

## Особенности функционирования и применений ассоциативной памяти

*А.С.Парфененко, А.В.Рыбачек*

ОАО «Ангстрем-М» (г. Москва)

*Е.С.Рыбалко*

Московский государственный институт электронной техники  
(технический университет)

Представлен алгоритм поиска с применением ассоциативной памяти. Рассмотрены особенности применения ассоциативной памяти в приборах для организации компьютерных сетей и особенности исполнения блоков ассоциативной памяти.

Запоминающие устройства (ЗУ) с ассоциативным доступом, или память, адресуемая по содержанию (САМ – Content Addressable Memory), отличаются тем, что в них поиск и считывание информации происходят не по ее адресу, а по некоторому признаку самой информации, хранящемуся в слове. В ассоциативных ЗУ многозарядное слово разбивается на две части: метка (или признак) и данные. Поиск информации осуществляется по значению метки. Применяются ЗУ в специальных процессорных устройствах для поиска, сортировки и упорядочивания информации.

Ассоциативная память относится к классу схем с параллельным сравнением признаков. Эти схемы работают как стандартные схемы памяти и могут использоваться для хранения двоичных данных. Однако в отличие от стандартных схем памяти они имеют специальный режим сравнения, позволяющий осуществлять параллельный поиск данных в САМ.

До недавнего времени САМ использовалась в ряде специализированных приложений, при встраивании в заказные проекты. Однако с появлением технологий уровня 0,35, 0,25 и 0,18 мкм стало возможным создание блоков ассоциативной памяти с большой информационной емкостью, а также возникла необходимость в совершенствовании архитектуры САМ. Ассоциативная память становится одним из ключевых звеньев в сетевых системах обработки и передачи информации.

**Алгоритм поиска с использованием САМ.** Принцип работы ассоциативной памяти легко понять при сравнении с памятью с произвольным доступом (RAM – Random Access Memory).

Алгоритм поиска информации с использованием памяти с произвольным доступом представлен на рис.1.

Цикл поиска информации в этом случае включает следующие шаги:

- выработка процессором адреса;
- считывание данных из микросхемы памяти по данному адресу;
- сравнение полученных данных с искомой информацией.

Этот набор операций будет повторяться для каждой записи в памяти до тех пор, пока не будет найдена искомая информация. Тогда цикл будет остановлен. Например, для нахождения фамилии Сергеев (см. рис.1) требуется три цикла, так как она находится в третьей строке. В случае, если фамилия находится в первой строке, то для ее обнаружения потребуется только один цикл. Таким образом, число циклов, требуемых для обнаружения информации, зависит от расположения искомой информации в накопите-

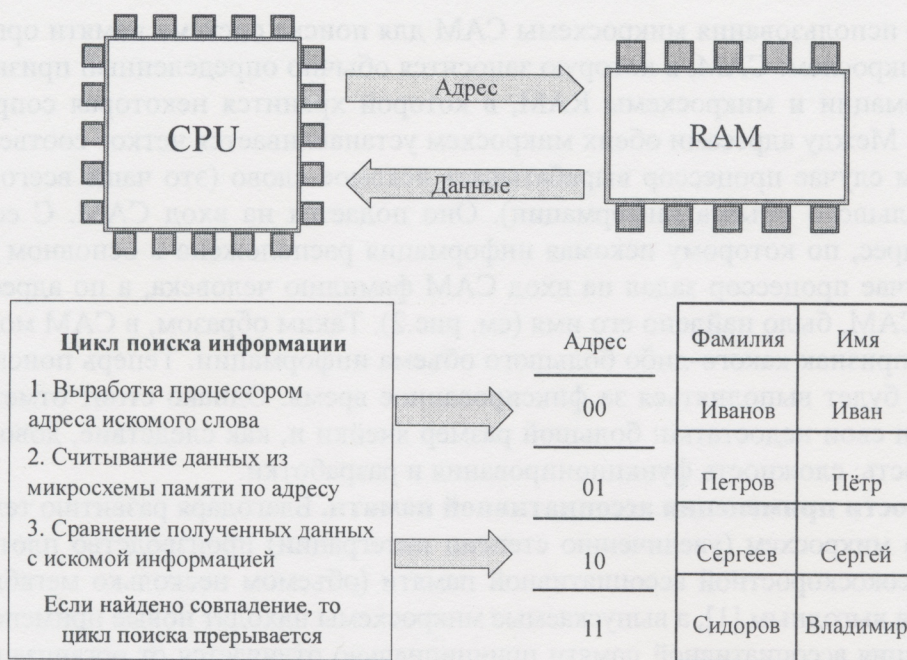


Рис. 1. Поиск информации с применением SRAM

ле. Зависимость времени поиска информации от того, насколько далеко она расположена от начала накопителя является существенным недостатком поиска при использовании RAM. Эта проблема весьма актуальна в связи с возросшими объемами обрабатываемой информации.

