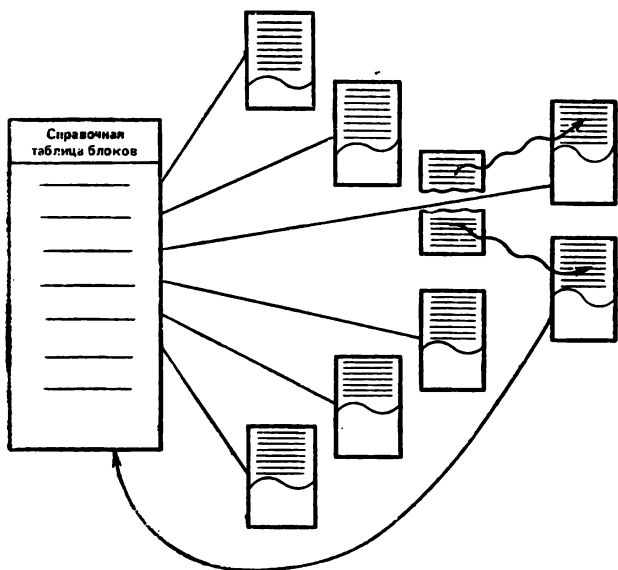


Рис. 15

ницы информации в другом блоке. (В справочную таблицу выносятся младшая единица информации из каждого блока.) Новые единицы информации поступают в один из полученных двух блоков в соответствии со справочной таблицей. Так продолжается до тех пор, пока какой-нибудь из блоков не переполнится, после чего он снова разделится, а справочная таблица пополнится еще одной единицей информации и т. д. При исключении единиц информации из массива может быть организовано слияние двух соседних блоков в один.

В определенных случаях можно исключить лишние обращения к внешней памяти, поскольку нет необходимости фактически заменять переполнившийся блок двумя новыми; достаточно лишь сформировать только один новый, включающий в себя единицу информации, вызвавшую переполнение. Исходный же блок, который в результате этого будет содержать избыточную дублирующую информацию, до поры до времени может быть оставлен без изменения. В справочной таблице блоков такой блок должен быть снабжен специальным признаком, указывающим на то, что в дальнейшем перед записью в него он должен быть предварительно отредактирован.

Возможны два основных правила деления блока.

По первому из них, который назовем делением по объему, содержимое переполняющего блока размещается таким образом, что объем каждого из вновь образуемых блоков используется примерно наполовину.

По второму правилу, который назовем делением по значению разрядов, заполнение новых блоков ведется на основе анализа значений двоичных разрядов поисковых признаков единиц информации; при первоначальном делении блока анализируется самый старший разряд, при последующих делениях — очередной, более младший. Единицы информации поступают в тот или иной блок в зависимости от значения анализируемого разряда (0 или 1). Если коды признаков распределены более или менее равномерно, каждый из этих блоков оказывается заполненным примерно наполовину.

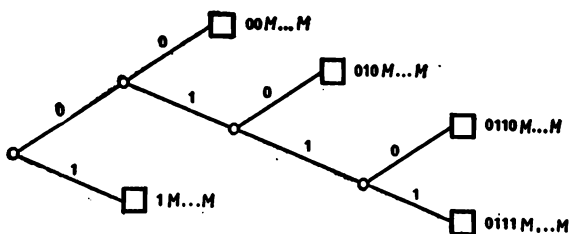
Применение второго правила деления позволяет упростить структуру справочной. Она принимает вид двоичного дерева (рис. 16), концевые его вершины можно отождествить с буквами некоторого алфавита. Тогда совокупность кодовых комбинаций для этих букв, задаваемых в виде путей от корня к концевым вершинам, будет обла-



дать префиксным свойством, подобно известным кодам Шеннона — Фано или Хаффмена. Процедура поиска требуемого блока в справочной с этой точки зрения является процедурой декодирования для поискового признака.

Если в процессе роста массива переполняется блок, содержащий справочную таблицу, то он сам подвергается делению по общему правилу. В этом случае строится «справочная справочной» и т. д. При этом массив представляется многоуровневой системой блоков; поиск информации в такой системе осуществляется путем обращения на каждом уровне только к одному блоку. Блоки более высокого уровня целесообразно размещать в более быстродействующих ЗУ; таким образом, массив, имеющий описанную структуру, «вкладывается» в иерархическую систему памяти современных вычислительных машин. Наиболее удобно, однако, реализовать рассмотренную схему с помощью двухуровневого устройства, используя внешнюю память, имеющую произвольный доступ к страницам (как в разрабатываемых оптических ЗУ и на цилиндрических магнитных доменах), или циклический доступ (как в обычных магнитных барабанах и дисках).

