Министерство образования и науки Российской Федерации
Государственное образовательное учреждение высшего профессионального образования
“ПЕНЗЕНСКАЯ ГОСУДАРСТВЕННАЯ ТЕХНОЛОГИЧЕСКАЯАКАДЕМИЯ”

КАФЕДРА “ВЫЧИСЛИТЕЛЬНЫЕ МАШИНЫ И СИСТЕМЫ”

Р.А. Бикташев, Н.И. Чернышев, М.Н. Шмокин

Организация ЭВМ

Учебное пособие для бакалавров
по направлению 230100
“Информатика и вычислительная техника ”

Пенза 2012

УДК 681.3 (075)
Б33
Рецензенты:
Доктор технических наук, профессор кафедры ВТ
Пензенского государственного университета
П.П. Макарычев;
Доктор технических наук, профессор, заведующий кафедрой ВМиС
Пензенской государственной технологической академии
И.И. Сальников
Бикташев Р.А., Чернышев Н.И., Шмокин М.Н.
Организация ЭВМ: Учеб. пособие. – Пенза: Издательство ЦНТИ, 2012.- 125с.

Учебное пособие посвящено вопросам организации структур и функционирования ЭВМ. Рассмотрены структуры и функционирование классических фон- неймановских машин, принципы организации процессоров, памяти, шин, систем прерывания, ввода-вывода, внешних запоминающих устройств, методы повышения производительности процессоров. Приведены примеры структур и реализации основных систем современных ПК.

Учебное пособие одобрено и рекомендовано методическим советом Пензенской государственной технологической академии для подготовки бакалавров по направлению 230100 «Информатика и вычислительная техника» .

УДК 681.3 (075)

1 ОБЩИЕ СВЕДЕНИЯ О ЭВМ

1.1 Этапы развития ЭВМ

Идея использования программного управления для построения устройств, автоматически выполняющих арифметические вычисления, была впервые высказана английским математиком Ч. Бэббиджем в 1833 г. Однако его попытки построить механическое вычислительное устройство с программным управлением не увенчались успехом.

Фактически эта идея была реализована спустя более чем 100 лет, когда в 1942 г. К. Цюзе в Германии и в 1944 г. Г. Айкен в США построили вычислительные машины на электромагнитных реле с управлением от перфоленты, на которую записывалась программа вычислений.

Идея программного управления вычислительным процессом была существенно развита американским математиком Джорджем фон Нейманом, который в 1945 г. сформулировал принцип хранимой в памяти программы. Первые ЭВМ с программным управлением и с хранимой в памяти программой появились практически одновременно в Англии, США и СССР.

На протяжении более шести десятилетий электронная вычислительная техника бурно развивается. Появились, сменяя друг друга, несколько поколений ЭВМ. Появление новых поколений ЭВМ вызывалось расширением областей и развитием методов их применения, требовавших более производительных, более дешевых и более надежных машин.

Поколение ЭВМ определяется совокупностью взаимосвязанных и взаимообусловленных существенных особенностей и характеристик, используемых при построении машин, конструктивно-технологической (в первую очередь элементной) базы и реализуемой в машине архитектуры.

Первое поколение образовали ламповые ЭВМ, промышленный выпуск которых начался в начале 50-х гг. В качестве компонентов логических элементов использовались электронные лампы. ЭВМ этого поколения характеризовались низкой надежностью и высокой стоимостью. Их быстродействие составляло всего 5 ÷ 8 тыс. опер/с.

Второе поколение ЭВМ появилось в конце 50-х годов. Элементной базой второго поколения ЭВМ были полупроводниковые приборы, благодаря чему повысилась их надежность, а производительность возросла до 30 тыс. опер/с. В рамках ЭВМ 2-го поколения академик Лебедев С.А. создал ЭВМ БЭСМ-6 с производительностью до 1 млн. опер/с.

С середины 60-х годов отсчитывается начало появления ЭВМ 3-го поколения. Их элементной базой стали интегральные микросхемы (ИМС). В рамках этого поколения фирма IBM создала систему машин IBM-360, в которых был использован ряд новых достижений в области вычислительной техники. Машины серии IBM-360, а затем и IBM-370, получили широкое распространение в мире. К этому времени в Пензенском научно- исследовательском институте математических машин (ныне АО “Рубин”) была разработана ЭВМ примерно такого же класса - Урал-16, однако заметное отставание СССР в области элементной базы не могло не сказаться на характеристиках отечественных ЭВМ. Поэтому правительством было принято решение о переходе на производство техники, разработанной фирмой IBM. В СССР она выпускалась под названием Единая Система ЭВМ (ЕС ЭВМ). Наиболее быстродействующая ЭВМ из этого ряда- ЕС 1065 выпускалась заводом ВЭМ (г. Пенза). Она выполняла до 5 млн. опер/с.

Конструктивно-технологической основой ЭВМ четвертого поколения являются большие (БИС) и сверхбольшие (СБИС) ИМС.

К четвертому поколению относятся реализованные на СБИС такие новые средства вычислительной техники, как микропроцессоры и создаваемые на их основе микро-ЭВМ и микропроцессорные контроллеры. Микропроцессоры и микро-ЭВМ нашли широкое применение в устройствах и системах автоматизации измерений, обработки данных и управления технологическими процессами, при построении различных специализированных цифровых устройств и машин.

Вычислительные возможности микро-ЭВМ оказались достаточными для создания на их основе в рамках ЭВМ четвертого поколения, нового по ряду эксплуатационных характеристик и способу использования типа вычислительных устройств - персональных компьютеров (ПК), получивших в настоящее время широкое распространение.

К четвертому поколению относятся также многопроцессорные вычислительные системы, имеющие быстродействие в несколько сотен миллионов, или даже миллиард операций в секунду. К этому же поколению относятся управляющие комплексы на их основе с повышенной живучестью и надежностью, получаемых путем автоматической реконфигурации при выходе из строя одного или нескольких процессоров или других устройств.

Примером ранних отечественных вычислительных систем, которые следует отнести к четвертому поколению, является многопроцессорный комплекс «Эльбрус-2» с суммарным быстродействием до 100 млн. опер/с. В центральном процессоре комплекса была реализована нетрадиционная система команд, приближенная к языкам высокого уровня. Представление программ осуществлялось в виде обратной польской записи. Для обработки программ применялся магазинный (стековый) механизм организации вычислений и обращений к памяти программ и данных.

В 90-е годы прошлого века определились контуры нового, пятого поколения ЭВМ. В значительной степени этому способствовали публикации сведений о проекте ЭВМ пятого поколения ведущих японских фирм, поставившими перед собой цель захвата в 90-х годах японской промышленностью мирового лидерства в области вычислительной техники. Поэтому этот проект часто называют “японским вызовом”. Согласно этому проекту ЭВМ и вычислительные системы пятого поколения, помимо более высокой производительности и надежности при более низкой стоимости, должны обладать качественно новыми свойствами. В первую очередь к ним относятся возможность взаимодействия с ЭВМ при помощи человеческой речи и графических изображений, способность системы обучаться, производить ассоциативную обработку информации, делать логические суждения, вести “разумную” беседу с человеком в форме вопросов и ответов. Вычислительные системы пятого поколения должны также “понимать” содержимое базы данных, которая при этом превращается в “базу знаний”, и использовать эти “знания” при решении задач. В настоящее время исследования по подобным проблемам ведутся и в России.

1.2 Характеристики ЭВМ

Важнейшими характеристиками ЭВМ являются быстродействие и производительность. Эти характеристики тесно связаны. Быстродействие характеризуется числом команд B, выполняемых ЭВМ за одну секунду. Быстродействие можно вычислить как

$$B = 1/ t_{K,}$$

где t_K – время выполнения одной машинной команды. Однако процессор ЭВМ выполняет множество различных команд, время выполнения которых значительно отличается друг от друга. Поэтому быстродействие оценивается обычно по времени выполнения самой короткой команды.

Команды в процессоре выполняются по тактам, причём длительность такта t_T зависит от частоты тактового генератора f_T и определяется по формуле

$$t_T = 1/f_T.$$

Если считать, что на выполнение машинной команды затрачивается w машинных тактов, то быстродействие можно определить как

$$B = f_T / w \qquad (1)$$

Формула (1) является основной для вычисления быстродействия.

Одной из единиц измерения быстродействия была и остается величина, измеряемая в MIPS (Million Instructions Per Second — миллион операций в секунду). В качестве операций здесь обычно рассматриваются наиболее короткие операции типа сложения.

При решении научно-технических задач в программах превалируют операции с плавающей точкой, поэтому в этом случае используется характеристика быстродействия, выраженная в MFLOPS (Million Floating Point Operations Per Second — миллион операций с плавающей запятой в секунду).

Производительность — это число эталонных программ, выполняемых ЭВМ в единицу времени.

$$P=1/T,$$

где Т- время выполнения эталонной программы, которое можно вычислить по формуле

$$T = \sum_{i=1}^{h} m_i t_i ,$$

где m_i- число команд i-го типа с временем выполнения t_i, h- число типов команд в эталонной программе. Тогда

$$P = 1 / \sum_{i=1}^{h} m_i t_i \qquad (2)$$

Таким образом, производительность зависит от характера эталонной программы, т.е. сколько в ней команд различного типа: сложения, умножения, логических операций и др., имеющих разное время выполнения.

В настоящее время применяется характеристика эффективности *E*, удобная при сравнении разных ЭВМ. Наиболее употребительное название этой характеристики - "стоимость-производительность"

$$E = S / P, \qquad (3)$$

где *S*- стоимость ЭВМ.

Несмотря на простоту формул (1), (2) и (3), определение характеристик быстродействия, производительности и эффективности представляет собой очень сложную задачу, поскольку в формулах в явном виде не учтены архитектурные параметры ЭВМ.

Для более точных комплексных оценок существуют тестовые наборы, которые можно разделить на три группы:

-наборы тестов фирм-изготовителей для оценивания качества собственных изделий (например, компания *Intel* для своих микропроцессоров ввела показатель *iCOMP - Intel Comparative Microprocessor Performance);*

-стандартные универсальные тесты для ЭВМ, предназначенных для крупномасштабных вычислений (например, пакет математических задач *Linpack*, по которому ведется список TOP 500, включающий 500 самых производительных компьютерных установок в мире);

-специализированные тесты для конкретных областей применения компьютеров (например, для тестирования ПК по критериям офисной группы приложений используется тест *Winstone 97-Business*, для группы «домашних компьютеров» — *WinBench 97-CPUMark 32*, а для группы ПК для профессиональной работы — *3D WinBench 97-UserScene*).

Результаты оценивания ЭВМ по различным тестам несопоставимы. Наборы тестов и области применения компьютеров должны быть адекватны. Наиболее применительной практикой для оценки производительности стало использование некоторого набора специально подобранных, реально используемых прикладных программ. Подбором таких приложений занимается некоммерческая организация под названием *System Performance Evaluation Corporation (SPEC)*. Она публикует списки программ для различных прикладных областей и результаты тестирования многих имеющихся на рынке моделей компьютеров. Тесты постоянно обновляются. Так для компьютеров общего назначения тест трижды модифицировался, а самая новая версия была опубликована в 2000 году.

Коэффициент производительности *SPEC* вычисляется по формуле:

$$SPEC = T_E / T_T ,$$

где T_E - время выполнения программы на эталонном компьютере, T_T – время выполнения программы на тестируемом компьютере. В качестве эталонного компьютера выбирается один из серийно выпускаемых компьютеров. Так для теста *SPEC 2000* в качестве эталонного применяется рабочая станция UltraSPARC10 с тактовой частотой процессора 300 МГц.

Для проведения полного тестирования по очереди компилируются и выполняются все программы из списка SPEC, а затем вычисляется среднее геометрическое полученных результатов

$$SPEC = \left[\prod_{i=1}^{n} SPECi \right]^{1/n} ,$$

где *n* - количество программ в тестовом наборе.

Таким образом, *SPEC*– коэффициент характеризует результат суммарного влияния всех факторов, от которых зависит производительность компьютера: компилятора, операционной системы, процессора и памяти.

Другой важнейшей характеристикой ЭВМ является емкость запоминающих устройств. Она измеряется количеством структурных единиц информации, которые одновременно можно разместить в памяти. Этот показатель позволяет определить, какой набор программ и данных может быть одновременно размещен в памяти.

Обычно отдельно характеризуют емкость оперативной памяти и емкость внешней памяти. Современные персональные ЭВМ могут иметь емкость оперативной памяти, равную 64- 256 Мбайтам и даже больше. Этот показатель очень важен для определения, какие программные приложения могут обрабатываться в машине.

Емкость внешней памяти зависит от ее типа. Емкость жесткого диска достигает нескольких десятков Гбайтов, емкость компакт-диска (*CD-ROM*) — сотни Мбайтов (640 Мбайт и выше). Емкость внешней памяти характеризует объем программного обеспечения и отдельных программных продуктов, которые могут устанавливаться в ЭВМ. Например, для установки операционной среды *Windows XP* (2002 г.) требуется объем памяти жесткого диска более 1,5 Гбайт и не менее 128 Мбайт оперативной памяти ЭВМ.

Надежность — это способность ЭВМ при определенных условиях выполнять требуемые функции в течение заданного времени (стандарт *ISO* – международной организации стандартов - 2382/14-78).

Точность — возможность различать почти равные значения (стандарт *ISO* — 2382/2-76). Точность получения результатов обработки в основном определяется разрядностью ЭВМ, которая в зависимости от класса ЭВМ может составлять 32, 64 и 128 двоичных разрядов.

Во многих применениях ЭВМ не требуется большой точности, например при обработке текстов и документов, при управлении технологическими процессами. В этом случае достаточно воспользоваться 8- и 16-разрядными двоичными кодами. При выполнении же сложных математических расчетов следует использовать высокую разрядность (32, 64 и даже более). Программными способами диапазон представления и обработки данных может быть увеличен в несколько раз, что позволяет достигать очень высокой точности.

Достоверность — свойство информации быть правильно воспринятой. Достоверность характеризуется вероятностью получения безошибочных результатов. Заданный уровень достоверности обеспечивается аппаратно-программными средствами контроля самой ЭВМ. Возможны методы контроля достоверности путем решения эталонных задач и повторных расчетов. В особо ответственных случаях проводятся контрольные решения на других ЭВМ и сравнение результатов.

1.3 Классификация средств ЭВТ

По виду представления обрабатываемой информации электронная вычислительная техника разделяется на аналоговую и цифровую.

В аналоговых вычислительных машинах (АВМ) обрабатываемая информация представляется аналоговыми значениями величин: тока, напряжения, угла поворота некоторого механизма и т.п. АВМ обеспечивают высокое быстродействие, но не высокую точность вычислений (0,001 — 0,01). Подобные машины мало распространены. Они используются в основном в проектных и научно-исследовательских учреждениях в составе стендов по отработке новых образцов техники. По назначению их можно рассматривать как специализированные вычислительные машины.

В настоящее время под словом ЭВМ обычно понимают цифровые вычислительные машины (ЦВМ), в которых числа и другая информация представляются в виде двоичных кодов. Именно эти машины из-за их универсальности являются самой массовой вычислительной техникой.

По быстродействию ЭВМ можно разделить на:

- суперЭВМ для решения крупномасштабных вычислительных задач, для обслуживания крупнейших информационных банков данных;
- большие ЭВМ для комплектования вычислительных центров различного уровня, а также для управления сложными производственными и технологическими процессами. ЭВМ этого типа могут использоваться и для управления распределенной обработкой информации в качестве сетевых серверов;
-персональные и профессиональные ЭВМ, позволяющие удовлетворять индивидуальные потребности пользователей. На базе этого класса ЭВМ строятся автоматизированные рабочие

места (АРМ) для специалистов различного уровня;

-встраиваемые микроконтроллеры, осуществляющие автоматизацию управления отдельными устройствами и механизмами.

С развитием сетевых технологий все больше начинает использоваться другой классификационный признак, отражающий место и роль ЭВМ в сети, а именно:

-мощные машины и вычислительные системы для управления сетевыми хранилищами информации;

-кластерные структуры;

-серверы;

-рабочие станции;

-сетевые компьютеры.

Мощные машины и вычислительные системы предназначаются для обслуживания крупных сетевых банков данных и банков знаний. По своим характеристикам их можно отнести к классу суперЭВМ, но в отличие от них они являются более специализированными и ориентированными на обслуживание больших потоков информации.

Кластерные структуры представляют собой многомашинные распределенные вычислительные системы, объединяющие несколько серверов. Это позволяет гибко управлять ресурсами сети, обеспечивая необходимую производительность, надежность, готовность и другие характеристики.

Серверы - это вычислительные машины и системы, управляющие определенным видом ресурсов сети. Различают файл-серверы, почтовые, коммуникационные, *Web*-серверы и др.

Рабочая станция ориентированна на работу профессиональных пользователей с сетевыми ресурсами. Персональная ЭВМ отличается от рабочей станции тем, что функционирует обычно в автономном режиме и предназначена в основном для непрофессиональных пользователей.

Сетевые компьютеры представляют собой упрощенные персональные компьютеры, вплоть до карманных ПК. Их основным назначением является обеспечение доступа к сетевым информационным ресурсам. Вычислительные возможности у них достаточно низкие.

1.4 Структуры ЭВМ

1.4.1 Обобщенная структура ЭВМ

Обобщенная структура ЭВМ приведена на рисунке 1.1. В состав ЭВМ входят: запоминающие устройства (ЗУ), процессор, устройства ввода и вывода (УВВ).

Процессор предназначен для обработки информации и управления ЭВМ в целом. Он состоит из 2-х частей: УУ - устройство управления (управляющий автомат), и АЛУ - арифметико-логическое устройство (операционный автомат). Обработку информации процессор осуществляет под управлением программы, хранящейся в запоминающем устройстве (ЗУ), которое является памятью ЭВМ.

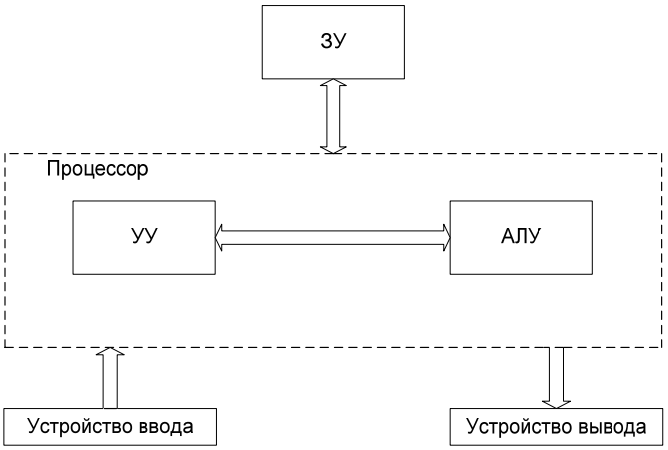


Рисунок 1.1- Обобщенная структура ЭВМ

Память вычислительной машины является многоуровневой и состоит из оперативной памяти (ОП) и внешней памяти (ВП или ВЗУ). Оперативная память является основной памятью машины. При выключенном питании информация в ОП не сохраняется, поэтому для выполнения программы ее копия и необходимых для нее данных загружаются перед началом работы из внешней памяти. Внешняя память хранит программы и данные все время их существования, т.е. до момента их удаления пользователем с внешней памяти.

Устройства ввода предназначены для ввода программ и данных в ЭВМ, а также для осуществления запуска программ на обработку пользователем. Устройства вывода используются для выдачи результатов выполнения программ пользователю в виде текстовых, графических документов или в ином виде.

Все устройства ЭВМ соединены с процессором с помощью связей (шин), показанных на рисунке линиями со стрелками. Однако реальное соединение устройств значительно отличается от расположения их в обобщенной структуре. Так внешняя память функционально относится к памяти ЭВМ, но электрически подключается подобно устройствам ввода-вывода, с которыми она образует единое пространство, называемое периферией.

От структурной организации, т.е. от того, какие устройства входят в состав ЭВМ и как они соединены между собой, зависят многие её качества. При самом общем подходе можно говорить о двух типах структур, одна из которых основана на использовании единственной общей шины, другая на множестве шин (иерархии шин).

1.4.2 Структура ЭВМ на основе общей шины

При организации ЭВМ на основе общей шины (ОШ) взаимодействие между ее устройствами осуществляется через общую шину, к которой подключены все устройства, входящие в состав ЭВМ- процессор, ОП, устройства ввода – вывода (УВВ) и внешние запоминающие устройства (ВЗУ).

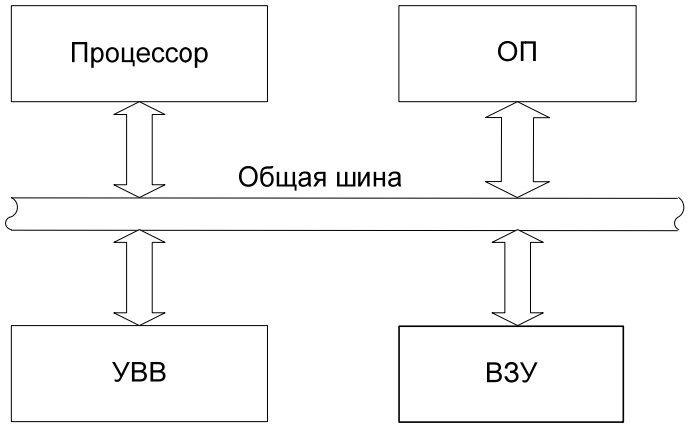
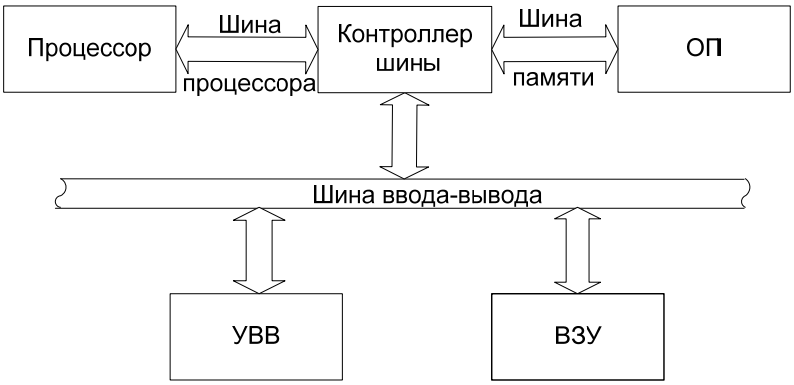


Рисунок 1.2 - Структура ЭВМ на основе ОШ

Взаимодействие между всеми устройствами ЭВМ осуществляется в режиме разделения времени общей шины (т.е. поочередно). Общая шина не обеспечивает высокой пропускной способности, что ограничивает число подключаемых устройств и общую производительность ЭВМ. Однако простота реализации обеспечили широкое использование такой структуры в ранних мини-ЭВМ и персональных компьютерах, а также в контроллерах - небольших специализированных микропроцессорных системах, предназначенных для управления производственными и бытовыми устройствами и приборами.

1.4.3 Структура ЭВМ на основе множества шин

По такому принципу построены современные компьютеры. На рисунке 1.3. показана 2-х шинная структура ЭВМ, в которой выделена одна шина для памяти, а вторая шина используется для подключения устройств ввода- вывода.



1.3 - Структура ЭВМ на основе множества шин

Поскольку общая шина может работать на частоте самого медленного устройства, подключённого к ней, а память и периферийные устройства значительно отличаются скоростными характеристиками (быстродействие памяти намного выше быстродействия УВВ и ВЗУ), поэтому разделение шин является логичным. При 2-х шинной организации низкоскоростные устройства не ограничивают скорость обмена высокоскоростных устройств, при этом шина памяти должна обладать более высокой пропускной способностью, чем шина ввода- вывода.

В некоторых компьютерах число шин достигает трех и даже более, причем они образуют иерархию. Одна шина выделяется для низкоскоростных устройств типа принтеров, модемов, другая шина, более скоростная, для высокоскоростных периферийных устройств типа магнитных и оптических дисков, графических адаптеров, и третья шина, наиболее быстродействующая, используется для взаимодействия процессора с памятью. На вершине иерархии находится шина памяти. К ней через блок сопряжения (мост) подключают высокоскоростную периферийную шину, к которой, в свою очередь, через другой мост подключают шину ввода-вывода. Подобную архитектуру ЭВМ называют мезанинной (т.е. с надстройкой). Она характерна для большинства современных ЭВМ, в том числе и компьютеров на основе процессоров *Pentium.*

В больших машинах для организации ввода - вывода используются специализированные процессоры, которые часто называют каналами или процессорами ввода-вывода (см. рисунок 1.4).

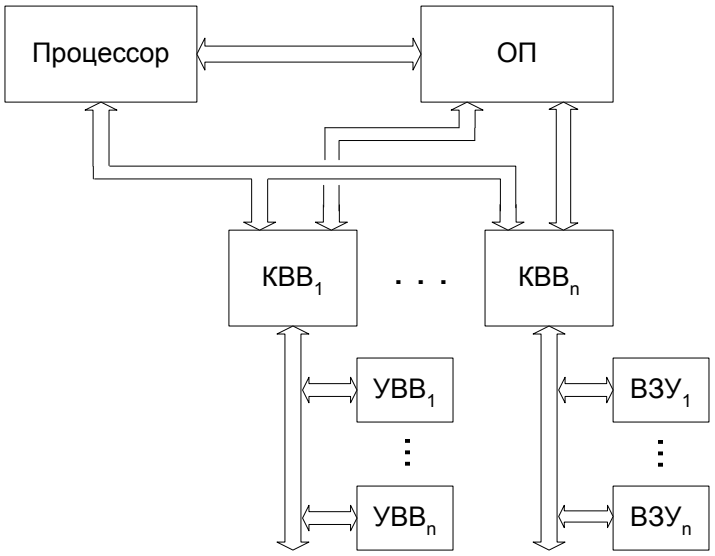


Рисунок 1.4 - Структура ЭВМ на основе каналов ввода-вывода

Каналы получают команды и параметры ввода- вывода от центрального процессора, после чего работают с периферийными устройствами самостоятельно. Процессор в это время может выполнять программы пользователя, обращаясь к памяти по высокоскоростной шине. Таким

образом, достигается параллельная работа процессора и периферийных устройств. Каналы ввода-вывода имеют собственные шины для подключения к основной памяти. При такой организации ЭВМ ОП должна быть либо многовходовой, либо иметь необходимые коммутаторы для подключения множества устройств.

В многомашинных и многопроцессорных системах применяют другие более высокоскоростные коммуникационные схемы для объединения процессоров, памяти и периферии.

1.5 Контрольные вопросы

1. Назовите основные этапы развития ЭВМ.
2. По каким признакам различают поколения ЭВМ?
3. Дайте определения быстродействия и производительности ЭВМ.
4. Расшифруйте термины MIPS и MFLOPS.
5. Какое назначение ОП и УВВ в ЭВМ?
6. По каким признакам классифицируются ЭВМ?
7. В чем различие структур ЭВМ на основе множества шин и общей шины?
8. В чем различие структур ЭВМ на основе множества шин и каналов ввода-вывода?
9. Каково назначение процессора в ЭВМ?
10. Перечислите достоинства ЭВМ на основе множества шин.
11. Недостатки ЭВМ на основе общей шины?

2. АРХИТЕКТУРА КЛАССИЧЕСКОГО ПРОЦЕССОРА

2.1 Обобщенная структура классического процессора с непосредственными и магистральными связями

Основными функциями процессора являются:

- организация обращений в ОП за командами и операндами;
- дешифрация и выполнение команд;
- инициация работы периферийных устройств;
- обработка внешних сигналов (запросы прерываний, прямого доступа в память и др.), поступающих из устройств машины.

Обобщенная структура процессора с непосредственными связями между его блоками приведена на рисунке 2.1. Такие процессоры имеют потенциально высокое быстродействие, поскольку пропускная способность множества шин велика. Однако большое число шин ограничивают возможность интеграции аппаратуры процессора в кристалл, поэтому они находят применение только в больших ЭВМ.

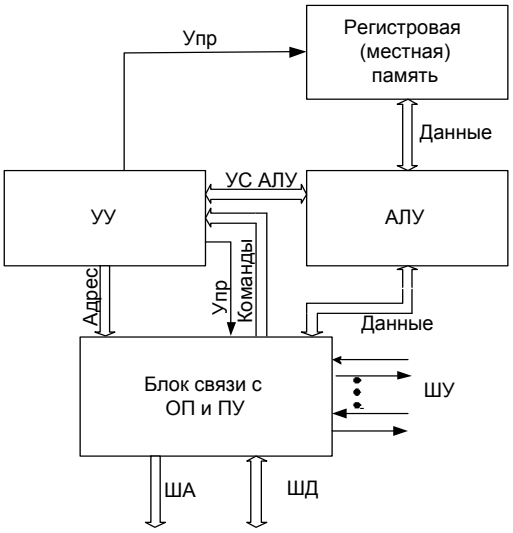


Рисунок 2.1 - Структура процессора с непосредственными связями

Структура процессора с магистральными связями между его блоками приведена на рисунке 2.2. Взаимодействие между блоками процессора производятся по общей магистрали, пропускная способность которой невелика, что ограничивает скорость передачи данных, и, следовательно, быстродействие процессора. Такая организация используется в большинстве современных микропроцессоров.

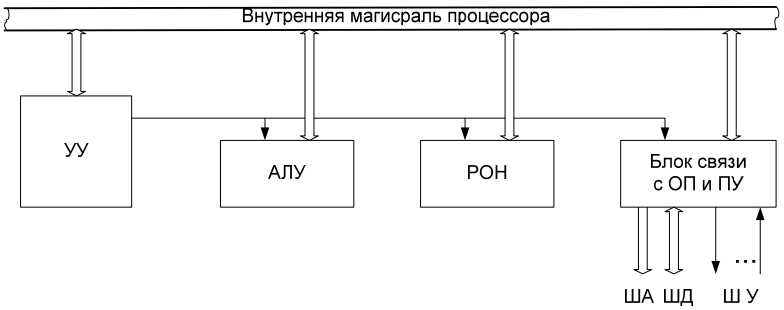


Рисунок 2.2 - Структура процессора с магистральными связями

В устройство управления (УУ) входят регистр команд, счетчик команд, дешифратор команд, регистр признаков (флагов), и др.

Арифметико-логическое устройство (АЛУ) предназначается для выполнения преобразований над логическими кодами фиксированной и переменной длины (над битами, байтами, словами), а именно: арифметических операций над числами с фиксированной и плавающей запятой, обработки алфавитно-цифровых слов переменной длины, а также операций преобразования кодов из одной системы счисления в другую и др.

Блок регистров общего назначения (РОН) позволяет увеличить производительность процессора и расширить его функциональные возможности. Обычно местная память имеет небольшой объем (8-16 байт), но выполняется на быстрых регистрах (на элементной базе самого процессора). Для адресации регистров используются укороченные команды, поэтому сокращается длина программ и время их выполнения. Расширение функциональных возможностей осуществляется путем введения в состав РОН базовых и индексных регистров, указателей стека и других регистров, что позволяет увеличить возможности адресации.

Блок связи с оперативной памятью и периферийными устройствами организует обмен с внешними по отношению к процессору устройствами памяти и ввода-вывода. В состав блока входит схема управления прерываниями и прямым доступом в память. Шины адреса (ША), данных (ШД) и управления (ШУ) предназначены для связи процессора с памятью и периферией, причём шина адреса является обычно однонаправленной, шина данных – двунаправленной, а линии в шине управления имеют разную направленность. Например, сигналы управления памятью (Чтение, Запись) и периферией (Ввод, Вывод) имеют направленность от процессора, а сигналы управления процессором (Готовность, Запрос на прерывание, Запрос на прямой доступ к памяти) направлены от внешних по отношению к процессору устройств.

Часто в состав процессора вводят блок контроля и диагностики, который служит для обнаружения и отказов аппаратуры процессора.

2.2 Декомпозиция процессора на УА и ОУ

Основу процессора составляют устройство управления (УУ или УА- управляющий автомат) и операционное устройство (ОУ или арифметико-логическое устройство- АЛУ) (см. рисунок 4.3).

Устройство управления реализует функции управления ходом вычислительного процесса, обеспечивая автоматическое выполнение команд программы.

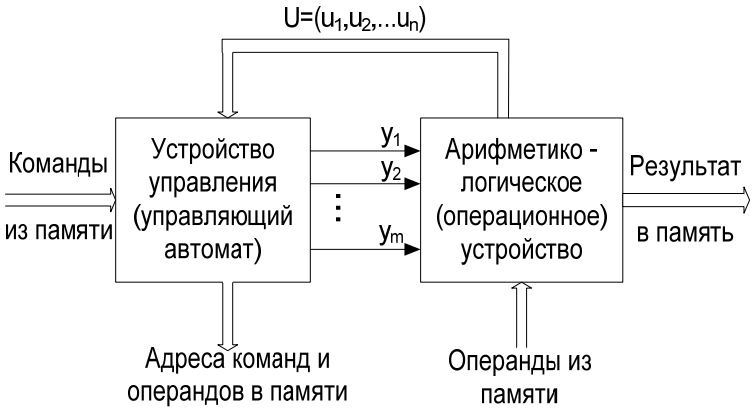


Рисунок 2.3- Разделение процессора на УА и АЛУ

Управляющие сигналы $Y=\{y_1,y_2\ldots y_m\}$ вырабатываются устройством управления в соответствии с КОП поступившей в процессор команды. Эти сигналы необходимы для выполнения операций в арифметическом устройстве. Каждый управляющий сигнал соответствует выполняемой микрооперации на некотором такте работы процессора. Сигналы $U=\{u_1,u_2\ldots u_n\}$ оповещают устройство управления о ходе выполнения операции в АУ. Такое представления процессора хорошо согласуется с теорией автоматов, при котором всякий автомат состоит из управляющего автомата и операционного устройства. Управляющее устройство может быть задано как автомат Мура или Мили.

2.3 Принцип программного управления

Принцип программного управления заключается в том, что алгоритм вычислений (например, вычисление некоторого выражения) представляется в виде упорядоченной последовательности команд, преобразующих исходные данные (операнды) в результат. Таким образом, действия предписанные алгоритмами, закладываются в команды (например, такие действия как сложение, вычитание, умножение, логические операции и т.д. представляются в виде соответствующих машинных команд). Последовательность команд называется программой. Программа управляет ходом вычислительного процесса.

Пусть, например, необходимо вычислить выражение: $Y = \frac{a \cdot b}{(a+b) \cdot c}$.

Возможная программа его вычисления содержит следующие команды:

1-я команда: умножить операнд *a* на *b*;

2-я команда: сохранить результат умножения ***(a*b)*** в памяти;

3-я команда: сложить операнды *a* и *b* ;

4-я команда: умножить результат сложения *(a+b)* на *c*;

5-я команда: считать из памяти *(a*b)*;

6-я команда: разделить результат *(a*b)* на *(a+b)* c*.

Если операнды и действия над ними представлены в двоичной системе счисления, то для реализации программы можно ввести следующую систему команд:

КОП	1-й операнд	2-й операнд

где КОП - код операции – поле, в котором заданы операции, выполняемые процессором, закодированные в двоичной системе счисления.

Подобная система команд применялась в самых первых ЭВМ, где для ввода программ и данных использовались перфоленточные устройства, в которых отсутствовала возможность возврата к ранее выполненным участкам программ. Это обстоятельство приводило к очень длинным (в буквальном смысле) программам. Пусть, например, необходимо вычислить выражение:

$$Y = \sum_{i=0}^{n} a_i b_i = a_0 b_0 + a_1 b_1 + a_2 b_2 + ... + a_n b_n .$$

Программа вычислений будет следующей:

1-я команда: умножить a_0 на b_0;

2-я команда: умножить a_1 на b_1;

3-я команда: сложить результат 1-й команды с результатом 2-й команды;

4-я команда: умножить a_2 на b_2;

5-я команда: сложить результат 3-й команды с результатом 4-й команды:

6-я команда: умножить a_3 на b_3**;**

7-я команда: сложить результат 5-й команды с результатом 6-й команды и т.д.

При использовании предложенной структуры команд программа будет состоять приблизительно из *n* - команд умножения и *n* - команд сложения, т.е. всего из 2*n* команд. Большая длина программы обусловлена невозможностью оперативного возврата к тем ее частям, которые могли бы выполняться многократно.

2.4 Принцип хранимой в памяти программы

Принцип хранимой в памяти программы был предложен Дж. фон Нейманом в 1945 году. Этот принцип стал основой современных вычислительных машин. В соответствии с этим принципом команды хранятся в памяти, также как и данные. При этом под программу отводится одна отдельная область памяти, под данные - другая область. В командах указываются не операнды, а их адреса, то есть номера ячеек памяти ОП, где они размещаются. Для вызова команд также надо указывать их адреса в ОП. При такой организации можно многократно вызывать из памяти одну и ту же команду или последовательность из нескольких команд и одни и те же

данные. Кроме этого, над командами и над данными можно производить операции, так как они с точки зрения обработки становятся равноценными. Структура команды для ЭВМ, организованной в соответствии с принципом фон-Неймана (фон- неймановской машины), будет следующей:

КОП	Адрес операнда

Такой тип команды оказался намного более универсальным и наряду с ранее приведенным он широко используется в современной вычислительной технике.

Программа вычисления выражения: $Y = \sum_{i=0}^{n} a_i b_i$ при использовании команд последнего типа намного сокращается.

1-я команда: i:= 0;
2-я команда: Y_i := 0:
3-я команда: умножение $a_i * b_i := X_i$;
4-я команда: сложение $Y_i + X_i := Y_i$;
5-я команда: $i := i+1$;
6-я команда: $i > n$? Если нет, то переход на 3-ю команду;
7-я команда: Конец.

Длина полученной программы, использующей введенный тип команды, не зависит от числа n слагаемых. Сокращение длины программы достигается благодаря возможности многократного вызова из памяти последовательности с 3-ей по 6-ю команд.

2.5 Обобщенный формат команд

Команды в ЦВМ могут быть одноадресными, двухадресными и трехадресными (в процессорах с так называемой естественной адресацией команд).

Формат одноадресной команды следующий:

КОП	А - адрес операнда

Формат двухадресной команды:

КОП	А1-адрес первого операнда	А2 - адрес второго операнда

Формат трехадресной команды:

КОП	А1- адрес первого операнда	А2 - адрес второго операнда	Ар - адрес результата

Каждая команда состоит из операционной части - кода операции (КОП) и адресной части. В операционной части указывается тип выполняемой операции в виде двоичного числа. В адресной части указывается адрес ячейки памяти, в которой размещается операнд (для одноадресной команды). Если в команде указывается адреса 1-го и 2-го операндов, то такая команда называется двухадресной. В трехадресном процессоре указывается еще и адрес результата, то есть номер ячейки ОП, куда помещается результат. В процессорах больших машин и суперкомпьютерах могут сочетаться все типы команд, в простых микропроцессорах адресность ограничивается одним-двумя полями.