На рис.2 представлен алгоритм поиска информации с использованием ассоциативной памяти. Для нее цикл поиска информации состоит из подачи искомого слова на вход микросхемы и получения адреса искомой информации. Эти операции выполняются за один рабочий цикл. В данном случае время поиска не зависит от расположения искомой информации в накопителе, потому что в ассоциативной памяти происходит сравнение поданной на вход информации сразу со всеми записями в накопителе, после чего выдается сигнал о том, что искомая информация найдена или отсутствует в блоке памяти.

В базе данных может быть организован поиск имени сотрудника компании по его фамилии (см. рис.2).

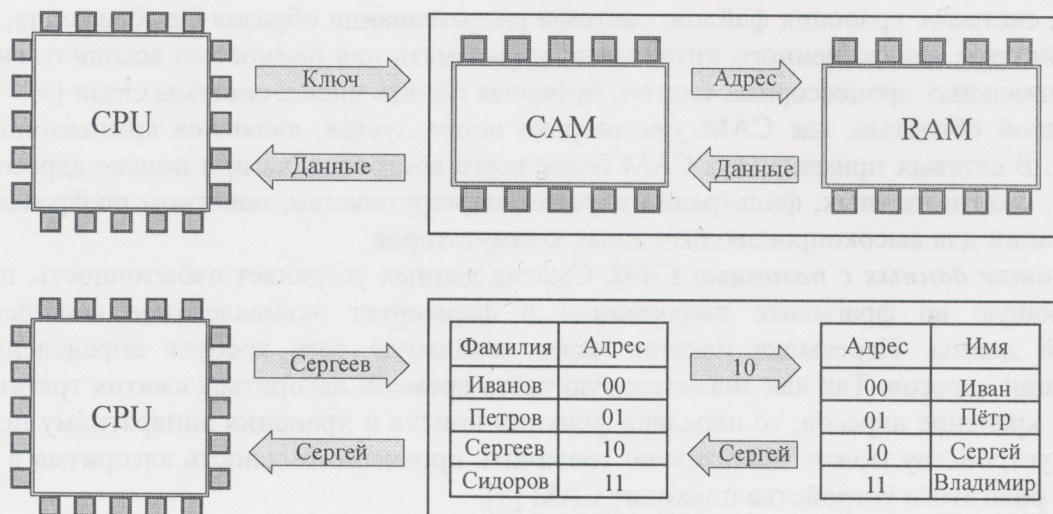


Рис.2. Поиск информации с использованием CAM

В случае использования микросхемы САМ для поиска система памяти организуется из двух микросхем: САМ, в которую заносится обычно определенный признак некоторой информации и микросхемы RAM, в которой хранится некоторая сопряженная информация. Между адресами обеих микросхем устанавливается четкое соответствие.

В данном случае процессор вырабатывает искомое слово (это чаще всего признак какого-то большого объема информации). Оно подается на вход САМ. С ее выхода снимается адрес, по которому искомая информация расположена в основном массиве. В нашем случае процессор задал на вход САМ фамилию человека, а по адресу, полученному из САМ, было найдено его имя (см. рис.2). Таким образом, в САМ можно хранить только признак какого-либо большого объема информации. Теперь поиск нужной информации будет выполняться за фиксированное время. Однако стоит отметить, что САМ имеет и свои недостатки: большой размер ячейки и, как следствие, довольно высокая стоимость; сложность функционирования и разработки.

**Особенности применения ассоциативной памяти.** Благодаря развитию технологии производства микросхем (увеличению степени интеграции) производство плотноупакованной и высокоскоростной ассоциативной памяти (объемом несколько мегабит) стало экономически выгодным [1], а выпускаемые микросхемы находят новые применения.

Организация ассоциативной памяти принципиально отличается от организации памяти с произвольным доступом. Данные в ассоциативной памяти хранятся в виде таблицы, которая может загружаться при включении системы и изменяться во время ее работы.