В динамическом методе не делается различия между регулярным построением массива и внесением изменений, причем редактирование носит чисто локальный характер, система все время находится в состоянии роста, положительного или отрицательного. Массив занимает во внешней памяти лишь примерно в полтора раза больше места, чем то, которое занимали бы все единицы информации, выписанные подряд. При выборе соответствующих размеров блока и справочной таблицы удовлетворение единичного запроса к массиву возможно провести всего за одно обращение к внешней памяти; наряду с этим искомая информация оказывается расположенной в блоках массива



Р и с. 16

пространственно компактно, благодаря чему массив является равным образом эффективно приспособленным и для обработки групповых запросов в режиме последовательного просмотра.

*Биологическая аналогия.* Можно усмотреть довольно очевидную аналогию между системой, функционирующей на основе динамического метода построения массивов, и биологическим объектом, растущим в результате деления клеток. Подобного рода аналогии могут быть найдены в самых различных областях, относящихся к изучению развивающихся систем, поскольку при этом используется принцип, согласно которому произвести сложную систему легче в виде набора инструкций, чем в виде полного описания ее в развитом состоянии. Подробное рассмотрение таких аналогий имеет смысл, когда оно приводит к какому-нибудь содержательным выводам, а именно когда черты одного явления могут быть использованы для объяснения другого. В данном случае представляется интересным то, что рассматриваемая аналогия вскрывает некоторые черты механизма дифференцировки клеток — важнейшего явления биологии развития.

Прежде всего отметим вариант, состоящий собственно не в делении, а в формировании нового дополнительного блока, включающего в себя единицу информации, вызвавшую переполнение. Оказалось, что аналогичное явление можно найти в биологии; имеются объекты (дрожжи), в которых клетки размножаются не делением, а путем отпочкования дочерней клетки от материнской.

С общей точки зрения проблемы биологии развития сводятся к построению модели детерминированного растущего автомата, которая позволяла бы объяснить основные черты процесса построения организмов. В первую очередь, такая модель должна прояснить, каким образом возникает несимметрия и каким образом происходит дифференцировка клеток, т. е. как одна клетка в ходе самовоспроизведения дает начало совокупностям клеток, совершенно различных по своим функциональным свойствам. Кроме того, подобного рода модель должна достаточно просто и эффективно реализовываться биохимическими механизмами клетки.

Известны различные модели развития, опирающиеся на небольшой набор аксиоматических положений, с помощью которых были изучены те или иные стороны данной проблемы.

Представляется плодотворным ввести в рассмотрение дополнительную аксиому, исходной предпосылкой для которой является аналогия со вторым правилом деления переполняющихся блоков информационного массива. Для модели клетки это означает, что предполагается существование механизма, который фиксирует историю делений клетки, т. е. две клетки, получающиеся в результате деления, не являются абсолютно идентичными; на одну из них навешивается ярлык с «0», на другую — с «1».

Состояние развивающегося организма в каждый текущий момент может быть отображено в виде двоичного дерева (см., например, рис. 16), концевые вершины которого представляют клетки, а пути к этим вершинам от корня в виде последовательностей «0» и «1» — ярлыки, навешенные на клетки и описывающие историю их возникновения. Количество делений, соответствующее клеткам разных типов, может быть различным, в результате чего ветви дерева могут иметь существенно неодинаковую длину. Число клеток организма в данный момент равно числу концевых вершин двоичного дерева.

С помощью предлагаемого механизма может быть дана, в частности, интерпретация следующим фактам.

1. Программа управляющих воздействий записана внутри самой клетки, так что организм сам управляет своим развитием.

2. Процесс дифференцировки клеток в ходе развития может происходить постепенно. К одному типу клеток могут быть отнесены все клетки, имеющие одинаковый начальный участок пути в дереве (рис. 16); длина этого участка для клеток разных типов может быть различной. Внутри данного типа клеток могут быть определенные подтипы и т. д. Разбиение клеток на типы в ходе развития может меняться в сторону более глубокой дифференцировки, при этом через некоторое количество делений после прохождения определенного начального участка в дереве дифференцировка может прекратиться.

3. Как было установлено Хайфликом, имеется предел возможного количества делений клетки, который для клеток человека равен примерно 50. Вызывает сомнение объяснение, которое обуславливает существование этого более или менее четко выраженного предела накоплением каких-то ошибок в процессе последовательных делений клетки. Более привлекательно объяснение существования предела Хайфлика тем, что при последовательных де-

лениях клетки исчерпывается информационная емкость механизма, фиксирующего историю делений клетки. Обратно говоря, этот механизм представляет собой регистр на 50 разрядов; в зародышевой клетке организма во всех разрядах регистра записаны безразличные символы «М», в результате последующих делений символы «М» будут заменяться в одной из клеток на «0», в другой — на «1».