Приведенные типы команд относятся к так называемым процессорам с естественной адресацией команд. В программах для таких процессоров команды размещаются в смежных ячейках памяти, т.е. команды следуют в памяти последовательно друг за другом. Адресация команд производится с помощью счетчика команд СчК *(PC- Program Counter)*.

Однако существует возможность принудительной адресации команд, когда очередная команда выбирается по адресу, указанному в предыдущей команде (такой способ адресации сохранен в настоящее время только в так называемых микропрограммных устройствах управления). Структура команд такого процессора имеет вид:

КОП	А1	А2	Ар	Аск

где Ар- адрес результата; Аск -адрес следующей команды. В такой команде уже используется четыре поля адреса: первые два для выборки операндов (А1 и А2), третий для сохранения результата, и четвертый для выборки следующей команды.

2.6 Способы адресации команд

При размещении команд в ОП для их вызова на исполнение в процессор необходимо формировать адреса ячеек ОП, в которых они хранятся. В соответствии с методом формирования адреса команды процессоры делятся на два типа: процессоры с принудительным порядком выполнения команд (принудительной адресацией команд) и с естественным порядком выполнения команд (естественной адресацией команд).

2.6.1 Процессоры с принудительной адресацией команд

Упрощенная структура процессора с принудительной адресацией команд приведена на рисунке 2.1.

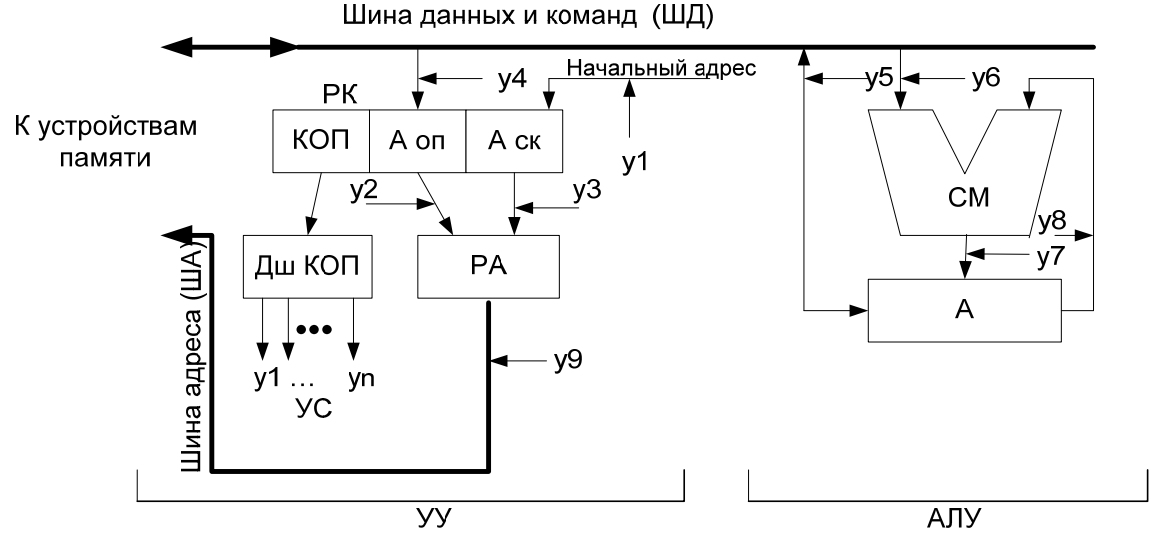


Рисунок 2.1- Упрощенная структура процессора с принудительной адресацией

Процессор содержит два устройства: устройство управления (УУ) и арифметико – логическое устройство (АЛУ).

Устройство управления предназначено для формирования управляющих сигналов (УС), необходимых для выполнения команды. В состав УУ входит регистр команд (РК), предназначенный для временного хранения текущей команды, т.е. команды, находящейся на этапе выполнения. Для нашего примера команда содержит следующие три поля: код операции (КОП), адрес операнда (Аоп) и поле адреса следующей команды (Аск). Поле кода операции указывает на тип операции, которую должна выполнить команда (сложение, вычитание, логическая операция и т.д.). Дешифратор кода операции (Дш КОП) предназначен для формирования управляющих сигналов y_i. Регистр адреса (РА) предназначен для адресации ячеек памяти, в которых хранятся команды и операнды.

Арифметико-логическое устройство в минимальной конфигурации состоит из сумматора (СМ) и аккумулятора (А). Сумматор реализует операции, заданные в команде, аккумулятор хранит один из операндов и результат.

Первоначально, например, при включении процессора, под действием управляющего сигнала y_1 в поле адреса следующей команды (Аск) заносится адрес первой выполняемой команды В простых компьютерах или контроллерах начальный адрес задается с помощью сигналов “Сброс” или “Пуск”. При этом обнуляются все регистры процессора, в результате чего выполнение программы начинается с нулевого адреса. В дальнейшем процесс выполнения команд

носит циклический характер. В начале цикла в регистр адреса помещается значение адреса из поля Аск (по сигналу y_3) . По указанному адресу из памяти читается команда (по сигналу y_9), которая по шине данных и команд (по сигналу y_4) помещается в регистр команд. Двоичный код операции дешифрируется, т.е. преобразуется в унитарный код, представляющий собой набор управляющих сигналов $y_1,...,y_n$, обеспечивающих межрегистровые передачи в процессоре, требуемые при выполнении команды. Поле адреса операнда Аоп через РА задает номер ячейки ОП (по сигналу y_2), в которой хранится операнд, участвующий в операции. Считанный из памяти операнд по сигналу y_5 помещается либо в аккумулятор, либо на один из входов сумматора (по сигналу y_6). Результат выполнения операции сохраняется в аккумуляторе (по сигналу y_7), откуда может быть направлен либо на шину данных и команд (по сигналу y_5), либо на один из входов сумматора для участия в качестве операнда при выполнении следующей команды (по сигналу y_8). Завершается цикл выполнения текущей команды формированием адреса следующей команды. Для этого значение поля Аск помещается в РА, после чего из памяти выбирается очередная команда и цикл повторяется.

Рассмотрим следующий пример. Положим, что процессор имеет следующую систему команд в машинных кодах (для более краткой записи представим систему команд в шестнадцатеричной системе счисления):

01- вызов операнда из ОП в аккумулятор;

02- запись содержимого А в ОП;

1A- команда сложения;

00- останов выполнения программы.

Пусть необходимо составить программу сложения 2-х чисел, находящихся соответственно в ячейках ОП с адресами 0841 и 0842 и записать результат в ячейку ОП с адресом 0843. Программа хранится в смежных ячейках памяти, начиная с адреса 1300. Разрядность памяти (ширина выборки данных) составляет 1 байт. Пример программы сложения 2-х чисел с использованием принудительной адресации приведен в таблице.

№ яч	КОП	Аоп	Аск	Комментарий
1300	*01*	*0841*	*1305*	Вызов 1-го операнда из ОП и переход к считыванию следующей команды из ячейки ОП с номером *1305H*.
1305	*1A*	*0842*	*130A*	Вызов 2-го операнда, сложение и переход к считыванию следующей команды из ячейки ОП с номером *130AH*.
130A	*02*	*0843*	*130F*	Запись результата в ОП и переход к ячейке *130FH*.
130F	*00*	*0000*	*0000*	Останов.

Нетрудно подсчитать, что при использовании принудительной адресации команд длина программы составляет 20 байт.

2.6.2 Процессоры с естественной адресацией команд

Упрощенная структура процессора с естественной адресацией команд приведена на рисунке 2.2.

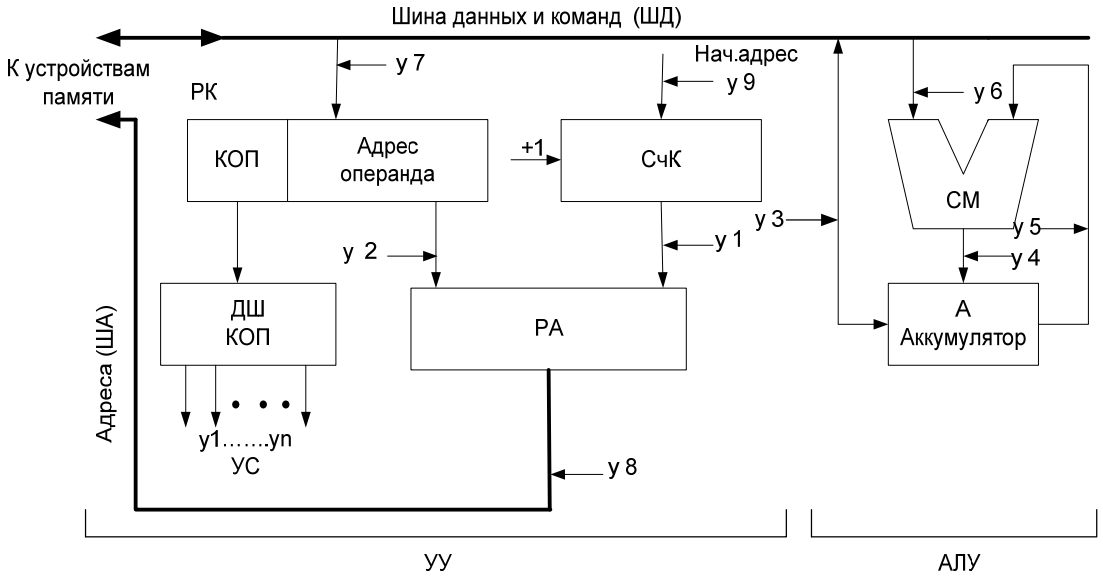


Рисунок 2.2 - Упрощенная структура процессора с естественной адресацией команд

В таких процессорах в регистре команд (РК) отсутствует поле с адресом следующей команды. Вместо этого поля РК в процессор включается счетчик команд (СчК) или программный счетчик *(PC- Program Counter)*, указывающий на адрес текущей команды. Счетчик команд при инициации процессора загружают начальным адресом программы, по которому выбирается первая команда. После выполнения очередной команды содержимое СчК модифицируется путем автоматического добавления к текущему значению адреса единицы (+1) или шага адресации – обычно это число, равное длине выполняемой команды в байтах. По модифицированному адресу выбирается на выполнение следующая команда и т.д.

Предыдущая программа для этого типа процессора будет иметь вид (см. таблицу 2.2):

Таблица 2.2- Пример программы сложения 2-х чисел с использованием естественной адресации

№ яч.	КОП	Аоп	Комментарий
1300:	*01*	*08 41*	Вызов 1-го операнда из ОП в аккумулятор и переход к считыванию следующей команды из ячейки ОП с номером *1303H*.
1303:	*1A*	*08 42*	Вызов 2-операнда, сложение его с 1- м операндом и переход к считыванию следующей команды из ячейки ОП с номером *1306H*.
1306:	*02*	*08 43*	Запись вычисленного результата в ячейку ОП с адресом *0843H* и переход к считыванию следующей команды.
1309:	*00*	*00 00*	Останов.

Нетрудно подсчитать, что длина последней программы составляет 12 байт. Уменьшение длины программы достигается благодаря отсутствию в командах поля с адресом следующей команды.

В процессорах с естественной адресацией длина программы и требуемый под неё объем памяти сокращаются, но система команд усложняется. Так, например, для организации ветвлений

в программах требуются специальные команды - условных и безусловных переходов. Однако первое обстоятельство оказалось более весомым и традиционные машины выполняются по второму способу.

2.5 Способы адресации операндов

Под способом адресации понимается правило нахождения адреса операнда по информации, заданной в команде. В современных ЭВМ используется большое число способов адресации операндов. Рассмотрим наиболее часто используемые из них.

2.5.1 Прямая адресация

При прямой адресации адрес операнда указывается в адресной части команды. Поле адреса может быть одно, двух и трехадресным. Длина адресного поля n_A должна быть такой, чтобы перекрывать все адресное пространство оперативной памяти– $n_A=log_2 M$, где M-емкость памяти в байтах.

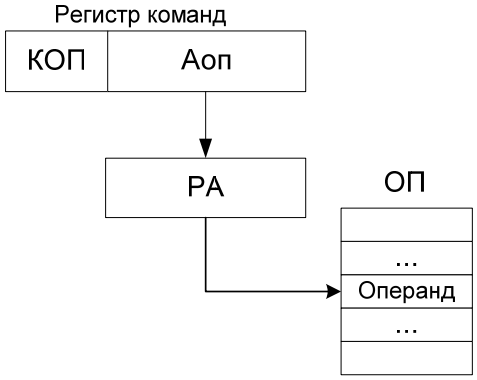


Рисунок 2.3 - Порядок выборки операнда из памяти при прямой адресации

Длина команд с прямой адресацией велика и может достигать размеров в несколько машинных слов. Например, для трёхадресной команды, состоящей из четырех полей (см. рисунок 2.4), общая длина N_K может составить до 4-х слов ($N_K = n_{КОП} + n_{A1} + n_{A2} + n_P$). Так например, 32-х разрядный процессор способен адресовать память объемом 4 Гбайт. Но тогда потребуется, чтобы все адресные поля были длиной по 32 разряда. В этом случае общая длина команды (с учетом поля кода операции) составит не менее 13 байтов.

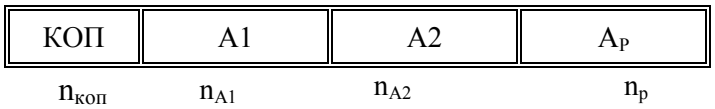


Рисунок 2.4 - Формат трехадресной команды с использованием прямой адресации

Для выборки такой команды из памяти потребуется несколько шинных циклов, поэтому для чтения команды и исполнения действий, предписанных командой, требуется значительное время.

2.5.2 Регистровая адресация

Регистровая адресация является укороченной. В поле адреса команды указываются адреса ячеек регистровой (сверхоперативной) памяти (РП), в которых находятся операнды. РП является быстродействующей памятью, реализуемой на кристалле (чипе) процессора. Отдельные регистры РП выполняется на той же элементной базе, что и процессор, а их часто называются регистрами общего назначения (РОН). Число РОН невелико, однако они обеспечивают значительное повышение производительности процессора (до 20-30 %). Это объясняется тем, что время выполнения команд с регистровой адресацией очень мало из-за близкого расположения данных. Время обращения к операндам может составлять доли процессорного такта. Следующий рисунок (см. рисунок 2.5) поясняет порядок выборки операндов при использовании 2-х адресной команды.

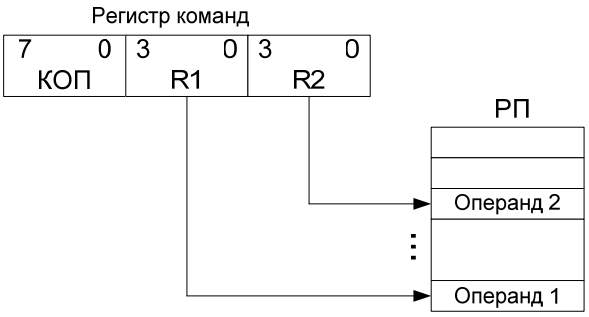


Рисунок 2.5 - Порядок выборки операндов при регистровой адресации: R1- адрес 1-го операнда, R2- адрес второго операнда

2.5.3 Косвенная адресация

При косвенной адресации в адресной части команды указывается адрес ячейки памяти, в которой находится адрес операнда (косвенная адресация - это адресация адреса). Косвенный адрес обычно хранится в регистровой памяти процессора. Разрядность адресных регистров РП должна быть достаточной для перекрытия адресного пространства памяти. На рисунке 2.6 показан порядок выборки операндов при косвенной адресации. Регистр команд содержит поле R, в котором хранится адрес косвенного регистра A_K, размещенного в регистровой памяти. Содержимое A_K помещается в регистр адреса процессора, и по его значению производится выборка из оперативной памяти или запись в оперативную память операнда.

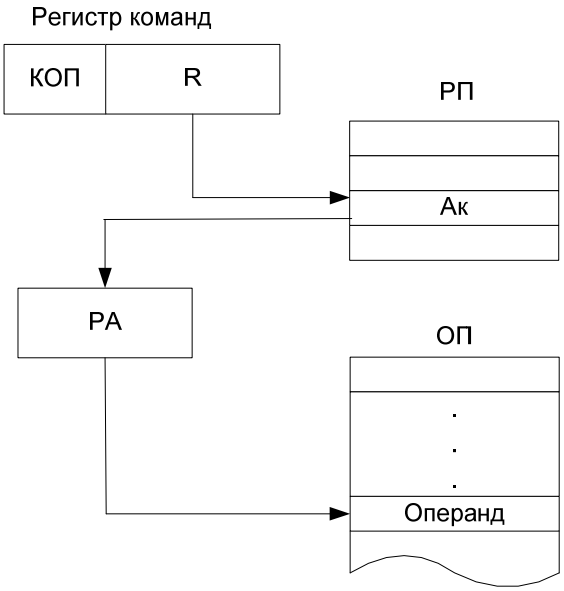


Рисунок 2.6 - Порядок выборки операнда при косвенной адресации

Команда с косвенной адресацией является укороченной, поэтому часто используется в микропроцессорах и микроконтроллерах, имеющих малую разрядность. Она позволяет при малой длине разрядной сетки адресовать большие объёмы оперативной памяти или весь её объём. Пусть, например, РП процессора состоит из шестнадцати 32-х разрядных регистров общего назначения. Для адресации регистровой памяти такого объема команда должна иметь 4-х разрядное поле адреса R. Но, поскольку регистр A_K 32-х разрядный, то объем адресуемой памяти составляет 4 Гбайта.

Недостатком такого способа является увеличенное время выполнения команды, которая по сравнению с прямой адресацией требует дополнительного цикла для выборки косвенного адреса.

2.5.4 Непосредственная адресация

В поле адреса команды находится не адрес, а сам операнд. Непосредственный операнд может иметь любую длину (байт, слово, 2-е слово). Этим и определяется длина команды. Формат команды с непосредственной адресацией показан на рисунок 2.7:

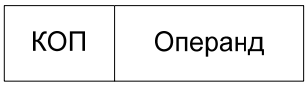


Рисунок 2.7 – Формат команды с непосредственной адресацией

В отличие от других типов команд, при выполнении команд с непосредственной адресацией отсутствует дополнительный цикл обращения в память за операндом. Используется для хранения и быстрой выборки констант из оперативной памяти.

2.5.5 Неявная адресация

Неявная (подразумеваемая) адресация является модификацией регистровой адресации. В команде нет явных указаний на адреса операндов, они подразумеваются, поскольку заключены в коде операции команды. Такая команда является самой короткой и используется в простейших микропроцессорах и микроконтроллерах. На рисунке 2.8 показан пример команды с неявной адресацией. Она позволяет адресовать регистровую память с восемью РОНами. Два первых разряда указывают на тип команды (например, посылочная, арифметическая, логическая), а поля R1 и R2 – для адресации регистров.

Рисунок 2.8 – Формат команды с неявной адресацией

2.5.6 Относительная (базовая) адресация

Адрес операнда определяется как сумма содержимого адресного поля команды и некоторого числа, называемого базовым адресом. Базовый адрес является косвенным; для указания его адреса в команде предусмотрено короткое поле B. Ещё одно поле команды, в котором находится адрес операнда, называют смещением. С помощью поля смещения адресуются области оперативной памяти большого объема, например, сегменты кода или данных объемом от нескольких Мбайт до сотен Мбайт. Адресация операнда производится полем смещения относительно базового адреса (Base).

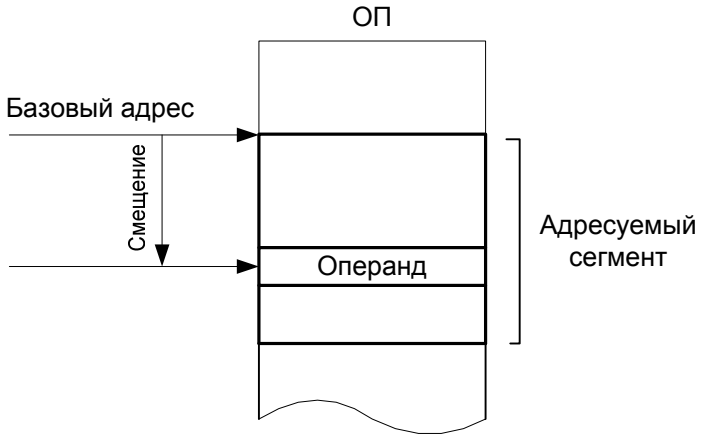


Рис 2.9- Принцип выборки операнда относительной адресации

На рисунке 2.10 показан принцип формирования адреса операнда при относительной адресации и порядок выполнения команды. Поле B используется для выборки базового адреса (Base) из регистровой памяти. Формирование адреса операнда EA производится специальным адресным сумматором, на один вход которого помещается базовый адрес, на другой - содержимое поля смещения, взятое из команды. Полученный адрес EA=[B]+D называется эффективным или исполнительным адресом. Прямые скобки при B ([B]) – означают, что первое слагаемое берется по адресу, указанному в поле B. При выборке некоторого участка данных базовый адрес является неизменным.

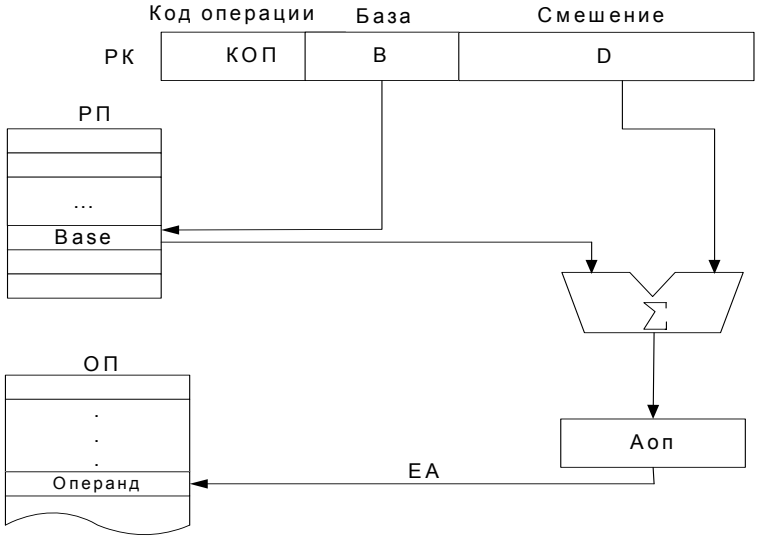


Рисунок 2.10 - Формирование адреса операнда при относительной адресации

Относительная адресация обеспечивает перемещаемость программ, то есть обладает возможностью передвижения программ в памяти без изменений внутри самой программы, что не требует в дальнейшем перетрансляции программы. Поскольку адресация внутри программы производится полем смещения, то программист может размещать программу, начиная, например, с нулевого адреса. Реальный адрес программы определяется базовым адресом B, созданным, например, операционной системой. При назначении новых адресов, изменяется только значение базового адреса. Относительная адресация широко используется в мультипрограммных вычислительных системах с виртуальной памятью.

2.5.7 Индексная (автоинкрементная или автодекрементная) адресация

При обработке больших массивов данных, выбираемых последовательно друг за другом, нет смысла каждый раз обращаться в память за новым адресом. Для этого достаточно автоматически менять содержимое специального регистра, называемого индексным, причем индексный регистр является косвенным (см. рисунок 2.11). Его загружают начальным адресом массива при задании параметров программы. Дальнейшая адресация осуществляется путем автоматического добавления или вычитания единицы или шага адресации из содержимого индексного регистра.

В некоторых процессорах применяют более сложную адресацию, которая сочетает индексную и относительную. Часто в команду с индексной адресацией включают признак, определяющий шаг индексации T (T=1,2,4 и т.д.), что позволяет осуществлять адресацию массивов через байт, слово, двойное слово и т.д. В современных процессорах (например, в Intel 80386 и выше) применяют все возможные сочетания из смещения, индексного адреса, относительного адреса и шага. Например:

- индексная адресация с шагом. Содержимое индексного регистра умножается на шаг и суммируется со смещением- EA=[X]·T+D, где T - величина шага;

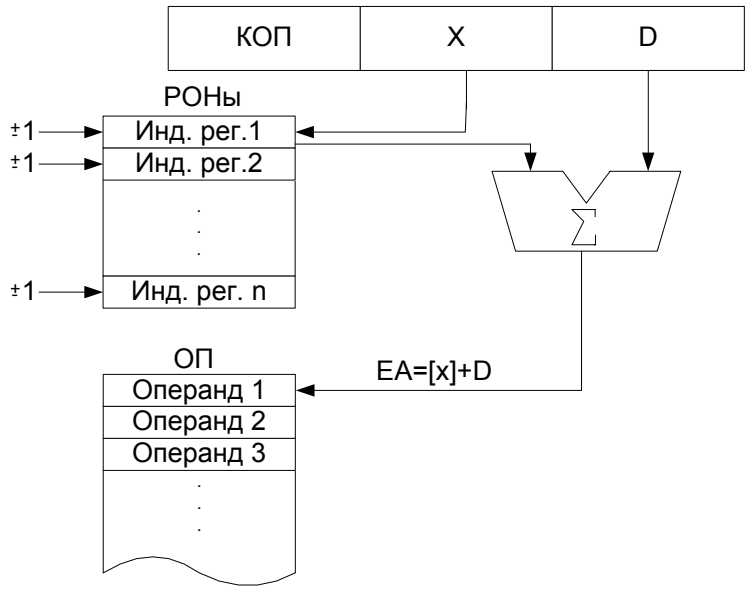


Рисунок 2.11 - Формирование адреса операнда при индексной адресации

- базово - индексная адресация EA=[B]+[X];
- базово - индексная адресация с шагом EA=[B]+[X]·T;
- базово - индексная адресация со смещением EA=[B]+[X]+D:
- базово - индексная адресация со смещением и шагом EA=[B]+[X]·T+D.

2.6 Контрольные вопросы

1. Какова структура команды?
2. Какие поля включает команда?
3. Чем определяется длина команды?
4. В чем заключается естественная адресация команд в ЭВМ?
5. В чем заключается принудительная адресация команд в ЭВМ?
6. Перечислите достоинства и недостатки естественной адресации?
7. Перечислите достоинства и недостатки принудительной адресации?
8. Какие существуют способы адресации операндов?
9. Достоинства неявной и регистровой адресации?
10. В чем заключается непосредственная адресация?
11. В чем заключается прямая адресация?
12. Какие преимущества косвенной адресации?
13. Каково назначение относительной адресации?
14. Каково назначение индексной адресации?
15. Что означает базово - индексная адресация с шагом?
16. Что означает базово - индексная адресация со смещением и шагом.

3 УСТРОЙСТВА ПАМЯТИ

3.1 Основные понятия

Программы и обрабатываемые ими данные хранятся в оперативной памяти компьютера. Для размещения в ОП больших программ необходима память соответствующего объема, при этом скорость выполнения программ напрямую зависит от скорости передачи данных между процессором и памятью.

Идеальная память должна обладать высокой скоростью чтения- записи информации, иметь большой объем и быть недорогой. Удовлетворить всем трем требованиям одновременно невозможно. Чем больше память и чем быстрее она работает, тем дороже она стоит.

Обычно память разрабатывается с учетом того, что данные записываются и считываются не только байтами, но и словами. Само понятие длины слова, чаще всего определяется как количество бит, сохраняемых или считываемых за одно обращение (шинный цикл) к памяти.

Максимальный размер оперативной памяти, который может использоваться процессором, определяется разрядностью его шин адреса и данных. Если разрядность шины адреса процессора - n бит, а шины данных - k бит, то максимальный размер памяти составляет 2n k-разрядных слов. За один шинный цикл обращения к памяти в процессор пересылается k бит данных. Поэтому процессор с 16-разрядной шиной адреса, может адресовать память объемом до 216 - 64 К k-разрядных слов, процессор, генерирующий 32-разрядные адреса, может использовать память объемом до 232 = 4 Г k-разрядных слов, а для процессоров с 40-разрядными адресами доступна память объемом до 240 = 1 Т единиц памяти.

Кроме шин адреса и данных для обмена информацией процессора и памяти используется шина управления. В простейшем случае она должна содержать линию для управления типом передачи данных: чтение или запись- Чт/Зп (Read/Write# - R/W#), которая часто дополняется линией готовности памяти к обмену (RDY или REDY). Могут использоваться и другие линии, с помощью которых, например, задается количество пересылаемых за один шинный цикл байт данных. Соединение процессора и ОП схематически показано на рисунке 3.1.

Рисунок 3.1- Организация связи ОП с процессором

Чтобы считать данные из ОП, процессор сначала выставляет адрес на шину адреса и устанавливает (с некоторой задержкой) линию R/W# в состояние “Лог. 1”. В ответ память помещает содержимое адресованной ячейки на линии данных и сообщает об этом процессору активизацией сигнала RDY. После получения сигнала RDY k-разрядное слово с шины данных вводится в процессор.

Для того чтобы записать данные в память, процессор выставляет адрес на ША, а данные- на ШД после чего устанавливает линию R/W# в состояние “Лог. 0” (знак # показывает, что активным уровнем сигнала W является “Лог. 0” или низкий уровень), указывая таким образом, что выполняется операция записи в память.

Если в операциях чтения (записи) производится обращение по последовательным адресам ОП, может быть выполнена операция блочной (пакетной) пересылки, при которой за один шинный цикл осуществляется пересылка нескольких (обычно 4-х) k-разрядных слов. При пакетных передачах повышается скорость обмена, при этом можно ограничиться выдачей на ША только адреса первого слова пакета.

3.2 Классификация ЗУ

При разработке системы памяти ЭВМ приходится решать противоречивую задачу создания памяти требуемой емкости и быстродействия при приемлемой стоимости. При этом наиболее оптимальными оказываются системы памяти, построенные по иерархическому признаку. Современные иерархические ЗУ подразделяются на сверхоперативные, КЭШ - память, оперативные и внешние (см. рисунок 3.2).

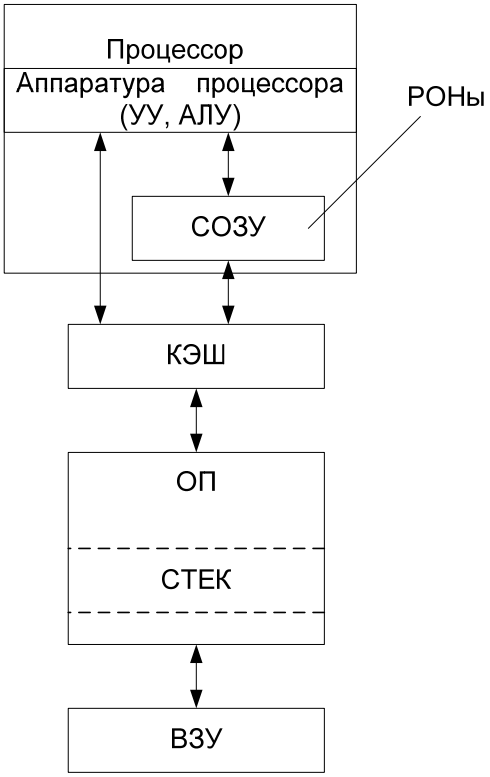


Рисунок 3.2 - Иерархическая структура ЗУ

Сверхоперативная память обычно входит в состав процессора, выполняется на его элементной базе и обеспечивает повышение быстродействия ЭВМ в целом. Она состоит из небольшого количества (до нескольких десятков) регистров общего назначения (РОН), в связи с чем команды процессора, использующие РОНы, имеют малый длину благодаря укороченному полю адреса. Вследствие этого длина программ сокращается.

Необходимость введения сверхоперативной памяти в состав процессора обусловлено тем, что процессор обрабатывает команды и данные быстрее, чем они выбираются из памяти. Время доступа процессора к командам и данным, размещенным в памяти, является узким местом системы в целом. Использование РОН для промежуточного хранения тех данных, которые необходимы при выполнении следующих команд программы, уменьшают количество обращений к памяти и, как следствие, уменьшают время выполнения всей программы.

Другим способом сокращения времени доступа к информации является использование кэш-памяти. Это быстрая память небольшого объема (несколько десятков или сотен килобайт), расположена между основной памятью ЭВМ и процессором. В ней хранятся копии тех участков оперативной памяти с программами и данными, которые интенсивно используются процессором в текущий момент времени. Территориально кэш- память может находиться внутри или вне кристалла процессора. Обмен между процессором и внешней кэш- память осуществляется несколько медленнее обмена с внутренней. Это обусловлено тем, что внешняя кэш- память реализуется на менее быстрой элементной базе и обращение к ней осуществляется через шину процессора.

Кэш- память реализуется на статической памяти с произвольным доступом (SRAM- Static Random Access Memory). Роль запоминающего элемента в статической памяти выполняет электронный триггер.

Оперативная память является основной памятью машины. В ОП хранятся копии запускаемых программ, а также данные, подлежащие обработке. Для уменьшения габаритов и стоимости ОП в настоящее время выполняется в основном на микросхемах динамической памяти (DRAM- Dynamic RAM).

Стековая память используется в ЭВМ для запоминания содержимого регистров процессора, при этом стековая память входит либо в состав процессора, либо под нее отводится часть ОП.

Важным звеном в иерархии ЗУ является внешняя память. Назначение внешнего ЗУ - хранение больших массивов информации. Внешние запоминающие устройства (ВЗУ чаще всего выполняются на магнитных и оптических дисках, а также ленточных накопителях.

Первые два типа ВЗУ называют устройствами прямого доступа (циклического доступа). Магнитные и оптические поверхности этих устройств непрерывно вращаются, благодаря чему обеспечивается быстрый доступ к хранимой информации (время доступа этих устройств составляет от нескольких мс до десятка мс). Накопители на магнитных лентах (МЛ) называют устройствами последовательного доступа, из-за последовательного просмотра участков носителя информации (время доступа этих устройств составляет от нескольких секунд до нескольких минут).

3.3 ОЗУ с произвольным доступом

В оперативных ЗУ с произвольным доступом (RAM) запись или чтение осуществляется по адресу, указанному регистром адреса (РА). Информация, необходимая для осуществления процесса записи - чтения (адрес, данные и управляющие сигналы), поступает из процессора (см. рисунок 3.3). В ЗУ с произвольным доступом, на обращение по любому адресу уходит одно и то же время. Этим RAM-память отличается от запоминающих устройств с последовательным доступом, таких

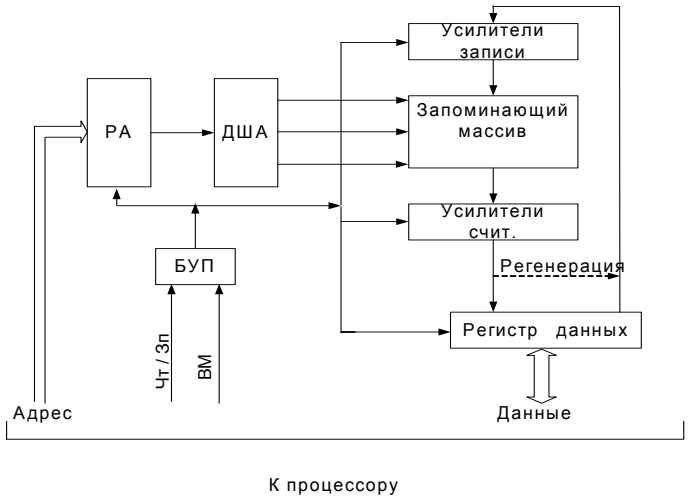


Рисунок 3.3 - Обобщенная схема модуля RAM

как магнитные ленты. Время доступа последних зависит от адреса (местоположения) данных.

Адрес, поступающий из процессора, фиксируется в регистре адреса РА микросхемы, дешифрируется с помощью ДшА и выбирает нужную ячейку запоминающего массива. По сигналу запись Зп производится запись данных в заданную ячейку памяти, по сигналу чтения Чт - выборка данных. Сигнал выборки ВМ или ВК (CS - Chip Selekt) предназначен для выбора адресуемой микросхемы памяти.

Усилители записи и считывания обеспечивают физический процесс записи - чтения запоминающего элемента массива при выработке соответствующих сигналов блока управления

БУП. При чтении содержимое адресованной ячейки памяти через регистр данных поступает на ШД процессора. Если при считывании содержимого ячейки памяти происходит его разрушение, то после выдачи данных на ШД процессора необходимо восстановление (регенерация) содержимого ячейки.

Запоминающий массив RAM содержит множество одинаковых запоминающих элементов статического либо динамического типов. Если запоминающие элементы памяти могут сохранять свое состояние до тех пор, пока на них подано питание (Vпит), то такая память называется статической (SRAM). Возможная реализация запоминающего элемента ячейки памяти на КМОП- транзисторах показана на рисунке 3.4.

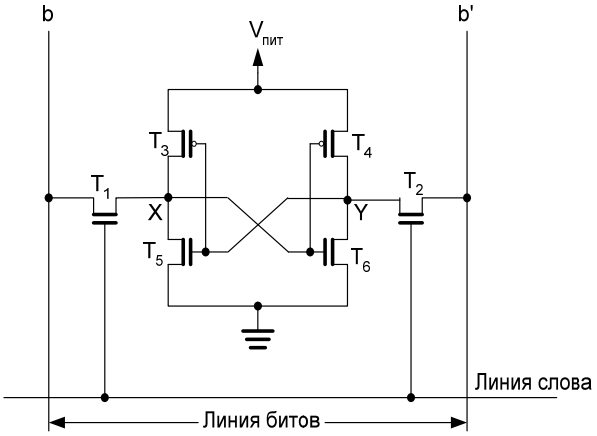


Рисунок 3.4 - Схема ячейки памяти на КМОП- транзисторах

Для запоминания одного бита информации в приведенной схеме используется триггер, который образуют транзисторы Т3, Т5 и Т4, Т6. В состоянии “лог 1” напряжение в точке X будет иметь высокий уровень напряжения за счет того, что транзисторы Т3 и Т6 открыты, а транзисторы Т5 и Т4 закрыты. Таким образом, если транзисторы Т1 и Т2, управляемые сигналами на линии слова, открыты, то напряжение на линии бита b будет высоким, а на линии бита b' - низким.

Время записи- считывания (доступа) информации современных микросхем статической памяти составляет несколько наносекунд. Поэтому статическая память используется в первую очередь в тех устройствах, где требуется малое время записи- чтения информации.

