Запоминающие устройства

Запоминающие устройства (ЗУ) служат для хранения информации и обмена ею с другими ЦУ. Микросхемы памяти в общем объеме выпуска ИС занимают около 40% и играют важнейшую роль во многих системах различного назначения. Микросхемы и системы памяти постоянно совершенствуются как в области схемотехнологии, так и в области развития новых архитектур. В настоящее время созданы и используются десятки различных типов ЗУ.

Важнейшие параметры ЗУ находятся между собой в противоречии. Так, например, большая информационная емкость не сочетается с высоким быстродействием, а быстродействие в свою очередь не сочетается с низкой стоимостью. Поэтому системам памяти свойственна *многоступенчатая иерархическая структура*, и в зависимости от роли того или иного ЗУ его реализация может быть существенно различной.

В развитой иерархии памяти ЭВМ можно выделить следующие уровни:

* ***регистровые ЗУ***, находящиеся в составе процессора, благодаря которым уменьшается число обращений к другим уровням памяти, реализованным вне процессора и требующим большего времени для операций обмена информацией;
* ***кэш-память***, служащая для хранения копий информации, используемой в текущих операциях обмена. Работа процессора с кэш-памятью высокого быстродействия повышает производительность ЭВМ;
* ***основная память***(оперативная, постоянная, полупостоянная), работающая в режиме обмена с процессором и по возможности согласованная с ним по быстродействию. Исполняемый в текущий момент фрагмент программы обязательно находится в основной памяти;
* ***специализированные виды памяти***, характерные для некоторых специфических архитектур (ассоциативные, видеопамять и др.),
* ***внешняя память***, хранящая большие объемы информации. Эта память обычно реализуется на основе устройств с подвижным носителем формации (магнитные и оптические диски, магнитные ленты и др.) В настоящем пособии устройства внешней памяти не рассматриваются

**Основные параметры**

***Информационная емкость***– максимально возможный объем хранимой формации. Выражается в битах или словах (в частности, в байтах). Бит хранится запоминающим элементом (ЗЭ), а слово – запоминающей ячейкой (ЗЯ), т.е. группой ЗЭ, к которым возможно лишь одновременное обращение.

***Организация ЗУ***– произведение числа хранимых слов на их разрядность Видно, что это произведение выражает информационную емкость ЗУ, однако при одной и той же информационной емкости организация ЗУ может быть различной, так что организация является самостоятельным важным параметром, выражаемым парой чисел. Примеры организации памяти: 32x8, I28Kx8.

***Быстродействие ЗУ*** оценивают временами считывания, записи, длительностями циклов чтения/записи и другими параметрами. ***Время считывания*** – интервал между моментами появления сигнала чтения и слова на выходе ЗУ. ***Время записи*** – интервал после появления сигнала записи, достаточный для установления ЗЯ в состояние, задаваемое входным словом. Минимально допустимый интервал между последовательными повторными операциями чтения или записи образует соответствующий цикл. Длительности циклов могут превышать времена чтения или записи, т.к. после этих операций до начала следующей может потребоваться время для восстановления необходимого начального состояния ЗУ.

Времена чтения, записи и длительности циклов – традиционные параметры, достаточные для оценки быстродействия простых структур ЗУ. Для многих современных ЗУ они должны быть дополнены новыми. Причиной является более сложный характер доступа к хранимым данным, когда обращение к первому слову некоторой группы слов (страницы, пакета) требует большего времени, чем обращение к последующим. Для таких режимов вводят параметры ***времени доступа при первом обращении*** (Latency) и ***темпа передач*** для последующих слов пакета.

Применительно к ЗУ используется также параметр, называемый ***полосой пропускания*** или***производительностью***, и определяемый как произведение числа считываемых (или записываемых) в секунду слов на их разрядность. Например, ЗУ с темпом передачи слов 50 МГц при их разрядности 8 бит имеет полосу пропускания (производительность) 400 Мбит/с.

Перечисленные динамические параметры являются *эксплуатационными* (измеряемыми). Кроме них существует ряд режимных параметров, обеспечение которых необходимо для нормального функционирования ЗУ, поскольку оно имеет несколько сигналов управления, сигналы адресации данных и самих данных и для них должно быть обеспечено определенное взаимное положение во времени. Для этих сигналов задаются длительности и ограничения по взаимному положению во времени.

Важным для микросхем памяти является ***свойство энергонезависимости***, т.е. способность ЗУ сохранять данные при отключении напряжения питания. Энергонезависимость может быть *естественной*, т.е. присущей самим ЗЭ, или *искусственной*, достигаемой введением резервных источников питания, автоматически подключаемых к накопителю ЗУ при снятии основного питания или же дополнением схемы ЗУ специальными вспомогательными энергонезависимыми элементами памяти.