4. Теории, объясняющие перерождение нормальных клеток в злокачественные, связывают это с ослаблением и утратой клетками дифференцировки. Данный процесс может быть интерпретирован как сбой механизма, фиксирующего историю делений клетки.

5. Появление однояйцевых близнецов может быть объяснено тем, что сбой рассматриваемого механизма, фиксирующего историю делений клетки, происходит на первом делении оплодотворенной клетки. В результате появляются две идентичные клетки, каждая из которых, развиваясь, дает начало различным организмам с одинаковой информационной структурой.

Биология развития характеризуется очень многими сложными процессами, и для всестороннего объяснения их требуется привлечение многих механизмов различных видов. Гипотеза о том, что специализация дифференцированных клеток связана с состоянием механизма, фиксирующего историю деления клетки, позволяет дать согласованное представление об определенной совокупности биологических явлений.

Ассоциативная виртуальная память с поиском по простому ключу. В качестве АЗУ с одним признаком фиксированной длины можно представить память адресного типа, если рассматривать в качестве адреса соответствующий ассоциативный признак. В этом случае удобным образом можно было бы осуществлять операции ассоциативного доступа — извлекать по значению признака требуемые данные за одно обращение к устройству. Однако непосредственное использование такой схемы совершенно нерационально, во-первых, потому, что для этого потребовался бы чрезмерный объем физического адресного пространства даже для сравнительно коротких поисковых признаков, во-вторых, коэффициент его использования был бы крайне ничтожен.

Для реализации данной схемы предлагается применить конструктивные принципы виртуальной памяти, исполь-

зую при этом динамический метод построения массивов. В распоряжение пользователя предоставляется фиктивное адресное пространство, емкость которого определяется разрядностью кодов признаков и может намного превосходить суммарный физический объем используемых запоминающих устройств. Рис. 17 иллюстрирует эту схему; на нем показана также структура ячеек виртуальной памяти, в которых фактическое заполнение полей признака и искомой информации отмечено штриховкой.

При организации ассоциативной виртуальной памяти создается массив, в который в качестве единиц информации заносятся математические адреса используемых ячеек и их содержимое.

Для блоков виртуальной памяти, хранящихся в буферной зоне оперативного ЗУ, целесообразно иметь возможность быстрого определения их местоположения. Для этого следует принять дополнительные меры, и в случае, когда вся справочная достаточно велика по объему, может оказаться необходимым выделить из нее сравнительно небольшую часть, связанную со страницами буферной зоны, и организовать ее таким образом, чтобы максимально повысить скорость поиска. Так как в оперативном ЗУ могут находиться произвольные блоки, то указанная часть справочной или таблица страниц должна характеризовать каждый блок как наименьшим, так и наибольшим адресами, относящимися к данному блоку. При

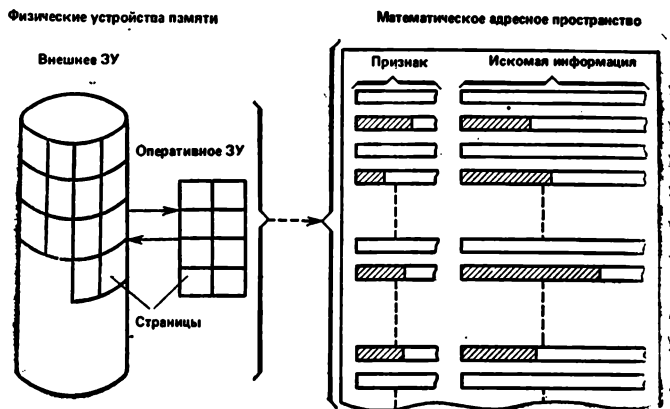
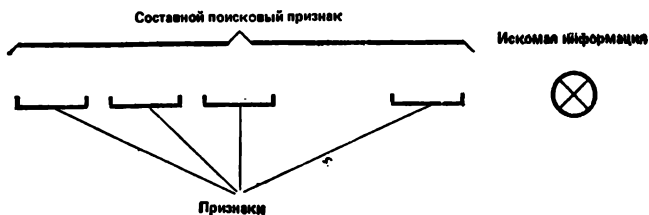