Статическая RAM является быстродействующей памятью, но ее габариты и стоимость не всегда приемлемы, поскольку каждая ее ячейка, хранящая 1 бит информации, реализуется на 6-ти транзисторах. Поэтому производителями выпускается дешевая память с более простой конструкцией запоминающих элементов. Однако они требуют постоянного обновления записанной информации, поскольку не способны долго сохранять свое состояние. Такая память называется динамической (DRAM- Dynamic RAM). В ячейке динамической памяти (см. рисунок 3.5)

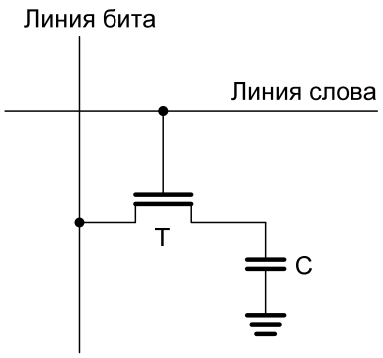


Рисунок 3.5 - Упрощенная схема запоминающего элемента ячейки динамической памяти

информация хранится в виде заряда конденсатора, и этот заряд может сохраняться всего несколько единиц или десятков миллисекунд из-за утечки заряда. Поскольку информация в памяти должна сохраниться все время, в течение которого на память подается напряжение питания, содержимое каждой ячейки динамической памяти должно периодически обновляться путем восстановления заряда конденсаторов запоминающих элементов. Такая операция называется регенерацией информации. Для регенерации информации достаточно выбрать строку памяти, т.е. указать адрес регенерируемой строки.

3.4 Организация микросхем SRAM

Полупроводниковая память реализуется в виде микросхем с очень разным быстродействием. Длительность их цикла варьируется от 100 нс до менее чем 10 нс. Появившиеся в конце 1960-х полупроводниковые схемы памяти первоначально были гораздо дороже доминирующей в это время памяти на магнитных (ферритовых) сердечниках. Однако стремительное развитие технологии СБИС позволило быстро снизить их стоимость, и в настоящее время практически вся память реализуется в виде полупроводниковых микросхем.

В каждом запоминающем элементе памяти может храниться один бит информации. Запоминающие элементы объединяются в ячейки памяти длиной в k- бит которые, в свою очередь, объединяются в массивы матричной структуры, состоящих из строк и столбцов. Каждая строка матрицы составляет слово памяти, а все ячейки строки выбираются общей линией, называемой линией слова, которая управляется входящим в состав кристалла памяти дешифратором адреса строки. Во время операции чтения информация из выбранной строки пересылается на выходные линии данных микросхемы. В процессе операции записи входная информация записывается в ячейки выбранной строки.

Рассмотрим организацию микросхемы памяти на 1 К (1024) бит. Память такого объема может быть организована в виде массива из 128 слов с длиной по 8 бит (Байт) каждое. Для адресации 128 слов потребуется $\log_2 128 = 7$ разрядов адреса. Вместе с 8-ю линиями данных при такой организации массива в микросхеме необходимо 15 внешних выводов (без учета выводов для питания и управляющих сигналов). В оптимальных (по количеству выводов) вариантах то же количество ячеек можно организовать в виде массивов 512x2 бит или 1 К x 1 бит. В последнем случае понадобится 10-разрядный адрес и одна линия данных, а, следовательно, 11 внешних выводов. Эта организация показана на рисунке 3.6. В ней 10-разрядный адрес делится на две части по пять разрядов в каждой, представляющих адреса строки и столбца массива ячеек. Пять разрядов адреса строки выделяют одну из 32-х строк массива длиной в 32 бита, позволяя им поступать через усилители записи- чтения на мультиплексор. Пять разрядов адреса столбца с помощью мультиплексора 32 x 1 соединяют с внешней линией данных только адресованный бит.

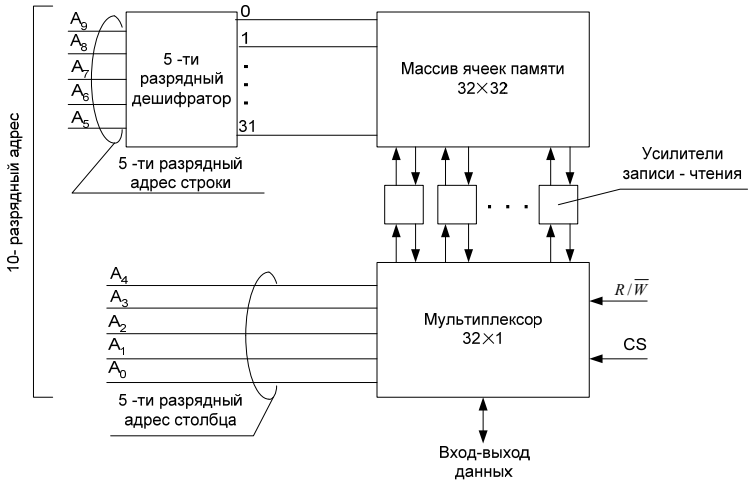


Рисунок 3.6 - Структура микросхемы SRAM с организацией 1К x 1 бит

Современные микросхемы памяти содержат гораздо большее количество ячеек. Микросхемы SRAM большей емкости и разрядности построены по такой же структуре, но со-

держат массивы большего размера и имеют большее число внешних соединений. Например, 4-мегабитовая микросхема памяти может иметь организацию 512 К х 8 бит, для которой понадобятся 19 адресных выводов и 8 выводов для ввода-вывода данных. В настоящее время выпускаются микросхемы SRAM емкостью в сотни кбит адресуемых единиц. Количество адресуемых слов микросхемы памяти определяет размер ее адресного пространства.

3.5 Организация динамической памяти

Структура микросхем динамической памяти (DRAM) в целом близка к структуре микросхем статической памяти. Однако для уменьшения количества выводов в микросхемах динамической памяти используется мультиплексированные входы адреса. Адрес, подаваемый с шины адреса процессора, делится на две части - адрес строки и адрес столбца. При адресации ячеек DRAM эти части адреса, последовательно во времени, подаются на адресные входы микросхемы памяти в сопровождении соответственно стробов адреса строки RAS (Row Address Strobe) и столбца CAS (Column Address Strobe). Сначала подаются адресные биты, выбирающие строку массива ячеек. По сигналу RAS эти биты сохраняются в защелке адреса строки внутри микросхемы, после чего на те же выводы микросхемы подаются младшие адресные биты.

Разделение полного адреса, выдаваемого процессором, и последовательную выдачу его на микросхему памяти осуществляет мультиплексор, входящий в схему управления (контроллер) динамической памятью.

Временные диаграммы ввода адреса запоминающего элемента в микросхемы DRAM приведены на рисунке 3.7.

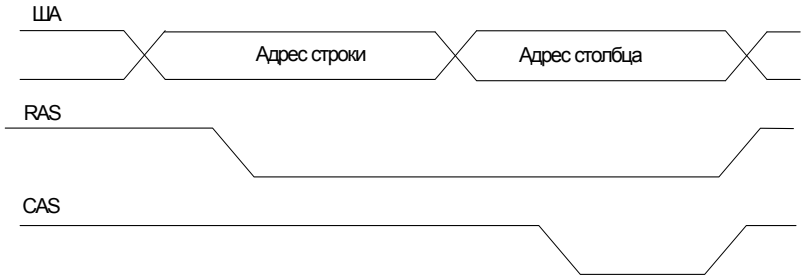


Рисунок 3.7 - Временные диаграммы сигналов ввода адреса в микросхему DRAM

Структура асинхронной динамической памяти DRAM конфигурации 2М х 8 емкостью 16 Мбит показана на рисунке 3.8. Ее ячейки организованы в массив 4 К х 4 К, в котором 4096 запоминающих элементов каждой строки разделены на 512 групп по 8 бит, так что в каждой строка хранит 512 байт данных. Для адресации 4096 строк необходимо 12 адресных разрядов. Еще 9 разрядов адреса необходимы для выбора из строки группы из 8 бит, поэтому для доступа к каждому байту в микросхеме с рассматриваемой организацией необходим 21-разрядный адрес. Старшие 12 и младшие 9 разрядов адреса образуют адреса строки и столбца байта.

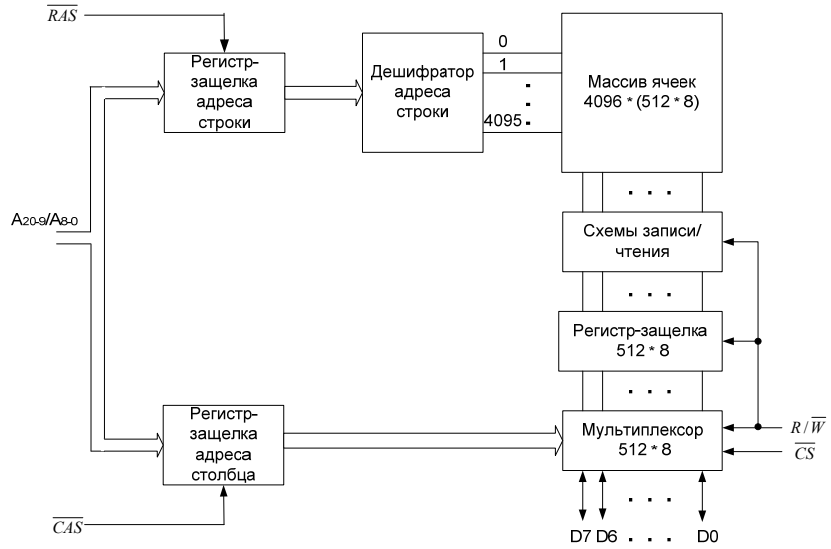


Рисунок 3.8 - Организация микросхемы динамической памяти 2M x 8

При выполнении операции чтения или записи сначала на адресные выводы микросхемы подается адрес строки. При активизации сигнала RAS он загружается в защелку (регистр) адреса строки микросхемы. Адрес строки дешифрируется и активизирует одну из линий выбора строки запоминающего массива 0,1,…,4095. Содержимое выбранной строки массива с помощью схемы записи- чтения считывается в регистр- защелку разрядностью 512x8 статического типа.

Через некоторое время после загрузки адреса строки на адресные выводы микросхемы подается адрес столбца, который загружается в регистр-защелку при активном сигнале строба CAS. 9 линий этого регистра выбирают адресованный байт из 512, хранящихся в защелке 512x8. Если управляющий сигнал R/W указывает на операцию чтения, выбранный байт выгружается на линию данных D7…D0.

Для осуществления записи в микросхему, информация с линий D7…D0 пересылается в соответствующую группу из 8-ми бит защелки 512x8 и замещает их прежнее содержимое. В микросхемах динамической памяти, как правило, активным сигналам RAS и CAS соответствует низкий уровень напряжения, поэтому стробирование адреса выполняется при переходе соответствующего сигнала от высокого уровня к низкому. На схемах эти сигналы обозначаются как RAS и CAS.

Особенностью запоминающих элементов динамической памяти является то, что при считывании информации из них их содержимое разрушается, в связи с чем, после окончания действий по записи или чтению информации в защелку 512x8, необходимо регенерировать содержимое считанной в эту защелку строки массива. Для этого ранее сохраненное в защелке содержимое строки (при чтении информации из микросхемы), или с частью модифицированных бит (при записи информации в микросхему) записывается в считанную строку.

Необходимость восстановления содержимого выбранной строки запоминающего массива является причиной еще одного недостатка микросхем DRAM- обязательного ожидания окончания процесса перезаписи информации из защелки в выбранную строку перед следующей операцией записи - чтения данных из другой строки. Это приводит к увеличению времени чтения- записи микросхемы. Однако, если последующий адрес записи- чтения относится к той же строке микросхемы, то адресуемый байт можно считать прямо из защелки, без повторной подачи адреса строки, с подачей на микросхему только адреса столбца в сопровождении строба CAS. Описанный режим записи - чтения информации из смежных ячеек памяти одной строки называется режимом быстрого страничного чтения (Fast Page Mode- FPM) и обеспечивает значительное снижение времени записи- чтения в режиме пакетного обмена.

В связи с вышеописанной организацией микросхем динамической памяти подача адреса строки в ходе операции считывания или записи приводит к обновлению содержимого всех ячеек выбранной строки. Поэтому, для того, чтобы поддерживать содержимое памяти DRAM, достаточно периодически обращаться к каждой ее строке. Обычно эти действия выполняется с

помощью специальной схемы, называемой схемой регенерации (Memory Refresh- “освежение” памяти), являющейся обязательным компонентом архитектуры ЭВМ, использующей динамическую память. Заметим однако, что основные действия по регенерации микросхем DRAM осуществляются внутри самих микросхем. Подсистема регенерации должна осуществлять только отсчет интервала регенерации и формировать адреса регенерируемых строк.

Благодаря своей высокой емкости (от 1 до 256 Мбит) и дешевизне, микросхемы DRAM широко используются в запоминающих устройствах ЭВМ. Микросхемы имеют различную организацию, благодаря чему из них можно легко компоновать память требуемой емкости и разрядности.

7.6 Регенерация динамической памяти

Как указывалось ранее, динамическая память требует регенерации (восстановления) информации, производимой через каждые несколько мс. Это связано с тем, что для хранения одного бита информации в ней используется заряд на конденсаторе, который с течением времени рассасывается. Регенерация памяти заключается в том, что содержимое каждой строки микросхемы DRAM считывается, усиливается и записывается вновь на прежнее место. При регенерации доступ к памяти со стороны процессора или других устройств запрещен, что приводит к снижению производительности ЭВМ.

В соответствии с ранее рассмотренной организацией микросхемы DRAM для регенерации хранимой в ней информации, на микросхему необходимо подать адрес строки (в сопровождении строба RAS) и сигнал чтения. Упрощенная структура системы регенерации содержимого памяти DRAM, состоящей из 256 строк, приведена на рисунке 7.10.

Таймер интервала регенерации следит за своевременным проведением регенерации всех строк микросхем DRAM. Если требуемое время регенерации составляет 4 мс, то для регенерации каждой строки 256- строковой микросхемы DRAM сигнал регенерации (REFRESH) должен вырабатываться через каждые 15,6 мкс. В каждом цикле регенерации осуществляется восстановление содержимого одной строки. При выработке нового сигнала REFRESH содержимое счетчика адреса строки инкрементируется, после чего вырабатываются строб RAS и сигнал чтения памяти MEMR. Основные действия по регенерации содержимого очередной строки осуществляются внутри самой микросхемы DRAM. Система регенерации отвечает лишь за

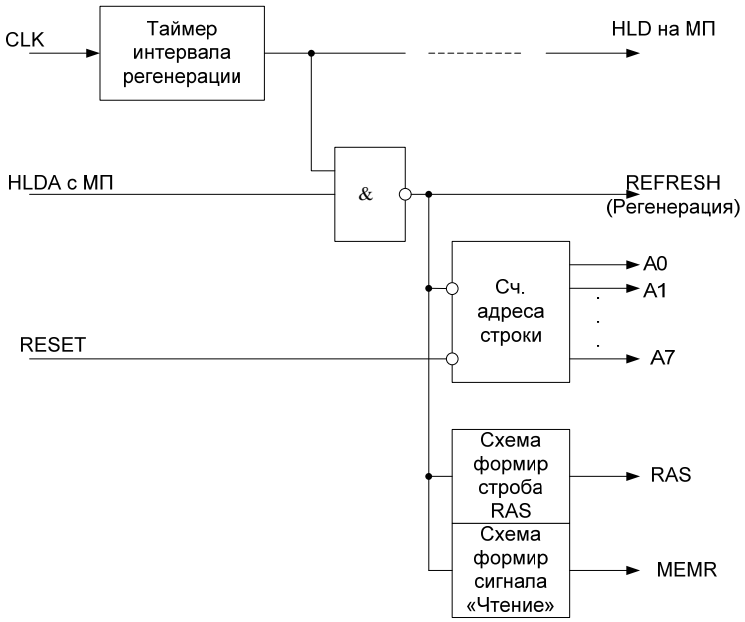


Рисунок 7.10 - Структура подсистемы регенерации динамической памяти

своевременную регенерацию всех строк, выработку адреса очередной строки и необходимых управляющих сигналов.

3.6 Особенности микросхем синхронной динамической памяти

Описанная динамическая память управляется в асинхронном режиме. Она тактируется только управляющими сигналами RAS и CAS и момент готовности микросхемы к обмену информацией с процессором, в общем, не известен. При использовании асинхронной памяти контроллер памяти должен учитывать задержку реакции памяти после ее адресации, конкретную для каждого типа микросхем. Обычно это осуществляется регулировкой длительности стробов RAS и CAS.

Результатом дальнейшего развития технологий DRAM стало создание синхронных DRAM (Synhronous DRAM или SDRAM). Они получили это название потому, что процессы чтения – записи данных в них синхронизированы тактовыми сигналами (Clok- Clk) . Запоминающие элементы в микросхемах SDAM точно такие же, как и в асинхронных DRAM. Однако благодаря синхронизации процессов чтения – записи данных с фронтами тактовых сигналов контроллер памяти “знает” моменты готовности данных. Это позволяет повысить скорость обмена между процессором и памятью при пакетных пересылках и упростить взаимодействие памяти и других устройств ЭВМ. Перечисленные и другие факторы позволяют современным SDRAM работать на высокой тактовой частоте, достигающей 400 и более МГц.

В состав SDRAM включают встроенную схему регенерации, которая содержит счетчик адреса, формирующий адрес строки, требующей регенерации. В типичной SDRAM данные регенерируются по меньшей мере каждые 64 мс. Включение схемы регенерации в состав SDRAM позволяет сохранять их содержимое в “спящем” режиме работы ЭВМ, когда для экономии энергопотребления включенного, но не используемого пользователем компьютера отключается часть его подсистем, при этом память переводится в режим саморегенерации.

7.4 Банковая память

Наличие в системе множества микросхем памяти позволяет использовать потенциальный параллелизм, заложенный в такой организации. Для этого микросхемы памяти часто объединяются в банки или модули, содержащие фиксированное число слов, причем только к одному из этих слов банка возможно обращение в каждый момент времени. Чтобы получить большую скорость доступа к данным, нужно осуществлять его одновременно ко многим банкам памяти.

Банком памяти называют комплект микросхем или модулей памяти, обеспечивающий требуемую для данного процессора разрядность хранимых данных. Типовая структура банковой памяти показана на рисунке 7.8. Адресное пространство памяти разбито на группы последовательных адресов, и каждая такая группа обеспечивается отдельным банком памяти. Для обращения к банку M_i используется k-разрядный адрес, а m младших разрядов поступают параллельно на все банки памяти и выбирают в каждом из них одну ячейку. Выбор банка обеспечивается с помощью дешифратора номера банка памяти (на рисунке не показан). В функциональном отношении такая память может рассматриваться и как единое ЗУ, емкость которого равна суммарной емкости составляющих банков.

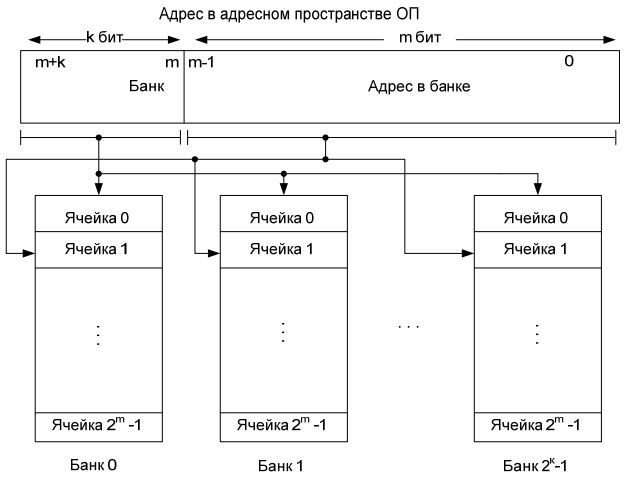


Рисунок 7.8 - Декодирование адреса при обращении процессора к ячейкам памяти с последовательными адресам

7.5 Память с чередованием адресов

Если установленная в ЭВМ память состоит из нескольких банков памяти, то появляется возможность повышения производительности системы в целом за счет использования режима чередования банков. Такой способ адресации банков приведен на рисунке 7.9. Младшие k бит адреса выбирают один из банков, а старшие m бит- ячейку памяти в этом банке. При таком способе адресации последовательные адреса адресного пространства процессора указывают на разные банки. Процессор, генерирующий запросы на доступ к последовательным адресам памяти, будет поддерживать одновременную активность нескольких банков.

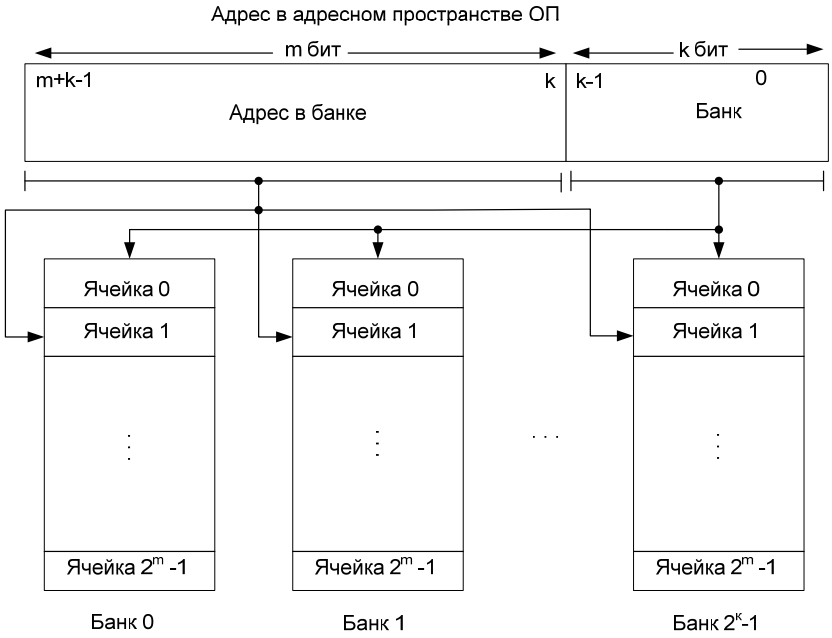


Рисунок 7.9 - Декодирование адреса при обращении процессора к ячейкам памяти с чередующимися адресами

При последовательных обращениях по смежным адресам адресного пространства памяти банки будут работать поочередно. Это позволяет осуществлять запись или чтение очередных данных не дожидаясь восстановления содержимого строки микросхемы DRAM ранее выбранного банка. Благодаря этому ускоряется доступ к блоку данных и эффективнее используется вся система памяти. Чередование адресов памяти возможно лишь при условии, что

количество модулей равно 2k, иначе в адресном пространстве будут области, не поддерживаемые установленной в систему памятью.

3.7 Основные характеристики ЗУ

1.*Емкость памяти*. Является важнейшей характеристикой ЗУ любого типа. Она определяет максимальное количество информации, которое может в ней храниться. Емкость может измеряться в битах, байтах или машинных словах. Наиболее распространенной единицей измерения является байт. При большом размере памяти ее емкость выражают в килобайтах (Кбайт) – 1024 байт, в мегабайтах (Мбайт) – миллион байт (точнее 1024*1024 байт), в гигабайтах (Гбайт) – миллиард байт.

2.*Время обращения к памяти.* Существует два вида времени обращения к памяти: при чтении и при записи. Их значения могут быть различными из-за того, что процессы чтения и записи отличаются процедурами, которые сопровождают эти операции.

Время обращения при чтении:

$$t_0^{чт} = t_д + t_{чт} + t_{рег},$$

где $t_д$ - время доступа (подготовительное время) - промежуток времени между началом операции обращения и моментом начала процесса чтения;

$t_{чт}$ - продолжительность физического процесса считывания;

$t_{рег}$ - время регенерации (восстановления), если в процессе чтения информации или из-за утечек заряда происходит её разрушение.

Время обращения при записи:

$$t^{зп}{}_0 = t_д + t_п + t_{зп},$$

где $t_п$ - время подготовки, расходуемое на приведение запоминающих элементов в исходном состоянии, если это необходимо;

$t_{зп}$ - время, необходимое для физического изменения состояния запоминающих элементов при записи информации.

3.*Цикл памяти.* Принимается равным минимально допустимому интервалу между двумя обращениями в память:

$$t_ц = \max\left(t_0^{чт}, t_0^{зп}\right).$$

4. *Время ожидания.* Под временем ожидания памяти или латентностъю (latency) понимается время, уходящее на пересылку в память или из памяти одного слова данных. Причем, если данные считываются и записываются пословно, то латентность полностью характеризует производительность памяти. Однако в современных компьютерах, использующих кэш, данные передаются из основной памяти блоками в несколько слов (строками кэш), которые называют пакетами. Для пакетных передач полное время, уходящее на выполнение операции, зависит от скорости пересылки отдельных слов и от размера блока данных. Поэтому при блочной пересылке под временем ожидания подразумевается время пересылки первого слова данных. Обычно это слово пересылается значительно дольше следующих слов блока.

5.*Пропускная способность.* Пропускную способность памяти можно определять количеством бит или байтов, пересылаемых за одну секунду. При пакетном способе обмена для оценки пропускной способности необходимо знать, сколько времени уходит на пересылку блока данных.

Пропускная способность подсистемы памяти, состоящей из одной или более микросхем, зависит от времени доступа к хранящимся в памяти данным, и от количества параллельно доступных бит. Однако реальная пропускная способность памяти определяется не только её быстродействием. Она зависит и от пропускной способности соединений между памятью и процессором, то есть в наиболее типичном случае от пропускной способности шины. Микросхемы

памяти обычно разрабатываются с учетом скорости функционирования шин, пропускная способность которых зависит от количества линий шины данных или, по другому, от ширины шины.

6.Тайминги памяти

Для современных микросхем памяти время задержки характеризуют *таймингами,* которые тесно связаны с задержками, возникающими при любых операциях с памятью. Измеряют тайминги в тактах системной шины. Схема доступа к данным микросхемы SDRAM состоит из нескольких действий:

1)Активизация строки. Перед осуществлением любой операции с данными, содержащимися в определенном банке микросхемы SDRAM, необходимо «активизировать» соответствующую строку в соответствующем банке. С этой целью, на микросхему подается команда активизации (ACTIVATE) вместе с номером банка (линии BA0-BA1 для 4-банковой микросхемы) и адресом строки (адресные линии A0-A12, реальное количество которых зависит от количества строк в банке.

Активизированная строка остается открытой (доступной) для последующих операций доступа до поступления команды подзарядки банка (PRECHARGE), по сути, закрывающей данную строку. Минимальный период «активности» строки — от момента ее активации до момента поступления команды подзарядки, определяется минимальным *временем активности строки* (Row Active Time, t_{RAS}).

Повторная активизация какой-либо другой строки того же банка не может быть осуществлена до тех пор, пока предыдущая строка этого банка остается открытой (т.к. усилитель уровня, содержащий буфер данных размером в одну строку банка является общим для всех строк данного банка микросхемы SDRAM). Таким образом, минимальный промежуток времени между активизацией двух различных строк одного и того же банка определяется *минимальным временем цикла строки* (Row Cycle Time, t_{RC}).

В то же время, после активизации строки выбранного банка микросхеме SDRAM возможна активация какой-либо другой строки другого банка (в этом и заключается рассмотренное выше преимущество «многобанковой» структуры микросхем SDRAM) на следующем такте шины памяти. Тем не менее, в реальных условиях производителями устройств SDRAM обычно здесь также умышленно вводится дополнительная задержка, именуемая *«задержкой от активации строки до активации строки»* (Row-to-Row Delay, t_{RRD}). Причины введения этой задержки не связаны с функционированием микросхем памяти как таковых и являются чисто электрическими — операция активизации строки потребляет весьма значительное количество электрического тока, в связи с чем частое их осуществление может приводить к нежелательным избыточным нагрузкам устройства по току.

2)Чтение/запись данных. Следующий временной параметр функционирования устройств памяти возникает в связи с тем, что активизация строки памяти сама по себе требует определенного времени. В связи с этим, последующие (после ACTIVATE) команды чтения (READ) или записи (WRITE) данных не могут быть поданы на следующем такте шины памяти, а лишь спустя определенный временной интервал, называемый *«задержкой между подачей адреса строки и столбца»* (RAS#-to-CAS# Delay, t_{RCD}).

Итак, по прошествии интервала времени, равного t_{RCD}, при чтении данных в микросхему памяти подается команда READ вместе с номером банка (предварительно активизированного командой ACTIVATE) и адресом столбца. Устройства памяти типа SDRAM ориентированы на чтение и запись данных в пакетном (Burst) режиме. Это означает, что подача всего одной команды READ (WRITE) приведет к считыванию из ячеек (записыванию в ячейки) не одного, а сразу нескольких подряд расположенных элементов, или «слов» данных (разрядность каждого из которых равна ширине внешней шины данных микросхемы — например, 8 бит). Количество элементов данных, считываемых одной командой READ или записываемых одной командой WRITE, называется *«длиной пакета»* (Burst Length) и обычно составляет 2, 4 или 8 элементов (за исключением экзотического случая передачи целой строки (страницы) — «Full-Page Burst», когда необходимо дополнительно использовать специальную команду BURST TERMINATE для прерывания сверхдлинной пакетной передачи данных).

Необходимо отметить, что существует две разновидности команды чтения. Первая из них является «обычным» чтением (READ), вторая называется «чтением с автоматической подзарядкой» (Read with Auto-Precharge, «RD+AP»). Последняя отличается тем, что после завершения пакетной передачи данных по шине данных микросхемы автоматически будет подана команда подзарядки строки (PRECHARGE), тогда как в первом случае выбранная строка микросхемы памяти останется «открытой» для осуществления дальнейших операций.

После подачи команды READ, первая порция данных оказывается доступной не сразу, а с задержкой в несколько тактов шины памяти, в течение которой данные, считанные из усилителя уровня, синхронизируются и передаются на внешние выводы микросхемы. Задержка между подачей команды чтения и фактическим «появлением» данных на шине считается наиболее важной и именуется *«задержкой сигнала CAS#»* (CAS# Latency, t_{CL}). Последующие порции данных (в соответствии с длиной передаваемого пакета) оказываются доступными без каких-либо дополнительных задержек, на каждом последующем такте шины памяти.

Операции записи данных осуществляются аналогичным образом. Точно также существуют две разновидности команд записи — простая запись данных (WRITE) и запись с последующей автоматической подзарядкой строки (Write with Auto-Precharge, «WR+AP»). Точно также при подаче команды WRITE/WR+AP на микросхему памяти подаются номер банка и адрес столбца. Точно также запись данных осуществляется «пакетным» образом. Отличия операции записи от операции чтения следующие. Во-первых, первую порцию данных, подлежащих записи, необходимо подать по шине данных одновременно с подачей по адресной шине команды WRITE/WR+AP, номера банка и адреса столбца, а последующие порции, количество которых определяется длиной пакета — на каждом последующем такте шины памяти. Во-вторых, вместо «задержки сигнала CAS#» (t_{CL}) важной здесь является иная характеристика, именуемая *«периодом восстановления после записи»* (Write Recovery Time, t_{WR}). Эта величина определяет минимальный промежуток времени между приемом последней порции данных, подлежащих записи, и готовности строки памяти к ее закрытию с помощью команды PRECHARGE. Если вместо закрытия строки требуется последующее считывание данных из той же самой открытой строки, то приобретает важность другая задержка, именуемая *«задержкой между операциями записи и чтения»* (Write-to-Read Delay, t_{WTR}).

3)Подзарядка строки. Цикл чтения/записи данных в строки памяти, который в общем случае можно обозначить «циклом доступа к строке памяти», завершается закрытием открытой строки банка с помощью команды подзарядки строки — PRECHARGE (которая может быть «автоматической», т.е. являться составной частью команд «RD+AP» или «WR+AP»). Последующий доступ к этому банку микросхемы становится возможным не сразу, а через некоторый интервал времени, называемый *«временем подзарядки строки»* (Row Precharge Time, t_{RP}). За этот период времени осуществляется собственно операция «подзарядки», т.е. возвращения элементов данных, соответствующих всем столбцам данной строки с усилителя уровня обратно в ячейки строки памяти.

4) *Схемы таймингов.* Четыре важнейших параметра таймингов памяти, расположенных в такой последовательности: t_{CL}-t_{RCD}-t_{RP}-t_{RAS}, принято называть «схемой таймингов». Такие схемы (например, 2-2-2-5 или 2.5-3-3-7 для памяти типа DDR; 3-3-3-9, 4-4-4-12 и 5-5-5-15 для памяти типа DDR2) достаточно часто можно встретить в спецификациях на модули оперативной памяти.
Строго говоря, такая последовательность не соответствует фактической последовательности возникновения задержек при доступе в микросхему памяти (так, t_{RCD} располагается перед t_{CL}, а t_{RAS} — «где-то посередине»), поэтому на самом деле она отражает основные тайминги памяти, расположенные в порядке их значимости. Действительно, наиболее значимой является величина задержки CAS# (t_{CL}), проявляющая себя при каждой операции чтения данных, тогда как параметры t_{RCD} и t_{RP} актуальны лишь при операциях на уровне строки памяти в целом (ее открытия и закрытия, соответственно).

3.8 ОЗУ магазинного типа (стек)

Магазинная или стековая память организуется по принципу “Первый пришел, первый вышел” (FIFO- First In First Out) или “Последний пришел, первый вышел” (LIFO- Last In First Out). (рисунок 3.9).

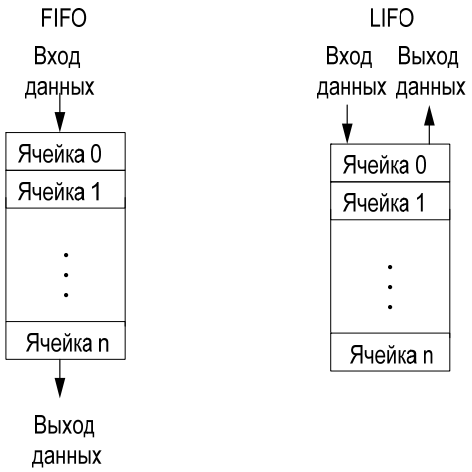


Рисунок 3.9 – Принцип занесения и выборки данных в магазинной (стековой) памяти

В памяти, выполненной по принципу FIFO, новое слово заносится в верхнюю ячейку, ранее записанные слова выталкиваются вниз. То есть это обычная очередь, которую можно наблюдать, например, в магазине. Возможно, поэтому такая организация памяти получила название магазинной.

В памяти типа LIFO новое слово заносится в верхнюю ячейку, ранее занесенные данные проталкиваются вниз. При считывании наоборот, последнее слово выталкивается вверх первым. Такую память называют стековой.

Стековая память широко используется в ЭВМ для сохранения содержимого регистров процессора при обработке запросов прерывания и вызове подпрограмм. Стековая память либо включается в состав процессора отдельным аппаратным блоком, либо реализуется аппаратно-программным путем.

Для адресации стека используется специальный регистр адреса, который называют указателем стека УС или SP- Steak Pointer (см. рисунок 3.10):

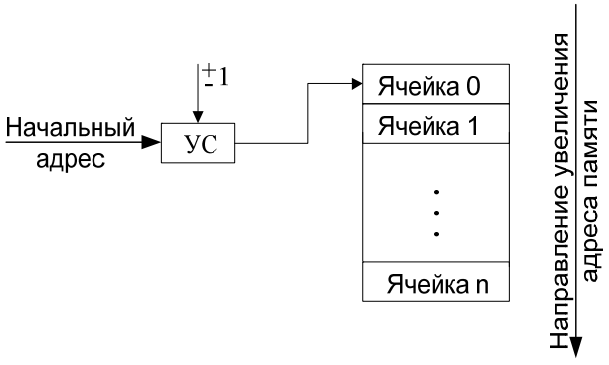


Рисунок 3.10 - Адресация стека с помощью УС

Перед началом работы в указатель стека заносится адрес ячейки ОП, в которую будет записываться первое слово (начальный адрес). Дальнейшая адресация осуществляется путем увеличения адреса на единицу при выполнении операций записи в стек (например, командой

PUSH). Уменьшение адреса на единицу производится при выполнении операции чтения содержимого стека (например, командой POP). Физический же процесс записи и считывания данных происходит точно так же, как в обычной памяти с произвольным доступом.

Возможные изменения состояния УС стековой памяти типа LIFO при записи- чтении показаны на следующем рисунке:

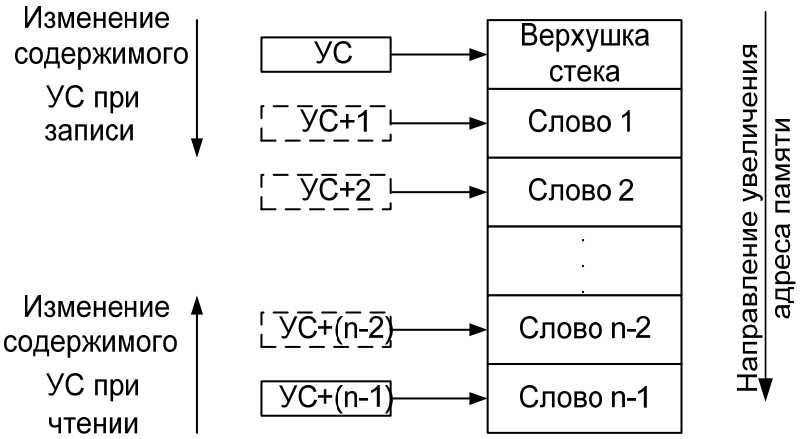


Рисунок 3.11 - Изменение состояния УС при записи и чтении стековой памяти типа LIFO

В результате правильного выполнения операций сохранения - восстановления регистров процессора, когда число записанных и считанных слов равны, стек приходит в исходное состояние. В том случае, когда число слов, записанных в стек и считанных из стека не равны, может произойти сбой в работе программы. Следует отметить, что верхушка стека при такой организации всегда остается пустой.

3.9 Ассоциативные ЗУ

В современных вычислительных системах широко используются операция поиска информации. При использовании обычной памяти с адресным принципом доступа к данным эта операция занимает много времени, поскольку операнды считываются из памяти поочередно (последовательно), после чего над каждым операндом производится операция сравнения. Это обстоятельство является фактором, увеличивающим время поиска. Решение проблемы заключается в том, чтобы эти операции выполнялись одновременно (параллельно). Принцип ассоциативного поиска поясняет рисунок 3.12.

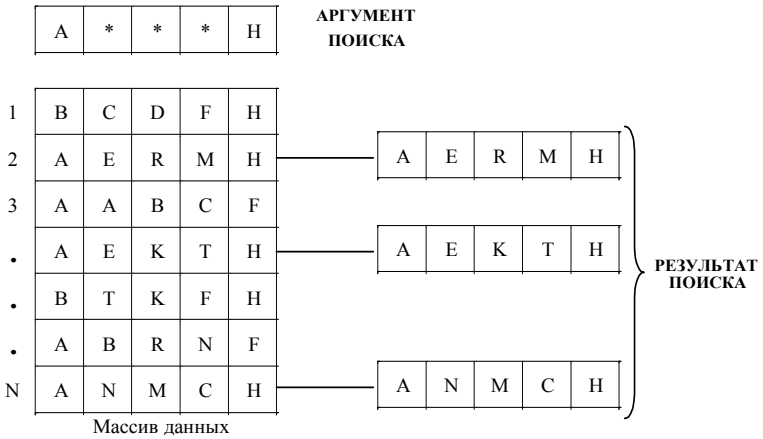


Рисунок 3.12 - Принцип ассоциативного поиска информации

С целью ускорения поиска данных используется адресация по содержанию, которая осуществляется путем одновременного доступа ко всем ячейкам памяти. Сущность принципа адресации по содержанию заключается в следующем (см. рисунок 3.12). Имеется массив данных

емкостью N слов. Требуется найти в массиве все слова, которые начинаются с символа "А" и кончаются символом "Н". В этом случае аргументом поиска (ключевым словом, компарандом) является слово А***Н, где значком * отмечены разряды, не влияющие на результат поиска. Запоминающий массив на аппаратном уровне строится таким образом, что бы на выходе ячеек памяти, содержимое которых совпадает со значением поступившего аргумента поиска, появлялся сигнал - указатель совпадений. В дальнейшем по выработанным сигналам выполняется выборка содержимого тех ячеек памяти, в которых произошло совпадение.