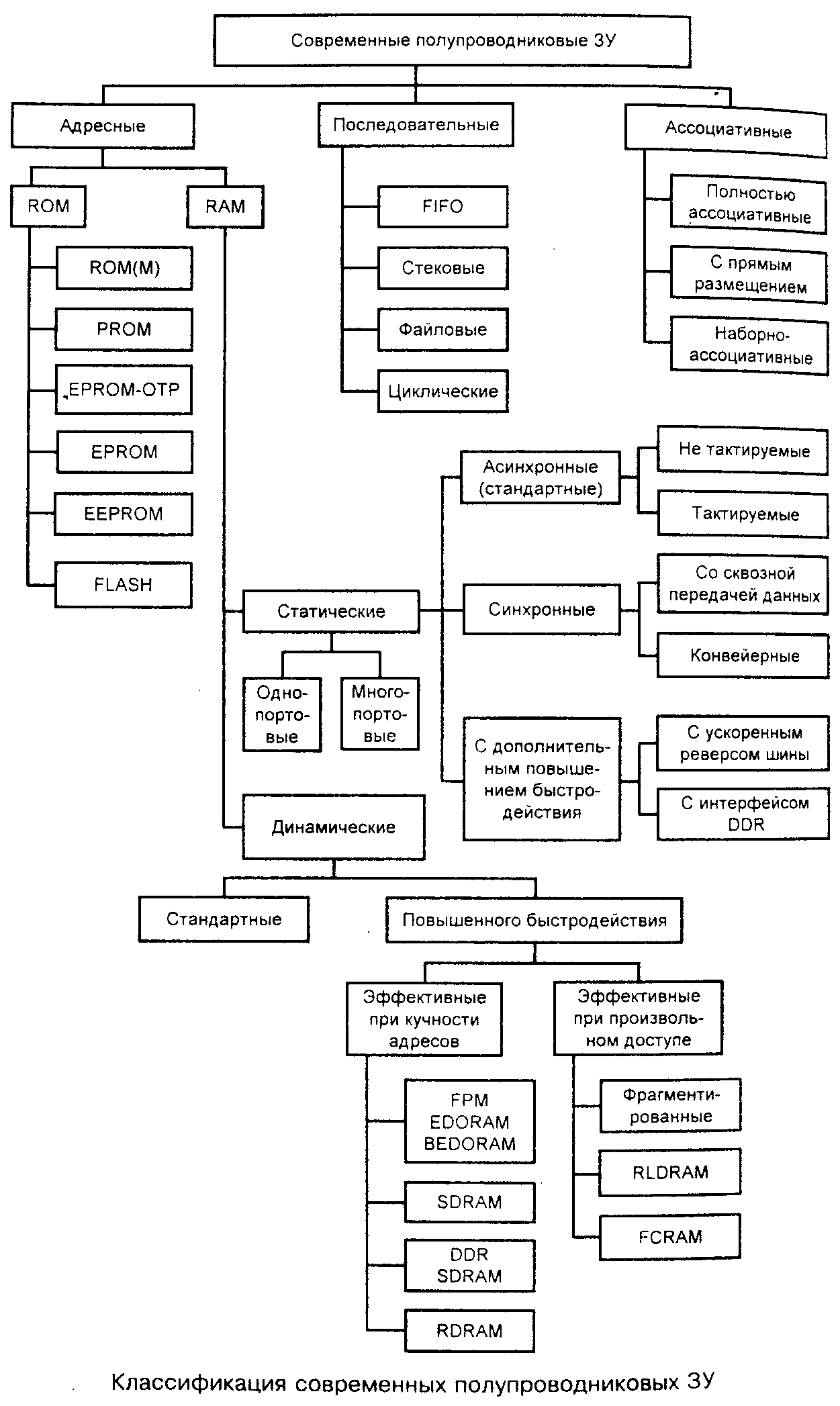
**Классификация современных ЗУ**

Для классификации ЗУ (см. рисунок на следующей странице) важнейшим признаком является способ доступа к данным.

При *адресном* доступе код на адресных входах указывает ячейку, с которой ведется обмен. Все ячейки адресной памяти в момент обращения равнодоступны. Эти ЗУ наиболее разработаны, и другие виды памяти часто строят на основе адресной с соответствующими модификациями.

***Адресные ЗУ делятся на RAM (Random Access Memory) и ROM (Read-Only Memoty).***Русские синонимы термина RAM: ОЗУ (оперативные ЗУ) или ЗУПВ (ЗУ с произвольной выборкой). Оперативные ЗУ хранят данные, участвующие в обмене при исполнении текущей программы, которые могут быть изменены в произвольный момент времени. Современные ОЗУ, как правило, не обладают энергонезависимостью (этим свойством обладают новые перспективные варианты ОЗУ, которые, возможно, вскоре начнут заменять существующие).

В ROM (русский эквивалент – ПЗУ, т.е. постоянные ЗУ) содержимое либо вообще не изменяется, либо изменяется, но редко и в специальном режиме. Для рабочего режима это "память только для чтения".



*Постоянная память*типа Mask ROM, обозначенная как ROM(M), программируется при изготовлении методами интегральной технологии с помощью масок. На русском языке ее можно назвать памятью типа ПЗУМ (ПЗУ масочные). Для потребителя это в полном смысле слова постоянная память, т.к. изменить ее содержимое он не может.

В следующих четырех разновидностях ROM в обозначениях присутствует буква Р (от Programmable) Это*программируемая пользователем память* (в русской терминологии ППЗУ). В память типов PROM и EPROM-OTP содержимое записывается однократно (ОТР означает One Time Programmable). В ЗУ типов EPROM, EEPROM и FLASH содержимое может быть изменено путем стирания старой информации и записи новой. В памяти EPROM (Erasable Programmable ROM) стирание выполняется облучением кристалла ультрафиолетовыми лучами, ее русское название РПЗУ-УФ (репрограммируемое ПЗУ с УФ-стиранием).

В EEPROM (Electrically Erasable Programmable ROM) стирание производится электрическими сигналами, ее русское название РПЗУ-ЭС (репрограммируемое ПЗУ с электрическим стиранием) или ЭСППЗУ (электрически стираемое программируемое ПЗУ). Запись данных в EPROM, EEPROM и FLASH производится электрическими сигналами.

Программирование PROM, EPROM и EEPROM производится в обычных лабораторных условиях. Для ЗУ типов PROM и EPROM это делается с помощью программаторов. Для EEPROM возможно также использование специальных режимов для программирования без изъятия микросхемы из устройства, в котором она используется. Запоминающие элементы памяти типа FLASH принципиально подобны применяемым в EPROM и EEPROM, но эта память имеет структурные и технологические особенности, позволяющие выделить ее в отдельный вид.