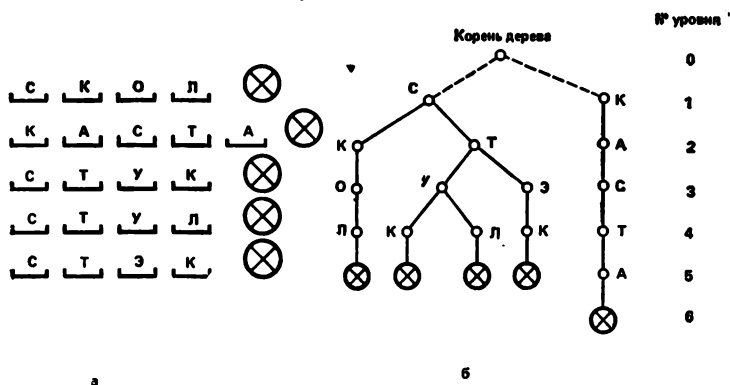
Рис. 17



Р и с. 13

этом поиск в таблице страниц будет осуществляться по критерию «в заданных пределах». В качестве аппаратного средства для ускорения выполнения процедур в справочных таблицах можно использовать АЗУ на трехзначных элементах, в котором операции поиска по критериям «ближайшее меньшее» и «в заданных пределах» могут быть реализованы за одно обращение к устройству. Емкость АЗУ, требуемого для этих целей, относительно невелика — порядка  $10^3 \div 10^4$  бит.

*Ассоциативная виртуальная память с поиском по составному ключу.* Структура единиц информации с составным поисковым признаком изображена на рис. 18. Пример массива, состоящего из единиц информации такого рода, показан на рис. 19, а; для удобства чтения рисунка значения признаков обозначены буквами, которые подобраны так, чтобы каждый набор признаков представлял собой некоторое осмысленное слово.



Р и с. 19

Так как в этом массиве значения простых признаков для многих единиц информации могут быть одинаковыми, то для экономного представления массива удобно использовать структуру дерева (рис. 19, б). В результате исключается повторение признаков, совпадающих на верхних уровнях дерева, однако для указания связей между признаками требуется придать каждой вершине адреса связи, один из которых определяет адрес соседней вершины «вниз» по дереву, а другой — «вправо» по дереву.

Нетрудно показать, что организация данных в виде дерева, как правило, дает экономию памяти в  $g$  раз:

$$g \approx \frac{w-1}{w} \cdot \frac{h}{1 + \frac{2 \log_2 B}{r}} \sim \frac{h}{2},$$

где  $w$  — средний коэффициент ветвления ( $\geq 2$ );  $h$  — количество уровней в дереве;  $B$  — размер блока;  $r$  — количество разрядов, кодирующих один признак.

Известные методы машинной организации древовидных структур, в которых вершины каждого иерархического уровня объединяются в отдельные блоки, размещаемые во внешней памяти, оказываются неэффективными. В этом случае для поиска некоторой вершины требуется вызывать из внешней памяти по меньшей мере столько блоков, сколько предшествующих вершин имеется на пути к ней от корня дерева. На самом деле число обращений к внешней памяти может быть значительно больше из-за того, что для размещения вершин некоторого иерархического уровня может потребоваться не один, а несколько блоков. Увеличение объема хранимых данных (рост дерева), как правило, сопровождается возрастанием времени поиска. Имеется еще один недостаток такой организации дерева. Так как признаки, характеризующие некоторую единицу информации, хранятся в разных блоках, то оказывается невозможной выборка единиц информации путем последовательного просмотра.

Организация дерева на основе динамического метода позволяет обойти указанные недостатки, на структуру дерева не накладывается практически никаких ограничений, и обработка единичного запроса производится за одно обращение к внешнему ЗУ, при этом оказывается возможным извлекать данные путем последовательного просмотра блоков в силу их взаимной независимости.

Организация дерева признаков в памяти машины про-

изводится следующим образом. Устанавливается лексикографическое упорядочение для структурных имен всех вершин. В соответствии с чем можно задать правило их обхода, по которому на множестве всех вершин устанавливается отношение глобального порядка, благодаря чему к ним можно применить общую схему динамического метода. Однако при этом нарушаются связи соподчиненных вершин одного уровня, и блоки такого массива будут представлять собой, вообще говоря, уже не дерево, а «лес» — совокупность несвязных компонент. Процедура обращения к дереву складывается из поиска блока путем нахождения в справочной таблице блоков ближайшего меньшего набора признаков и из определения необходимой компоненты внутри блока.

Рассмотрим это на конкретном примере (рис. 20). К каждому блоку следует отнести совокупность вершин, заключенных между соседними сечениями, которые на рисунке отмечены штрих-пунктирными линиями (I — V). Каждый блок можно характеризовать какой-либо граничной вершиной, например лексикографически младшей. Совокупность блоков размещается во внешнем ЗУ, а таблица путей, ведущих в граничные вершины, помещается в оперативную память и используется в качестве справочной блоков дерева (рис. 21, а).