Практически в любом блоке ассоциативной памяти имеется набор битов, которые определяют статус данной записи в накопителе. Если данная запись больше не нужна пользователю, то она может быть перезаписана. Запись информации в ассоциативную память может производиться как по адресу, поданному на адресную шину, так и в первое свободное место в памяти.

В режиме сравнения информация, поданная на вход микросхемы, сравнивается со всеми записями в памяти. Если произошло совпадение, то ассоциативная память выдает соответствующий флаг.

Одним из важнейших свойств ассоциативной памяти (тернарного типа) является ее способность хранить в ячейке одно из трех состояний: логический ноль, логическую единицу и «безразличное» состояние. Это свойство является наиболее востребованным для сетевых применений памяти [2].

Основными областями применения ассоциативной памяти являются: поиск в базах данных, системах хранения файлов; системы распознавания образов (изображения, голоса); системы искусственного интеллекта; кэш-память для полностью ассоциативных или специальных процессорных систем; буферная память диска; системы связи [3].

Важной областью, где САМ уже широко используется, являются компьютерные сети [4]. В сетевых применениях САМ более всего востребована при поиске адресов в Ethernet, сжатии данных, фильтрации адресов, маршрутизации, пакетном шифровании информации для высокопроизводительных коммутаторов.

**Сжатие данных с помощью САМ.** Сжатие данных устраняет избыточность, присутствующую во фрагменте информации и формирует эквивалентное сообщение меньшей длины. Пересылка пакетов через локальную сеть требует определенной трансляции адресов. Так как значительную часть времени алгоритмы сжатия тратят на поиск и хранение адресов, то передача функций поиска и хранения аппаратному поисковому устройству может значительно увеличить производительность алгоритма в целом. На роль этого устройства подходит САМ [3].

При сжатии поиск в САМ осуществляется для каждого слова из оригинального выражения. Если найден код, соответствующий шаблону во входном регистре, то соответствующий символ или знак подается на выход, а входной регистр очищается, если код не найден, то на вход подается очередное слово. САМ будет генерировать результат за одну операцию, независимо от размера таблицы. Данное свойство делает ассоциативную память идеальным кандидатом для применения в алгоритмах сжатия.

**Использование САМ в сетевых коммутаторах.** САМ может использоваться в сетевых коммутаторах для выделения и обработки адресной информации из входящих пакетов [3]. Для того чтобы перенаправить пакет в правильный выходной порт, выполняется сравнение адреса назначения с таблицей адресов, записанной в САМ. Например, САМ могла бы хранить Ethernet-адреса и номера портов коммутатора. Полученные данные сравниваются с таблицей, записанной в САМ, и если при сравнении выявлено совпадение, то выдается номер порта и пакет направляется к верному порту или адресу (рис.3).