***RAM делятся на статические и динамические***. В статических RAM запоминающими элементами являются триггеры, сохраняющие свое состояние, пока схема находится под питанием и нет новой записи данных. В динамических RAM данные хранятся в виде зарядов конденсаторов, образуемых элементами МОП-структур. Саморазряд конденсаторов ведет к разрушению данных, поэтому они должны периодически (каждые несколько миллисекунд) регенерироваться, что усложняет эксплуатацию ЗУ. В то же время плотность упаковки элементов динамической памяти в несколько раз превышает плотность упаковки, достижимую в статических RAM, поэтому динамические ЗУ имеют намного более высокую информационную емкость и в несколько раз дешевле более быстродействующих статических.

Разработаны также ЗУ с динамическими запоминающими элементами, имеющие внутреннюю встроенную систему регенерации, у которых внешнее поведение становится аналогичным поведению статических ЗУ.

Статические ОЗУ в английской и международной терминологии называются SRAM (Static RAM), а динамические - DRAM (Dynamic RAM).

Статические ОЗУ разделены в классификации на ***асинхронные*** и ***синхронные.*** Асинхронные ОЗУ названы также ***стандартными***, т.к. до недавнего времени они были практически единственными представителями статических микро схем памяти и наиболее привычны для потребителя. В асинхронных ЗУ после произвольного по времени обращения к памяти до выдачи данных ходит определенное время, которое является параметром самой памяти не связанным с параметрами системы синхронизации процессора. Ввиду отсутствия увязки моментов обращения к памяти и моментов выработки ею готовых данных с синхросигналами процессора могут возникать дополнительные задержки обмена данными между ЗУ и процессором.

Асинхронные статические ОЗУ можно разделить на ***нетактируемые*** и ***тактируемые***. В нетактируемых сигналы управления могут задаваться как импульсами, так и уровнями. В тактируемых ЗУ некоторые сигналы обязательно должны быть импульсными (например, сигнал разрешения работы CS в каждом цикле должен переходить из пассивного состояния в активное).

В синхронных ОЗУ длительности этапов работы памяти жестко связаны с синхросигналами системы, и это позволяет исключить неоправданные потери времени при обмене данными между памятью и процессором, а также организовать конвейерную обработку данных. Таким образом, синхронность памяти является средством повышения ее быстродействия. Это важный способ повышения быстродействия, применяемый как в статических, так и в динамических микросхемах памяти.

Статические ОЗУ выполняются как ***однопортовые*** (обычные) и ***многопортовые***. Многопортовые ЗУ специализированы для определенных применении. В них возможны одновременные обращения более чем к одной ячейке, например, в двупортовых ЗУ возможно считывание информации из одной ячейки и одновременная запись в другую. Подобные режимы полезны при разделении памяти между двумя или более абонентами.

***Динамические ЗУ характеризуются наибольшей информационной емкостью и невысокой стоимостью, поэтому именно они используются как основная память ЭВМ.***Базовая структура динамических ЗУ названа стандартной. Поскольку желательно получить от основной памяти ЭВМ максимально возможное быстродействие, разработаны многочисленные способы его повышения.

Статические ЗУ в 4-5 раз дороже динамических и приблизительно во столько же раз меньше по максимально достижимой информационной ёмкости. Их достоинством является высокое быстродействие, а типичной областью использования - схемы кэш-памяти, буферы FIFO и LIFO память данных небольшой емкости для микроконтроллеров, быстродействующих коммуникационных устройств и т.п.

В ЗУ с ***последовательным доступом*** записываемые данные образуют некоторую ***очередь***. Считывание происходит из очереди слово за словом либо в порядке записи, либо в обратном порядке. Моделью такого ЗУ является последовательная цепочка запоминающих элементов, в которой данные передаются между соседними элементами.

Прямой порядок считывания имеет место в ***буферах FIFO*** с дисциплиной "первый пришел – первый вышел" (First In – First Out), а также в файловых и циклических ЗУ.

Разница между памятью FIFO и файловым ЗУ состоит в том, что в FIFO запись в пустой буфер сразу же становится доступной для чтения, т.е. слово поступает в конец цепочки (модели ЗУ). В файловых ЗУ данные поступают в начало цепочки и появляются на выходе после некоторого числа обращений, равного числу элементов в цепочке. При независимости операций считывания и записи фактическое расположение данных в ЗУ на момент считывания не связано с каким-либо внешним признаком. Поэтому записываемые данные объединяют в блоки, обрамляемые специальными символами конца и начала (файлы). Прием данных из файлового ЗУ начинается после обнаружения приемником символа начала блока.

В ***циклических ЗУ*** слова доступны одно за другим с постоянным периодом, определяемым емкостью памяти. К такому типу среди полупроводниковых ЗУ относится видеопамять (VRAM).

Считывание в обратном порядке свойственно ***стековым ЗУ***, для которых реализуется дисциплина "последний пришел – первый вышел". Такие ЗУ называют ***буферами LIFO*** (Last In – First Out).

Время доступа к конкретной единице хранимой информации в последовательных ЗУ представляет собою случайную величину. В наихудшем случае для такого доступа может потребоваться просмотр всего объема хранимых данных.

***Ассоциативный доступ***реализует поиск информации по некоторому признаку, а не по ее расположению в памяти (адресу или месту в очереди). В наиболее полной версии все хранимые в памяти слова одновременно проверяются на соответствие признаку, например, на совпадение определенных полей слов (тегов – от англ. tag) с признаком, задаваемым входным словом (теговым адресом). На выход выдаются слова, удовлетворяющие признаку. Дисциплина выдачи слов, если тегу удовлетворяют несколько слов, а также Дисциплина записи новых данных могут быть разными. Основная область применения ассоциативной памяти в современных ЭВМ – кэширование данных.

**Основные структуры запоминающих устройств**

Многочисленные варианты ЗУ имеют много общего с точки зрения структурных схем, что делает рациональным изучение некоторых обобщенных структур с последующим описанием особенностей и запоминающих элементов для конкретных ЗУ.