Первоначально, пока вершин мало, все дерево представляет собой один блок. При переполнении блока производится его деление. Например, блок III на рис. 20

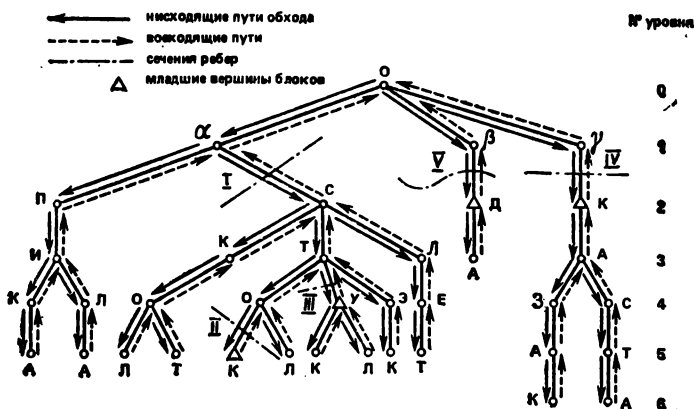


Рис. 20

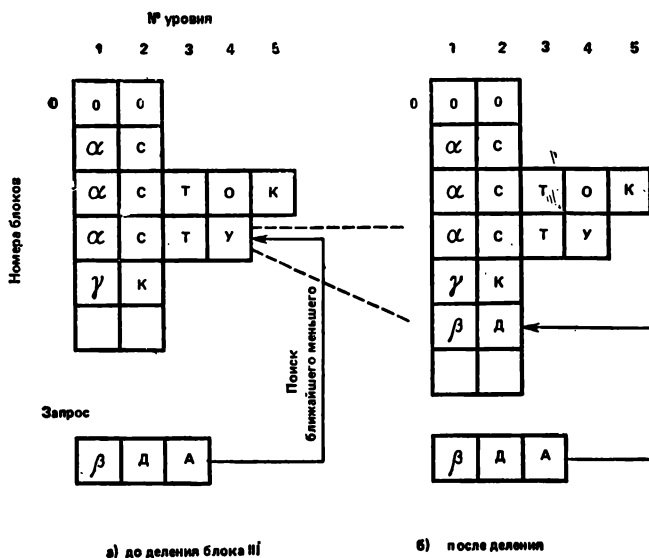


с помощью сечения V (волнистая линия) разбивается на два блока; в справочную вносятся соответствующие изменения (рис. 21, б).

При организации поиска с помощью справочной определяется вершина, структурное имя которой является ближайшим меньшим по отношению к поступившему запросу. Например, на рис. 21, а ближайшим меньшим к запросу ВДА является структурное имя  $\alpha$ СТУ, соответствующее блоку III; после деления блока III (рис. 21, б) этому же запросу соответствует блок V. Найденный блок вызывается в оперативное ЗУ, после чего запрос удовлетворяется.

Описанная организация массива позволяет непосредственно выполнять операцию поиска, если для искомой единицы информации набор признаков задан полностью. В случае, когда набор признаков задается неполностью, для обеспечения возможности эффективного извлечения информации в различных разрезах в систему необходимо ввести соответствующие ассоциативные связи.

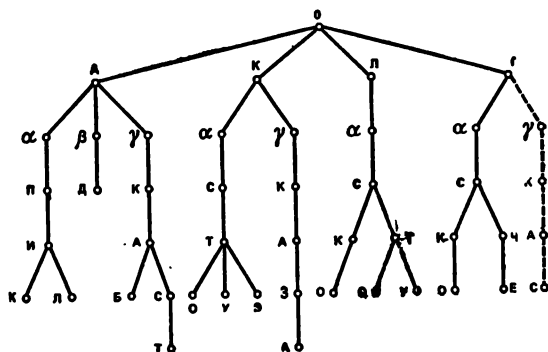
В дереве связи между вершинами могут быть установлены путем задания в одной из вершин ссылки на положение другой одноименной. Такая ссылка представляется



Р и с. 21

Рассмотрим построение инвертированной справочной на следующем примере. Пусть в дереве (см. рис. 20) необходимо установить ассоциативные связи для всех концевых вершин. Для этого надо каждому наименованию, используемому для обозначения концевых вершин, сопоставить совокупность путей, которые ведут в вершины с этим наименованием. В рассматриваемом дереве имеются всего четыре разных наименования концевых вершин: «А», «К», «Л», «Т»; эти наименования образуют четыре вершины первого уровня в справочном массиве (рис. 22), представленном в виде дерева. К указанным вершинам «подвешиваются» соответствующие структурные имена, взятые из дерева (рис. 20). Например, три структурных имени ( $\alpha$ СКО,  $\alpha$ СТО,  $\alpha$ СТУ), соответствующие трем концевым вершинам с наименованием «Л», подвешиваются в инвертированном дереве к вершине первого уровня с наименованием «Л».