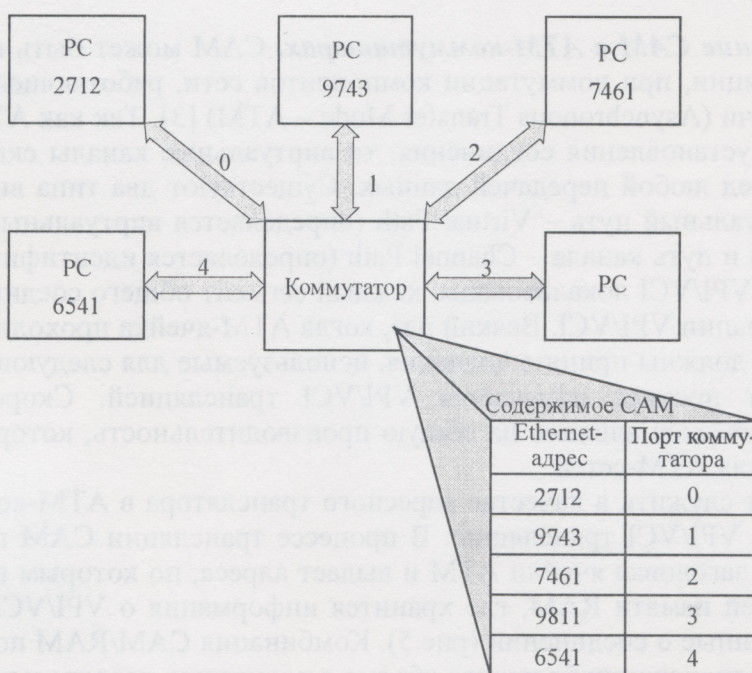


Рис. 3. Использование САМ в сетевых коммутаторах

**Использование САМ для фильтрации IP-адресов.** Фильтр IP-адресов служит для ограничения неавторизованного доступа к ресурсам локальной сети или трафика, проходящего через маршрутизатор, а также для управления доступом в Internet из локальной сети. Например, рабочим станциям локальной сети можно разрешить выполнение только специальных служб (таких как e-mail). САМ может быть использована как IP-фильтр, который блокирует все попытки доступа кроме тех случаев, когда пакетам даны исключительные права [3]. В этом случае происходит сравнение пакетов, поступающих в маршрутизатор, с типами пакетов, хранящимися в САМ. Если обнаружено совпадение, то пакету либо разрешается доступ, либо выдается отказ в доступе в соответствии с правилами фильтра (рис.4).

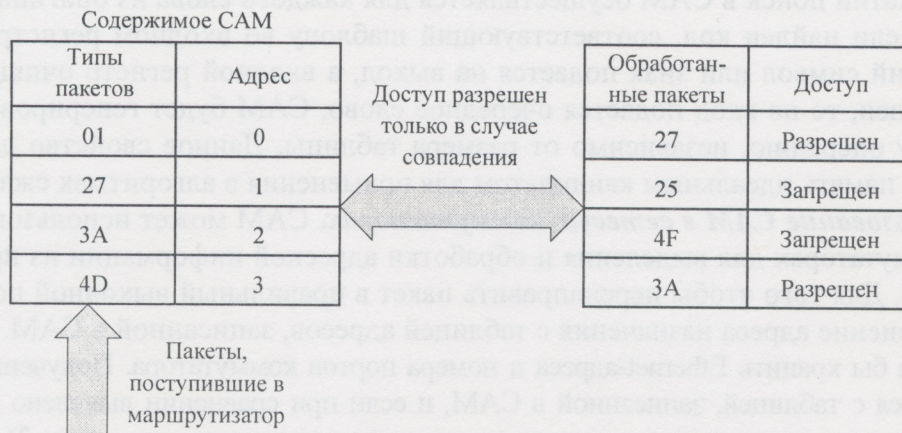


Рис. 4. Использование САМ для фильтрации IP-адресов

**Использование САМ в АТМ-коммутаторах.** САМ может быть использована как таблица трансляции, при коммутации компонентов сети, работающей в режиме асинхронной передачи (Asynchronous Transfer Mode – АТМ) [3]. Так как АТМ-сети – это сети, требующие установления соединения, то виртуальные каналы сквозь них должны создаваться перед любой передачей данных. Существуют два типа виртуальных каналов АТМ: виртуальный путь – Virtual Path (определяется виртуальным идентификатором пути – VPI) и путь канала – Channel Path (определяется идентификатором канала – VCI). Значения VPI/VCI локализованы, каждый сегмент общего соединения имеет уникальные комбинации VPI/VCI. Всякий раз, когда АТМ-ячейка проходит через коммутатор, ее VPI/VCI должны принять значения, используемые для следующего сегмента соединения. Этот процесс называется VPI/VCI трансляцией. Скорость трансляции оказывает критическое влияние на общую производительность, которая является важным фактором для АТМ-сетей.

САМ может служить в качестве адресного транслятора в АТМ-коммутаторе и быстро выполнять VPI/VCI трансляцию. В процессе трансляции САМ принимает значения VPI/VCI из заголовка ячейки АТМ и выдает адреса, по которым происходит обращение к внешней памяти RAM, где хранится информация о VPI/VCI отображении, а также другие данные о соединении (рис.5). Комбинация САМ/RAM позволяет реализовывать таблицы трансляции большого объема с полностью параллельным поиском.