В виду высокой стоимости и сложности технической реализации такого способа адресации, ассоциативная память используется не везде, а в технически обоснованных случаях, например, в устройствах буферизации данных при выполнении обменных операций (в КЭШ- памяти и подобных устройствах). Кроме того, существуют специальные ассоциативные процессоры (сопроцессоры) где аппаратно реализуются операции свертки, поиска, сортировки, часто встречающиеся в программах пользователя или операционной системы.

В параллельных ассоциативных ЗУ (АЗУ) процесс поиска данных по содержанию организуется следующим образом. Каждая ячейка модуля памяти АЗУ обеспечивает выполнение функций приема, хранения данных, сравнения хранимой информации с аргументом поиска и выработку сигналов о результате сравнения. Модуль памяти АЗУ организован таким образом, что на каждом такте работы аргумент поиска поступает параллельно во все ячейки памяти. В результате в модуле памяти АЗУ выполняется массовая операция сравнения содержимого ячеек памяти с аргументом поиска и установка - сигналов указателей о совпадении на выходе.

Каждая строка модуля памяти такого АЗУ содержит (см. рис. 3.13) регистр для хранения слова данных (как в обычных ОЗУ) и специальные комбинационные логические схемы для сравнения текущего содержимого регистра с ключевым словом, которое поступает одновременно на все ячейки. При поиске формируется сигнал чтения из всех ячеек строки АЗУ с одновременным сравнением прочитанного слова с ключевым

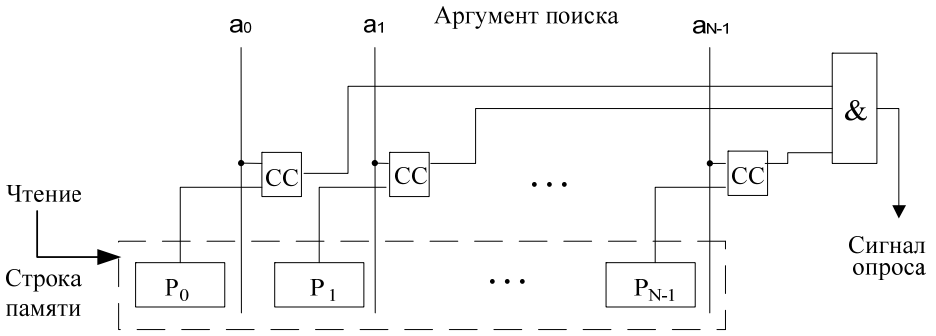


Рисунок 3.13 - Устройство ячейки АЗУ

словом (аргументом поиска). Сигнал опроса появится на выходе ячейки, содержимое которой совпадает с ключевым словом.

При построении АЗУ с маскированием, в операции сравнения участвуют не все разряды ключевого слова, а только та их часть, которая активизируется управляющим вектором-маской. Ячейка памяти переводится в активное состояние, если соответствующий ей разряд маски единичный, в противном случае она в операции сравнения не участвует.

Аппаратная реализация ассоциативного поиска должна обеспечить, как минимум, следующие функции:

-обеспечивать запись исходных данных в модуль памяти;

-осуществлять операцию поиска элемента по некоторому ключевому слову;

-выполнять занесение ключевого слова в регистр компаранда и код маски в регистр маски;

-возвращать в исходное состояние память отклика (регистр признаков);

-анализировать многократность совпадений и осуществлять выборку по совпадению.

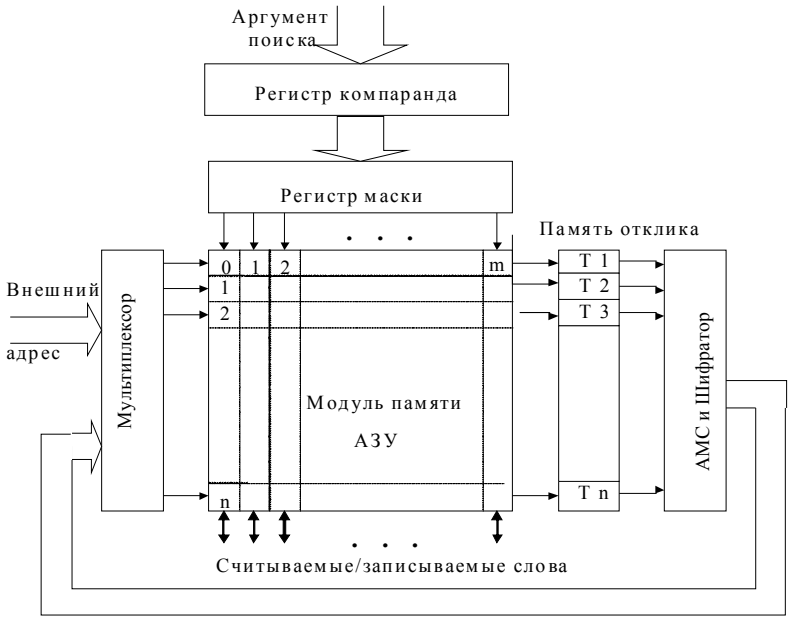


Рисунок 3.14 - Структура АЗУ

При выборке данных из АЗУ в регистр компаранда (см. рис. 3.14) вводится аргумент поиска, в регистр маски – код вектора управления. Содержимое всех разрядов регистра компаранда (если не применяется маска) параллельно поступают в ячейки всех строк модуля памяти АЗУ и выполняется операция сравнения. По окончании переходных процессов на выходах тех строк модуля памяти АЗУ, в которых произошло совпадение с заданным аргументом поиска, устанавливаются сигналы совпадения. Каждая ячейка модуля памяти связана с памятью отклика (регистром признаков) с помощью тегового разряда Ti. Перед началом работы все разряды регистра признаков устанавливаются в состояние “0”. По команде процессора “Сравнить” любая ячейка, содержащая слово, которое совпадает с компарандом, формирует сигнал, устанавливающий соответствующий разряд Ti в состояние “1”. Эта информация является адресной для линейной выборки. В анализаторе многократных совпадений (АМС) выполняется приоритетный разбор теговых разрядов и по его результатам последовательное формирование сигналов для шифратора. В принципе сигналы с АМС можно подавать непосредственно на адресные шины модуля памяти АЗУ, что обеспечило бы доступ ко всем "откликнувшимся" ячейкам одновременно. Однако на практике модули памяти АЗУ изготовляются из микросхем с встроенными дешифраторами адреса. Это позволяет использовать АЗУ в режиме прямо адресуемой памяти, что необходимо для занесения исходного массива данных перед началом операции ассоциативного поиска.

Ввод данных в АЗУ выполняется после предварительного поиска в модуле памяти АЗУ свободных строк. С этой целью в каждой строке модуля памяти предусматривают специальный бит занятости, который устанавливается в единицу при вводе слова данных в АЗУ и обнуляется при выводе соответствующего слова.

7.7 КЭШ-память

Повышение производительности процессоров привело к тому, что основная память, построенная на микросхемах DRAM, стала замедлять дальнейшее повышение производительности ЭВМ в целом. Реализация ОП на микросхемах SRAM технически и экономически не оправдана, так как габариты и стоимость микросхем SRAM на 1 бит хранимой информации существенно выше, чем аналогичные показатели у DRAM. Разумным компромиссом для построения экономичных и быстродействующих систем явилось сочетание памяти большого объёма на DRAM и небольшой на микросхемах SRAM.

Слово Cache означает склад, тайник. КЭШ- память не имеет отдельного адресного пространства и не доступна для пользователя. Она является дополнительным и быстродействующим хранилищем копий тех областей информации ОП, к которым, вероятно, в ближайшее время будет обращение. В их число попадают в первую очередь области, примыкающие к выполняемой в данный момент команде, а во вторую – области связанные с ней командами перехода (см. рисунок 7.11).

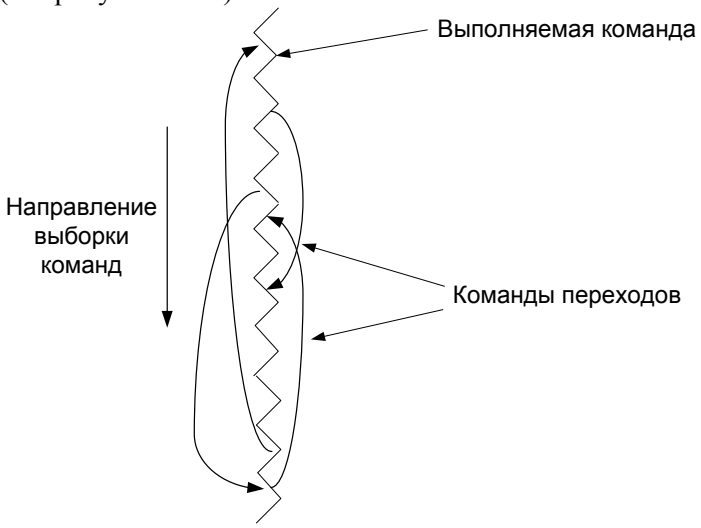


Рисунок 7.11- Возможная область кэширования ОП

Приведенный рисунок, а также анализ хода выполнения различных программ показывают, что большую часть времени в них выполняются определенные группы команд, которые многократно повторяются. Это свойство программ называется локализацией ссылок. Локализация ссылок происходит во времени и в пространстве. Первое означает, что недавно выполненные команды скорее всего будут затребованы снова. Локализация в пространстве означает, что скорее всего в последующие моменты времени будут выполняться команды, расположенные (по значениям адресов) в непосредственной близости от выполняемой. Заметим, что последовательность выполнения команд для работы КЭШ- памяти не имеет значения.

КЭШ не может хранить копию всей основной памяти, так как её размер во много раз меньше ОП. Поэтому она хранит копии части содержимого ОП. Для записи информации о текущем соответствии содержимого КЭШ-памяти конкретным областям (блокам) ОП используется каталог, находящийся в дополнительной тэговой (ТЭГ) памяти, входящей в состав КЭШ- памяти. При обращении к ОП контроллер КЭШ- памяти (ККП) с помощью каталога в ТЭГ проверяет, есть ли копия затребованных данных (или команды) в КЭШе. Если она там есть, то это случай так называемого КЭШ- попадания и данные берутся из КЭШа. Если нет (случай КЭШ - промаха), то данные берутся из основной памяти, вводятся в процессор и записываются в КЭШ. При попадании в КЭШ время доступа к подсистеме памяти КЭШ+DRAM уменьшается и основная память представляется процессору более быстрой, чем есть на самом деле.

7.7.1 КЭШ прямого отображения

Принцип работы КЭШ прямого отображения проиллюстрирован на рисунке 7.12.

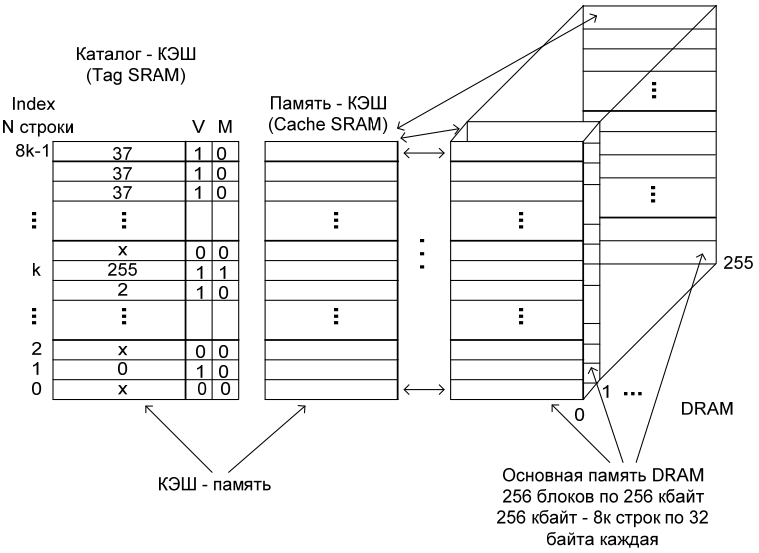


Рисунок 7.12- Структура КЭШ прямого отображения

В таблице 7.2 приведено разбиение полного адреса ячейки DRAM на поля.

Таблица 7.2- Распределение полного адреса ячейки DRAM

Разряды адреса ячейки памяти		
TAG	INDEX	
25... 18	17… 5	4… 0
Номер страницы (блока) ОП	Номер строки в блоке ОП	Выбор байта в строке

Рассмотрим принцип работы КЭШ прямого отображения на примере КЭШ объёмом 256 Кбайт с размером строки 32 байта (4 х 8 байт) и объёмом кэшируемой основной памяти 64 Мбайт. Кэшируемая основная память при этом условно разбивается на блоки, размер которых равен размеру КЭШ – памяти. Для рассматриваемого случая количество блоков равно: 64 Мбайт/ 256 Кбайт = 256. В свою очередь КЭШ- память делится на строки, длина которых равна количеству байт, передаваемых процессором в одном пакетном цикле (пакете) (4 х 4 байт =16 для процессора INTEL 486 и 4 х 8 байт = 32 байта для Pentium - процессоров). При размере КЭШ в 256 Кбайт число строк КЭШ для Pentium будет равно 256 Кбайт / 32 байт = 8 К .

В КЭШ прямого отображения средняя часть адреса (Index), по которому производится обращение, однозначно определяет строку КЭШ, в которой может находиться отображение соответствующей строки блока DRAM. На запись в каждую строку КЭШа могут претендовать только одноимённые строки всех блоков основной памяти (откуда и название данного типа КЭШ), но конкретная строка КЭШа в некоторый момент времени может содержать копию только одной строки некоторого блока ОП. Номер блока основной памяти, строка которого отображается в КЭШ, или старшая часть адреса (А18…А25 для КЭШ и ОП рассматриваемых размеров) оперативной памяти называется тегом (Tag) и хранится в дополнительной памяти тегов (Tag SRAM). Младшие пять разрядов адреса (для пакета в 32 байта) определяют номер байта в строке КЭШ памяти и для работы КЭШ памяти несущественны, т.к. минимальной единицей кэширования является строка.

Память тегов должна иметь количество ячеек, равное количеству строк КЭШа (объём КЭШ делится на длину строки КЭШ в байтах), а её разрядность должна вмещать старшие биты адреса кэшируемой памяти.

Кроме адресной части Tag с каждой строкой связаны биты признака действительности строки (V-Valid) и модифицированности данных (M-Mod).

В начале каждого обращения к кэшируемой памяти контроллер ККП считывает содержимое ячейки Tag с заданным индексом и сравнивает его со старшими битами D18…D25, адреса строки DRAM, подлежащей чтению. Сравнение осуществляется цифровым компаратором, входящим в состав КЭШ- контроллера. Если результат сравнения отрицателен (случай КЭШ- промаха), то вырабатывается цикл чтения основной памяти и считанные с нее данные вводятся в процессор, помещаются в соответствующую строку КЭШ, в Tag строки записывается старшая часть адреса, а бит V - достоверности, устанавливается в 1. В случае попадания в КЭШ при значении бита достоверности равным 1 данные берутся из КЭШ- памяти и обращение к DRAM не производится.

В начале цикла записи работа подсистемы памяти (КЭШ+DRAM) не отличается от цикла чтения. В случае промаха запись осуществляется сразу в строку DRAM, в случае попадания - в строку КЭШ (без изменения содержимого Tag соответствующей строки) и с установкой в 1 бита модифицируемости M строки КЭШ. Однако при этом нарушается согласованность (когерентность) данных в КЭШ и DRAM, что может привести к сбою работы всей системы и ККП должен выровнять содержимое КЭШ и DRAM до следующего обращения к этой строке DRAM. Поведение ККП в этом случае определяется его политикой записи.

Существуют два основных алгоритма записи данных из КЭШ - памяти в основную: сквозная запись WT (Write Through) и обратная (отложенная) запись WB (Write Back).

Алгоритм WT предусматривает выполнение каждой операции записи сразу и в КЭШ память и в DRAM. При этом каждая операция записи равносильна записи в DRAM и использование КЭШ- памяти не даёт никакого выигрыша при операциях записи (бит M в этом случае в составе КЭШ не используется).

При алгоритме WB запись со стороны процессора осуществляется только в строку КЭШ, при этом бит M устанавливается в 1, т. е. соответствующая строка в КЭШ отмечается как модифицированная или грязная, т. е. требующей записи в основную память. После копирования в DRAM строка становится чистой и бит M снова устанавливается в 0.

ККП старается осуществить копирование в промежутках между обращениями процессора к остальной системе, и только в крайнем случае (например, при попытке повторной записи в модифицированную строку КЭШ), копирование выполняется в первую очередь.

В системах с несколькими ведущими запись в DRAM могут осуществлять и другие устройства, например подсистема ПДП. В этом случае содержимое соответствующих строк КЭШ и DRAM также может оказаться различным, но уже из-за изменения строки DRAM.

Для устранения такой возможности адрес изменяемой строки DRAM при захвате шины другим ведущим передаётся в КЭШ второго (L2) и первого (L1) уровней. При этом процессор, находящийся в состоянии захвата его шины отслеживает адреса изменённых строк DRAM и в результате устанавливает биты их достоверности V в ноль.

7.7.2 Наборно-ассоциативный КЭШ

Его можно рассматривать как набор нескольких (например 4-х) КЭШ прямого отображения. В наборно-ассоциативной КЭШ - памяти каждая строка DRAM может размещаться в одной из нескольких строк КЭШ. В этом случае в состав TAG вводят ещё дополнительное поле, по которому ККП определяет, к какой строке КЭШ было самое давнее обращение, и которая следовательно, может быть заменена. Дополнительным усложнением является то, что старшая часть текущего адреса системы должна сравниваться с содержимым нескольких ТЭГ. КЭШ такого типа используются как внутренняя (L1) КЭШ память процессоров.

В процессорах Pentium внутренняя КЭШ- память имеет объем в 32 Кбайта. Она разбита на две равные части по 16 Кбайт - КЭШ кода программы и данных.

7.8 Контрольные вопросы

1. По каким признакам классифицируются запоминающие устройства?
2.Назначение ВЗУ и СОЗУ?

3.Назовите признаки ЗУ прямого и последовательного доступов?
4.Расшифруйте сокращения ПЗУ и ЗУПВ.
5.Перечислите основные характеристики ЗУ.
6.Что такое “Цикл памяти”?
7.Каковы преимущества ЗУ с произвольной выборкой?
8.Перечислите основные узлы ЗУПВ.
9.Какие отличия между памятью статического и динамического типа?
10.Перечислите основные узлы подсистемы регенерации.
11.Что такое регенерация DRAM?
12.Какова организация стековых ЗУ и где они применяются?
13.Что общего в работе стековой памяти типов LIFO и FIFO?
14.В чем заключается принцип действия ассоциативных ЗУ?
15.Назначение КЭШ- памяти?
16.Что обозначает название- КЭШ прямого отображения?
17.Назначение битов V и M в тэге КЭШ- памяти?
18.Отличия алгоритмов сквозной и отложенной записи?
19.Какими показателями КЭШ- памяти определяется объем кэширования ОП?

4. Арифметико- логические устройства

4.1 Классификация арифметико-логических устройств

Арифметико-логические устройства (АЛУ) предназначаются для выполнения арифметических и логических преобразований над операндами.

По способу действия над операндами АЛУ делятся на последовательные и параллельные. В последовательных АЛУ операнды поступают в последовательном коде (побитно), т.е. они являются одноразрядными. Недостатком последовательных АЛУ является низкое быстродействие, достоинством - простота реализации. В параллельных АЛУ операнды поступают в параллельном коде в виде 8-, 16-, 32- х и т.д. разрядных слов. Операция производится одновременно над всеми битами слова.

По структуре АЛУ делятся на АЛУ с непосредственными связями и АЛУ с магистральной структурой. В первом типе межрегистровые связи внутри АЛУ осуществляются непосредственно друг с другом; во втором - обмен между регистрами осуществляется через общую шину.

По способу организации работы различают асинхронные и синхронные АЛУ. В асинхронных АЛУ определяется момент окончания текущей операции, после чего может начинаться следующая операция. В синхронных АЛУ на выполнение любых операций отводится фиксированный интервал времени, независимо от того, какая операция выполняется- “короткая” (сложение, вычитание) или “длинная” (умножение, деление и др.). Первый тип АЛУ более быстродействующий, чем второй.

По характеру использования элементов и узлов АЛУ делятся на блочные и многофункциональные. В блочных АЛУ для каждого типа операции применяют специализированный блок обработки, например, блок умножения, блок деления, блок сложения - вычитания, блок логических операций, блок десятичной арифметики, блок арифметики с плавающей точкой и т.д. В многофункциональных АЛУ для выполнения всех типов операций используется один блок, выполняющий все виды операций. В блочных АЛУ быстродействие выше, поскольку каждый блок может выполнить операции параллельно с операциями в других блоках. Однако затраты оборудования при этом возрастают. В многофункциональных АЛУ используется общее оборудование для выполнения различных типов операций. Параллелизм при выполнении операций в многофункциональных АЛУ невозможен, поэтому их быстродействие невысокое, но затраты оборудования минимальны.

4.3.2 АЛУ для сложения и вычитания чисел с фиксированной запятой

Операция сложения в АЛУ обычно сводится к арифметическому сложению кодов чисел путём применения инверсных кодов - дополнительного или обратного для представления отрицательных чисел. Обратный код имеет два представления нуля (+0, -0), что затрудняет анализ результата операции. Поэтому чаще используется дополнительный код.

Алгоритмы выполнения в АЛУ арифметических операций зависят от того, в каком виде хранятся в памяти ЭВМ отрицательные числа - в прямом или дополнительном. В последнем случае сокращается время выполнения операции за счёт исключения преобразования получаемого в АЛУ дополнительного кода отрицательного результата в прямой код, при этом несколько усложняется операция умножения.

Алгоритм сложения двоичных чисел с фиксированной запятой, при использовании дополнительного кода для представления отрицательных чисел, заключается в следующем. Производится сложение двоичных кодов, включая разряды знаков. Если при этом возникает перенос из знакового разряда суммы при отсутствии переноса в этот разряд, или возникает перенос в знаковый разряд при отсутствии переноса из разряда знака, то происходит переполнение разрядной сетки. Такой результат получается как при отрицательной, так и положительной суммах. Если нет переносов из знакового и в знаковый разряд суммы или есть оба эти переноса, то переполнения нет. Если получен 0 в знаковом разряде, то сумма положительна и представлена в прямом коде. Если получена 1 в знаковом разряде, то сумма отрицательна и представлена в дополнительном коде.

На рис. 4.4 представлена упрощённая структурная схема

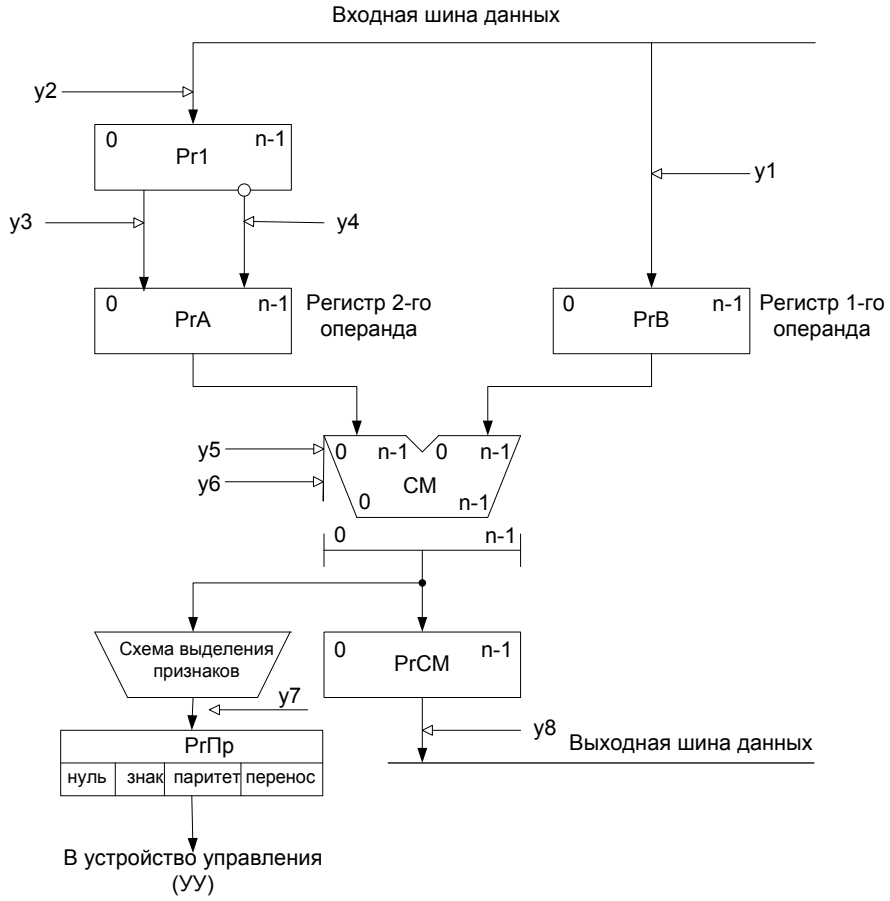


Рисунок 4.4- Структура АЛУ для операций сложения и вычитания

АЛУ для операций сложения и вычитания n-разрядных (нулевой разряд знаковый) двоичных чисел с фиксированной запятой. Предполагается, что отрицательные числа хранятся в памяти в дополнительном коде.

В состав АЛУ входят n-разрядный параллельный комбинационный сумматор См, регистр сумматора РгСМ, входные регистры сумматора РгА и РгВ, входной регистр АЛУ Рг1.

Операнды читаются из оперативной памяти и поступают в АЛУ по входной ШД (в дальнейшем- ШВХ). Для рассматриваемой структуры АЛУ положительные числа должны быть представлены прямым кодом, а отрицательные - дополнительным. Первый операнд размещается в РгВ (первое слагаемое или уменьшаемое), а второй в РгА (второе слагаемое или вычитаемое); Рг1 связан с РгА цепями прямой и инверсной передачи кода. Прямая передача используется при операции сложения, а инверсная - при операции вычитания. Результат операции выдается из АЛУ в оперативную память по выходной ШД (в дальнейшем- ШВЫХ).

При выполнении операции в АЛУ помимо результата операции формируется код признака результата ПР, который, например, может принимать следующие значения :

Результат операции	Регистр признаков			
	нуль	знак	паритет	перенос
0	1	0	0	0
<0	0	1	0	0
>0	0	0	0	0
Перенос	0	0	0	1

Примем, что код признака результата формируется комбинационной схемой выделения признаков, на входы которой поступают сигналы, соответствующие значениям всех разрядов сумматора, а также сигналы переносов из знакового разряда ПнСм[0] и в знаковый из старшего цифрового разряда ПнСм [1]. Признак переноса формируется при переполнении разрядной сетки машины (ПР=11), т. е. если булева функция

ПнСм[0] ^ ПнСм[1] v ПнСм[0] ^ ПнСм[1]=1.

Признак нулевого значения результата формируется если

$$\bigcap_{i=1}^{n-1} \overline{CM[i]} = 1$$

.

Условия выработки признаков положительного и отрицательного результатов имеют соответственно вид:

См[0] ^ (ПнСм[0] ^ ПнСм[1] v ПнСм[0] ^ ПнСм[1]);

См[0] ^ (ПнСм[0] ^ ПнСм[1] v ПнСм[0] ^ ПнСм[1]).

При выполнении операции сложения поступившие в АЛУ коды операндов находятся во входных регистрах РгВ и РгА сумматора. Код суммы формируется на выходах схемы СМ и фиксируется в регистре сумматора РгСм.

Операция вычитания операнда Y из операнда X Z=X-Y=X+(-Y) сводится к изменению знака вычитаемого Y и последующим применением операции сложения к операндам X и (-У). Изменению знака соответствует следующая процедура: принятый в Рг1 код числа передается в РгА по цепи инверсной передачи кода (линия с кружком на рисунке 4.4), а при сложении осуществляется подсуммирование 1 в младший разряд сумматора.

Передачи информации в регистрах АЛУ производятся отдельными микрооперациями, инициируемыми показанными на рис. 4.3.2 управляющими сигналами yi. Например, слово из Рг1 в РгА может быть передано в прямом (управляющий сигнал y3) или в инверсном (управляющий сигнал y4) кодах. Пусть для рассматриваемого АЛУ имеется набор следующих микроопераций:

y1: Рг В := ШВХ; прием 1-го операнда в Рг В с входной ШД.

y2: Рг 1 := ШВХ; прием 2-го операнда в Рг 1 с входной ШД.

y3: Рг А := Рг 1; прямая передача 2-го операнда из Рг1 в РгА.

y4: Рг $A := \overline{\text{Рг 1}}$; инверсная передача 2-го операнда из Рг1 в РгА.

y5: СМ := СМ+1;

y6: Рг СМ := СМ; передача результата из сумматора в Рг СМ.

y7: РгПр := Сх.Пр.; регистру признаков присвоить значение признака результата.

у8: ШВЫХ := Pr СМ; передача результата из РгСМ на выходную ШД.

Ниже приведены описания микропрограмм сложения и вычитания на языке регистровых передач, причем микрооперация у7 состоит в выдаче в регистр признаков (находится в управляющем устройстве) кода признака и в формировании запроса прерывания при переполнении разрядной сетки.

Микропрограмма сложения :

у1: Pr B := ШВХ; прием 1-го операнда в Pr B.

у2: Pr 1 := ШВХ; прием 2-го операнда в Pr 1.

у3: Pr A:= Pr 1; прямая передача 2-го операнда в РгА.

у6: Pr См:= Pr A + Pr B.

у7: РгПр:=СхПр; регистру признаков присваивается значение полученного признака результата, причем если Пр = 11, то прерывание программы по переполнению, иначе

у8: ШВЫХ := Pr См.

Микропрограмма вычитания :

у1: Pr B := ШВХ

у2: Pr 1 := ШВХ

у4: $\text{Pr}\,A := \overline{\text{Pr}\,1}$

у5: Pr См := Pr A + Pr B +1

у7: РгПр:=СхПр

Если Пр = 11, то прерывание программы по переполнению, иначе

у8: ШВЫХ := Pr См

4.3.3 АЛУ для умножения двоичных чисел

В ЭВМ операция умножения чисел с фиксированной запятой сводится к последовательности операций сложения и сдвига. Произведение двух (n- 1)- разрядных чисел может иметь до 2 (n - 1) значащих разрядов, поэтому для выполнения операции умножения целых чисел в АЛУ необходимо предусмотреть возможность формирования произведения, имеющего двойную, по сравнению с сомножителями, длину. В ЭВМ, в которых числа с фиксированной запятой являются дробями, младшие n- 1 разрядов произведения часто отбрасываются, при этом при отбрасывании может производиться операция округления произведения.

Для выполнения умножения АЛУ должно содержать регистры множимого, множителя и сумматор частичных произведений, в котором путем соответствующей организации передач производится последовательное суммирование частичных произведений.

Операция умножения состоит из n циклов, где n- число цифровых разрядов множителя. В каждом цикле анализируется очередная цифра множителя, и если она равна 1, то к сумме частичных произведений прибавляется множимое, в противном случае прибавление не происходит. Цикл завершается сдвигом множимого относительно суммы частичных произведений либо сдвигом суммы частичных произведений относительно неподвижного множимого.

В зависимости от способа формирования суммы частичных произведений различают четыре основных метода выполнения умножения и соответственно четыре структуры АЛУ для этой операции.

1. Умножение, начиная с младших разрядов множителя, со сдвигом суммы частичных произведений вправо и при неподвижном множимом (см. рисунок 4.5).

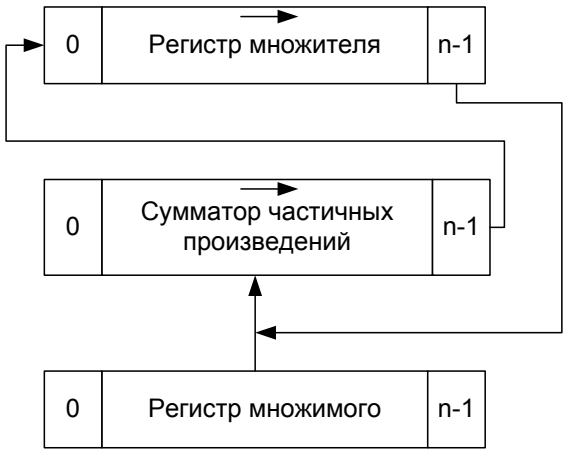


Рисунок 4.5 - Метод умножения, начиная с младших разрядов множителя со сдвигом суммы частичных произведений вправо

Регистр множителя и сумматор частичных произведений при этом должны иметь цепи сдвига вправо. Регистр множимого может не иметь цепей сдвига.

Последовательность действий в каждом цикле выполнения умножения определяется младшим разрядом регистра множителя, куда последовательно одна за другой поступают цифры множителя.

Поскольку по мере сдвига множителя вправо старшие разряды регистра множителя освобождаются, он может быть использован для хранения младших разрядов произведения, поступающих из младшего разряда сумматора частичных произведений по мере выполнения умножения. Для этого при выполнении сдвига младший разряд регистра сумматора частичных произведений соединяется со старшим разрядом регистра множителя. После выполнения умножения старшие разряды произведения находятся в регистре сумматора, младшие — в регистре множителя.

При данном методе умножения все три регистра имеют одинаковую длину, равную числу разрядов сомножителей. Этот метод умножения нашел наибольшее применение в ЭВМ.

2. Умножение, начиная с младших разрядов множителя, при сдвиге множимого влево и неподвижной сумме частичных произведений (см. рисунок 4.6).

Регистр множителя при этом должен иметь цепи сдвига вправо, регистр множимого — цепи сдвига влево, а сумматор частичных произведений не содержит цепей сдвига.

Последовательность действий определяется, как и в первом варианте, младшим разрядом регистра множителя.

При этом методе регистр множимого и сумматор частичных произведений должны иметь двойную длину. Этот метод требует больше оборудования, но никаких преимуществ не дает, и поэтому применение его нецелесообразно.

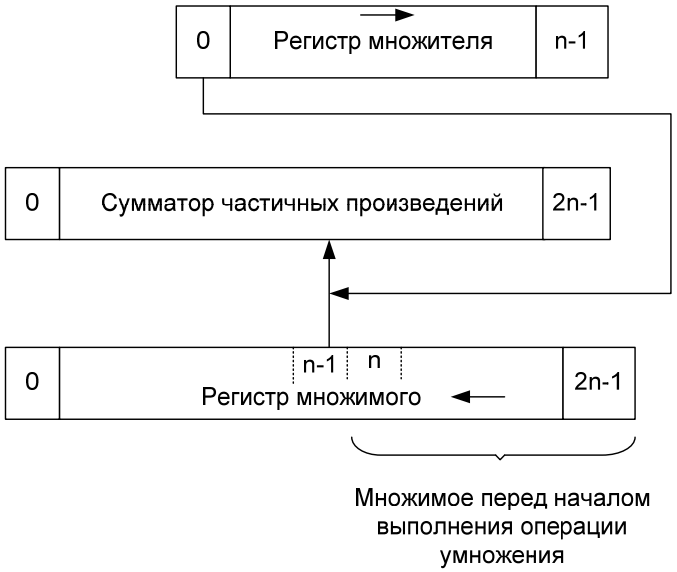


Рисунок 4.6 - Метод умножения, начиная с младших разрядов множителя, при сдвиге множимого влево

3. Умножение, начиная со старших разрядов множителя, при сдвиге суммы частичных произведений влево и неподвижном множимом (см. рисунок 4.7).

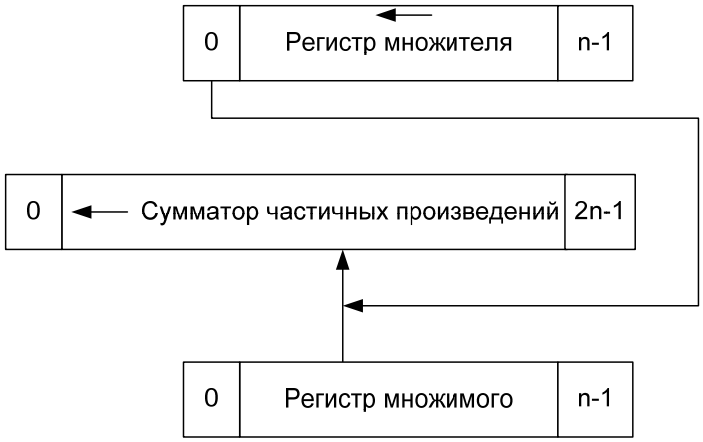


Рисунок 4.7 - Метод умножения, начиная со старших разрядов множителя при сдвиге суммы частичных произведений влево

Регистр множителя и сумматор частичных произведений должны иметь цепи сдвига влево. Регистр множимого не имеет цепей сдвига. Последовательность действий в каждом цикле выполнения умножения определяется старшим разрядом регистра множителя.

При использовании этого метода сумматор частичных произведений должен иметь двойную длину. Данный метод требует дополнительного по сравнению с первым методом оборудования. Несмотря на это, он применяется в некоторых АЛУ, так как позволяет без дополнительных цепей сдвига выполнять и деление.

Для выполнения операций деления в АЛУ, реализующем первый метод умножения, необходимы дополнительные цепи сдвига влево в регистре множимого (частного) и в сумматоре частичных произведений (разностей).

Умножение, начиная со старших разрядов множителя, при сдвиге вправо множимого и неподвижной сумме частичных произведений (см. рисунок 4.8).