Из инвертированной справочной можно извлекать сведения о количестве в массиве единиц информации со



**Рис 22**

значением признака, заданным в запросе. На основании этих сведений можно выбирать режим доступа к массиву: либо путем последовательного просмотра, либо путем прямого доступа к вершинам, структурные имена которых предварительно извлекаются из инвертированной справочной.

В силу возможности извлечения единиц информации из рассматриваемой структуры путем последовательного просмотра основное дерево и инвертированная справочная могут быть при необходимости достаточно легко перестроены.

Таким образом, можно организовать ассоциативную виртуальную память с несколькими независимыми фиксированными признаками и обеспечить эффективное выполнение многоаспектного поиска по составному ключу в различных разрезах.

## ЗАКЛЮЧЕНИЕ

Количество областей применения ЭВМ неуклонно возрастает, по темпам роста вычислительная техника опережает другие отрасли. Крупнейшие достижения современности в атомной и ракетной технике, в физике, биологии и исследовании космоса были бы немыслимы без помощи электронных вычислительных машин. К одной из наиболее актуальных областей применения ЭВМ относят создание автоматизированных систем управления сложными комплексами. Примерами могут служить экономические системы планирования и оперативного управления, системы управления технологическими объектами, резервирования авиационных билетов, системы обработки радиолокационных сигналов, управления космическими объектами, обработки данных непосредственно в ходе проведения экспериментальных исследований и многие другие.

Представляется, что прогресс человечества требует непрерывного увеличения количества обрабатываемой информации, и, таким образом, в числе прочего он оказывается связанным с совершенствованием средств вычислительной техники.

В течение минувшей четверти века сменилось несколько поколений ЭВМ, причем эффективность машин последних поколений на несколько порядков превосходила

эффективность своих предшественников. Но уже сегодня скорость выполнения вычислительных операций достаточно близка к своему принципиальному пределу. Определенные возможности повышения производительности ЭВМ имеются на пути распараллеливания вычислительных процессов. Однако разделение процесса на необходимое количество ветвей, каждая из которых может выполняться независимо, с одной стороны, не всегда допустимо, с другой — зачастую требует слишком сложной организации.

Одной из определяющих тенденций в развитии вычислительной техники является разработка более эффективных систем памяти. Необходимость совершенствования памяти ЭВМ определяется в значительной степени потребностями в создании больших информационных систем и банков данных, представляющих собой непреходящую составную часть автоматизированной системы управления.

Повышение эффективности систем памяти ЭВМ предполагает не только построение физических устройств памяти большой емкости и высокого быстродействия, но и совершенствование логических средств манипулирования данными. Это направлено на увеличение производительности процесса обработки информации и упрощение больших программных систем, которые в настоящее время являются характерным примером наиболее сложных и несовершенных продуктов человеческой деятельности.

Разработка концепции ассоциативной организации памяти может способствовать решению всевозможных аспектов рассматриваемой проблемы. Реализация больших ЗУ с ассоциативной организацией чисто аппаратными средствами наталкивается на известные технологические трудности, в связи с чем важнейшую роль в этом деле при использовании ЗУ с произвольным, циклическим и последовательным доступом играют программные средства.

Описанная в брошюре виртуальная ассоциативная память, основанная на динамическом методе построения информационных массивов, обладает рядом существенных преимуществ.

Реализация такой виртуальной памяти программным способом позволяет ввести в состав математического обеспечения существующих ЭВМ удобное и эффективное средство работы с информационными массивами.

Покажем на примерах возможности использования ассоциативной виртуальной памяти.

Информационную структуру многих объектов различных типов удобно изображать посредством графов общего вида. В общем случае информационное содержание графа составляют наименования вершин и ребер и комментарии к ним. Примерами объектов, которые описываются с помощью графов, могут служить: транспортные и коммуникационные сети, сетевые графики комплекса работ, классификаторы экономических терминов и т. д. В частности, для транспортной сети наименованиями графа являются названия пунктов, наименованиями ребер — наименования дорог, связывающих соответствующие пункты; комментариями к вершинам являются характеристики пунктов, а комментариями к ребрам — характеристики соответствующих дорог.

В случае больших графов удобно применить представление информационного содержания графа путем указания наименования вершины и ее комментария с перечислением исходящих ребер и их комментариев; наименования ребер задаются посредством указания наименований примыкающих вершин, поскольку начальная вершина у всех этих ребер общая. Коль скоро количество и длина таких единиц информации будут переменными, для их накопления и хранения целесообразно использовать описанную структуру виртуальной памяти.