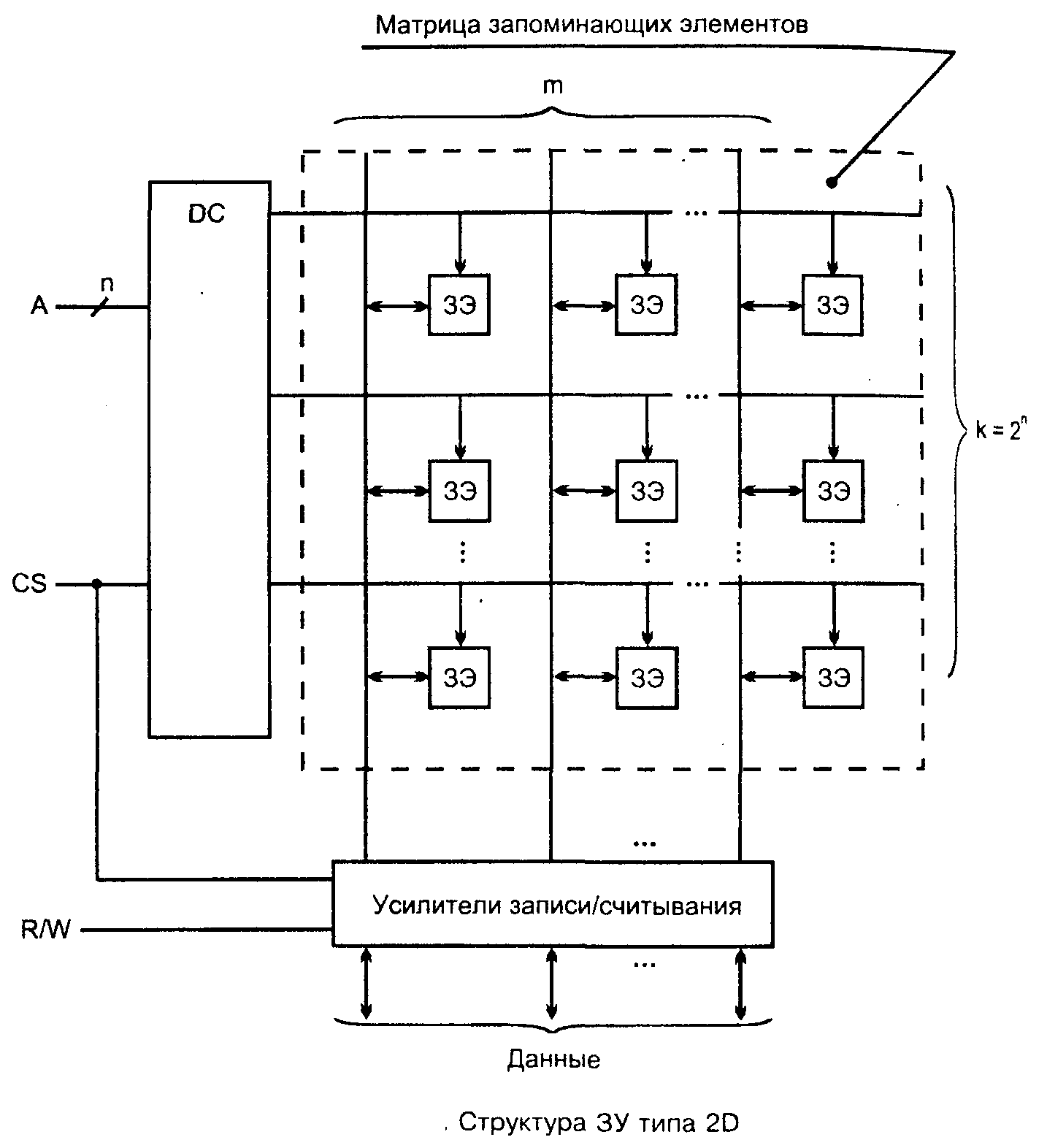
Общность структур адресных ЗУ особенно проявляется для статических ОЗУ и памяти типа ROM. Для статических ОЗУ и памяти типа ROM характерны структуры 2D, 3D, 2DM и блочные структуры на их основе.

**Структура 2D**

В структуре 2D (см. рисунок на следующей странице) запоминающие элементы ЗЭ организованы в прямоугольную матрицу размерностью *М = k x m*, где *М* – информационная емкость памяти в битах; *k* – число хранимых слов; *m* – их разрядность.

Дешифратор адресного кода DC при наличии разрешающего сигнала CS (Chip Select) активизирует одну из выходных линий, разрешая одновременный доступ ко всем элементам выбранной строки, хранящей слово, адрес которого соответствует номеру строки. Элементы столбца соединены вертикальной линией – внутренней линией данных (разрядной линией, линией записи/считывания). Элементы столбца хранят одноименные биты всех слов. Направление обмена определяется усилителями чтения/записи под воздействием сигнала R/W (Read – чтение, Write – запись).

Структура типа 2D применяется лишь в ЗУ малой информационной емкости, т.к. при росте емкости проявляется несколько ее недостатков, наиболее очевидным из которых является чрезмерное усложнение дешифратора адреса (число выходов дешифратора равно числу хранимых слов).



**Структура 3D**

Структура 3D позволяет резко упростить дешифраторы адреса с помощью двухкоординатной выборки запоминающих элементов. Принцип двухкоординатной выборки поясняется на примере ЗУ типа ROM (см. рисунок на следующей странице), реализующего только операции чтения данных.

Здесь код адреса разрядностью ***n*** делится на две половины, каждая из которых декодируется отдельно. Выбирается запоминающий элемент, находящийся на пересечении активных линий выходов обоих дешифраторов. Таких пересечений будет как раз

***2n/2 x 2n/2 = 2n***

Суммарное число выходов обоих дешифраторов составляет

***2n/2 + 2n/2 = 2n/2+1***,

что гораздо меньше, чем ***2n*** при реальных значениях ***n***.