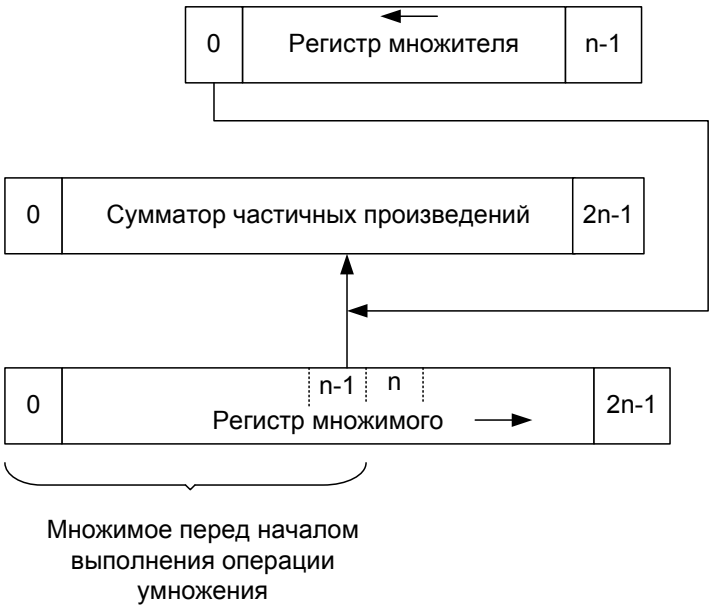


Рисунок 4.8 - Метод умножения, начиная со старших разрядов множителя при сдвиге множимого вправо

Регистр множителя должен иметь цепи сдвига влево, регистр множимого - цепи сдвига вправо. Сумматор частичных произведений не имеет цепей сдвига. Последовательность действий на каждом шаге умножения определяется старшим разрядом регистра множителя.

При этом методе умножения и регистр множимого, и сумматор частичных произведений должны иметь двойную длину. Однако, как и третий метод, он не требует дополнительных цепей сдвига для выполнения деления.

При четвертом методе, в котором сумма частичных произведений неподвижна, можно совмещать во времени операции сдвига и сложения и за этот счет увеличить быстродействие АЛУ при выполнении умножения (а также деления).

Если необходимо образование произведения двойной длины, например, при операциях с целыми числами, наиболее экономичным является первый из рассмотренных методов умножения, так как он позволяет использовать все регистры одинарной длины.

Если в результате умножения достаточно иметь произведение одинарной длины, то целесообразно использовать либо первый, либо четвертый метод умножения. При использовании первого метода требуется введение дополнительных цепей сдвига для реализации деления, а при использовании четвертого метода необходимо удлинение сумматора. Выбор одного из этих методов умножения определяется соотношением затрат оборудования на реализацию цепей сдвига и дополнительных разрядов сумматора.

При образовании произведений одинарной длины простое отбрасывание младших разрядов вносит погрешность, которая будет накапливаться, так как произведение будет всегда вычисляться с недостатком. Поэтому для повышения точности вычислений часто производят округление результата умножения, вследствие чего погрешность становится знакопеременной.

Для округления произведения длина сумматора частичных произведений обычно увеличивается на один разряд. После образования произведения к этому дополнительному разряду прибавляется 1. Если дополнительный разряд произведения был равен 0, то произведение в основных разрядах сумматора получается с недостатком. Если дополнительный разряд был равен 1, то в результате переноса 1 из дополнительного разряда к основным разрядам сумматора добавляется единица и произведение получается с избытком, при этом максимальное значение погрешности произведения равно половине 1 младшего разряда.

Следует отметить, что при любом методе умножения операция обычно начинается с анализа на 0 сомножителей. При равенстве нулю хотя бы одного сомножителя умножение не производится, а произведению присваивается нулевое значение.

Рассмотрим алгоритм наиболее распространенного метода умножения целых чисел, начиная с младших разрядов, со сдвигом суммы частичных произведений вправо.

1. Берутся модули от сомножителей (числа представляются без знака).

2. Исходное значение суммы частичных произведений принимается равным 0.

3. Если анализируемая цифра множителя равна 1, то к сумме частичных произведений прибавляется множимое; если эта цифра равна 0, прибавление не производится.

4. Производится сдвиг суммы частичных произведений вправо на один разряд.

5. Пункты 3 и 4 последовательно выполняются для всех цифровых разрядов множителя, начиная с младшего.

6. Произведению присваивается знак плюс, если знаки сомножителей одинаковы, в противном случае — знак минус.

Деление в ЭВМ обычно сводится к выполнению последовательности вычитаний делителя сначала из делимого, а затем из образующихся в процессе деления частичных остатков, и сдвига частичных остатков.

4.3.4 Методы ускорения умножения

Методы ускорения умножения делятся на аппаратурные и логические. Как те, так и другие требуют дополнительных затрат оборудования. При использовании аппаратурных методов дополнительные затраты оборудования прямо пропорциональны числу разрядов в операндах. Эти методы вызывают усложнение схемы АЛУ.

Дополнительные затраты оборудования при реализации логических методов ускорения умножения не зависят от разрядности операндов. Усложняется в основном схема АЛУ. В ЭВМ для ускорения умножения часто используется комбинации этих методов.

К аппаратурным методам ускорения умножения относятся ускорение выполнения операций сложения и сдвига, введение дополнительных цепей сдвига, позволяющих за один такт производить сдвиг информации в регистрах сразу на несколько разрядов, совмещение по времени операций сложения и сдвига, построение комбинационных схем множительных устройств, реализующих «табличное» умножение.

Среди логических методов наиболее распространены в настоящее время методы, позволяющие за один шаг умножения обработать несколько разрядов множителя.

Рассмотрим один из способов умножения на два разряда множителя, начиная с его младших разрядов. В зависимости от результата анализа пары разрядов множителя предусматриваются следующие действия. При 00 производится простой сдвиг на два разряда вправо суммы частичных произведений. При 01 к сумме частичных произведений прибавляется одинарное множимое и сумма частичных произведений сдвигается на два разряда вправо. Тогда в первых трех случаях результат получается правильный, а в последнем неправильный, он должен быть скорректирован на следующем шаге.

Поскольку при 11 из суммы частичных произведений вычитается одинарное множимое вместо прибавления утроенного множимого, для корректировки результата к сумме частичных произведений перед выполнением сдвига надо было бы прибавить учетверенное множимое. Но после сдвига на два разряда вправо сумма частичных произведений уменьшается в 4 раза, так что для корректировки его на следующем шаге должно быть прибавлено одинарное множимое.

Это учитывается при обработке следующей пары разрядов. Если следующая пара 00, то она обрабатывается как 01, если 01, то как 10, если 10, то как 11, если 11, то как 00, и фиксируется необходимостью коррекции при обработке следующей пары. Удвоенное множимое может быть получено его сдвигом. Признак необходимости коррекции может запоминаться в отдельном триггере коррекции.

Правила обработки пар разрядов множителя с учетом признака коррекции сведены в таблице 4.1

Таблица 4.1 - Правила обработки пар разрядов множителя с учетом признака коррекции

Пара разрядов множителя	Признак коррекции из предыдущей	Признак коррекции для следующей	Знак действия	Кратность множимому
00	0	0		0
01	0	0	+	1
10	0	0	+	2
11	0	1	-	1
00	1	0	+	1

Продолжение таблицы 4.1

01	1	0	+	2
10	1	1	-	1
11	1	1		0

После обработки каждой комбинации содержимое регистра множителя и сумматора частичных произведений сдвигается на два разряда вправо.

Данный метод умножения требует корректировки результата, если старшая пара разрядов множителя 11 или 10 и состояние триггера коррекции является единичным. В этом случае к полученному произведению должно быть добавлено множимое. Аналогичным образом можно организовать умножение с обработкой за шаг большего числа разрядов множителя.

4.3.5 Особенности операций десятичной арифметики

Арифметические операции над десятичными числами (сложение, вычитание, умножение, деление) выполняются аналогично операциям над целыми двоичными числами. Основой АЛУ десятичной арифметики является сумматор двоично-десятичных кодов. Такой сумматор, как правило, строится на основе двоичного путем добавления некоторых цепей.

Рассмотрим, каким образом можно выполнить сложение двоично-десятичных кодов. Пусть необходимо сложить модули двух двоично-десятичных чисел X и Y. Первое слагаемое X преобразуем в код с избытком 6 (обозначим X6), получаемый путем прибавления к каждой цифре X двоичного числа 6. Переход от X к X6 изменяет все тетрады X так, что в каждой тетраде X6 находится число 6 - 15.

Складывая X6 и Y по правилам двоичного сложения, получаем результат Z'. В Z' одни тетрады совпадают, а другие не совпадают с тетрадами двоично-десятичной суммы Z.

Если результат сложения в i-м разряде X [i] + Y[i] + P [i] >= 10, где P [i] - десятичный перенос в i-й разряд, то i-я десятичная цифра Z [i] = X [i] + Y[i] + P [i] – 10 и P [i + 1] = 1, где P [i + 1] - десятичный перенос в (i + 1)-й разряд. Для Z'[i] в этом случае получаем

Z'[i] = X6[1] + У[i] +P[i] - 16= 6+ X[i] + Y[i] + P[i] - 16 = Z[i].

При этом возникает перенос в (i+1)-ю тетраду.

Если 1-я десятичная цифра Z(i) должна получаться из X [i] + Y[i] + P [i] < 10, то Z[i] = X [i] + У[i] + P [i] и P[i+1] = 0.

Для Z'(i) в этом случае получаем

Z’[i] =X6[i] + Y[i] + P[i] = 6 + X[i] + У[i] + P[i] =Z[i] + 6.

Перенос в (i + 1)-ю тетраду здесь не возникает (P[i+1] = 0), так как

Z' [i] < 16.

Таким образом, складывая X6 и Y как двоичные числа, получаем Z'. В Z' тетрады, из которых возникал перенос, совпадают с тетрадами двоично-десятичного результата Z, а тетрады, из которых не было переноса при сложении, представлены с избытком 6. Для получения суммы Z необходимо откорректировать Z' путем уменьшения на 6 тех тетрад Z , из которых не было переноса при сложении X6 и Y.

Вычитание числа 6 из тетрад, требующих коррекции, можно реализовать путем подсуммирования числа 10 с одновременным игнорированием переноса, возникающего при этом из тетрад. Если Z' [i] нуждается в коррекции, то Z' [i] = Z[i]+ 6. Поэтому Z'[i] + 10 >= 16, значит, после прибавления 10 из тетрады возникнет перенос, т. е. в тетраде останется (Z' [i] + 10)-16 = Z[i]-6.

Вычитание двоично-десятичных модулей X-Y осуществляется следующим образом.

Все разряды Y инвертируются, что дает дополнение каждой цифры Y до 15, при этом получается обратный код двоично-десятичного Y с избытком 6, обозначенный Уобр6. Затем, складывая X + Уобр6 и прибавляя 1 к младшему разряду, получаем Z. Результат Z' является положительным числом, если из старшей тетрады его возникает перенос, при этом Z' корректируется по тем же правилам, что и при сложении модулей.

Если из старшей тетрады 2' нет переноса, то получен отрицательный результат, представленный в дополнительном коде. В этом случае код Z' инвертируется и к нему прибавляется 1 младшего разряда. Новое Z' корректируется, при этом к тетрадам, из которых

возникал перенос при получении (X + Уобр6 + 1), прибавляется 10, а к остальным не прибавляется.

Выполнение сложения и вычитания чисел со знаками сводится к выполнению сложения или вычитания модулей путем определения фактической выполняемой операции по знакам операндов и виду выполняемой операции. Знак результата определяется отдельно. Например, при $X < 0$ и $Y < 0$ вычитание X- Y заменяется вычитанием | Y | - | X |. Затем знак результата меняется на противоположный знаку (| Y | - | X |).

Двоично-десятичное умножение сводится к образованию и многократному сложению частичных двоично-десятичных произведений. Умножение двоично-десятичных чисел выполняется следующим образом:

1) сумма частичных произведений полагается равной нулю;

2) анализируется очередная цифра (тетрада) множителя, и множимое прибавляется к сумме частичных произведений столько раз, какова цифра множителя;

3) сумма частичных произведений сдвигается вправо на 1 тетраду, и повторяются действия, указанные в п. 2, пока все цифры множителя не будут обработаны.

Для ускорения умножения часто отдельно формируются кратные множимого 8X, 4X, 2X и IX, при наличии которых уменьшается число сложений при выполнении п. 2.

Двоично-десятичное деление выполняется путем многократных вычитаний, подобно тому, как это делается при обычном делении.

4.3.6 Операции над числами с плавающей запятой

Арифметические операции над числами с плавающей запятой более сложны, чем операции над числами с фиксированной запятой. Алгоритм сложения и вычитания чисел с плавающей запятой выглядит следующим образом:

Производится выравнивание порядков чисел. Порядок меньшего (по модулю) числа принимается равным порядку большего, а мантисса меньшего числа сдвигается вправо на число разрядов, равное разности порядков чисел.

Производится сложение (вычитание) мантисс, в результате чего получается мантисса суммы (разности).

Порядок результата принимается равным порядку большего числа.

Полученная сумма (разность) нормализуется. Выравнивание порядков начинается с их сравнения. Мантисса числа с меньшим порядком при выравнивании сдвигается вправо на число разрядов, равное разности порядков.

Сложение (вычитание) мантисс производится по правилам сложения (вычитания) чисел с фиксированной запятой.

При умножении чисел с плавающей запятой порядки сомножителей складываются, а мантиссы перемножаются. Произведение нормализуется, и ему присваивается знак плюс, если сомножители имеют одинаковые знаки, и знак минус, если знаки разные.

Если мантисса множимого или множителя равна 0, то произведению можно присвоить значение 0 без выполнения умножения мантисс. Если при суммировании порядков возникло переполнение и порядок отрицательный, то это означает, что произведение меньше минимального представляемого в машине числа, и в качестве результата операции может быть записан 0 без перемножения мантисс.

Если при суммировании порядков возникает переполнение и порядок положительный, может оказаться, что результат все-таки находится в диапазоне чисел, представляемых в машине, так как при умножении мантисс возможно нарушение нормализации вправо, и после нормализации мантиссы переполнение в порядке может исчезнуть.

При делении чисел с плавающей запятой мантисса частного равна частному от деления мантиссы делимого на мантиссу делителя, а порядок частного - разности порядков делимого и делителя. Частное нормализуется, и ему присваивается знак плюс, если делимое и делитель имеют одинаковые знаки, и знак минус, если разные.

Если делимое равно 0, то в частное может быть записан 0 без выполнения деления. Если при вычитании порядков образовалось переполнение с положительным знаком или если делитель равен 0, то деление не производится и формируется сигнал прерывания.

При делении нормализованных чисел с плавающей запятой может оказаться, что мантисса делимого больше мантиссы делителя, и мантисса частного образуется с переполнением. Для устранения этого явления перед делением мантисс нарушают нормализацию делителя сдвигом на разряд влево. Тогда нарушения нормализации частного влево не возникает.

5. Устройства управления

4.4.1 Классификация УУ

Устройство управления (УУ) управляет работой процессора, обеспечивая автоматическое выполнение команд программы. Выполнение команды процессором представляет собой последовательность следующих действий (иногда называемых машинными циклами):

-выборка команды из памяти и ее декодирование (дешифрация кода операции);

-формирование адреса следующей команды;

-формирование исполнительного адреса операнда и выборка его из памяти;

-исполнение операции и запись результата в память.

Для выполнения каждого машинного цикла необходим ряд управляющих сигналов, формируемых устройством управления.

В зависимости от способа формирования управляющих сигналов различают два основных типа УУ:

- аппаратные (с жесткой или схемной логикой);

- микропрограммные (с хранимой в памяти логикой).

В аппаратных УУ для каждой операции, задаваемой кодом операции команды, строится набор схем, которые в нужных тактах формируют соответствующие управляющие сигналы.

В УУ с микропрограммным управлением каждой операции соответствует набор микрокоманд, хранимых в памяти микрокоманд. Каждая микрокоманда несет информацию о микрооперациях, подлежащих выполнению в течение машинного такта и указания, какая микрокоманда должна быть выбрана из памяти следующей. Последовательность микрокоманд, выполняющая одну машинную команду или некоторую процедуру, образует микропрограмму.

4.4.2 Аппаратные УУ

Управляющие устройства с жесткой логикой представляют собой логические схемы, вырабатывающие распределенные во времени управляющие сигналы. В отличие от управляющих устройств с хранимой в памяти логикой в аппаратных УУ нельзя изменить логику работы без изменения их схемы. Типичная структурная схема управляющего автомата с жесткой логикой показана на рисунке 4.9. Данный УА можно рассматривать в качестве автомата с конечным числом состояний (конечный автомат), который на каждом такте переходит из одного

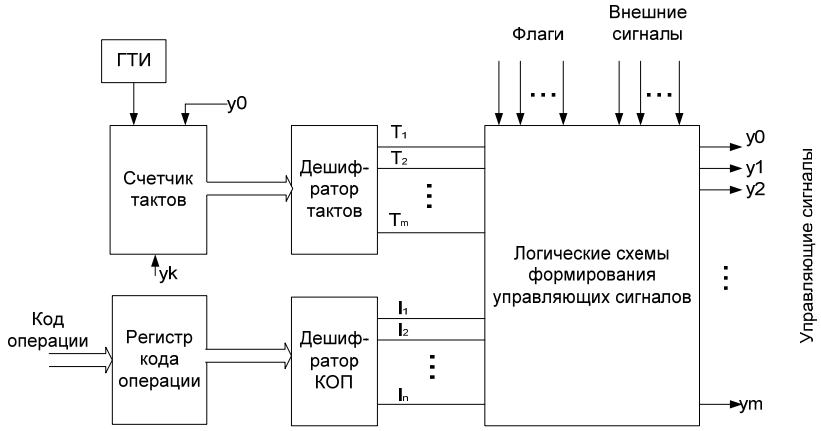


Рисунок 4.9 - Схема блока формирования сигналов управления

состояния в другое, определяемое содержимым регистра команды, кодами условий и внешними сигналами. Выходами такого автомата являются управляющие сигналы. Формируемая им последовательность операций задается физическими связями между логическими элементами.

В состав схемы входят регистр кода операции, являющейся частью регистра команд процессора, счетчик тактов, дешифратор тактов и дешифратор кода операции (Дешифратор КОП), а также логические схемы формирования управляющих сигналов. На счетчик тактов поступают сигналы от генератора тактовых импульсов (ГТИ). Состояние счетчика представляют собой

номера тактов, изменяющие от 1 до m. Дешифратор тактов формирует на i-м выходе единичный сигнал при i-м состоянии счетчика тактов, т.е. во время i-го такта.

Принцип построения логических схем формирования управляющих сигналов поясняется на рисунке 4.10. На нем изображен фрагмент схемы, обеспечивающий выработку управляющих сигналов y1, y2 и y3 выполнения команды сложения на тактах T_1,T_2, T_6.

В общем случае значения управляющих сигналов зависят еще от оповещающих сигналов $U=\{u_1, u_2 \dots u_n\}$, отражающих ход вычислительного процесса. Для реализации этих зависимостей логические элементы, представленные на рисунке 4.10, берутся многовходовыми и на них подаются требуемые сигналы логических условий.

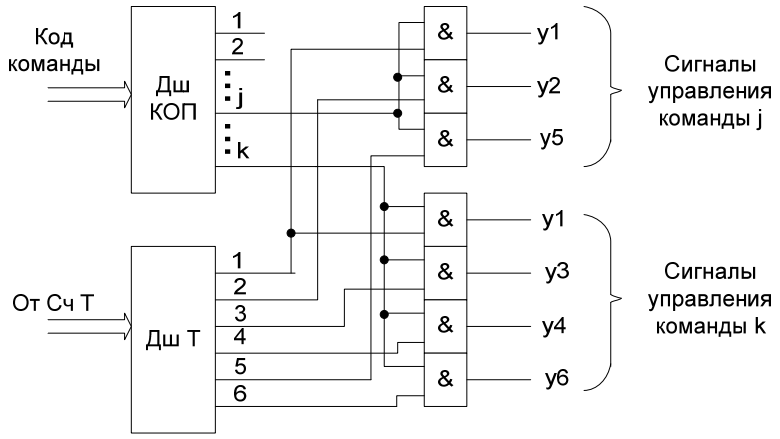


Рисунок 4.10 - Фрагмент логической схемы формирования управляющих сигналов

Сигналы y0 и yk (см. рисунок 4.9) обычно используются для определения моментов начала и окончания выполнения команд. С этой целью они используются для управления работой счетчика тактов. Управляющий сигнал y0 (Пуск) указывает на начало выполнения команды. Когда он установлен в 1, в конце каждого тактового цикла значение счетчика увеличивается на 1. Если же его значение становится равным 0, отсчет шагов прекращается. Сигнал yk (Останов) сбрасывает счетчик тактов в начальное состояние, обеспечивая начало нового цикла выборки команды.

Серьезным недостатком рассмотренных схем является одинаковое число тактов, необходимых для выполнения всех команд, значение которого выбирается по наиболее длинной команде. Это приводит к непроизводительным затратам времени и, как следствие, к уменьшению быстродействия процессора. Для устранения этого недостатка применяют УУ с переменным числом тактов, в котором используют счетчик тактов с изменяемым модулем счета. Для коротких команд используют счетчик с небольшим модулем счета и наоборот.

При реализации простой системы команд узлы устройства управления с жесткой логикой экономичны и позволяют обеспечить наибольшее быстродействие среди всех возможных методов построения УУ. Однако с возрастанием сложности системы команд усложнялись и схемы автоматов с жесткой логикой, при этом их быстродействие уменьшалось.

4.4.3 Микропрограммные УУ

Альтернативой аппаратного способа реализации УУ является микропрограммное управление, согласно которому сигналы генерируются под управлением программы, подобной программе, написанной на машинном языке для ЭВМ. Этот принципиально иной подход был предложен английским ученым М. Уилксом в начале 50-х годов. Его называют принципом микропрограммного управления. Он позволяет преодолеть сложности реализации УУ с жесткой логикой. В основу принципа микропрограммного управления заложен тот факт, что каждой машинной команде соответствует уникальный код, называемый микрокомандой. Последовательность микрокоманд, реализующих машинную команду, образует микропрограмму. Микропрограммы размещаются в специальной управляющей памяти, называемой памятью микропрограмм. Выполнение любой команды в процессоре реализуется путем последовательного

извлечения микрокоманд из памяти микропрограмм с последующей их дешифрацией для формирования управляющих сигналов, необходимых при выполнении конкретной команды.

Идея заинтересовала многих конструкторов ЭВМ, но на момент возникновения она была нереализуема, поскольку требовала использования быстрой памяти относительно большой емкости. Вернулись к ней в 1964 году, в ходе создания системы 360 фирмой IBM. С этого времени устройства управления с программируемой логикой стали чрезвычайно популярными и были встроены во многие компьютеры.

Структура блока микропрограммного управления (БМУ) с принудительной адресацией микрокоманд (МК) приведена на рис. 4.11. В состав БМУ входят память микрокоманд (ПМК), регистр адреса микрокоманд (РАМК), регистр микрокоманд (РМК), дешифратор микроопераций (ДшМО), генератор тактовых импульсов (ГТИ).

Код операции (КОП) поступает из ОП системы на регистр кода операции (РКОП), который задает начальный адрес микропрограммы. Адрес микропрограммы формируется устройством формирования адреса МК (УФАМК) и хранится в РАМК. По этому адресу из памяти микрокоманд (ПМК) БМУ считывается микрокоманда и фиксируется в регистре МК (РМК).

Микрокоманда содержит два основных поля: код микрооперации (КМО) и адрес следующей МК (АСМК)

КМО дешифрируется и преобразуется в набор сигналов $y_1 \ldots y_m$, управляющих функционированием процессора. Поле адреса следующей микрокоманды заносится в УФАМК, в результате чего производится выборка следующей МК.

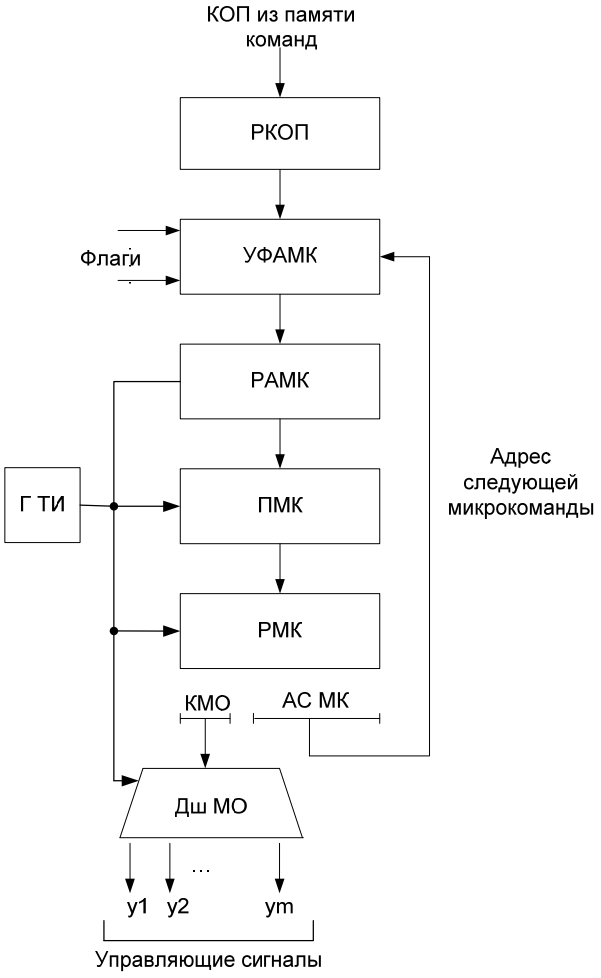


Рисунок 4.11- Структура БМУ с принудительной адресацией МК

Структурная схема БМУ, использующего естественную адресацию микрокоманд, показана на рис. 4.12. Последовательное чтение слов из управляющей памяти обеспечивается счетчиком

микропрограмм (СчМК). При каждой загрузке в регистр команд (РК) новой машинной команды в счётчик загружается выходное значение из блока формирования начального адреса. После этого на очередном такте производится автоматическое приращение содержимого СчМК для выбора из управляющей памяти очередной команды. Благодаря этому управляющие сигналы поступают в разные части процессора в необходимой, для выполнения конкретной команды, последовательности и в нужные моменты времени.

Для поддержки ветвления в микропрограммах блок формирования начального адреса имеет входы, на которые заводятся коды условий (флаги) из регистра признаков и внешние сигналы, например, сигналы прерываний. Необходимые проверки кодов условий выполняются с помощью микрокоманд условного перехода.

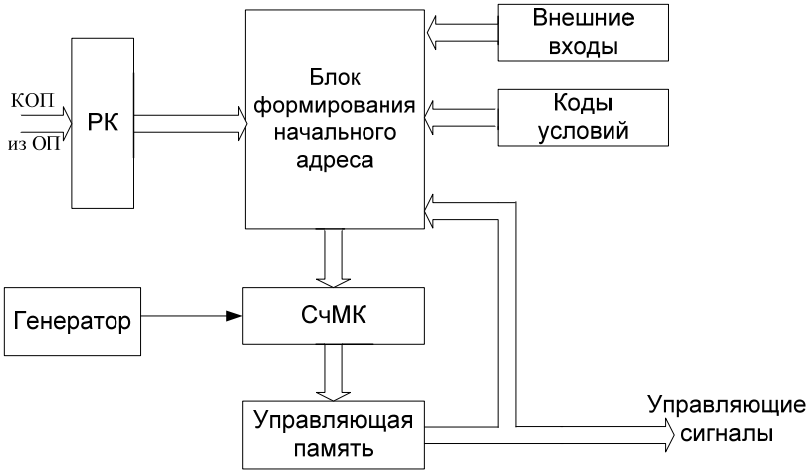


Рисунок 4.12 - Структура БМУ с естественной адресацией МК

На формирователь начального адреса в этом случае возлагается дополнительная функция - генерирование адреса перехода. По указанию микрокоманды этот блок загружает в счетчик СчМК новый адрес. Для поддержки условных переходов на входы данного блока подаются не только КОП команды с РК, но и сигналы с внешних входов, а также коды условий.

В БМУ с естественной адресацией МК после выборки очередной микрокоманды из управляющей памяти происходит приращение значения СчМК. Естественный порядок выборки МК нарушается в следующих случаях:

- если в РК загружается новая команда, в счетчик СчМК загружается начальный адрес ее микропрограммы;
- когда в последовательности выбираемых МК встречается микрокоманда перехода и условие перехода выполняется, в СчМК загружается адрес перехода;
- в случае появления микрокоманды “Останов” в СчМК загружается адрес микропрограммы для последующего выполнения очередной машинной команды.

БМУ с принудительной адресацией используются чаще, чем БМУ с естественной адресацией. Их общим достоинством является то, что замена одной системы команд на другую производится путем установки памяти микропрограмм с новой прошивкой. Кроме того, легко осуществляется эмуляция системы команд любого серийно выпускаемого процессора, что особенно важно при разработке контроллеров и специализированных ЭВМ. В этом случае можно использовать наработанное системное программное обеспечение серийной машины для отладки программ новой ЭВМ. Кроме того, микропрограммное управление позволяет легко создавать проблемно ориентированную систему команд в терминах управляемого объекта.

4.5. Структурно - функциональная организация процессоров

В состав процессора (см. рисунок 4.13) входят арифметико- логическое устройство (АЛУ), регистры общего назначения (РОН), устройство управления (УУ), а также интерфейс ОП и ПУ. Операнды из памяти и данные из периферийных устройств передаются через внешнюю двунаправленную магистраль данных, формируются буферным регистром данных (БРД) и помещаются на внутреннюю магистраль данных (и команд).

Выполнение некоторой программы начинается с загрузки счетчика команд (СчК) начальным адресом. Содержимое СчК передается в буферный регистр адреса (БРА) и используется для выборки команды из памяти. Команда по магистрали данных поступает в РК. Поле КОП команды дешифрируется ДшКОП (используется для выборки микропрограммы из ПЗУ микрокоманд) и служит для формирования сигналов, управляющих ходом выполнения команды, а также для формирования внешних управляющих сигналов шины управления (ШУ). Адресная часть команды передается в РгАоп для выборки операндов. Операнды передаются из памяти по внешней ШД, помещаются на внутреннюю магистраль процессора и, в зависимости от типа команды, заносятся либо в аккумулятор, либо в один из РОН, либо в регистр операнда РгОп. Результаты выполнения команд с выхода сумматора поступают в магистраль данных и далее пересылаются либо в память, либо в один из регистров процессора (А или РОН). Выбор конкретного РОН производится селектором адреса (СА) РОН, после записи в него соответствующего поля КОП РК.

После завершения процесса исполнения текущей команды, содержимое СчК модифицируется и производится выборка следующей команды.

В качестве внешних управляющих сигналов используются выходные сигналы чтения (Чт) и записи (Зп) для управления памятью (формируются при выполнении команд обращения к памяти), сигналы ввода (Вв) и вывода (Выв) (формируются при выполнении команд обращения к УВВ); входной сигнал запрос прерывания ЗПр, обеспечивающий прерывание выполнения основной программы и переход к выполнению подпрограммы, соответствующей внешнему запросу. Часто в процессорах формируют сигналы внутренних прерываний (например, при попытке деления на нуль или при недопустимых переполнениях).

Указатель стека УС предназначен для адресации стековой памяти, которая чаще всего реализуется в некоторой области оперативной памяти. Эта область определяется либо операционной системой, либо программистом путем загрузки начального адреса области стека в УС.

В интерфейс оперативной памяти и периферийных устройств входят буферные регистры адреса (БРА) и данных (БРД), предназначенные для хранения текущих значений адреса и данных. Дополнительно эти регистры должны обеспечивать увеличение токовой нагрузки внешних ША и ШД, а также высокоимпедансное состояние этих шин.

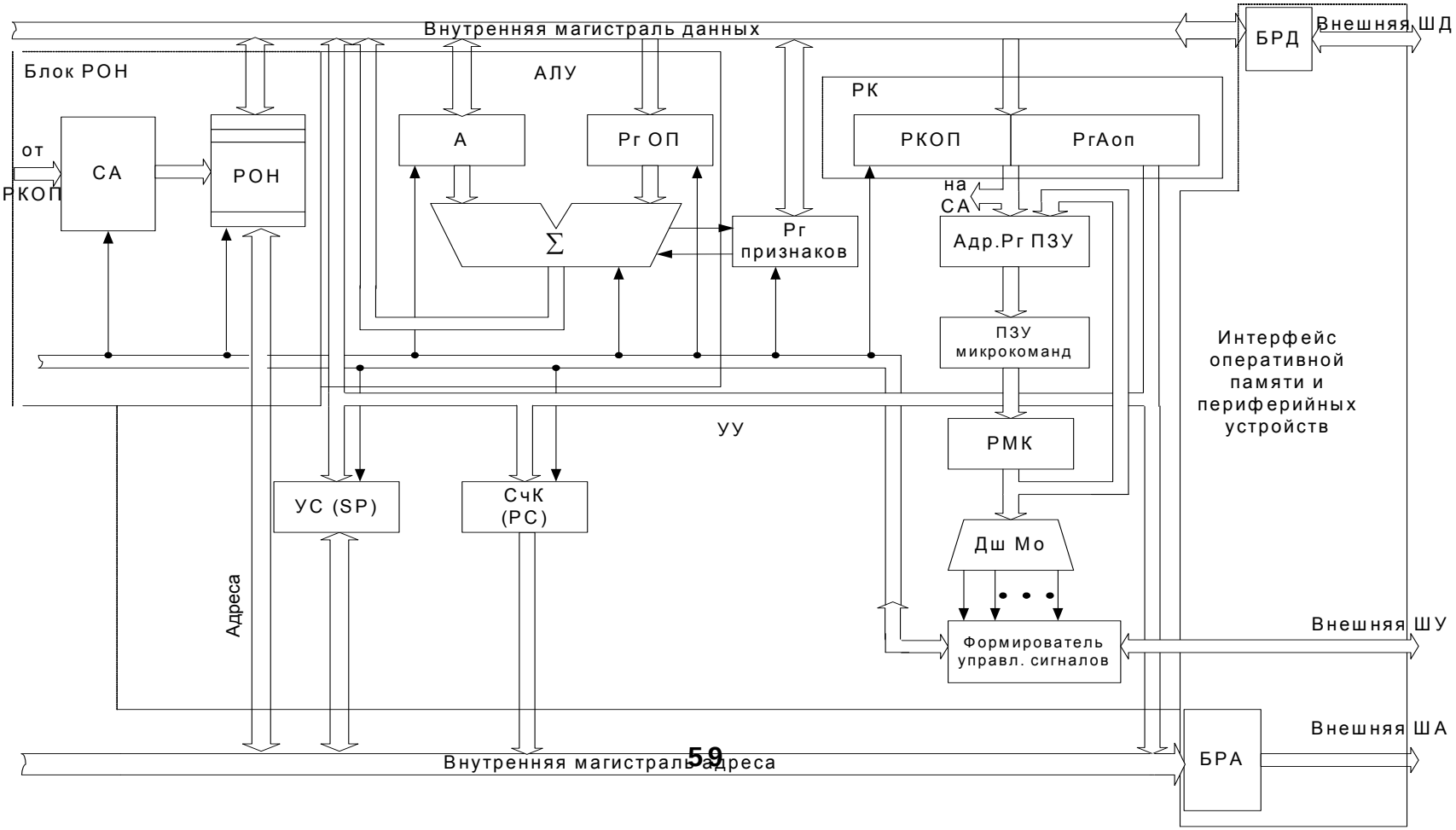


Рисунок 4.13 - Структура процессора с микропрограммным управлением

4.6 Рабочий цикл процессора

Функционирование процессора состоит из повторяющихся рабочих циклов, каждый из которых соответствует выполнению либо целой команды, либо её части. Завершив очередной рабочий цикл процессор переходит к выполнению следующего рабочего цикла. Предположим, что процессор может выполнять четыре типа команд:

- основные (арифметические, логические, пересылочные операции);
- передачи управления;
- ввода-вывода;
- системные (установка маски прерываний, состояния процессора и др.).

Рассмотрим рабочий цикл, выполняющийся по командно (существуют рабочие циклы, выполняющиеся по машинным циклам).

Рабочий цикл начинается (см. рисунок 4.14) с определения состояния процессора - СЧЕТ или ОЖИДАНИЕ. Процессор никаких действий в состоянии ожидания не выполняет и может выйти из него только при активизации некоторых внешних сигналов, например, запроса на прерывание ЗПр.

В состоянии СЧЕТ (в этом режиме происходит последовательная выборка и выполнение команд), если сигнал ЗПр отсутствует, то последовательно выполняются этапы рабочего цикла: формирование исполнительных адресов операндов, выборка операндов, выполнение операций и запоминание результата. После этого процессор переходит к выборке следующей команды и цикл повторяется. Если поступил запрос прерывания, процессор сбрасывает триггер прерывания ТПр, запоминает свое текущее состояние (например, путем записи адреса следующей команды в стек) и переходит к выполнению подпрограммы обработки прерывания путем передачи адреса подпрограммы в СчК.

При выполнении большинства команд формируются признаки операций, которые используются в командах условного перехода.

При выполнении команд передачи управления проверяется условие перехода по вышеуказанным признакам для команд условных переходов. Если условие перехода не выполняется, то выбирается следующая по порядку команда по продвинутому адресу, хранящемуся в СчК. Если условие выполняется, то в СчК заносится адрес перехода.

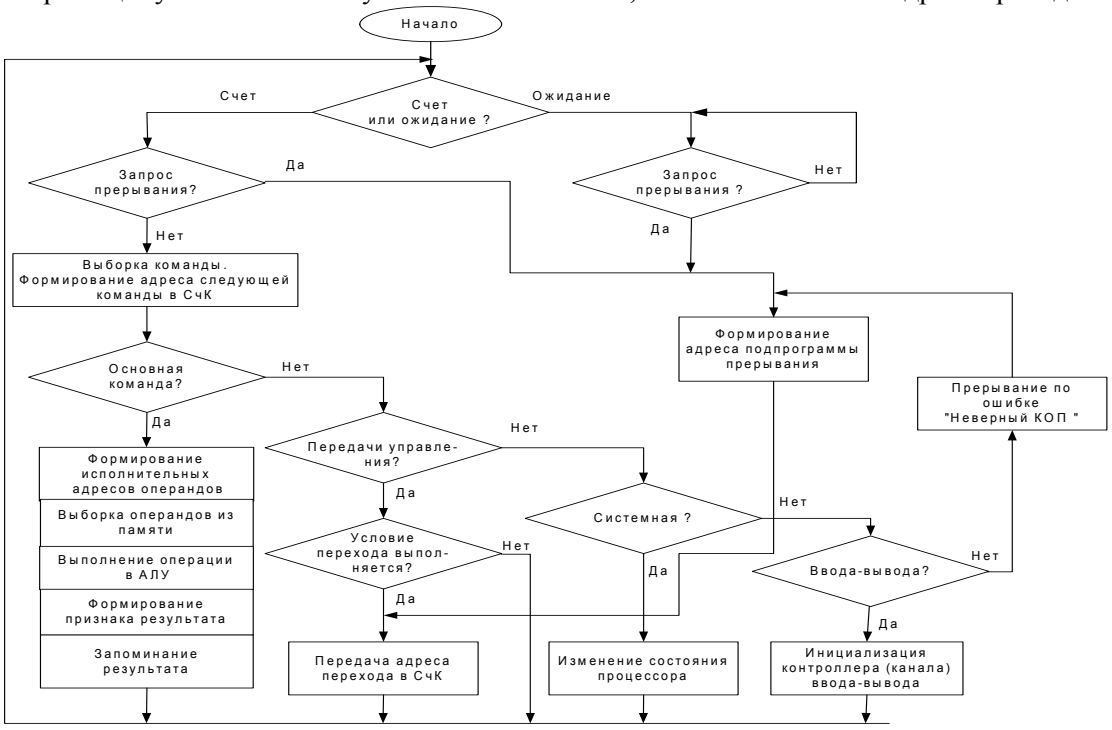


Рисунок 4.14- Рабочий цикл процессора

Команда вызова подпрограмм и переход к подпрограмме выполняется так же, как и команда перехода, но при этом дополнительно запоминается состояние процессора. Системные команды

производят переключение состояния процессора. Команды ввода-вывода инициируют обращение процессора к периферийным устройствам (ПУ).