Другим примером может служить организация поиска в многомерных пространствах. Рассматривается следующий тип задач. Имеются координаты  $(x_1, x_2, \dots, x_n)$  множества точек  $X$  в многомерном пространстве и задаются координаты какой-то точки  $A (a_1, a_2, \dots, a_n)$ . Требуется найти точки, принадлежащие множеству  $X$  и удовлетворяющие некоторому условию близости к точке  $A$ , например, требуется найти точку, ближайшую в смысле евклидовой метрики, или точки, находящиеся в заданной окрестности, и т. п. Такого рода задачи возникают при распознавании образов, моделировании систем и ряде других проблем.

В общем случае для решения поставленной задачи необходимо полностью просмотреть все точки множества  $X$  и сопоставить их с заданной точкой  $A$ . Однако во многих конкретных задачах следует изучить возможности локализовать просмотр. Для этого разобьем многомерную область, в которой содержатся точки множества  $X$ ,

на многомерные параллелепипеды со сторонами, параллельными координатным осям. Для получаемого разбиения легко установить правило нумерации параллелепипедов, и, таким образом, каждому параллелепипеду можно поставить в соответствие ячейку ассоциативной виртуальной памяти. Все точки, попадающие в один и тот же параллелепипед, помещаются в ячейку памяти, имеющую номер этого параллелепипеда. Такое представление информации для рассматриваемого типа задач хорошо согласуется со структурной схемой ассоциативной виртуальной памяти. При поиске сначала вычисляется номер параллелепипеда, в который попадает точка  $A$ , а затем для точек, помещенных в ячейку с этим номером, путем их последовательного просмотра выполняется требуемая процедура выбора.

Организация эффективного поиска данных по различным критериям представляет собой центральную проблему в деле построения информационных систем. Сложность этой проблемы вызывает стремление разграничить физический и логический уровни манипулирования данными. Конструируется некая абстрактная схема, в рамках которой могут быть достаточно удобно представлены данные различной природы. Следует заметить, что на логическом уровне любые схемы информационных структур эквивалентны. Это утверждение можно рассматривать как аналог тезиса Чёрча, согласно которому являются эквивалентными всевозможные формулировки понятия вычислительного алгоритма. Вопрос заключается только в том, насколько эффективна реализация той или иной схемы с точки зрения проведения поисковых операций. В связи с этим ограничения, имеющиеся на физическом уровне, неявно довлеют над возможностями выбора на логическом уровне. В универсальных информационных системах перед программистом возникает весьма непростая задача нахождения рациональных путей использования возможностей, предоставляемых системой. Выбор удачной организации требует большого опыта работы с системой; в случае неудачного выбора функционирование информационной системы может оказаться крайне неэффективным, что может потребовать полной перестройки информационных массивов.

Рассмотренные способы организации виртуальной ассоциативной памяти на основе динамического метода позволяют эффективно применить модульный принцип

построения информационных систем. Путем выбора соответствующего способа организации виртуальной ассоциативной памяти для различных информационных объектов можно построить свой программный модуль; каждый из таких модулей имеет определенный набор параметров, варьируя которые можно получить желаемую проблемную ориентацию банка данных.

Такой подход, в частности, был реализован в информационной системе для экономических задач, разработанной в Институте проблем управления. В состав этой системы включены банки данных для информационных объектов трех типов: для объектов, имеющих структуру графа общего вида, для объектов, имеющих структуру дерева, и для неструктурированных объектов. Объекты первого типа отображают классификационные отношения на множестве экономических терминов, объекты второго типа служат для представления составных поисковых признаков экономических показателей. При наличии банка данных для объектов, имеющих структуру графа общего вида, создание банка данных для объектов с древовидной структурой имеет смысл постольку, поскольку в первом случае реализуются лишь поисковые операции, связанные с переходами между смежными вершинами, а во втором случае требуется эффективный поиск по запросам, формулируемым в виде цепочки вершин на пути от корня к конечным вершинам. В рассматриваемой информационной системе банк неструктурированных объектов используется для построения словаря идентификационных шифров экономических терминов. Этот банк основывается на некоторой разновидности виртуальной ассоциативной памяти с поиском по простому ключу, структура сопутствующей информации в нем не определяется, и программа для ее обработки в случае надобности должна создаваться самим пользователем. Применение словаря идентификационных шифров решает проблему автоматического кодирования. При этом открываются дополнительные возможности: устранение синонимической неоднозначности, коррекция определенных типов ошибок, согласование системного и подсистемного кодирования, которое позволяет организовать межсистемные обмены без перекодировки. Реализация указанных возможностей не вызывает существенного уменьшения скорости выполнения поисковых операций. Следует также отметить, что в словаре удалось организовать проведение

поиска в двух направлениях — от слова к идентификатору и от идентификатора к слову — без инвертирования исходного массива.