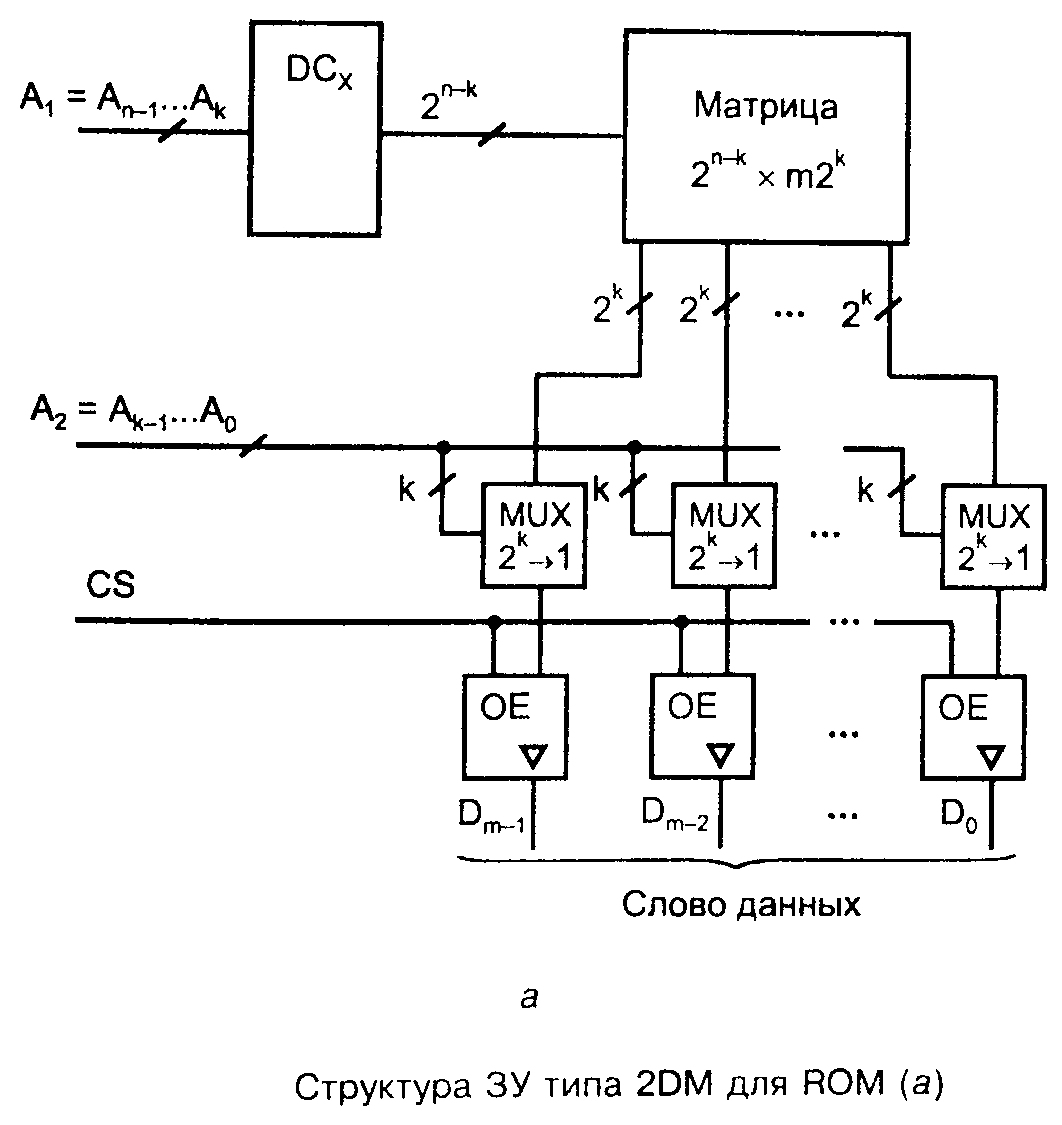


Уже для ЗУ небольшой емкости видна эта существенная разница: для структуры 2D при хранении 1К слов потребовался бы дешифратор с 1024 выходами, тогда как для структуры типа 3D нужны два дешифратора с 32 выходами каждый. Недостатком структуры 3D в первую очередь является усложнение элементов памяти, имеющих двухкоординатную выборку.