4.7 Понятие о слове состояния программы

В ходе функционирования процессора постоянно меняется состояние его внутренних регистров. Сигнал “Запрос на прерывание”, а также команда “Вызов подпрограммы” приводят к прекращению выполнения основной программы и переходу к выполнению другой программы, которую часто называют подпрограммой. По окончании подпрограммы необходимо вернутся к основной программе, и продолжить её выполнение. Этот возврат должен быть выполнен корректно, так чтобы не было потери содержимого регистров процессора, отражающих состояние основной программы на момент переключения процессора. С этой целью осуществляют запоминание состояния процессора в момент переключения программ.

Содержимое регистров, обеспечивающих восстановление состояния вычислительного процесса, составляет слово состояния программы ССП (PSW- Program Status Word). Чаще всего в информацию о состоянии программы включают содержимое счетчика команд, содержимое регистра признаков и аккумулятора (см. рисунок 4.15).

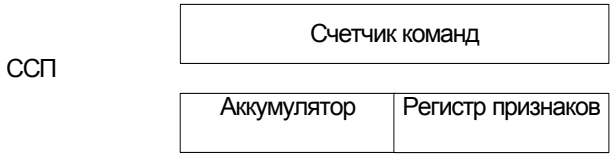


Рисунок 4.15- Структура слова состояния программы

Слово состояния обычно сохраняют в специально отведенной области памяти ЭВМ или стековой памяти. Сохранение производится либо автоматически (т.е. аппаратно) в начале обслуживания запроса на прерывание программы, либо программным путем.

4.8 Процедура выполнения команд перехода (условного и безусловного)

При естественной адресации адрес следующей команды получается из адреса выполняемой команды увеличением его на шаг адресации (1, 2, 3 и т.д. в зависимости от количества байт в команде). Производится эта операция путем автоматической модификации содержимого СчК после выполнения текущей команды.

Для управления ходом выполнения программ и организации ветвлений в систему команд процессоров с естественной адресацией вводятся команды условных и безусловных переходов.

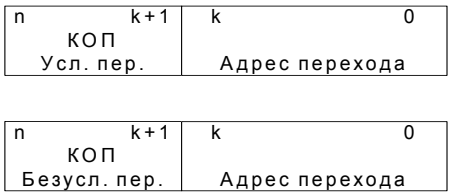


Рисунок 4.16- Форматы команд условного и безусловного переходов

В разных машинах реализация этих команд различная, однако общий подход следующий: содержимое поля адреса перехода команды загружается в счетчик команд, после чего процессор продолжает выполнение программы с нового адреса.

Команды безусловного перехода предписывают совершать переход по программе независимо от каких-либо условий. Существуют команды безусловного перехода по косвенному адресу. В этом случае в коде команды указывают адрес ячейки, в которой хранится адрес перехода.

При условном переходе адрес следующей команды зависит от некоторого условия, полученного в результате выполнения предыдущей. Если условие выполняется, то процессор переходит к выполнению программы по адресу, указанному в адресной части команды, если нет, то к команде, следующей непосредственно за командой условного перехода.

4.9 Процедура выполнения команд вызова подпрограмм

Другим типом команд передачи управления являются команды вызова подпрограмм. Их особенность заключается в том, что по окончанию выполнения подпрограммы они должны обеспечить возврат к выполнению программы, из которой подпрограмма была вызвана. Для этого адрес возврата должен быть запомнен, для чего в счетчике команд формируется продвинутый адрес, который затем сохраняется в памяти (или в стеке). Для перехода к выполнению подпрограммы в СчК заносится адресная часть команды ее вызова. По окончанию выполнения подпрограммы адрес следующей команды основной программы, ранее сохраненный в стеке, вызывается из него, заносится в СчК и выполнение программы продолжается. Для организации возврата в основную программу подпрограмма должна оканчиваться командой "Возврат" ("RETURN"). Кроме нее существует также и команда "Условного возврата".

Формат команды "Перехода к подпрограмме" приведен на рисунке 4.17.

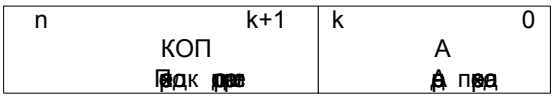


Рисунок 4.17- Формат команды "Переход к подпрограмме"

Процесс выполнения команд "Вызов подпрограмм" проиллюстрирован на рисунке 4.18. Короткий отрезок прямой на

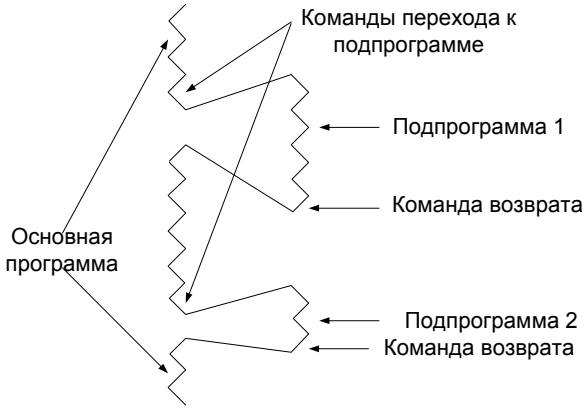


Рисунок 4.18 - Процесс выполнения команды "Вызов подпрограммы"

этом рисунке соответствует одной команде, длинный - переходу к выполнению подпрограммы или возврату из нее.

4.10 Контрольные вопросы

1.Перечислите функции процессора.

2.Каковы функции РК и СчК в процессоре?

3.Назначение АЛУ процессора?

4.Что дает введение в состав АЛУ РОНов?

5.Назначение УУ процессора?

6.Основное отличие между аппаратными и микропрограммными УУ?

7.Назначение РАМК УУ?

8.Перечислите основные узлы блока микропрограммного управления.

9.Опишите последовательность выполнения команды пересылки данных между РОН, используя структуру процессора с микропрограммным управлением.

10.Что такое ССП (PSW)?

11.Опишите процедуру выполнения команд условного и безусловного переходов.

12.Опишите процедуру выполнения команды вызова подпрограммы.

13.Какое основное отличие процедур выполнения команд вызова подпрограмм и выполнения команд условного и безусловного переходов?

5. Методы повышения производительности процессоров

5.1. Конвейерная реализация команд

Машинные команды выполняются в процессоре с помощью ряда последовательных элементарных действий: выборки команды (IF), декодирования команды (D), формирования адресов операндов (OA), выборки операндов (OF), выполнения операции (EX) и запоминания результата (S). В машине с простой структурой аппаратной части выборка следующей машинной команды производится лишь после завершения выполнения предыдущей команды (рисунок 1.19а). С другой стороны, в машинах с конвейерной организацией команд, как показано на рис. 1.19б, допустимо одновременное выполнение нескольких команд путем совмещения во времени различных микроопераций этих команд.

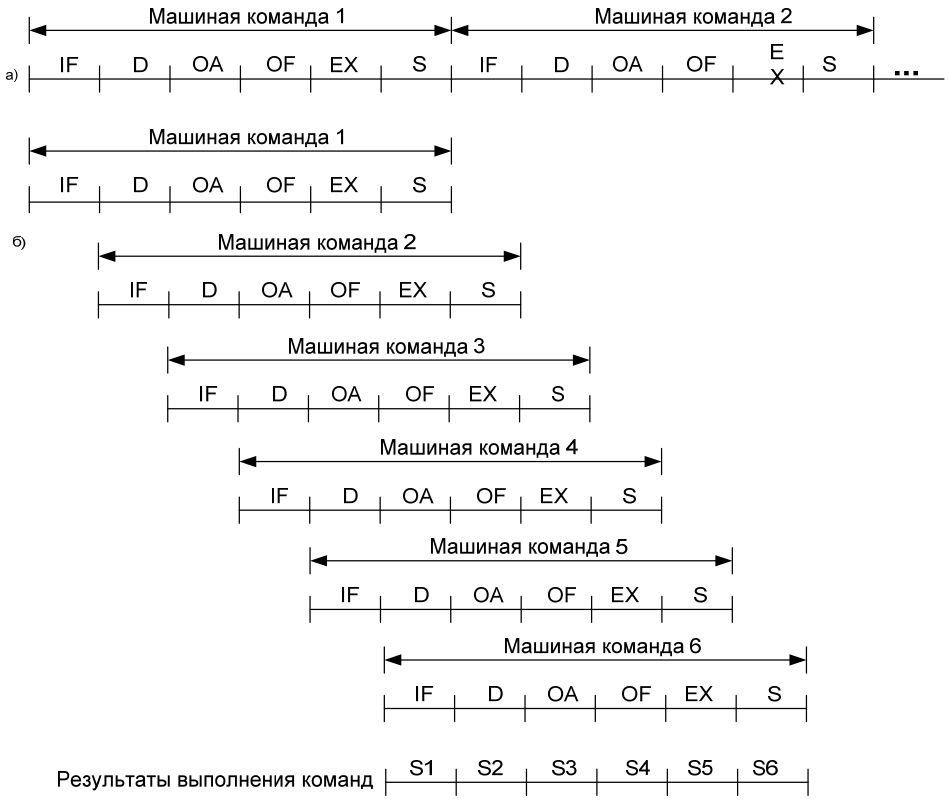


Рис. 1.19 - Конвейер микроопераций. а) процессор без конвейера; б) процессор с конвейером. Микрооперации: IF-выборка команды, D-декодирование, OA-формирование адресов операндов, OF-выборка операндов, EX-выполнение операции, S-запоминание результата.

Указанное выше совмещение возможно, если в процессоре выделить самостоятельные блоки для выполнения отдельных микроопераций (рисунок 1.20).

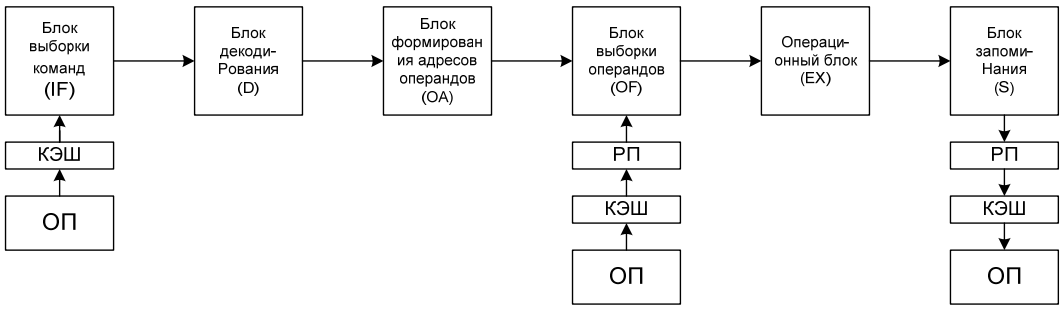


Рисунок 1.20 – Структура конвейера микроопераций

Если считать, что на выполнение каждой микрооперации затрачивается одинаковое время, равное машинному такту, то в идеальном случае можно получать результаты операций в каждом машинном такте. Конвейерное выполнение команд основано на тех же принципах, что и поточные линии сборки производстве. Оно имеет максимальную эффективность, когда продолжительность выполнения всех этапов команд одинакова, бесперебойно подаются команды и данные и на каждом этапе отсутствуют «мертвые» временные зоны, нарушающие непрерывность конвейерной реализации команд. К этим факторам можно отнести следующие:

1) когда для выполнения следующей команды требуется результат от предыдущей команды, или когда предыдущей командой определяется адрес операнда следующей команды (модификация адреса), то возникает задержка начала выполнения следующей команды, связанная с ожиданием выборки операнда или с преобразованием адресов;

2) при ветвлении программы по результатам проверки условий командой условного перехода команды, находящиеся в процессе конвейерной обработки, остаются невыполненными, и требуется повторная загрузка конвейера, начиная с момента выборки команды условного перехода;

3) когда в кэш-памяти отсутствуют требуемые данные или команды, необходимо еще передать их в кэш-память из основной памяти. При конфликтном обращении к кэш-памяти (например, при наложении друг на друга этапов IF, OF, S команд, следующих одна за другой) запросы с относительно низкими приоритетами будут находиться в стадии ожидания;

4) когда предшествующая команда изменяет содержание последующей или когда изменяется содержимое регистров, определяющих состояние программы, последующая команда должна ожидать завершения предшествующей команды;

5) в случае возникновения прерывания и перехода к программе его обработки команды, находящиеся в это время на командном конвейере, остаются незавершенными (стадии, на которых прерывается их выполнение, зависят от вида прерывания), и приходится заново загружать конвейер командами, входящими в программу обработки прерывания;

6) когда операция, реализуемая машинной командой, имеет сложный характер, как, например, операция с плавающей точкой или действие команды операционной системы, и для ее выполнения требуется много машинных циклов, последующая команда долго не может достичь стадии выполнения операции.

Чтобы предельно ограничить влияние перечисленных выше факторов, нарушающих работу конвейера команд, в современных высокопроизводительных компьютерах и суперЭВМ совершенствуется структура аппаратной части и оптимизируется компиляция потока информации, поступающего на конвейер команд.

Так второй фактор можно устранить применением специального блока предсказания ветвлений, который формирует адрес выбираемой команды до того, как будет определено условие перехода. Для реализации такого подхода в состав процессора вводят ассоциативную память, называемую буфером адресов ветвлений (BTB – Branch Target Buffer), в которой хранятся адреса ранее выполненных переходов. Предсказание ветвлений основано на свойстве цикличности алгоритмов, основанном на том, что в теле цикла программы многократно происходит обращение к одним и тем же адресам ветвлений.

Фактор 6 устраняют применением сокращенного набора команд (RISK -процессоров), в которых специально делают так, чтобы время выполнения всех команд было одинаковым. Однако это не всегда возможно реализовать, поэтому современные процессоры являются суперскалярными.

5.2. Суперскалярные процессоры.

Суперскалярный процессор содержит множество операционных устройств (см. рисунок 1.20).

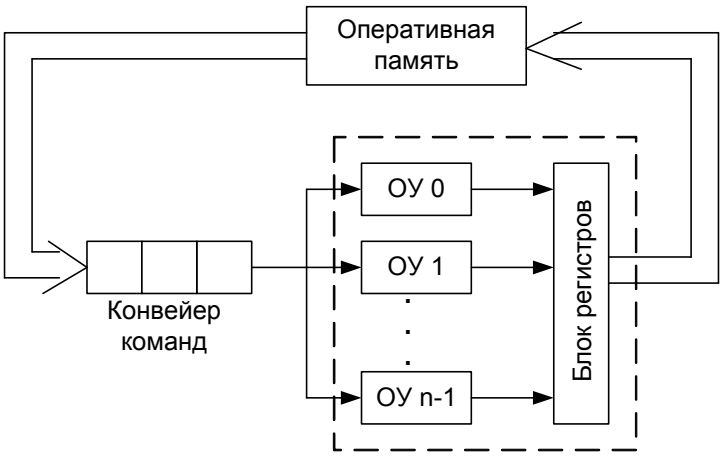


Рисунок1.20. Структура суперскалярного процессора

В соответствии с используемым принципом, команды, последовательно формируемые конвейером команд, выполняются свободными операционными блоками, причем результаты операций могут выдаваться с очередностью, отличной от очередности команд в командном конвейере. Операционные блоки параллельно выполняют соответствующие операции, совместно используя одни и те же регистры. Причём в процессе выполнения команд производится проверка занятости необходимых операционных блоков, управление поступлением требуемых данных (операндов) из других операционных блоков и памяти и передается управление определенным операционным блокам, когда для выполнения ими команд все готово.

Если новая команда поступает в каждом машинном такте, а быстродействие операционных блоков позволяет произвести суммирование с фиксированной точкой, например, за три машинных такта, умножение с плавающей точкой за 10 машинных тактов. Другими словами, выполнение обработки операционным блоком отстает от темпа поступления команд, поэтому введением определенного числа дополнительных операционных блоков устанавливается баланс между темпом поступления команд и скоростью выполнения операций.

Основной недостаток в том, что если длительность операции, задаваемой последующей командой, меньше и связь между ее данными и результатами выполнения предыдущей команды отсутствует, то результат этой операции окажется в регистре раньше, чем результат операции предыдущей команды. Подобная ситуация называется внеочередным завершением команд, при этом результаты при этом записываются во временные регистры. Позднее содержимое временных регистров в нужном порядке пересылается в постоянные регистры процессора. Причем временный регистр исполняет роль того постоянного регистра, и даже имя ему присваивается такое же. Для реализации подобного принципа необходимо, чтобы в процессоре присутствовал специальный управляющий блок, называемый *блоком сохранения.* Этот блок выбирает следующую команду для сохранения, используя очередь, называемую *буфером реорганизации.* Команды записываются в эту очередь в том порядке в каком они следуют в программе. Когда команда достигает начала очереди и завершается, соответствующие результаты пересылаются из временных регистров в постоянные. После этого команда удаляется из очереди, а все выделенные ей ресурсы, включая временные регистры очищаются, итак, команда теперь считается *покинувшей конвейер.* Описанный алгоритм позволяет выполнять команды в любой последовательности и обеспечивает их выход с конвейера в строгом соответствии с порядком расположения в программе.

Подобный принцип управления вычислительным процессом применён в процессорах Pentium, причём в зависимости от семейства число операционных блоков может достигать пяти. Так в процессорах семейства P6 в операционной части конвейера используются два целочисленных блока, один блок для операций с плавающей точкой, один блок MMX для обработки потока целочисленных данных (Pentium, Pentium II) и один блок SSE для обработки потока чисел с плавающей точкой (Pentium III).

5.3 Процессоры с длинным командным словом.

Процессоры с длинным командным словом (VLIW) используют параллелизм на уровне команд. В соответствии с этой концепцией, как показано на рис.1.20, сравнительно длинная

команда делится на множество полей и каждый операционный блок управляется отдельным полем.

В отличие от суперскалярных процессоров, где возможность распараллеливания операций выясняется в процессе их выполнения, в процессорах данного типа это выяснение происходит в ходе компиляции программы. Компилятор извлекает из программы команды, которые могут быть выполнены параллельно, и из них формируется одна команда.

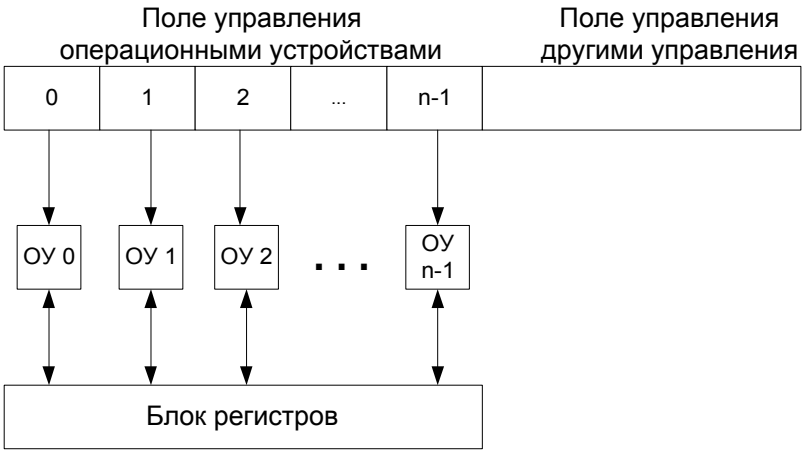


Рис.1.21. Структура VLIW- процессора.

Достоинства VLIW заключаются в следующем. Во-первых, компилятор может более эффектно выявлять зависимости между командами и выбирать параллельно исполняемые команды, чем это делает аппаратура суперскалярного процессора, ограниченная размером окна исполнения. При этом за счет обеспечения степени параллельности, близкой к числу операционных блоков, можно достичь высокой скорости обработки. Во-вторых, VLIW- процессор имеет более простое устройство управления и потенциально может иметь более высокую тактовую частоту.

Однако у VLIW процессоров есть серьезный фактор, снижающий их производительность. Это команды ветвления, зависящие от данных, значения которых становятся известны только в динамике вычислений. Число одновременно выполняющихся команд VLIW-процессора, не может быть очень большим в виду отсутствия у компилятора информации о зависимостях, формируемых динамически, в процессе выполнения.

5.4 Матричные процессоры

Как показано на рис. 1.21, в матричном процессоре команда, выделяемая управляющим устройством, одновременно передается множеству процессорных элементов с одинаковой структурой (ПЭ), и все ПЭ параллельно выполняют одну и ту же операцию. Управляющее устройство разрешает или запрещает выполнение операций на основе информации о состоянии каждого операционного элемента. Информация о состоянии хранится в специальном внутреннем регистре ПЭ. Выполнение операций разрешается только тем процессорным элементам, в которых выполняются определенные условия.

Поскольку поток команд является одиночным, то в случае необходимости условного перехода по результатам проверки выполнения условий заданные операции выполняются только теми элементами, для которых результаты проверки подтверждают выполнение условий, а затем только теми элементами, для которых результаты проверки говорят о том, что условия не выполняются. Следовательно, сначала запрещается выполнение операций процессорным элементам с отрицательными результатами

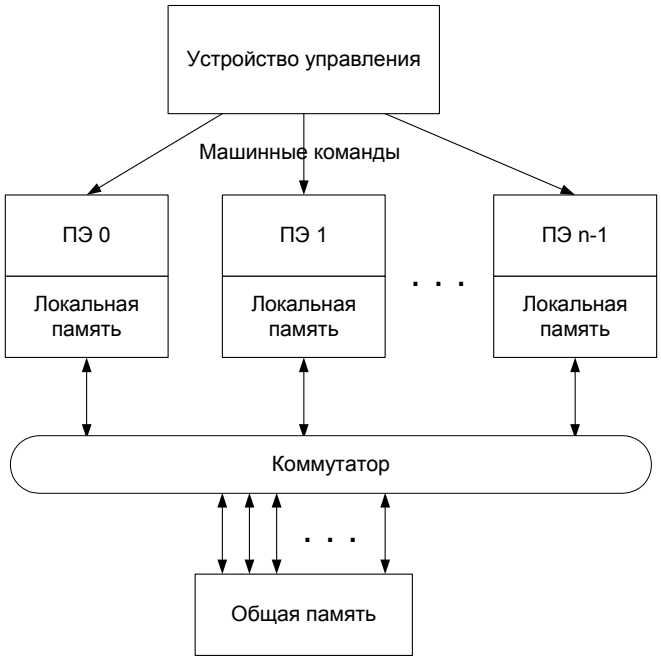


Рис. 1.22. Архитектура матричного процессора.

проверки условий, а затем — элементам, в которых проверяемые условия выполняются.

Управляющее устройство контролирует взаимосвязь между всеми операционными блоками и управляет обращением к общей памяти. Принцип матричных вычислений используется в современных микропроцессорах путем встраивания в кристаллы относительно самостоятельных блоков, обеспечивающих технологии MMX (Pentium, Pentium-II) и SSE (Pentium-III и выше), которые значительно ускоряют операции обработки изображений.

5.5 Векторные процессоры

Под *вектором* понимается одномерный массив однотипных данных (обычно в форме с плавающей запятой), размещенных в памяти ВС. Если обработке подвергаются многомерные массивы, их также рассматривают как векторы. Одной из характеристик вектора является число составляющих его элементов – *длина вектора.*

Векторный процессор – это процессор, в котором операндами некоторых команд могут выступать упорядоченные массивы данных – векторы. Векторный процессор может быть реализован в двух вариантах. В первом варианте он представляет собой дополнительное устройство к скалярному (основному) процессору, которое называют *ускорителем* или *сопроцессором* векторных операций. Во втором - векторный процессор является основой самостоятельной ВС.

Векторный процессор является расширением обычного скалярного процессора с регистровой памятью. В такой процессор включается большая группа регистров, обеспечивающая хранение векторов длиной *l.* Необходимым условием является наличие достаточно длинных векторных регистров, чтобы минимизировать число обращений в память за векторными данными.

Высокая скорость вычислений в векторном процессоре обеспечивается тем, что с помощью одной команды можно выполнить операцию над множеством данных. Структура типичной команды следующая:

КОП	R1	R2	R3

Поле КОП – код векторной операции;
R1- адрес векторного регистра, хранящего 1-й операнд;
R2- адрес векторного регистра, хранящего 2-го операнд;

R3- адрес векторного регистра, хранящего результат.

Векторный процессор выполняет операции типа R3 = R1* R2, где знак * указывает на тип операции. Например, необходимо вычислить вектор Y = AxB, где вектор A = $a_0,a_1,a_2,\ldots,a_{n-1}$; вектор B = $b_0,b_1,b_2,\ldots,b_{n-1}$. Вектор A помещается в VR1, вектор B – в регистр VR2. Полученный результат-вектор Y- заносится в регистр VR3.

Структура связей векторного процессора показана на рисунке 1.23.

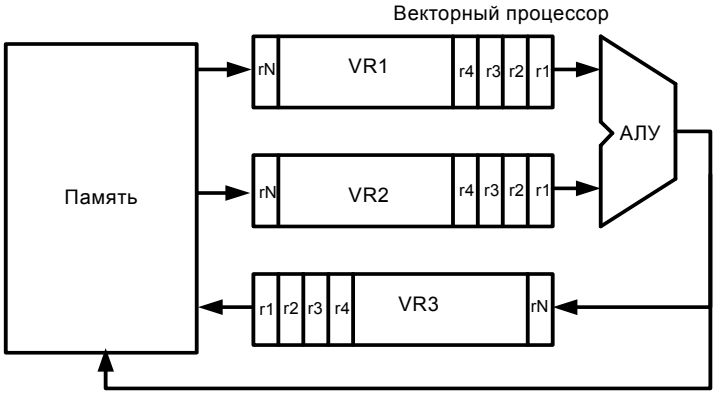


Рисунок 1.23- Структура векторного процессора

Векторные регистры представляются наборами обычных регистров для хранения двоичных чисел разрядностью в одно или двойное слово. Число регистров может составлять 256, 512, 1024 или более.

Наиболее частое применение получило векторные процессоры в качестве встроенных ускорителей векторных операций. Скалярный процессор играет роль главного. Он выполняет все типы операций, кроме векторных. Как только в программе появляется векторная команда, она перенаправляется в векторный сопроцессор. Результат возвращается в скалярный процессор для завершения необходимых действий.

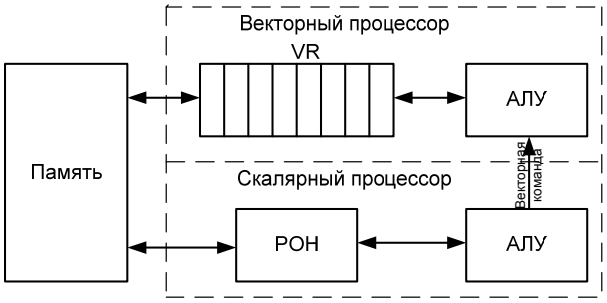


Рисунок 1.24 – Подключение векторного сопроцессора

Современные процессоры имеют в своем составе блоки векторной обработки. Такими блоками оснащен и процессор Pentium, содержащий два ускорителя векторных операций: MMX и SSE. Первый работает с 64- разрядными словами и способен выполнять операции над векторами длиной в 8, 4 или 2 символа, разрядностью 8, 16 и 32 соответственно. Блок SSE имеет в два раза большую разрадность.

6 СИСТЕМЫ ПРЕРЫВАНИЯ ПРОГРАММ

6.1 Общие сведения

Во время функционирования вычислительной системы (ВС) возможны различные ситуации, которые требуют немедленной реакции со стороны машины. При этом реакция ВС состоит в том, что процессор прерывает обработку текущей программы и переходит к выполнению некоторой другой программы, связанной с этой ситуацией. По завершению этой программы процессор должен вернуться к выполнению прерванной программы. Такой процесс называется прерыванием программ.

Необходимость введения систем прерывания в ЭВМ вызвано тем, что часть событий происходят в моменты времени, которые заранее не известны, поэтому их невозможно учесть в программе. Например, аварийное завершение программы, связанной с попыткой деление на ноль, выход за границы допустимых адресов памяти, появление сбоев в работе. Это так называемые внутренние события. Прерывания возможны и от внешних событий, например прерывания от устройств ввода- вывода, таймера, аппаратуры передачи данных, датчика технологического процесса и т. д.

Каждое прерывание совершается по сигналу, оповещающему ЭВМ о его возникновении. Такие сигналы называются запросами на прерывание.

Программа, к выполнению которой переходит ЭВМ в результате получения сигнала запроса прерывания, называется прерывающей программой. Программа, которая выполнялась в ЭВМ до появления запроса на прерывание, называется прерываемой программой. Процесс прерывания программ можно пояснить рисунком 5.1. Аппаратные и программные средства, участвующие в организации обработки запросов на прерывание, получили название системы прерывания программ. Они выполняют следующие функции:

-запоминание состояния прерванной программы и осуществление перехода к выполнению прерывающей программы;

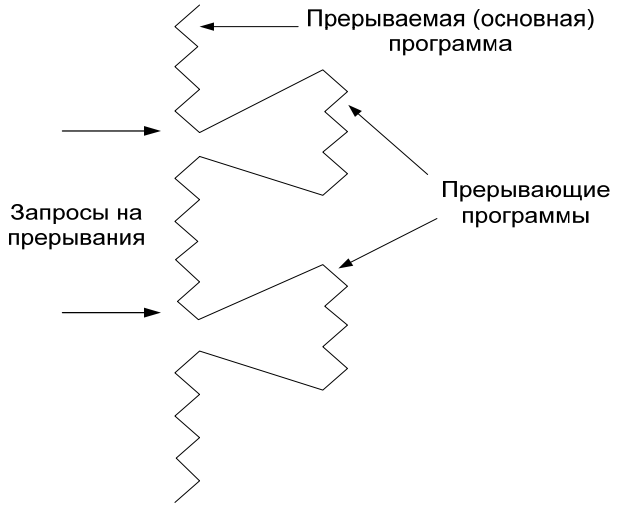


Рисунок 5.1 - Процесс обслуживания запросов на прерывание

-восстановление состояния прерванной программы и переход к ее выполнению.

Существующие системы прерывания программ строятся по приоритетному принципу. В этом случае при наличии нескольких источников, вырабатывающих запросы независимо, порядок их обслуживания, в случае одновременного поступления запросов прерывания, определяется присвоенным приоритетом. Если одновременно поступит несколько запросов, то будет обслуживаться тот запрос, приоритет которого выше. Приоритеты бывают абсолютные и относительные. В системах с абсолютными приоритетами, запрос с более высоким приоритетом прерывает прерывающую программу с низким приоритетом. В системах с относительным приоритетом обслуживание предыдущего запроса продолжается до конца.

Системы прерывания с относительными приоритетами называются одноуровневыми. Системы прерывания с абсолютными приоритетами - многоуровневыми (вложенными).

Временные диаграммы выполнения прерывания программ в одноуровневой системе прерываний приведены на рисунке 5.2.

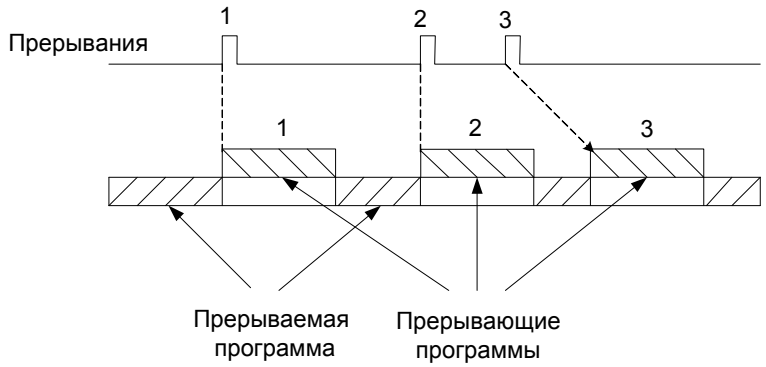


Рисунок 5.2- Временные диаграммы выполнения прерывания программ в одноуровневой системе прерываний. Запрос 1 имеет низший, 3 - высший приоритет

В соответствии с рисунком, выполнение программы обработки прерывания 1 закончилось до появления запроса прерывания 2. Появление запроса 3 во время выполнения программы обработки прерывания 2 не приводит к прекращению ее выполнения.

Временные диаграммы выполнения прерывания программ в многоуровневой системе прерываний приведены на рисунке 5.3. В такой системе прерывания запрос с более высоким приоритетом прерывает прерывающую программу с низким приоритетом.

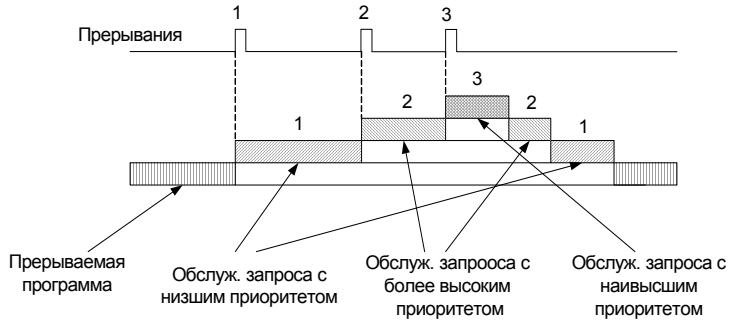


Рисунок 5.3 – Временная диаграмма выполнения программ в много уровневой системе прерываний. Запрос 1 имеет низший, 3 - высший приоритет.

5.2 Характеристики систем прерываний

1. Время реакции - время между появлением запроса прерывания и началом выполнения прерывающей программы (см. рисунок 5.4).

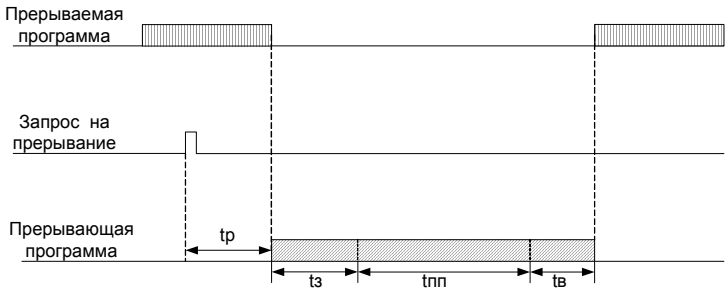


Рисунок 5.4 - Составляющие времени обслуживания прерывания

На приведенном рисунке используются следующие обозначения:

tp - время реакции;

tз - время, необходимое для запоминания состояния прерванной программы;

tв - время, необходимое для восстановления состояния прерванной программы;

tпп - время выполнения собственно прерывающей программы.

2. Время обслуживания прерывания t0 - время, затрачиваемое на переключение программ:

to=tз+tв.

3. Глубина прерывания - максимальное число программ, которые могут прервать друг друга. В одноуровневых системах глубина прерывания равна 1, в многоуровневых - n.

4. Насыщение системы прерывания. Если к моменту прихода нового запроса от одного и того же источника окажется не обслуженным предыдущий запрос, то говорят, что произошло насыщение системы. В этом случае новый запрос будет утрачен, что является недопустимым. Поэтому при проектировании систем прерывания необходимо учитывать это явление.

5. Допустимые моменты прерывания программ. Чаще всего прерывание допускается после окончания выполнения текущей команды, в момент выполнения которой поступил запрос. При этом необходимо запоминание состояния программы, т.е. всех программно-доступных регистров процессора, которые могут быть изменены прерывающей программой. Время реакции системы прерывания в этом случае определяется в основном длительностью выполнения одной команды.

Однако в некоторых случаях такое время реакции может оказаться недопустимо большим. В машинах с уменьшенным временем реакции допускается прерывание в любом такте исполнения команды, но в этом случае возрастает количество информации, которая требуется запомнить (например, дополнительно запоминается содержимое СчТ, регистра КОП и др.).

5.3 Схема выполнения процедуры прерывания

Взаимодействие процессора с контроллером прерываний при обработке запросов прерываний показано на рисунке 5.5.

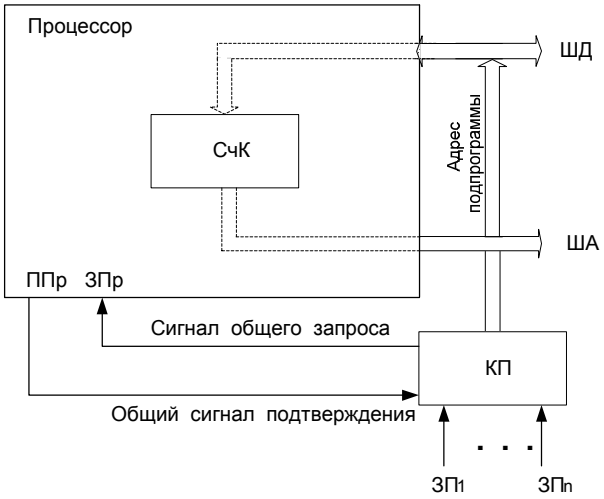


Рисунок 5.5 - Схема выполнения процедуры прерывания

Запросы ЗП1…ЗПn от внешних источников прерываний поступают на контроллер прерываний КП. Он формирует общий сигнал запроса прерывания ЗПр (INTR), с учетом приоритета запросов ЗП1…ЗПn. Процессор, выполнив команду и запомнив состояние процессора (ССП) выдает сигнал ППр (INTA). КП формирует адрес подпрограммы (вектор прерывания) запроса ЗПi, приоритет которого среди одновременно появившихся запросов ЗП1…ЗПn (или их части) выше. Адрес подпрограммы через ЩД процессора заносится в СчК, в результате чего процессор переключается на выполнение программы (подпрограммы) обслуживания прерывания.

5.4 Способы реализации систем прерываний

Для вызова подпрограммы обслуживания прерывания, соответствующего номеру запроса ЗПi, подсистема прерывания должна обеспечивать возможность определения (идентификации) номера запроса ЗПi , приведшего к появлению общего сигнала запроса прерываний ЗПр.

По способу идентификации источника запроса системы прерывания делятся на:

-системы с программным опросом источников;

-с циклическим опросом источников;

-с последовательным опросом (по принципу “ дейзи-цепочки ”);

-с опросом по вектору;

Три первых типа систем прерывания являются одноуровневыми. Приоритеты прерываний в них фиксированы и определяются их местом в схеме системы прерывания. Все они обладают одним общим недостатком - большим временем поиска источника запроса.

В настоящее время наиболее распространенными являются системы прерывания с опросом по вектору.