Можно было бы привести ряд других примеров использования разработанного метода построения ассоциативной виртуальной памяти. Этот метод целесообразен в особенности, когда имеют дело с большим количеством данных переменного объема и меняющейся структуры.

Принципы ассоциативной организации памяти, описанные в брошюре, могут служить основой для построения эффективных систем управления памятью современных и будущих ЭВМ.

## ЛИТЕРАТУРА

Крайзмер Л. П., Бородаев Д. А., Гутенмахер Л. И., Кузьмин Б. Н. и Смелянский И. Л. Ассоциативные запоминающие устройства. Л., «Энергия», 1967.

Беркович С. Я. О размерности информационных структур в пространстве восприятия. — «Биофизика», 1976, т. 21, вып. 6.

Китов А. И. Программирование информационно-логических задач. М., «Сов. радио», 1967.

Хэнлон Э. Ассоциативные запоминающие устройства. — В сб.: «Запоминающие устройства современных ЭЦВМ». М., «Мир», 1968.

Минский М. и Пейперт С. Перцептроны. М., «Мир», 1971.

Беркович С. Я., Кочин Ю. Я. и Молчанов В. А. Об эффективности применения ассоциативной памяти в вычислительных процедурах. — В сб.: «Вычислительные системы». Новосибирск, 1975, вып. 62.

Прангшвили И. В., Попова Г. М., Смородинова О. Г. и Чудин А. А. Однородные микроэлектронные ассоциативные процессоры. М., «Сов. радио», 1973.

Многопроцессорные вычислительные системы. Сб. под ред. В. Л. Арлазарова и А. Ф. Волкова. М., «Наука», 1975.

Кочин Ю. Я. Логический анализ ассоциативных структур памяти. — «Кибернетика», 1970, № 4.

Беркович С. Я., Кочин Ю. Я. и Лапир Г. М. Алгоритмы групповой обработки слов в ассоциативной памяти и их реализация с помощью однородных вычислительных структур. — «Автоматика и телемеханика», 1974, № 8.

Беркович С. Я. и Кочин Ю. Я. О поиске чисел, ближайших к заданному. — «Автоматика и телемеханика», 1975, № 2.

Бремер Дж. Сверхпроводящие устройства. М., «Мир», 1964.

Лихарев К. К., Ульрих Б. Т. Системы с джозефсоновскими контактами. М., МГУ, 1976.



Беркович С. Я. и Лапир Г. М. Запоминающая ячейка на криотронах. А/С № 443410. — «Бюлл. изобр.», 1974, № 34.

Беркович С. Я. О машинной организации растущего поискового дерева. — ДАН СССР, 1972, т. 202, № 2.

Аптер М. Кибернетика и развитие. М., «Мир», 1970.

Orgel L. E. Ageing of clones of Mammalian Cells. «Nature» 1973, June 22, v. 243.

Арлазаров В. Л., Беркович С. Я., Кочин Ю. Я., Леман А. А. и Шалаев Ю. В. Об организации словаря идентификаторов в наращиваемой информационной системе. — «Автоматика и телемеханика», 1972, № 4.

Иванов Ю. Н. и Беркович С. Я. Принципы организации банка данных для экономических систем. Сб. докладов сессии Отделения механики и процессов управления АН СССР. М., «Наука», 1976.

**Семен Яковлевич БЕРКОВИЧ**

**Юрий Яковлевич КОЧИН**

## **АССОЦИАТИВНАЯ ПАМЯТЬ**

Редактор В. И. Ковалев

Обложка Л. П. Ромасенко

Худож. редактор В. Н. Конюхов

Техн. редактор Т. В. Самсонова

Корректор В. И. Гуляева

А 03268. Индекс заказа 64308.

Подписано к печати 21/VII 1976 г.

Бумага типографская № 3. Бум. л. 1. Печ. л. 2.

Тираж 45 230 экз.

Заказ 1362

Цена 11 коп.

Сдано в набор 10/VI 1976 г.

Формат бумаги 84×108<sup>1</sup>/<sub>32</sub>.

Усл. л. 3,36. Уч.-изд. л. 3,36

Издательство «Знание», 101835, Москва Центр, проезд Серова, д. 4,

Чеховский полиграфический комбинат Союзполиграфпрома

при Государственном комитете Совета Министров СССР

по делам издательств, полиграфии и книжной торговли

г. Чехов Московской области