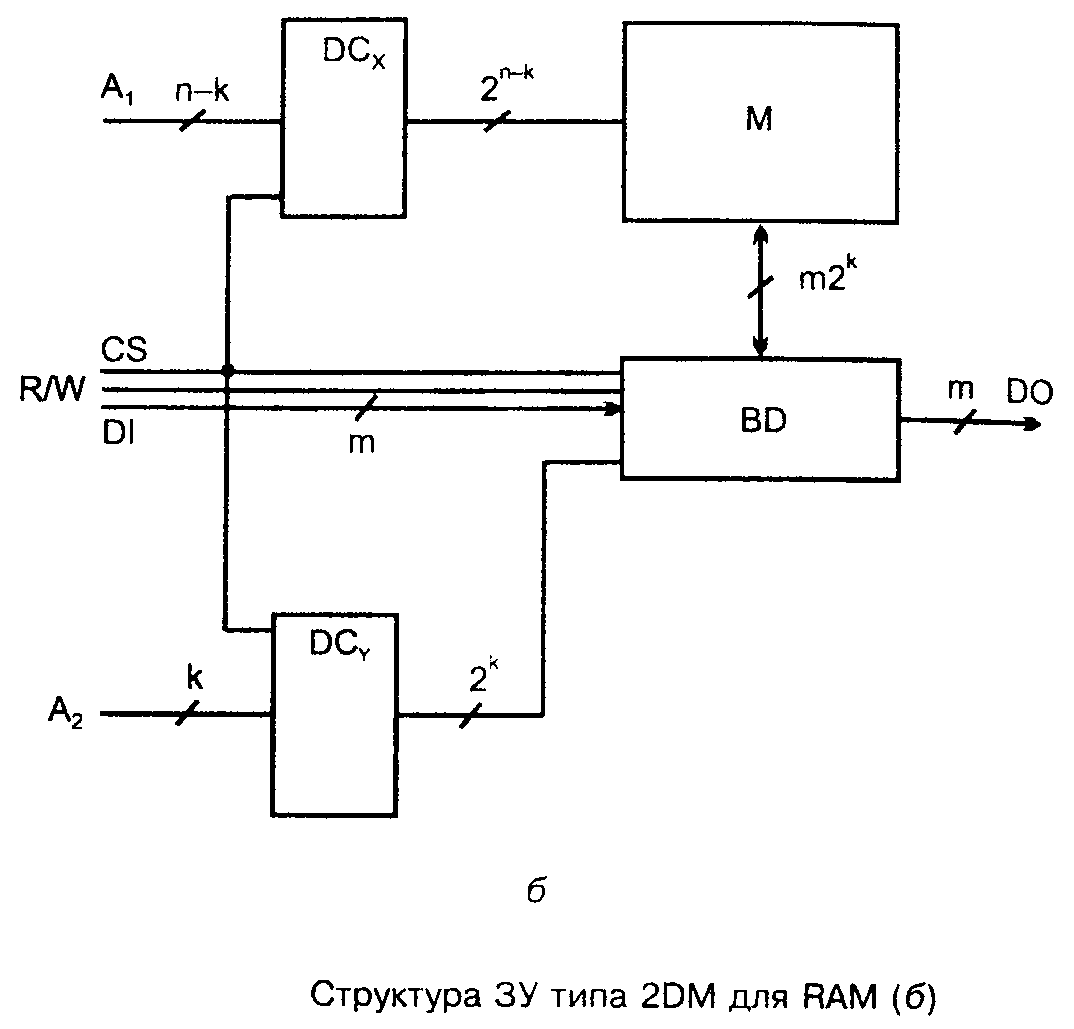
Структуры типа 3D имеют также довольно ограниченное применение, поскольку в структурах типа 2DM сочетаются достоинства обеих рассмотренных структур – упрощается дешифрация адреса и не требуются запоминающие элементы с двухкоординатной выборкой.

**Структура 2DM**

ЗУ структуры 2DM (2D модифицированная) (см. следующий рисунок) для матрицы запоминающих элементов с адресацией от дешифратора DCx имеет как бы характер структуры 2D: возбужденный выход дешифратора выбирает целую строку. Однако в отличие от структуры 2D, длина строки не равна разрядности хранимых слов, а многократно ее превышает. При этом число строк матрицы уменьшается и, соответственно, уменьшается число выходов дешифратора. Для выбора одной из строк служат не все разряды адресного кода, а их часть An-1...Ak. Остальные разряды адреса (от Аk-1 до А0) используются, чтобы выбрать необходимое слово из того множества слов, которое содержится в строке. Это выполняется с помощью мультиплексоров, на адресные входы которых подаются коды Аk-1...А0. Длина строки равна m2k, где m - разрядность хранимых слов. Из каждого "отрезка" строки длиной 2k мультиплексор выбирает один бит. На выходах мультиплексоров формируется выходное слово По разрешению сигнала CS, поступающего на входы ОЕ управляемых буферов с тремя состояниями, выходное слово передается на внешнюю шину.



На предыдущем рисунке для большей наглядности структура 2DM показана на примере ROM. На следующем рисунке структура 2DM в более общем виде показана для ЗУ типа RAM с операциями чтения и записи. Из матрицы М по-прежнему счи-тывается "длинная" строка.



Данные в нужный отрезок этой строки записываются (или считываются из нее) управляемыми буферами данных BD, воспринимающими выходные сигналы второго дешифратора DCY, и выполняющими не только функции мультиплексирования, но и функции изменения направления передачи данных под воздействием сигнала R/W.