5.4.1 Схема прерывания с опросом по вектору

В системах, в которых запросы требуют быстрого обслуживания, используется многоуровневые прерывания с опросом по вектору.

Под вектором будем понимать либо начальный адрес, либо идентификатор, связанный с начальным адресом прерывающей программы. Его использование не требует дополнительного опроса источников прерывания для поиска активного из них. Информация о номере запроса вводится с КП на ШД процессора по сигналу ППр (см. рисунок 5.6).

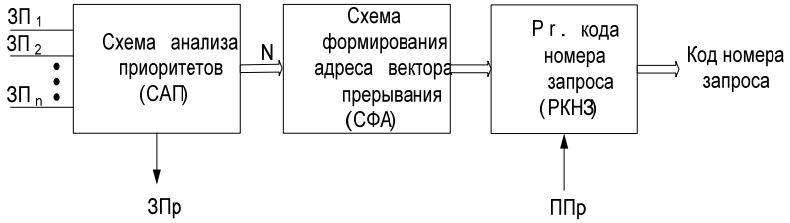


Рисунок 5.6 - Схема прерывания с опросом по вектору

Схема анализа приоритетов САП выделяет запрос с наивысшим приоритетом. Выделенный запрос поступает в блок СФА, представляющий собой шифратор. Он формирует код номера запроса, связанный с адресом ячейки памяти, в которой находится вектор прерывания. По сигналу ППр этот код вводится в процессор через ШД (см. рисунок 5.5).

5.4.2 Прерывания с программно - управляемым приоритетом

В случаях, когда в процессе выполнения программы необходимо изменять приоритеты прерываний, используются схемы прерываний с программно-управляемым приоритетом (см. рисунок 5.7).

Код маски запрещает или разрешает прерывание от соответствующего запроса. Он загружается командой процессора в регистр маски Рг.М и приоритеты прерываний устанавливаются путем программного изменения кода маски. Каждая прерывающая программа может установить свою маску. Рг.М представляется обычно как порт ввода- вывода и загружается командами вывода в порт.

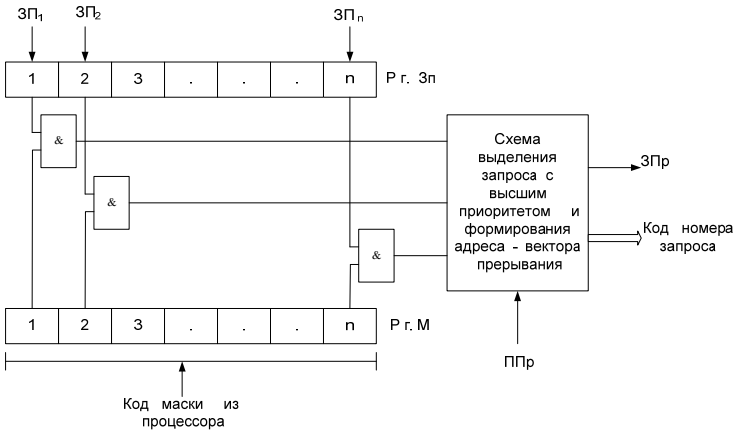


Рисунок 5.7 - Схема прерываний с программно- управляемым приоритетом

5.5 Контрольные вопросы

1. Что такое прерывание программы?
2. Дайте определения прерываемой и прерывающей программ.
3. Перечислите функции, выполняемые системой прерывания.
4. Перечислите основные характеристики систем прерываний.
5. Объясните смысл термина “Приоритет прерывания”.
6. Назовите отличия систем прерывания с абсолютным и относительным приоритетом.
7. Как реализуются системы прерываний?
8. В чем суть прерываний с программно-управляемым приоритетом?
9. Как реализуются схемы прерывания с опросом по вектору?
10. Перечислите функции КП.
11. Действия КП после получения от процессора сигнала ППр (INTA)?
12. Через какую шину вводится в процессор адрес- вектор прерывания?

7 ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

7.1 Общие сведения о вводе-выводе в ЭВМ

Ввод-вывод предназначается для связи центра (процессора и ОП) с периферийными устройствами ПУ (УВВ), которые используются либо для хранения больших объемов информации (ВЗУ), либо для ввода в ЭВМ или вывода из нее информации (программ и данных).

Передача информации из ПУ в центр называется операцией ввода, а передача из центра в ПУ - операцией вывода.

От организации системы ввода-вывода в значительной степени зависит общая производительность ЭВМ.

6.2 Основные способы ввода-вывода

Существуют два основных способа ввода-вывода: программный (программно-управляемый) и прямой доступ в память (ПДП).

Программно управляемая передача данных осуществляется при участии и под непосредственным управлением процессора. Данные между памятью и периферийными устройствами пересылаются через процессор.

При вводе- выводе в режиме ПДП процессор не участвует в обмене и либо приостанавливает свою работу на время обмена, либо выполняет параллельно с обменом обработку команд и данных, не требующих обращения к ОШ. Обмен данными между УВВ и ОП осуществляется напрямую, минуя процессор. Ввод-вывод в режиме ПДП является (принципиально) более быстродействующим, нежели программно управляемый.

Программный способ осуществляется либо по прерыванию, либо без прерывания. При вводе-выводе с прерыванием программы инициация ввода-вывода осуществляется сигналом запрос на прерывание от ПУ (см. рисунок 6.1). При вводе-выводе без прерывания его инициация осуществляется текущей командой программы.

Ввод-вывод без прерывания бывает синхронным и асинхронным. При синхронном вводе-выводе готовность ПУ к обмену не проверяется, при асинхронном - проверяется.

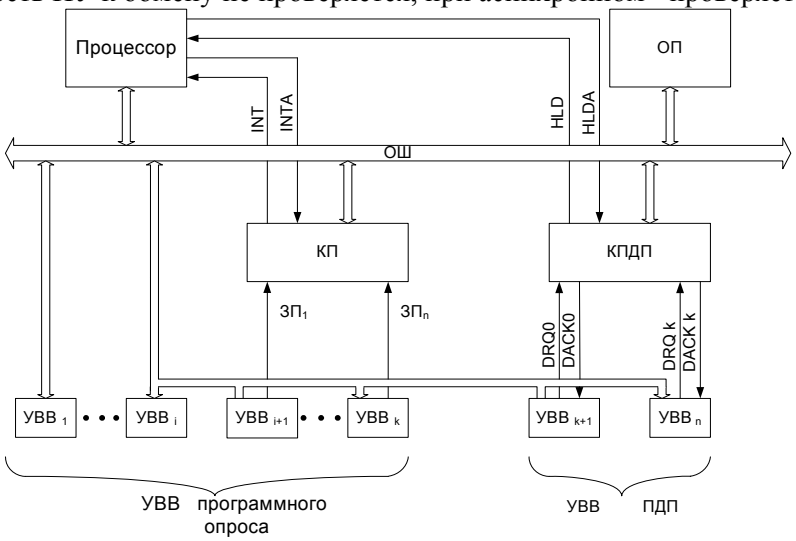


Рисунок 6.1- Организация ввода-вывода в ЭВМ на основе ОШ

6.2.1 Программно - управляемый ввод - вывод

При программно - управляемом вводе – выводе данные между ПУ и ОП пересылаются через процессор и под его управлением.

Если ПУ быстродействующее и работает в темпе процессора (т.е. успевает выдавать или принимать данные со скоростью выполнения процессором команд программы), то такой программно-управляемый ввод-вывод называется синхронным. При проведении операции обмена с такими ПУ процессор не осуществляет каких либо дополнительных действий, кроме записи в регистр данных ПУ или его чтения.

Однако часть ПУ является менее быстродействующими чем процессор (например- печать и клавиатура) и им требуется большее время для ввода и вывода, нежели процессору для выполнения одной команды.

Ввод-вывод с таких ПУ называется асинхронным. Для исключения потерь информации при асинхронном вводе- выводе, процессор при начале обмена проверяет готовность устройства ввода-вывода к обмену, путем считывания содержимого его регистра состояния. Регистр состояния должен содержать информацию о функционировании устройства ввода-вывода и является дополнительным регистром асинхронного ПУ. Регистры данных и состояния таких ПУ подключаются к ОШ и каждому из них присваиваются уникальные адреса. При неготовности ПУ к обмену процессор выполняет другие действия.

Процесс взаимодействия процессора с асинхронными УВВ проиллюстрирован на рисунке 6.2.

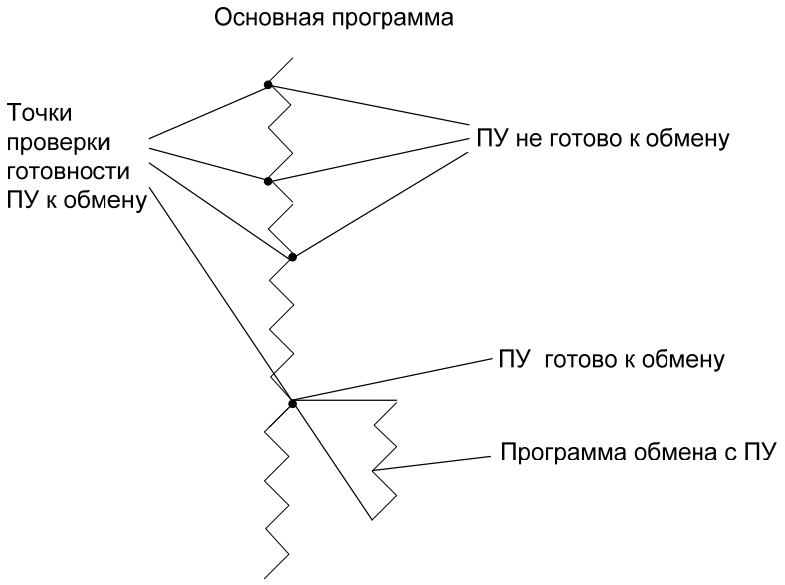


Рисунок 6.2 - Взаимодействие процессора с асинхронными УВВ

Асинхронный ввод-вывод приводит к непроизводительным затратам времени работы процессора или к простоям ПУ. Увеличение в программе количества команд анализа состояния ПУ увеличивает длину программы и время работы процессора, уменьшение- к увеличению простоя ПУ. Тем не менее, такой способ ввода-вывода широко используется в ЭВМ.

6.2.2 Ввод - вывод с прерыванием программы

При вводе-выводе по прерыванию используется уже рассмотренный подход: по сигналу ЗПр процессор переходит к выполнению программы (подпрограммы), осуществляющей ввод-вывод. ПУ само информирует процессор о своей готовности, при этом процессор не тратит времени на анализ готовности ПУ (т.е. в программе нет соответствующих команд). Он осуществляет выполнение основной программы. При появлении сигнала готовности (ЗПi) от ПУ, процессор прерывает выполнение основной программы и начинает выполнение подпрограммы обслуживания ПУ. По её завершению процессор продолжает выполнение прерванной программы. Взаимодействия процессора с УВВ при вводе- выводе по прерыванию показано на рисунке 6.1.

В связи с тем, что при таком способе ввода-вывода процессор прерывает выполнение основной программы, его называют вводом-выводом по прерыванию.

Подсистема ЭВМ, реализующая ввод-вывод по прерываниям ПУ называется подсистемой прерывания. При наличии у неё нескольких входов прерываний ЗПi от ПУ (см. рисунок 6.1) она так же выдаёт процессору информацию о номере входа, вызвавшему прерывание программы.

Ввод- вывод по прерываниям позволяет снизить время простоя ПУ и непроизводительной работы процессора, однако это достигается ценой усложнения аппаратной части ЭВМ.

6.2.3 Ввод - вывод в режиме ПДП

Недостатком программного ввода-вывода является то, что все операции ввода-вывода осуществляются через процессор. Это приводит к уменьшению скорости ввода- вывода, особенно

ощутимой при пересылке больших массивов информации, например, при вводе кода программы в ОП с ВЗУ. Для пересылки каждого байта данных в асинхронном режиме в программе, кроме команд проверки готовности ПУ к обмену и пересылки данных, должны быть также команды для изменения адреса ячейки памяти и отслеживания количества пересылок. При использовании ввода-вывода по прерываниям издержки еще больше, поскольку приходится сохранять содержимое счетчика команд, регистра признаков, части РОНов и другую информацию о состоянии процессора на момент прихода сигнала запроса прерывания.

При наличии в составе ЭВМ подсистемы ПДП (DMA-Direct Memory Access) обмен данными между ПУ и ОП осуществляется без постоянного участия процессора. При выполнении ПДП процессор отключается от управления системой, в связи с чем подсистема ПДП должна иметь в своём составе схемы для управления ОШ. Для осуществления блочной передачи между ВЗУ и ОП процессор заносит в подсистему ПДП данные о номере первой ячейки ОП, с которой начнётся обмен, размер передаваемого блока и направление изменения адресов ячеек ОП (уменьшение или увеличение). После этого подсистема ПДП (см. рисунок 6.1) выдаёт процессору сигнал запроса ЗЗх (HOLD) на захват ОШ (запрос на право управления ОШ). Реакцией процессора на этом сигнал является выработка сигнала подтверждения захвата ПЗх ОШ (HLDA) и перевод большинства своих выходов в высокоимпедансное состояние. Вслед за этим подсистема ПДП формирует адрес ячейки ОП, сигналы записи и чтения ОП, и организует обмен с ВЗУ. ВЗУ при готовности приема или передачи очередного байта или слова информации вырабатывает сигнал готовности данных DRQ, ответом которому со стороны подсистемы ПДП является сигнал подтверждения приема данных DACK.

При перемещении каждого очередного байта, значение внутреннего счетчика в подсистеме ПДП (контроллере ПДП -КПДП), указывающего на размер передаваемого блока, уменьшается на 1. При обнулении счетчика цикл ПДП заканчивается и управление ОШ возвращается процессору. Для этого КПДП переводит сигнал HOLD в неактивное состояние.

Скорость обмена в режиме ПДП определяется пропускной способностью оперативной памяти и ОШ и может достигать нескольких Мбайт/с.

6.3 Интерфейсы

Связь устройств ЭВМ друг с другом осуществляется через сопряжения, которые в вычислительной технике называются интерфейсами. Интерфейс -это совокупность линий, схем и алгоритмов, предназначенных для осуществления обмена информацией между устройствами ЭВМ. Более короткое определение: интерфейс- это совокупность аппаратных и программных средств, предназначенных для организации взаимодействия различных устройств ЭВМ между собой.

Для обеспечения возможности изменения конфигурации ЭВМ интерфейсы унифицирует, т. е. делают единым для всех или части устройств, которые могут быть подключены к нему. Это дает возможность легкого изменения конфигурации ЭВМ.

В высокопроизводительных ЭВМ используют несколько различных интерфейсов, значительно отличающихся по характеристикам. При такой организации возможно параллельная пересылка нескольких данных. Благодаря этому системы с несколькими интерфейсами работают быстрее, но стоимость их выше.

6.3.1 Характеристики интерфейсов

Интерфейсы имеют следующие характеристики :

- пропускная способность- максимальное количество информации, передаваемой по интерфейсу в единицу времени;
- максимально допустимое расстояние между соединяемыми устройствами;
- общее число линий интерфейса;
- ширина шин- число бит или байт информации, передаваемых параллельно через интерфейс за один шинный цикл;
- связность; интерфейс может быть односвязным или многосвязным. В первом случае существует один путь передачи данных от ПУ к центру, во втором- множество. Многосвязный интерфейс повышает живучесть и надежность ЭВМ.

Кроме этих характеристик существуют динамические характеристики интерфейсов, например время передачи слова или блока данных с учетом продолжительности процедур подготовки и завершения передачи (своего рода время реакции).

6.3.2 Шины интерфейсов ввода-вывода

Группа линий, обеспечивающих соединение устройств между собой, называется шиной (Bus). Линии шины обычно подразделяются на несколько групп: данных (ШД), адреса (ША), управления (ШУ) и др.

Организация обмена различных устройств ЭВМ через интерфейс невозможна без некоторого набора правил, задающих поведение соединенных шиной устройств, а именно: последовательности помещения информации на шину, выдачи управляющих сигналов и т.п.- так называемого шинного протокола.

Наиболее сложной (в понимании функционирования) группой линий являются линии шины управления ШУ. Для задания типа текущей операции шины (типа шинного цикла) используется линия (сигнал) Зп/Чт (R/W#). Значение “лог 1” на этой линии, как правило, соответствует операции чтения, а значение “лог 0”- операции записи. Если шина допускает пересылку операндов разных размеров (байт, слово и т.д.), размер пересылаемых данных также указывается управляющими линиями.

При обмене данными по шине одно из устройств ЭВМ инициирует пересылку данных по шине и называется инициатором (хозяином- Host) шины. Обычно его роль выполняет процессор, но хозяином может быть любое другое устройство, взявшее на себя (захватившее) управление шиной. Устройство, к которому обращается хозяин шины, называется подчиненным или целевым.

Для задания момента выдачи данных на шину инициатор использует специальные сигналы ШУ. По виду задания момента выдачи данных шины подразделяются на синхронные и асинхронные.

6.3.2.1 Синхронные шины

В случае синхронных шин все устройства получают синхронизирующую информацию по общей линии, на которую подаются тактовые импульсы фиксированной частоты. Промежуток времени, в течение которого выполняется одна операция пересылки данных, называется длительностью цикла шины. В простейшем случае она равна одному периоду тактовой частоты шины.

Временные диаграммы шинного цикла “Вывод данных” синхронной шины приведены на рисунке 6.3.

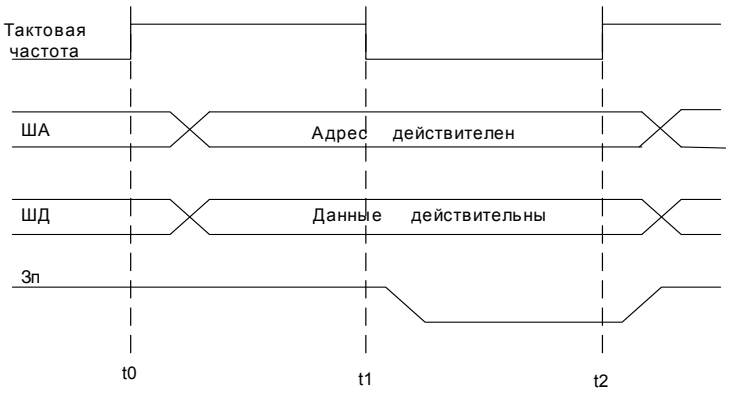


Рисунок 6.3 – Временные диаграммы шинного цикла ”Вывод данных ”

Будем считать, что изменение тактового сигнала все подключенные к шине устройства замечают одновременно (точнее- практически одновременно). По нарастающему фронту тактового сигнала в момент времени t0 хозяин шины выставляет адрес устройства и передаваемые для него данные. Однако из-за задержки распространения сигналов в логических схемах инициатора и переходных процессов в линиях шины сигналы адреса и данных придут к установившимся значениям спустя некоторое время после t0.

Для обозначения того, что инициатор выводит данные, он в момент времени t1 формирует сигнал записи Зп. Адресованное устройство к этому моменту времени должно сравнить адрес, передаваемый по ША, со своим внутренним (с адресом, назначенным устройству в системе) и быть готовым к транзакции (обмену). Ввод данных в регистр данных РД адресованного устройства осуществляется по нарастающему фронту тактовой частоты в момент времени t2.

Недостатком синхронной шины является то, что при подключении к ней нескольких устройств, скорость обмена будет определяться самым медленным из них. Кроме этого, инициатор не может определить, какое из адресуемых устройств ответило на запрос и ответило ли оно вообще, т.к. в синхронной шине отсутствуют ответные сигналы от адресованных устройств. В последнем случае инициатор обмена даже не обнаружит ошибку.

6.3.2.2 Асинхронные шины

В асинхронных шинах при пересылке данных используется механизм квитирования-подтверждение связи между инициатором обмена и подчиненным устройством. Механизм квитирования аналогичен порядку пересылки почтовых отправлений с уведомлением о вручении, при котором по получению отправления в адрес отправителя направляется уведомление о вручении (квитанция). В асинхронных шинах линия тактирования заменяется двумя управляющими линиями синхронизации: готовности хозяина (Master-ready) и готовности подчиненного устройства (Slave-ready). Первая принадлежит хозяину шины, который выставляет на ней сигнал готовности к транзакции, по второй отвечает подчиненное устройство.

Временные диаграммы сигналов асинхронной шины при вводе приведены на рисунке 6.4.

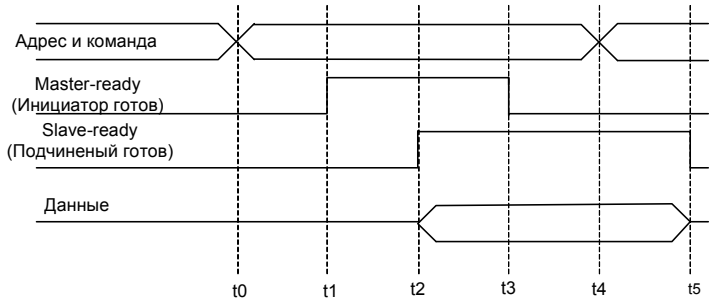


Рисунок 6.4 - Временные диаграммы шинного цикла “Ввод данных ” асинхронной шины

В момент времени t0 инициатор выдает, на линии шины адрес подчиненного устройства и информацию о типе пересылки (команду). В момент времени t1 он сообщает об этом всем подчиненным устройствам по линии Master-ready (инициатор готов). Все подключенные к шине устройства, при активации этой линии, декодируют адрес. Устройство, для которого предназначена команда, выполняет ее и информирует об этом хозяина сигналом квитирования Slave-ready (подчиненный готов), после чего хозяин может начинать новый шинный цикл.

Важнейшим преимуществом асинхронной шины является то, что квитирование избавляет от необходимости жесткой синхронизации скоростей работы инициатора и подчиненного устройства, что упрощает шину. Никакие задержки, связанные с распространением сигнала по шине и интерфейсным схемам, не отражаются на работе шины.

6.4 Контрольные вопросы

1. Дайте определение интерфейса.
2. Какие основные способы ввода-вывода применяются в ЭВМ?
3. Что такое программно- управляемый ввод- вывод и ввод- вывод в режиме ПДП?
4. Отличия синхронных и асинхронных УВВ? Достоинства и недостатки синхронного и асинхронного способов ввода- вывода.
5. Достоинства и недостатки ввода- вывода в режиме ПДП.
6. Перечислите последовательность действия устройств при выполнении процедуры ввода- вывода в режиме ПДП.
7. Перечислите характеристики интерфейсов ввода-вывода.
8. Назовите основной принцип организации синхронной шины.

9. Недостатки синхронной шины?
10. Назовите принципы организации асинхронной шины.
11. Недостатки асинхронной шины?
12. Какое назначение сигнала Master-ready?

7 Организация памяти вычислительных систем

7.1 Виртуальная память

Система виртуальной адресации

Виртуальная память представляет собой единое адресное пространство, в котором физическая ограниченность емкости основной памяти скрыта от программиста. Таким образом, для программиста создается видимость произвольной адресации с отсутствием ограничений на емкость используемой памяти, что значительно облегчает программирование. Кроме того, использование виртуальной организации памяти способствует повышению взаимозаменяемости программ между вычислительными системами.

Реально существующую основную память называют физической, а ее адреса - физическими, логическую память - виртуальной, а ее адреса - виртуальными (логическими). Соответствие между физическими и виртуальными адресами устанавливается совместно аппаратными средствами ЭВМ и ее операционной системой. Обычно виртуальное адресное пространство размещается во внешней памяти, например на магнитных дисках. Часть этого пространства, необходимая для выполнения программ в данный момент, копируется в основной памяти.

Для реализации виртуальной памяти необходимо разделить все адресное пространство памяти на части и организовать соответствующий обмен между основной и внешней памятью. При этом память разбивается на страницы или сегменты.

При разбиении на страницы виртуальное и реальное адресные пространства делятся на части фиксированной длины, называемые страницами.

Рис.1.3. Страничная организация памяти.

Адрес каждой страницы виртуального пространства ставится в соответствие адресу страницы физического адресного пространства. Взаимосвязь между адресами обоих типов устанавливается таблицей преобразования адресов (таблицей страниц), пример которой приведен на рис.1.3. Адрес слова внутри страницы определяют конкатенцией адреса страницы и содержимого поля смещения.

Перенос страницы виртуального пространства в основную память называется загрузкой страницы (подкачкой страницы в оперативную память), а обратное действие - удалением страницы (откачкой страницы из оперативной памяти).

Основной недостаток страничной организации памяти заключается в том, что она требует большого объема физической памяти под страничную таблицу, если велико виртуальное адресное

пространство (большой объем внешней памяти). Так при объеме внешней памяти в 100 Гбайт под страничную таблицу требуется выделить около 25 Мбайт, которые отнимаются у программ и данных, готовых к выполнению.

Сегментация

Сегментацией называется разделение адресного пространства памяти на части (сегменты) по логическим признакам, устанавливаемым программистом. Обычно величина сегмента соответствует объему программы или подпрограммы и в отличие от страницы имеет переменную длину.

Виртуальный адрес состоит в этом случае из номера сегмента и относительного адреса в пределах сегмента; он преобразуется в физический адрес по таблице сегментов (рис.1.4). Преобразование виртуального адреса в физический адрес производится аналогично страничному преобразованию, только таблица преобразования состоит из адресов сегментов. Также в каждом сегменте адресное пространство является линейным, виртуальное адресное пространство в целом оказывается двумерным. Физический адрес выбираемого слова получают суммированием адреса сегмента и адреса, указанного полем смещения. С учетом того, что сегмент является логической единицей, можно организовывать защиту информации и управление для коллективного использования сегментированной информации.

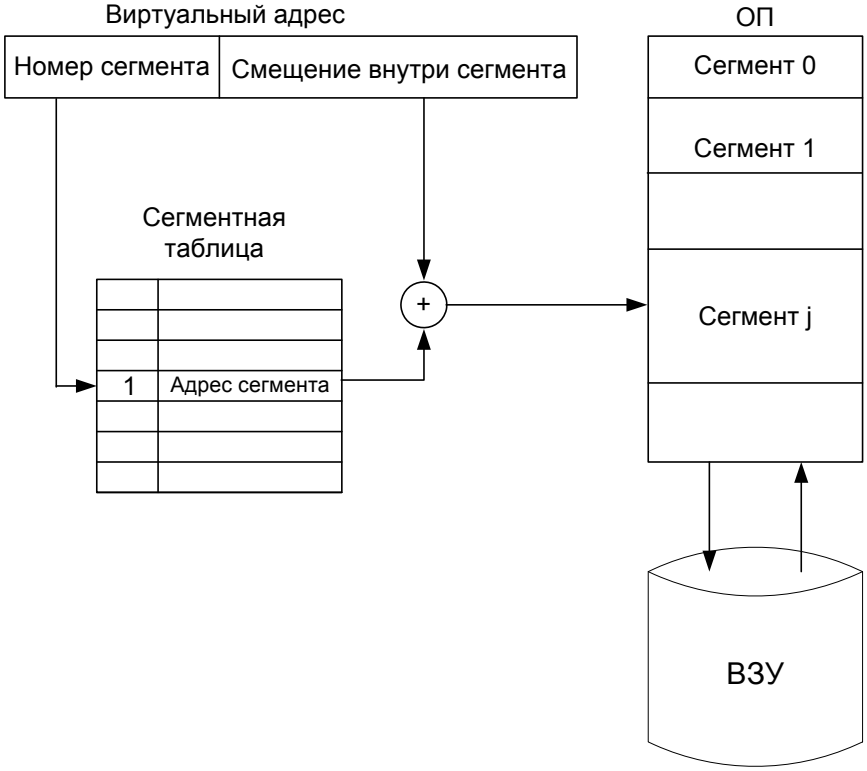


Рисунок 1.4- Сегментное преобразование адреса

Сегментная таблица занимает гораздо меньший объем физической памяти, поскольку число сегментов в памяти гораздо меньше, чем страниц. Но здесь присутствует другая проблема, которая заключается в так называемой фрагментации памяти. Когда загружаемый из ВЗУ сегмент небольшого размера помещается в свободный сегмент памяти большего размера, то часть памяти остается свободной, образуя т.н. «дыру». В процессе функционирования таких дыр набирается множество, что приводит к неэффективному использованию основной (оперативной) памяти.

Сегментно- страничная организация памяти

Как было отмечено выше, расширение виртуального пространства влечет за собой увеличение таблицы страниц. Одним из способов устранения этого неудобства служит многоуровневое разбиение на страницы. Суть этого разбиения состоит в том, что одномерное виртуальное пространство подразделяется на два уровня - сегментов и страниц, а преобразование виртуального адреса производится по двухуровневой таблице. Это дает возможность обойтись без ведения таблицы неиспользуемых страниц и, следовательно, экономит объем памяти, выделяемый для таблицы страниц. Сегмент в этом случае не является полностью двумерным пространством, но при необходимости в процессе использования его можно сделать двумерным.

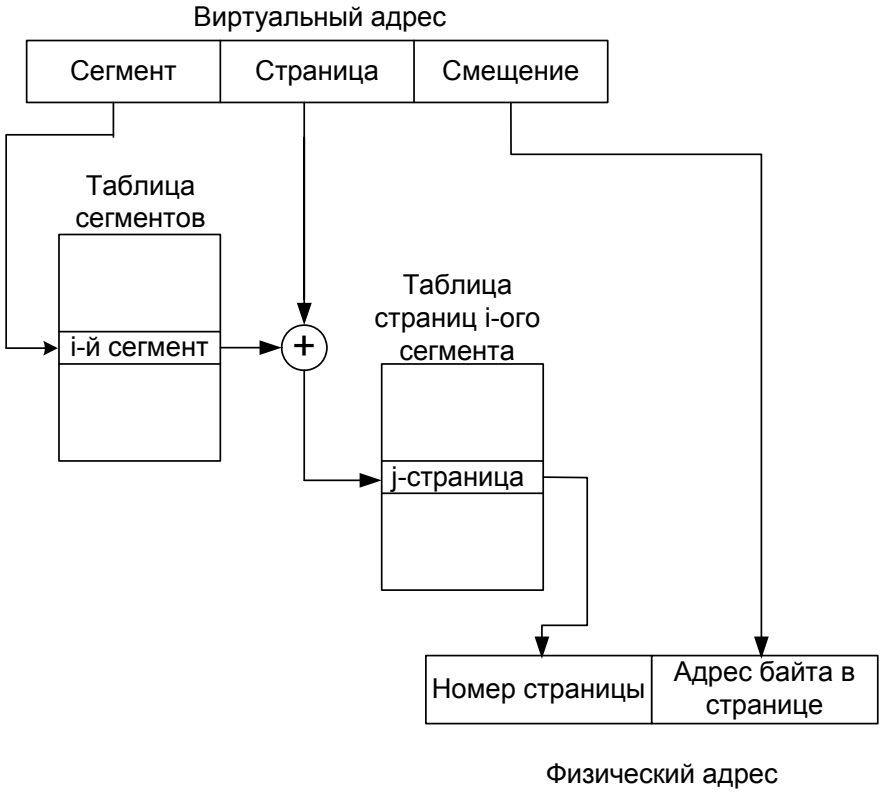


Рис.1.5.Сегментно-страничная организация памяти.

В большинстве вычислительных систем мультипрограммирование ориентировано на параллельную обработку нескольких задач и реализуется посредством системы виртуальной памяти следующими двумя методами. Первый метод основан на разделении виртуального пространства между несколькими задачами. По второму методу для каждой задачи создается отдельное виртуальное пространство адресов. Такая память называется мультиплексной виртуальной памятью.

Для управления мультиплексным виртуальным адресным пространством организуется несколько таблиц преобразования адресов, которые переключаются при переходе от задачи к задаче. Для этого в один из регистров устройства управления процессора вводится указатель, по которому выбирается соответствующая таблица преобразования адресов.

Преимущество этой системы заключается в том, что пространство памяти, используемое каждой задачей, полностью заполняет рамки виртуального пространства. Одновременно обеспечивается высокоэффективная защита памяти, так как никакая задача не может сформировать адрес, относящийся к другой задаче. Однако время обращения к таблице при преобразовании виртуального адреса в физический является относительно большим.

Для ускорения этой процедуры на основе использования аппаратных средств разработан так называемый механизм динамического преобразования адресов (Рис.1.6).

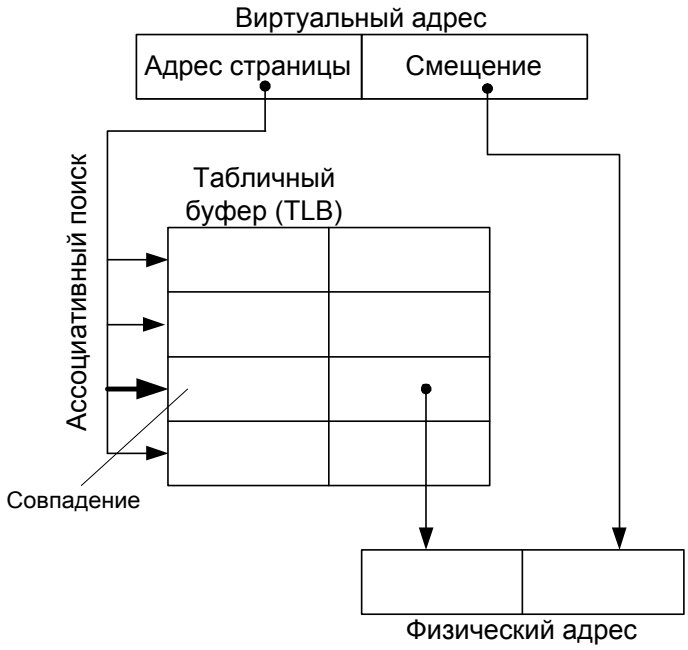


Рис.1.6. Механизм динамического преобразования адресов.

Смысл этого механизма состоит в том, что в ассоциативной памяти заранее записываются номера наиболее часто используемых в данное время страниц номера блоков, соответствующих этим страницам в основной памяти; в ходе преобразования адресов прежде всего проверяется эта ассоциативная память, и если в ней обнаруживаются сведения о необходимых страницах, эти сведения могут быть сразу же использованы, т. е. сокращается длительность процедуры преобразования адресов.

Подобный буфер для высокоскоростного преобразования адресов называется буфером динамической трансляции адресов виртуальной памяти.

Процедура замены страниц

Когда требуемая страница в основной памяти отсутствует, она переписывается в нее из внешней памяти. Если же в основной памяти не оказывается свободного блока для загрузки страницы, то необходимо удалить какую-либо из страниц, находящихся в ней. Связанные с этим действия называются заменой страниц.

Известны следующие стратегии замены страниц:

1) стратегия FIFO, в соответствии с которой из основной памяти удаляются страницы, раньше других занесенные в нее.

2) стратегия LRU, при использовании которой удаляется та страница, обращение к которой имело место раньше, чем к другим.

3) стратегия WS (Working Set - рабочее множество), в соответствии с которой удаляются страницы, не содержащиеся в так называемом рабочем множестве, т. е. наборе страниц, к которым за определенный истекший интервал времени зафиксировано обращение.

Две из этих стратегий замены страниц - LRU и WS - основаны на предположении, что страницы, использовавшиеся в последний период, будут часто использоваться и впредь. По сравнению с ними реализация стратегии FIFO проще, но эффективность ее относительно ниже. На практике обычно используются стратегии LRU и WS, а также их сочетание и модификации.

Управление распределением основной памяти

Одна из проблем параллельной обработки состоит в том, каким образом распределить блоки основной памяти между всеми программами, участвующими в этой обработке, В общем случае, чем больше объем распределяемой основной памяти, тем реже приходится производить ее перераспределение. Для повышения эффективности выполнения каждой из параллельно обрабатываемых программ необходимо обеспечить их определенным объемом основной памяти. При чрезмерном увеличении уровня мультиплексирования программ уменьшается объем памяти, отводимой каждой из них, повышается частота обмена страниц, что приводит к резкому снижению эффективности всей системы мультипрограммной обработки. Возникает так

называемое дробление памяти. Для устранения этого явления управление заметной страниц осуществляется с учетом обеспечения наибольшей эффективности использования центрального процессора. Хотя реализация этого управления занимает время, ситуация в целом улучшается благодаря своевременному сокращению степени мультиплексирования.

7.2 Построение памяти микро-ЭВМ

В микро-ЭВМ на основе общей шины возможны два способа организации адресного пространства памяти и устройств ввода- вывода:

- изолированное адресное пространство. Реализовано в ЭВМ типа IBM PC.
- совмещенное адресное пространства памяти и ввода- вывода. Реализовано в ЭВМ фирмы DEC и Motorola.

Изолированное адресное пространство памяти и ввода- вывода

В изолированной системе используется раздельная адресация памяти (ОП) и периферийных устройств (ПУ). Для адресации памяти используются все линии ША, а для адресации периферийных устройств - только их часть, начиная с младших разрядов. Адресацию и обращение к ОП и ПУ в изолированной системе адресного пространства можно пояснить с помощью рисунка 7.1.

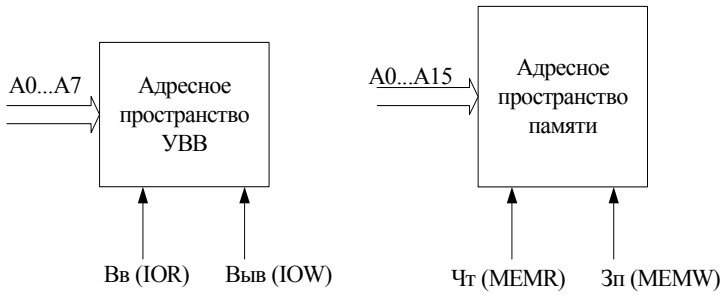


Рисунок 7.1 - Адресация и обращение к адресным пространствам памяти и УВВ

Признаком обращения к памяти являются сигналы чтения Чт (MEMR) и записи Зп (MEMW). Соответственно, при обращении к УВВ используются сигналы ввод Вв (IOR) и вывод Выв (IOW), формируемые процессором в шинных циклах ввода-вывода. Для обращения к ОП используются одни команды, для обращения к ПУ- другие, причем последних как правило очень мало (две команды для МП КР580ВМ80А и четыре для МП1810ВМ86). Размеры адресных пространств памяти и УВВ ряда процессоров приведены в таблице 7.1.

Таблица 7.1- Размеры адресных пространств памяти и УВВ различных процессоров

Тип процессора	УВВ		Память	
	Использ. линии ША	Объем адресного простран-ства	Использ. линии ША	Объем адресного простран-ства
580ВМ80А	A0…A7	256 байт	A0…A15	64 Кбайт
1821ВМ85	A0…A7* A8…A15	256 байт	----“------	----“-----
1810ВМ86	*	64 Кбайт	A0…A19	1 Мбайт
Intel80286	A0…A15	----“-----	A0…A23	16 Мбайт
Intel80386	----“-----	----“-----	A0…A31	4 Гбайт
Pentium	----“-----	----“-----	A0...A31	4 Гбайт
Pentium II	----“-----	----“-----	A0...A33	16 Гбайт

- На линии A8…A15 выдаются копии состояния линий A0…A7.