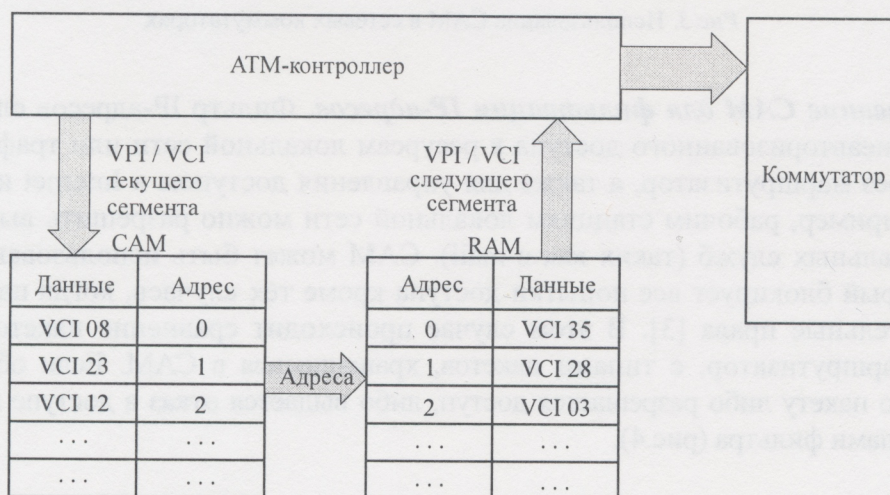


Рис. 5. Использование САМ в АТМ-коммутаторах

**Аппаратные решения для САМ.** Для САМ существует два типа аппаратной реализации: отдельные (или дискретные) компоненты и исполнение в виде систем-на-кристалле. Начиная с 1980 г. отдельные компоненты САМ производились как относительно специализированные устройства памяти. Такой подход позволяет разработчику использовать преимущества САМ, но при этом расходуется площадь на печатной плате. К тому же снижается общая производительность системы из-за внутренних задержек блока САМ, а также из-за задержек на соединениях между памятью и другими устройствами, расположенными на плате.

Реализация САМ на кристалле в виде встроенного функционального блока появилась сравнительно недавно. Примером такой реализации является использование встроенной САМ в последних поколениях программируемых логических устройств (PLD) [3]. Преимуществом такого использования САМ является эффективное использование площади и увеличение производительности. Увеличение производительности в данном случае достигается за счет сокращения длины связей на кристалле. Такие PLD-устройства называются встроенными системными блоками.

Данные в САМ могут загружаться во время конфигурации PLD-устройства или в процессе его работы.

Использование ассоциативной памяти в приложениях с интенсивным поиском информации имеет ряд существенных достоинств и дает проектировщикам возможность создания быстродействующих систем.

Результаты анализа алгоритмов поиска, основанных на использовании RAM и на САМ, позволяют утверждать, что использование САМ предпочтительнее, так как в этом случае время поиска не зависит от расположения информации в накопителе. Данное отличие делает САМ оптимальным решением для любой задачи, требующей быстрого поиска в памяти.

Важнейшей областью применения САМ являются компьютерные сети, где САМ используется для сжатия данных, маршрутизации и фильтрации IP-адресов, а также для коммутации различных сегментов ATM-сетей.

Перспективным направлением является использование САМ в базах данных, системах распознавания образов или для управления кэш-памятью в современных микропроцессорах.

Ассоциативная память реализуется как в виде отдельных микросхем, так и в качестве компонента систем-на-кристалле. Примером такой реализации может быть встраивание САМ в программируемые логические устройства.

Развитие и совершенствование технологий микроэлектроники, а также разработка новых архитектурных и схематических решений позволит существенно расширить круг использования ассоциативной памяти, добавив к уже существующим, например, и мобильные применения.

### Литература

1. BCC Research. Computer Memory: Important trends and directions. – Режим доступа: <http://www.bccresearch.com/RepTemplate.cfm?ReportID=345&cat=itt&RepDet=SC&target=repdetail.cfm>.
2. Mohan N., Sadchev M. A static power reduction technique for ternary content-addressable memories // Proc. of the IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), Niagara Falls. – May 2–5. – 2004. – P. 711–714.
3. Sherri Azgomi. Using content-addressable memory for networking applications. Altera International Ltd. – Режим доступа: <http://www.commsdesign.com/main/1999/11/9911feat3.htm>.
4. Kostas Pagiamtzis, Ali Sheikholeslami. Content-Addressable Memory (CAM) circuits and architectures: a tutorial and survey // IEEE J. of Solid-States Circuits. – Vol. 41, N 3, March 2006.

Статья поступила 22 апреля 2008 г.

**Парфененко Александр Сергеевич** – аспирант ОАО «Ангстрем-М» (г. Москва). *Область научных интересов:* статическая память, ассоциативная память, компиляторы памяти.

**Рыбачек Андрей Владимирович** – руководитель дизайн-группы Front-End проектирования ОАО «Ангстрем-М» (г. Москва). *Область научных интересов:* вычислительная техника, защита от воздействия статического электричества.

**Рыбалко Егор Сергеевич** – магистрант Института проектирования приборов и систем (Cadence) МИЭТ. *Область научных интересов:* ассоциативная память, методики уменьшения потребляемой мощности в блоках ассоциативной памяти, компиляторы памяти.