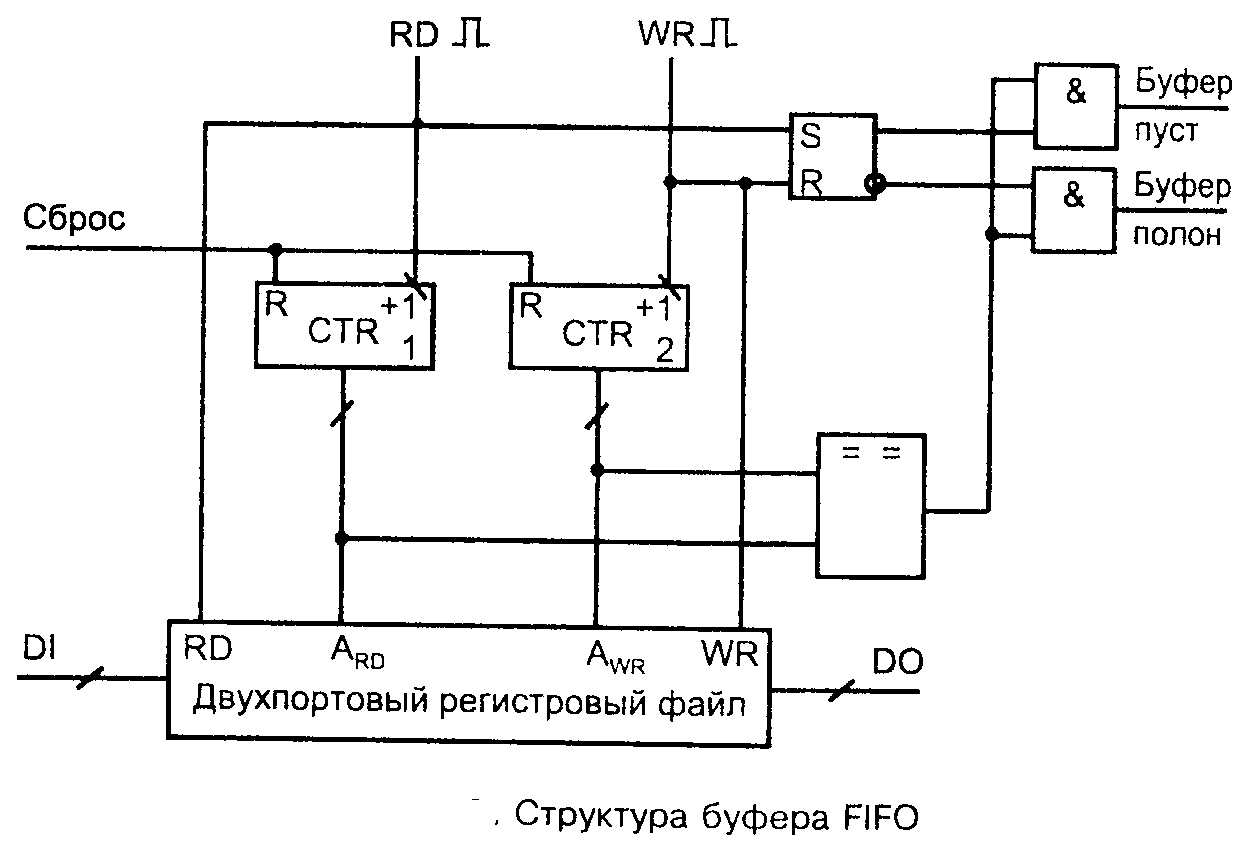
**Память с последовательным доступом**

Память с последовательным доступом строится либо с использованием продвижения данных в цепочке элементов (по подобию с регистрами сдвига). либо с хранением данных в адресном ЗУ при необходимом управлении адресом доступа.

Основными представителями этого вида памяти являются видеопамять, буфер FIFO и стек (буфер LIFO).

**Буфер FIFO**

Буфер FIFO, пример структуры которого приведен на следующем, представляет собою ЗУ для хранения очередей данных (списков) с порядком выборки слов, таким же, что и порядок их поступления. Интервалы между словами могут быть совершенно различными, т.к. моменты записи слова в буфер и считывания из него задаются внешними сигналами управления независимо друг от друга.



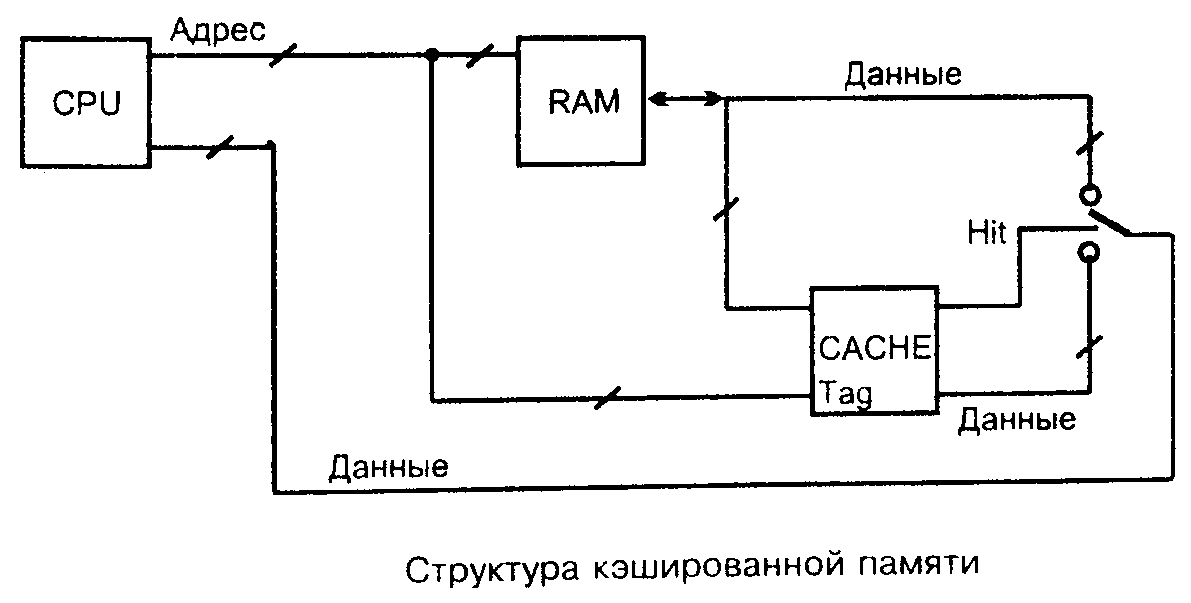
Возможность иметь разный темп приема и выдачи слов необходима, например, если приемник способен принимать данные, поступающие регулярно некоторой частотой, а источник информации выдает слова в более быстром темпе и, может быть, к тому же не регулярно. Такие данные поступают в их темпе в буфер FIFO, а из него считываются регулярно с необходимой для приемника данных частотой. Новое слово ставится в конец очереди, считывание осуществляется с начала очереди.

В схеме (см. рисунок выше) перед началом работы оба счетчика адресов CTR1, и CTR2 сбрасываются. При записи адреса увеличиваются на единицу при каждом обращении, т.е. возрастают, начиная с нулевого. То же происходит при чтении слов, так что адрес чтения всегда "гонится" за адресом записи. Если адреса сравняются при чтении, то буфер пуст. Если адреса сравняются при записи, то буфер полон (адресами занята вся емкость счетчика). Эти ситуации отмечаются соответствующими сигналами. Если буфер полон, то нужно прекратить прием данных, а если пуст, то нужно прекратить чтение. Очередь удлиняется или укорачивается в зависимости от разности чисел записанных и считанных слов. Переход через нуль осложнений не вызывает. Задачу построения стека можно решить принципиально аналогичным способом. Эта задача встречается в дальнейшем изложении при рассмотрении структуры микропроцессора.

**Кэш-память**

Кэш-память запоминает копии информации, передаваемой между устройствами (прежде всего между процессором и основной памятью). Она имеет небольшую емкость в сравнении с основной памятью и более высокое быстродействие (реализуется на триггерных элементах памяти).

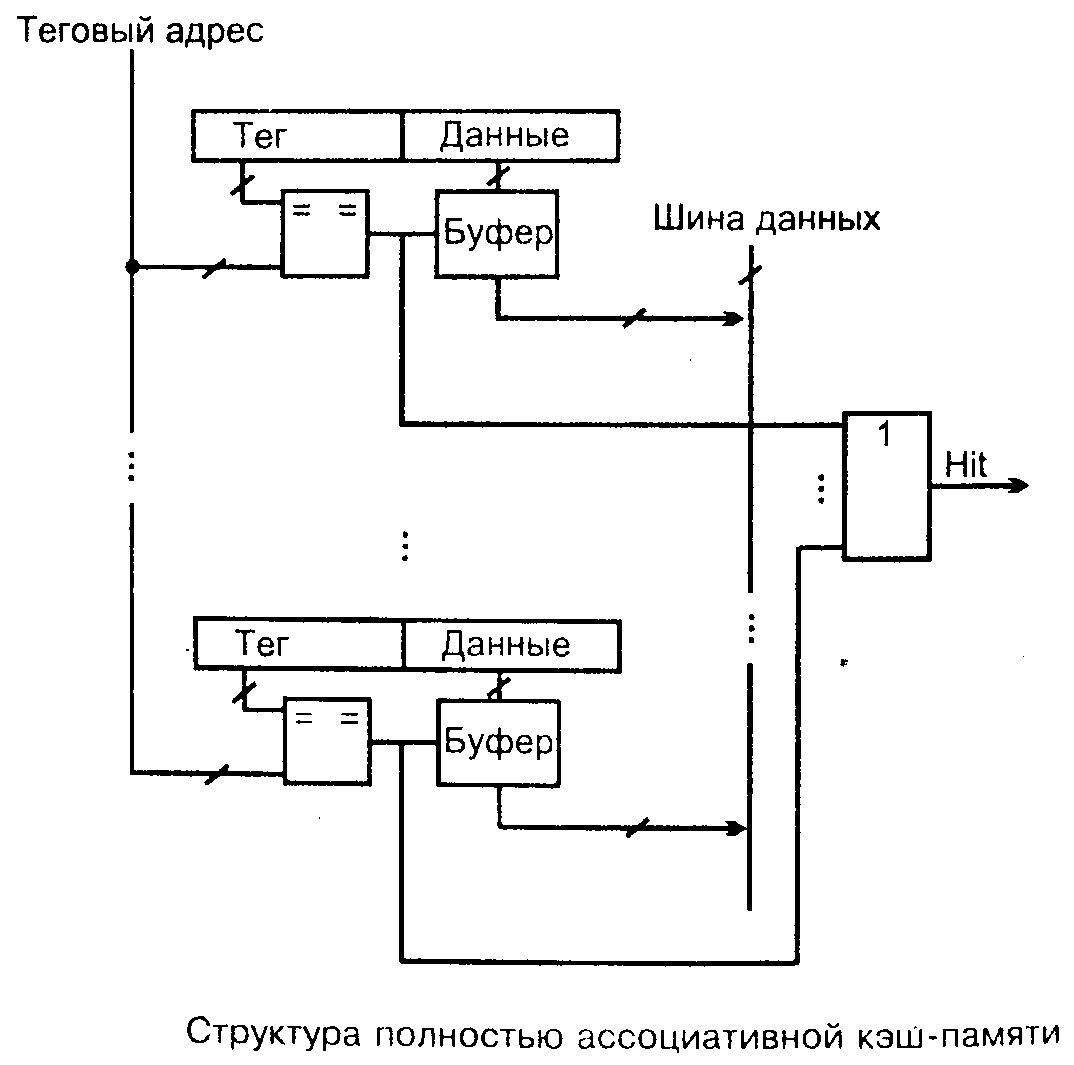
При чтении данных сначала выполняется обращение к кэш-памяти (см. следующий рисунок). Если в кэше имеется копия данных адресованной ячейки основной памяти, то кэш вырабатывает сигнал Hit (попадание) и выдает данные на общую шину данных. В противном случае сигнал Hit не вырабатывается и выполняется чтение из основной памяти и одновременное помещение считанных данных в кэш.



Эффективность кэширования обуславливается тем, что большинство прикладных программ удовлетворяют принципу локальности или, иначе говоря, имеют гнездовой характер обращений, при котором адреса последовательных обращений к памяти образуют, как правило, компактную группу. Поэтому после первого обращения к относительно медленной основной памяти повторные обращения (уже к кэшу) требуют меньше времени; К тому же при использовании процессором кэш-памяти основная память освобождается, и могут выполняться регенерация данных в динамическом ЗУ или использование памяти другими устройствами.

Объем кэш-памяти много меньше емкости основной памяти и любая единица информации, помещаемая в кэш, должна сопровождаться дополнительными данными (тегом), определяющими, копией содержания какой ячейки основной памяти она является.

***В полностью ассоциативной***кэш-памяти (FACM, Fully Associated Cache Memory), см. рисунок на следующей странице, каждая ячейка хранит данные, а в поле "Тег" – полный физический адрес информации, копия которой записана. При любых обменах физический адрес запрашиваемой информации сравнивается с полями "Тег" всех ячеек и при совпадении их в любой ячейке устанавливается сигнал Hit.



При чтении и значении сигнала Hit = 1 данные выдаются на шину данных, если же совпадений нет (Hit = 0), то при чтении из основной памяти данные вместе с адресом помещаются в свободную или наиболее давно не используемую ячейку кэш-памяти.

При записи данные вместе с адресом сначала, как правило, размещаются в кэш-памяти (в обнаруженную ячейку при Hit = 1 и свободную при Hit = 0) Копирование данных в основную память выполняется под управлением специального контроллера, когда нет обращений к памяти.

Вопрос №12