В соответствии с таблицей, размер адресного пространства памяти перечисленных процессоров увеличился с 64 Кбайт до 16 Гбайт. В то же время размер адресного пространства УВВ не превышает 64 Кбайт, из которого в ПК обычно задействован только первый килобайт.

Совмещенное адресное пространство памяти и ввода- вывода

В совмещенной системе используется общая адресация ОП и УВВ, при этом регистры УВВ с точки зрения операций записи-чтения идентичны ячейкам ОП. При адресации ПУ и ОП используются все линии ША, но под адресное пространство ОП отводится одна область, а под адресное пространство УВВ- другая область адресного пространства процессора, различающиеся значениями адресов (см. рисунок 7.2).

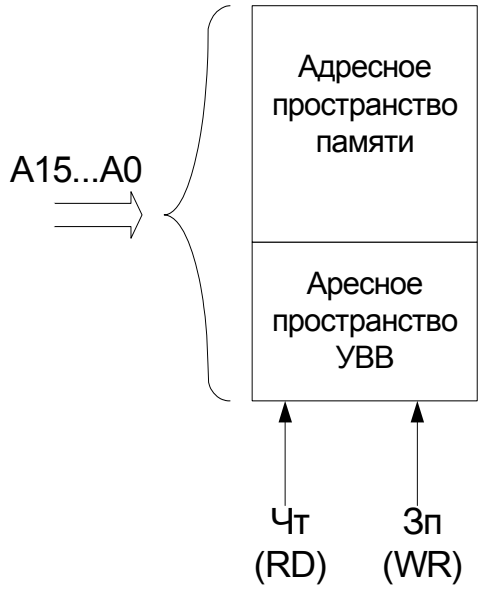


Рисунок 7.2- Адресация и обращение к адресным пространствам памяти и УВВ при совмещенной системе адресных шин

Достоинством совмещенного адресного пространство памяти и ввода- вывода является возможность обращения к УВВ с помощью тех же команд, что и к ОП. Недостатками его являются:

- уменьшение адресного пространства памяти;
- усложнение схем дешифрации адреса УВВ.

Построение постоянной памяти

Полупроводниковая память ЭВМ обычно включает в себя постоянную ПЗУ (Read Only Memory - ROM) и оперативную память ОЗУ (Random Access Memory - RAM). Постоянная память способна хранить информацию при отключении питания, поэтому предназначается для хранения программ и констант. В оперативной памяти при отключении питания информация теряется, поэтому в ней хранят промежуточные результаты, например данные, подлежащие обработке.

С точки зрения занесения информации в ПЗУ, т.е. их программирования, можно выделить следующие типы постоянной памяти:

- программируемые с помощью маски – программируются на стадии изготовления ПЗУ заводом-изготовителем;
- электрически программируемые ПЗУ. Программирование таких ПЗУ заключается в разрушении импульсом тока плавких перемычек неиспользуемых выводов транзисторов;
- перепрограммируемые ПЗУ (ППЗУ) называемые также репрограммируемыми (РПЗУ). Для их изготовления используют МОП- транзисторы с плавающим затвором. При записи в них информации (электрически) за счет создания большой разности потенциалов между стоком и истоком в области затвора формируется заряд, получаемый инжекцией электронов из области истока. Этот заряд может сохраняться десятки лет. Для стирания записанной информации в

микросхемах РПЗУ-УФ (EPROM) используется ультрафиолетовое излучение, а в микросхемах РПЗУ-ЭС (EEPROM) стирание производится электрическим способом;

- флэш – память (Flash - Memory). По принципу действия подобна РПЗУ-ЭС. Особенностью флэш – памяти является процесс стирания информации, которое производится либо для всего объема памяти, либо для достаточно больших блоков. Это позволяет упростить управление и повысить степень интеграции микросхем флэш – памяти.

Типичная емкость микросхем ПЗУ средней степени интеграции (СИС), составляет 256, 512, 1К четырех или восьми разрядных слов. Используем условное графическое обозначение (УГО) таких ПЗУ, приведенное на рисунке 7.3.

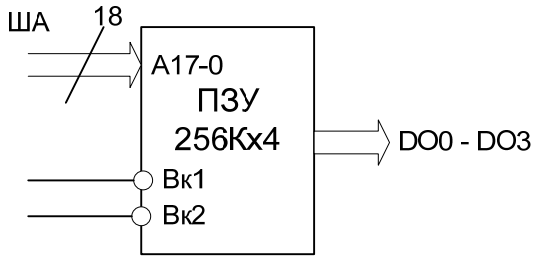


Рисунок 7.3 - Обозначение микросхемы ПЗУ 256К х 4

Входы микросхемы Вк1 и Вк2 используются для подачи сигналов выборки кристалла. Наличие нескольких сигналов Вк (Chip Select - CS) облегчает объединение микросхем в модули большей емкости и разрядности.

При проектировании памяти ЭВМ разработчиком должна решаться задача построения памяти требуемой емкости и разрядности. Для этого предварительно выбирается микросхема (или микросхемы) памяти, удовлетворяющей требуемым условиям по емкости, быстродействию, совместимости уровней и другим характеристикам. Затем определяется необходимое число К микросхем по формуле:

$$K = \frac{N \bullet Q_N}{n \bullet q_n}$$

где N - требуемая разрядность памяти; Q_N - требуемая емкость памяти;

n - разрядность выбранной микросхемы памяти; q_n - ее емкость.

Рассмотрим на примере организацию страницы памяти емкостью 256 Кбайт из микросхем, имеющих организацию 256К х 4 (см. рисунок 7.4).

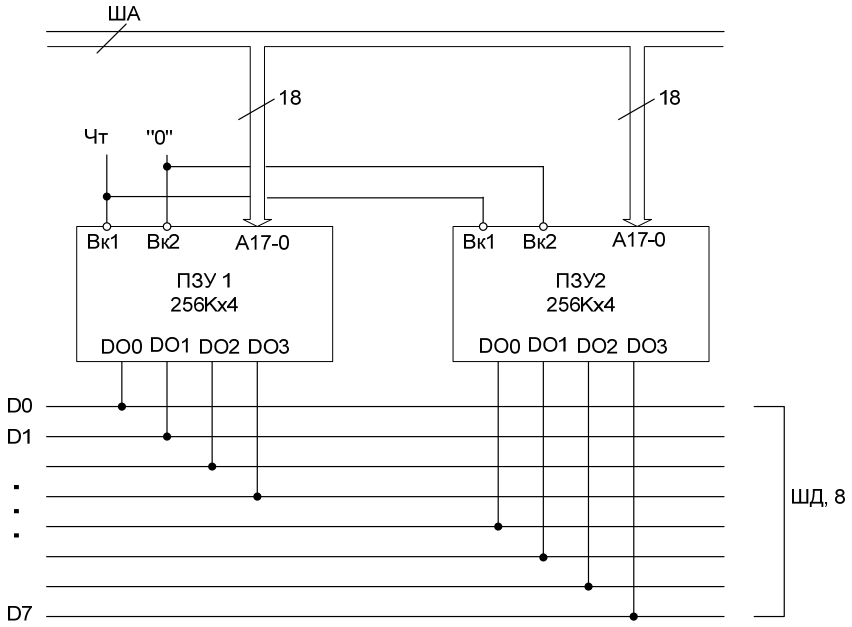


Рисунок 7.4- Построение страницы памяти требуемой разрядности из микросхем меньшей разрядности

Для адресации страницы в 256 Кбайт достаточно ША шириной восемнадцать двоичных разрядов. Если требуется большая емкость памяти, она набирается из нескольких страниц. Страницы подключаются к шине данных в режиме временного мультиплексирования. На рисунке 7.5 показан пример построения ПЗУ емкостью 512 Кбайт из 4-х микросхем с организацией 256К х 4. Две верхних микросхемы ПЗУ 256К х 4 составляют “Страницу 0” памяти, две нижних - “Страницу 1”. Для адресации памяти емкостью 512 Кбайт требуется 19 двоичных разрядов, из них 18 разрядов (А17… А0) используются для адресации ячеек памяти внутри станиц, а разряд А18 используются для выборки страниц.

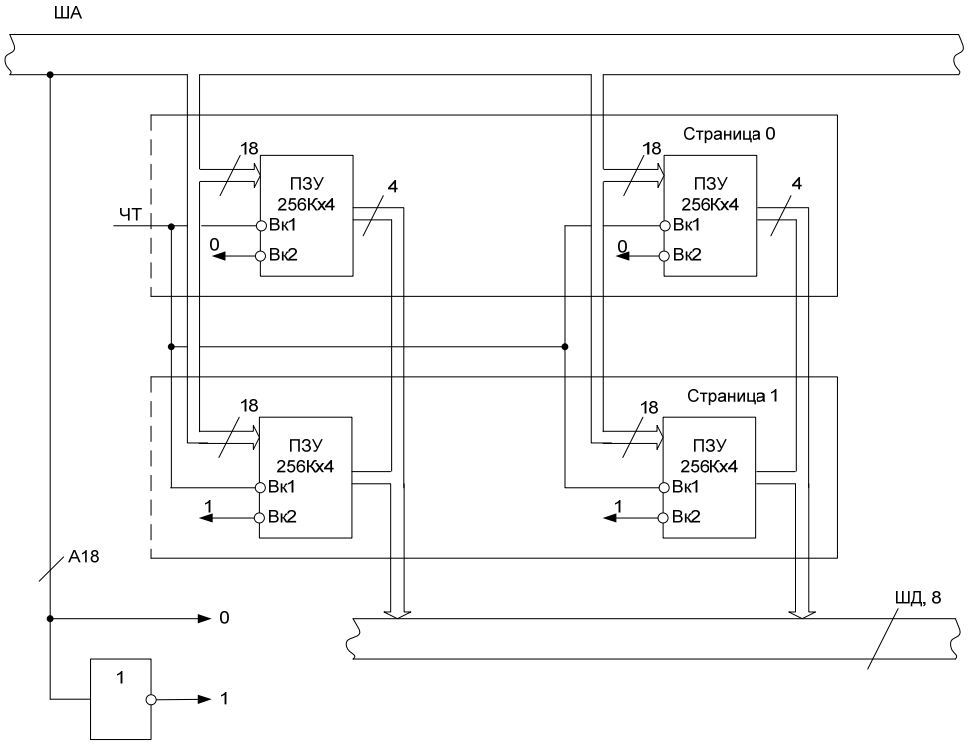


Рисунок 7.5 - Построение ПЗУ 512К*8 из микросхем памяти 256К*4

Для построения ПЗУ емкостью 1024 Кбайт из микросхем с той же организацией потребуется 8 микросхем, разбитых на 4 страницы по 256 Кбайт. Для выбора нужной страницы необходима дешифрация уже двух старших адресных разрядов А19 и А18.

7.3 Построение подсистемы памяти

Первые микросхемы ОЗУ при проведении операций записи и считывания использовали раздельные линии для входных и выходных данных: входные линии данных - DI (Data Input) и выходные линии - DO (Data Output). Выходы данных таких микросхем, как правило, не имели третьего (высокоимпедансного) состояния. Для создания двунаправленных шин применяют шинные формирователи (ШФ), усиливающие мощность выходных сигналов микросхемы памяти и обеспечивающие режим высокого выходного сопротивления (высокого импеданса).

В современных ОЗУ и для записи и для чтения используется двунаправленная шина данных (на схемах обозначается как DIO или DB), сформированная внутри микросхемы.

На рисунке 7.6 показан пример построения памяти ЭВМ, состоящей из оперативной памяти емкостью 512 Кбайт и постоянной памяти емкостью 2 Мбайта. Для построения подсистемы памяти использованы микросхемы ОЗУ с организацией 256К × 4 и двунаправленной ШД, а микросхемы ПЗУ – 1М × 8.

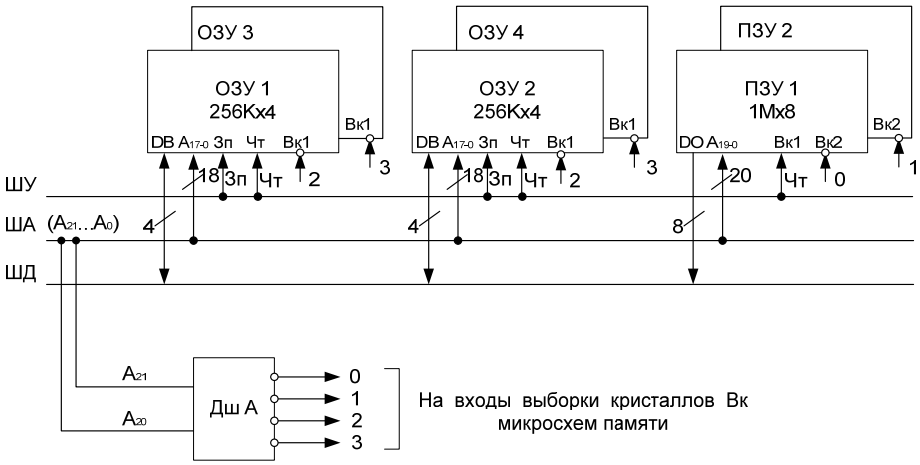


Рисунок 7.6 - Подсистема памяти, состоящая из микросхем ОЗУ и ПЗУ

Микросхемы ОЗУ1 и ОЗУ2 образуют первую страницу оперативной памяти, а микросхемы ОЗУ3 и ОЗУ4- вторую. Заметим, что при использовании простого дешифратора страниц памяти, между первой и второй страницей ОП будет разрыв адресного пространства. Для избегания от такой нежелательной ситуации потребуется включение большего числа старших разрядов адреса в процедуру дешифрации и, соответственно, применения более сложного дешифратора.

Аналогичные схемотехнические решения используются и для построения памяти ЭВМ большей емкости и разрядности.

7.4 Построение оперативной памяти на микросхемах DRAM

Как уже отмечалось ранее, для уменьшения количества внешних контактов (выводов) адресные входы в микросхемах динамической памяти мультиплексируются. Полное количество разрядов ША, подаваемое на микросхему DRAM делится на две части- адрес строки и адрес столбца. При адресации ячеек DRAM эти части адреса должны, последовательно во времени, подаваться на адресные входы микросхемы в сопровождении стробов RAS и CAS.

Однако процессор при выполнении шинных циклов чтения и записи памяти выдает весь адрес целиком, одновременно помещая на свою ША все его разряды. Разделение полного адреса запоминающего элемента и последовательную выдачу его на микросхему DRAM осуществляет мультиплексор, являющийся частью контроллера динамической памяти, расположенный между процессором и DRAM (см. рисунок 7.7). При осуществлении операция доступа (чтения или записи) к памяти, контроллер принимает от процессора полный адрес и сигналы ШУ, информирующие его о типе шинного цикла. После этого контроллер последовательно во времени выдает две части адреса на адресные входы микросхемы в сопровождении стробов RAS и CAS. Он же вырабатывает сигналы управления R/W# (знак # обозначает, что активным уровнем сигнала является лог. 0). Данные подаются с процессора на DRAM и обратно через ДШФ, который может также входить в состав контроллера.

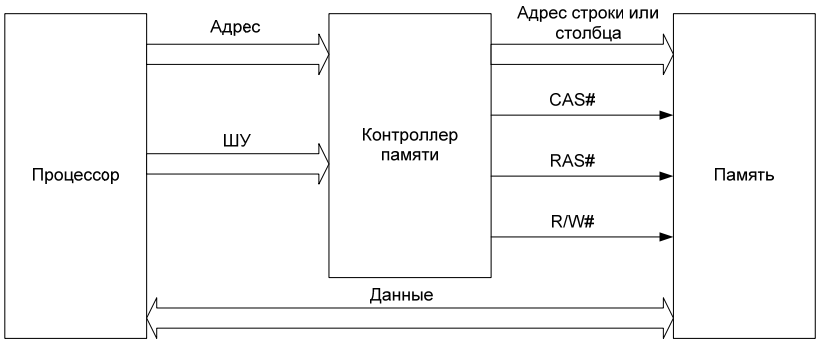


Рисунок 7.7- Организация подсистемы динамической памяти

При вводе адреса строки выбранная строка считывается в регистр-защелку микросхемы DRAM. Подача адреса столбца в сопровождении строба CAS выбирает в регистре- защелке, в зависимости от организации микросхемы DRAM, бит, тетраду, байт и т.д. При появлении сигнала

чтения выбранная информация выдается на ШД, после чего его содержимое регистра- защелки переписывается в прежнюю строку микросхемы DRAM.

При записи информация, поступившая на микросхему DRAM с ШД, записывается сначала в соответствующие разряды регистра- защелки, после чего его содержимое также переписывается в прежнюю строку микросхемы DRAM.

1. В чем заключается понятие виртуальной памяти и какие она создает возможности?
2. Как влияет виртуальная организация на производительность вычислительной системы?
3. Каким образом осуществляется преобразование виртуальных адресов в физические?
4. Какие преимущества страничной организации по отношению к сегментной и наоборот?
5. Какие преимущества странично-сегментной организации по отношению к сегментной и страничной?
6. Какие отличия в организации изолированной и совмещенной систем адресных шин?
7. Перечислите достоинства и недостатки изолированной и совмещенной систем адресных шин.
8. Перечислите основные типы ПЗУ.
9. Как определяется необходимое число микросхем для построения памяти нужной емкости и разрядности?

9 ОРГАНИЗАЦИЯ ВНЕШНЕЙ ПАМЯТИ ЭВМ

Иерархически организуемая память ЭВМ наряду с основной (оперативной) памятью высокого быстродействия, но сравнительно небольшой емкости (до нескольких мегабайт) содержит внешнюю память, намного медленнее работающую, но способную хранить практически сколь угодно большой объем необходимой для функционирования вычислительной установки информации. В основном, внешняя память реализована на основе электромеханических ЗУ с носителем информации в виде движущейся поверхности, покрытой тонким слоем магнитного материала. Внешние ЗУ (ВЗУ) являются устройствами с произвольным обращением, допускающими многократное считывание информации и запись новой информации на место ранее записанной. Электромеханические ВЗУ компактны, сравнительно дешевы (в пересчете стоимости на 1 бит хранимой информации), могут хранить в одном устройстве (модуле) объемы информации, достигающие сотен миллионов байт. Необходимый объем внешней памяти может достигаться подсоединением к ЭВМ соответствующего количества ВЗУ.

В настоящее время, наиболее распространенные следующие виды ВЗУ - это магнитные и оптические диски, магнитоленточные устройства, а также твердотельные ЗУ.

9.1 Характеристики ЗУ внешней памяти

Рассмотрим основные характеристики ЗУ внешней памяти:

- общий объем хранимой информации (емкость);
- быстродействие;
- метод доступа;
- единица пересылки;
- физический тип;
- энергонезависимость ;
- место расположения;
- достоверность функционирования;
- относительная стоимость.

Емкость ЗУ внешней памяти достаточно велика, поэтому ее характеризуют такими крупными единицами, как мегабайт, гигабайт, терабайт и петабайт.

Быстродействие ЗУ. Из-за большого различия быстродействия ОП и ВЗУ обращения к внешней памяти порождают потери производительности ЭВМ. Поэтому быстродействие ВЗУ является показателем не менее, а в ряде случаев даже более важным, чем его емкость.

Обращение к внешней памяти в общем случае предполагает последовательное выполнение двух процессов:

а) доступа к ВЗУ - подведения к головке (головкам) участка носителя, где находится нужная информация или куда информация должна быть записана;

б) считывания и передачи информации из ВЗУ в ОП или передачи информации из ОП в ВЗУ и записи ее на носитель.

Соответственно быстродействие ВЗУ определяется двумя показателями - средним или максимальным временем доступа и скоростью передачи информации (скоростью записи-считывания).

При оценке быстродействия необходимо учитывать применяемый в данном типе *ЗУ метод доступа* к данным.

В ВЗУ используется два основных метода доступа:

Последовательный доступ (позволяет обращаться к ячейкам ЗУ в определенной последовательности). ЗУ с последовательным доступом ориентировано на хранение информации в виде последовательности блоков данных, называемых записями. Для доступа к нужному элементу (слову или байту) необходимо прочитать все предшествующие ему данные. Время доступа зависит от положения требуемой записи в последовательности записей на носителе информации и позиции элемента внутри данной записи. Примером может служить ЗУ на магнитной ленте.

Прямой доступ. Каждая запись имеет уникальный адрес, отражающий ее физическое размещение на носителе информации. Обращение осуществляется по адресу к началу записи, с последующим последовательным доступом к определенной единице информации внутри записи. В результате время доступа к определенной позиции является величиной переменной. Такой режим характерен для магнитных дисков.

Говоря о *физическом типе* запоминающего устройства, необходимо упомянуть три наиболее распространенных технологии внешних ЗУ - это полупроводниковая память, лежащая в

основе твердотельных дисков, память с магнитным носителем информации, используемая в магнитных дисках и лентах, и память с оптическим носителем - оптические диски.

Энергонезависимость - определяется физическими особенностями ВЗУ. Магнитная и оптическая память - энергонезависимы. Полупроводниковая память может быть как энергозависимой, так и нет, в зависимости от ее типа. В твердотельных дисках, используется оба варианта.

Место расположения. Внешние ЗУ обычно выполнены в виде отдельных устройств или встраиваемых блоков. В последнем случае они монтируются в специально отведенных для них местах корпуса вычислительной машины. К центральным устройствам ЭВМ внешние ЗУ подключаются аналогично устройствам ввода/вывода.

Достоверность функционирования ВЗУ. Обычно достоверность работы ВЗУ оценивают числом правильно воспроизводимых в режиме записи-считывания двоичных знаков на один ошибочный знак.

Относительная стоимость ЗУ, принято оценивать отношением общей стоимости ЗУ к его емкости в битах, то есть стоимостью хранения одного бита информации.

9.2 Внешняя память на основе магнитных дисков

Информация в ВЗУ на магнитных дисках хранится на плоских дисках из немагнитного материала, покрытых намагничивающимся материалом. Традиционно подложка выполнялась из алюминия или его сплавов. В последнее время стали применяться подложки из стекла, позволяющие улучшить ряд характеристик ВЗУ на МД. Данные записываются и считываются с диска с помощью *магнитной головки считывания/записи,* представляющую из себя электромагнитную катушку, которая в процессе считывания и записи неподвижна, в то время как диск вращается относительно нее, с некоторой угловой скоростью. В результате данные образуют концентрические окружности на поверхности диска, называемые дорожками.

Запись основана на том, что ток, проходящий через катушку, создает магнитное поле. При записи на головку подаются электрические импульсы, намагничивающие участок поверхности под ней. Характер намагниченности поверхности различен в зависимости от направления тока в катушке. Одна полярность соответствует логической единице, а другая - логическому нулю. Считывание основано на электрическом токе, наводимом в катушке головки, под воздействием перемещающегося относительно нее магнитного поля. Когда под головкой проходит участок поверхности диска, в катушке наводится ток той же полярности, что использовался для записи информации.

9.2.1 Процедуры чтения и записи

В большинстве систем имеются две независимые головки: головка считывания и головка записи (рис.9.1).

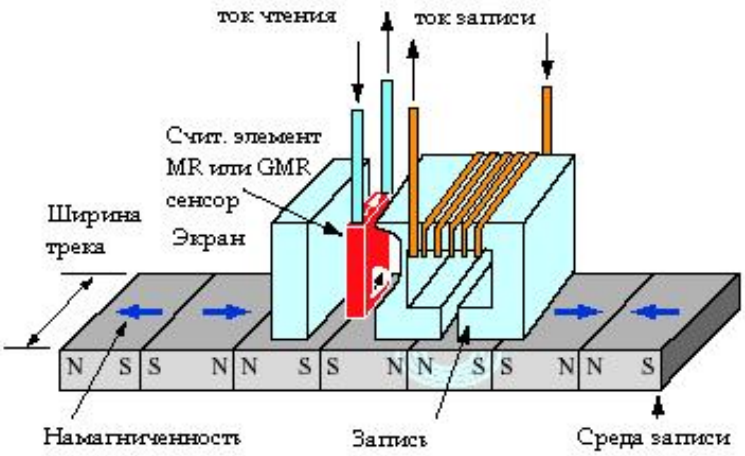


Рисунок 9.1-Принцип продольной записи на магнитный носитель

В ВЗУ на МД, где вектор намагниченности расположен в плоскости диска (продольная запись), головка записи имеет вид прямоугольной катушки с прорезью со стороны магнитного диска и несколькими витками провода с противоположной стороны. Электрический ток в проводе создает магнитное поле в прорези, которое, в свою очередь, намагничивает маленький участок на

поверхности диска (битовую ячейку). Изменение направления тока меняет на противоположное намагничивание магнитной поверхности. Структура головки считывания практически такая же, что и у головки записи, поэтому одна и та же головка может быть использована для обеих операций. Такие совмещенные головки применяются в накопителях на гибких магнитных дисках, а ранее - также в системах с жесткими дисками. Современные системы жестких дисков используют разные механизмы считывания, требующие отдельной головки считывания, расположенной по возможности ближе к головке записи. Головка считывания состоит из магниторезистивного материала, электрическое сопротивление которого зависит от направления намагниченности среды, движущейся под головкой. Изменения сопротивления фиксируются как изменение напряжения, вызванное пропусканием тока через сенсор с изменяющимся сопротивлением.

В последнее время для увеличения плотности записи используют технологию перпендикулярной записи (рис.9.2).

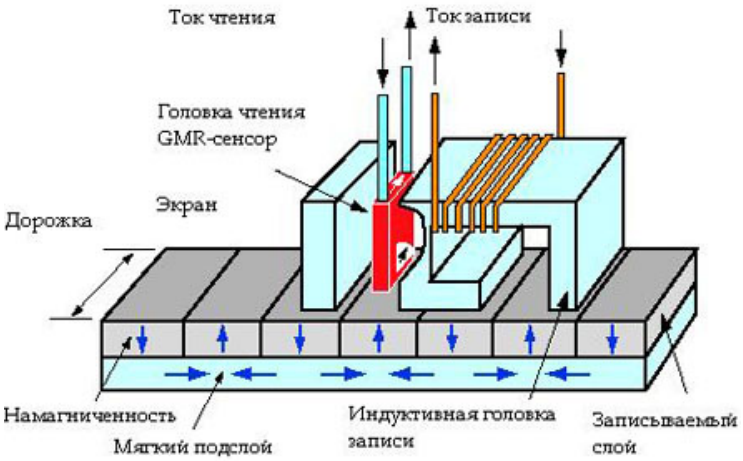


Рисунок 9.2-Принцип перпендикулярной записи на магнитный носитель

При перпендикулярной записи на диск магнитные частицы располагаются под углом 90° к плоскости магнитного диска. Благодаря этому домены, хранящие разные значения, не отталкиваются друг от друга, потому что намагниченные частицы повернуты друг к другу разными полюсами. Увеличение плотности, означающее уменьшение размера частиц, при этом не требует уменьшения толщины слоя, что обеспечивает стабильность магнитного материала. Для перпендикулярной записи на магнитный слой используется головка новой конструкции. Если при продольной записи магнитное поле генерируется в металлическом кольце с помощью индукции, то при перпендикулярной используется поле, генерируемое между срезом полюса головки записи и магнито-мягким подслоем на диске. Поэтому частицы записывающего слоя намагничиваются вертикально, а частицы магнитного подслоя – горизонтально.

Это обеспечивает дополнительную стабильность частиц относительно друг друга.

Важное отличие перпендикулярной записи от продольной заключается в характере и расположении сигнала чтения. Продольный магнитный слой без подслоя испускает магнитный сигнал только с границы перехода бит (с границы между одной магнитной частицы и другой) под прямым углом к плоскости диска. Перпендикулярный магнитный слой испускает сигнал по всей площади частицы, а благодаря подслою вектор этого сигнала направлен параллельно плоскости диска. Для считывания требуются принципиально новые головки чтения, которые позволяют значительно увеличить соотношение сигнал/шум и мощность самого сигнала. Поэтому некоторые компании уже начинают применять новое поколение головок, использующее туннельный магниторезистивный эффект (TMR Heads).

При перпендикулярной записи используется намного более сложный состав магнитного слоя. Под тонким защитным слоем расположен записывающий слой состоящий из окисленного сплава кобальта, платины и хрома. Подложка состоит из двух слоев сложного химического состава, называемых антиферромагнитносвязанными слоями. Именно они позволяют снять внутренние напряженности магнитного поля. В настоящее время использование перпендикулярной записи позволило создать диски с плотностью записи до 500 Гбит на дюйм. При этом емкость 3,5” накопителя составляет 2 Тбайт, 2,5” – до 640 Гбайт, 1” – 50 Гбайт.

9.2.2 Характеристики дисковых систем

Положение головок относительно поверхности дисков может быть фиксированным или изменяющимся. В ВЗУ с фиксированными головками на каждую дорожку приходится по одной головке считывания/записи. Головки смонтированы на жестком рычаге, пересекающем все дорожки диска. В дисковом ЗУ с nодвижными головками имеется только одна головка, также установленная на рычаге (рис. 9.3), однако рычаг способен перемещаться в радиальном направлении над поверхностью диска, обеспечивая возможность позиционирования головки на любую дорожку.

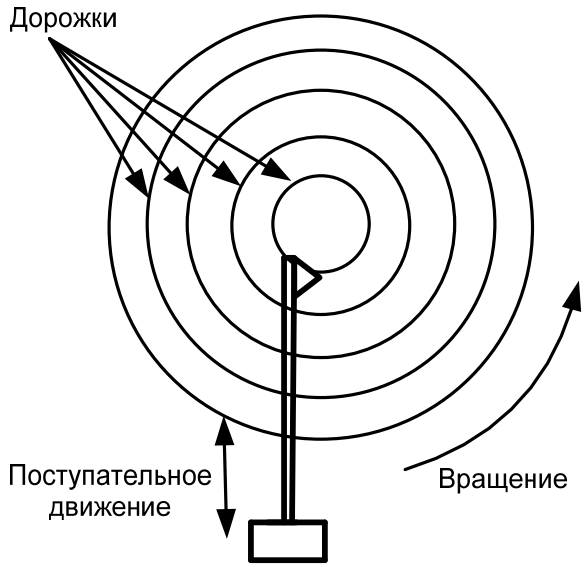


Рисунок 9.3- Вариант организации дисковой системы с подвижной головкой

Диски с магнитным носителем устанавливаются в дисковод, состоящий из рычага, шпинделя, вращающего диск, и электронных схем, требуемых для ввода и вывода двоичных данных. Диски могут быть несъемными либо съемными. Несъемный диск зафиксирован на дисководе. Съемный диск может быть вынут из дисковода и заменен на другой аналогичный диск. Преимущество системы со съемными дисками - возможность хранения неограниченного количества данных при ограниченном числе дисковых устройств. Кроме того, такой диск может быть перенесен с одной ЭВМ на другую. На оси может располагаться один или несколько (рис. 9.4) дисков. В последнем случае используют термин дисковый пакет. В современных накопителях обычно устанавливается несколько дисков, и данные записываются на обеих сторонах каждого из них. В большинстве накопителей имеются два или три диска (что позволяет выполнять запись на четырех или шести сторонах), но существуют также устройства, содержащие до 11 и более дисков. Однотипные (одинаково расположенные) дорожки на всех сторонах дисков объединяются в цилиндр. Для каждой стороны диска предусмотрена своя дорожка чтения/записи, но при этом все головки смонтированы на общем стержне, или стойке. Поэтому головки не могут перемещаться независимо друг от друга и двигаются только синхронно.

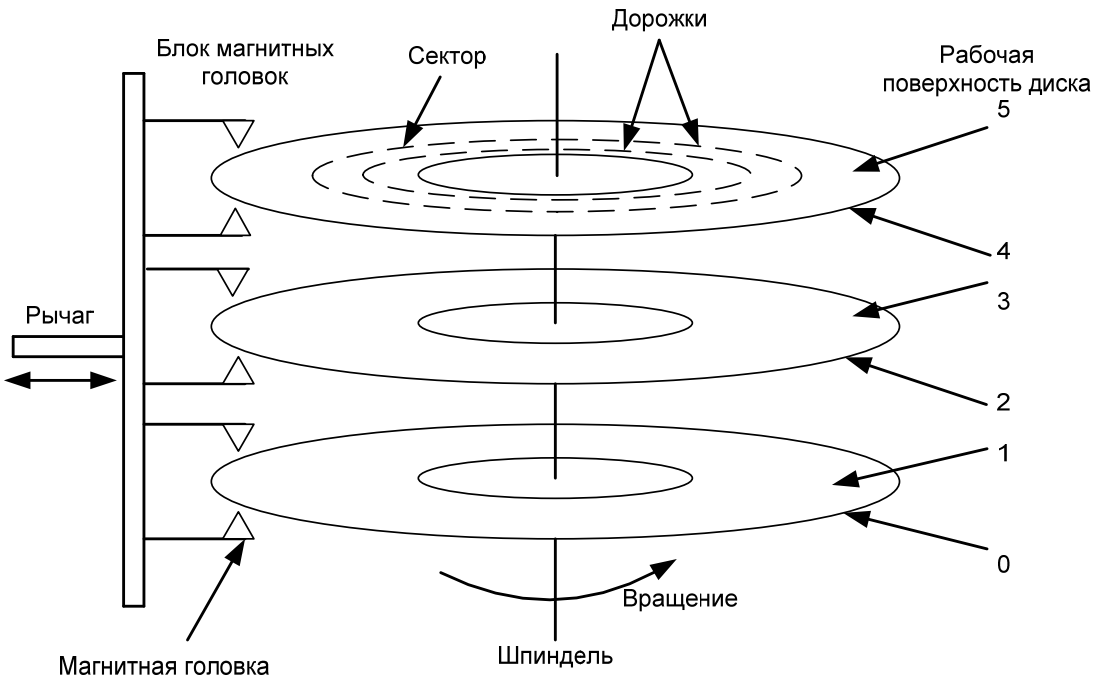


Рисунок 9.4- ВЗУ с пакетом дисков

Жесткие диски вращаются достаточно быстро. Частота их вращения даже в большинстве первых моделей составляла 3600 об/мин (то есть в 10 раз больше, чем в накопителе на гибких дисках). В настоящее время частота вращения жестких дисков возросла до 6400, 7200, 10000 об/мин и даже 15 000 об/мин. В зависимости от применяемой головки считывания/записи можно выделить три типа дисковых ЗУ. В первом варианте головка устанавливается на фиксированной дистанции от поверхности (плавающая головка) так, чтобы между ними оставался воздушный промежуток (бесконтактный способ записи/чтения). Второй вариант -это когда в процессе чтения и записи головка и диск находятся в физическом контакте. Такой способ используется, например, в накопителях на гибких магнитных дисках (дискетах), и накопителях на магнитной ленте. Для правильной записи и считывания головка должна генерировать и воспринимать магнитное поле определенной интенсивности, поэтому чем уже головка, тем ближе должна она размещаться к поверхности диска (более узкая головка означает более узкие дорожки, а значит, и большую емкость диска). Однако приближение головки к диску означает и больший риск ошибки за счет загрязнения и дефектов. В процессе решения этой проблемы был создан диск типа винчестер (рис.9.6).

Рисунок 9.5-Дисковое ЗУ типа винчестер.

Головки винчестера и диски помещены в герметичный корпус, защищающий их от загрязнения. Кроме того, головки имеют очень малый вес и аэродинамическую форму. Они спроектированы для работы значительно ближе к поверхности диска, чем головки обычных жестких дисках, тем самым повышается плотность хранения данных. При остановленном диске головка прилегает к его поверхности. Давления, возникающего при вращении диска, достаточно для подъема головки над поверхностью. В результате создается неконтактная система с узкими головками считывания/ записи, обеспечивающая более плотное прилегание головки к поверхности диска, чем в обычных жестких дисках.

9.2.3 Организация данных и форматирование

Данные на диске организованы в виде набора концентрических окружностей, называемых дорожками образованными на поверхности МД с помощью магнитной головки при форматировании низкого уровня. Они нумеруются с 0 от внешнего края МД. (рис. 9.6). На поверхности диска располагаются тысячи дорожек. Каждая из них имеет ту же ширину, что и головка. Соседние дорожки разделены промежутками. Это предотвращает ошибки из-за смещения головки и из-за интерференции магнитных полей. Как правило, для упрощения схемы управления принимается, что на всех дорожках может храниться одинаковое количество информации. Таким образом, плотность записи увеличивается от внешних дорожек к внутренним.

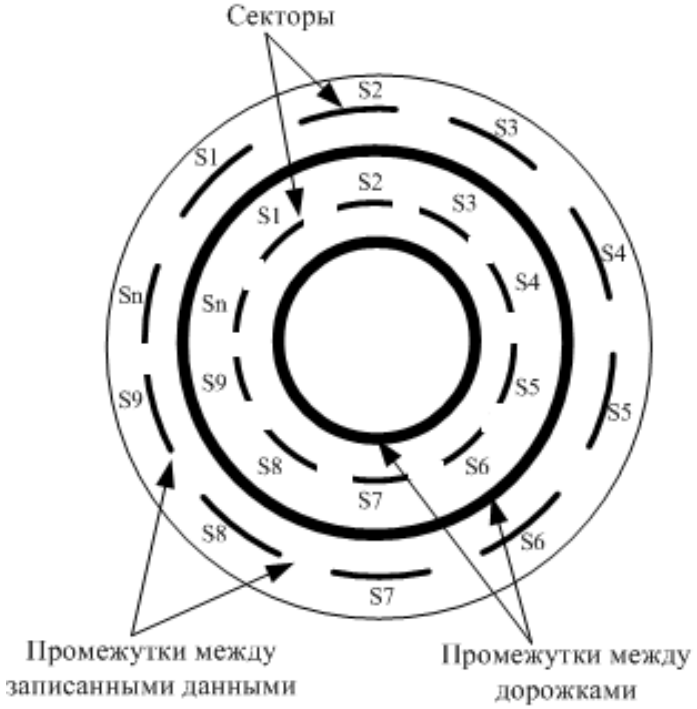


Рисунок 9.6- Порядок размещения информации на магнитном диске

Дорожка записи на диске слишком велика, чтобы использовать ее в качестве единицы хранения информации. Во многих накопителях ее емкость превышает 100 тыс. байтов, и отводить такой блок для хранения небольшого файла нецелесообразно. Поэтому дорожки на диске разбивают на нумерованные отрезки, называемые секторами. Обмен информацией с ВЗУ на МД осуществляется секторами. Количество секторов может быть разным в зависимости от плотности дорожек и типа накопителя. Например, дорожка жесткого диска может содержать от 380 до 700 секторов. Секторы, создаваемые с помощью стандартных программ форматирования, имеют емкость 512 байтов, но не исключено, что в будущем эта величина изменится. В общем виде структура сектора имеет следующий вид, см.рис.9.7

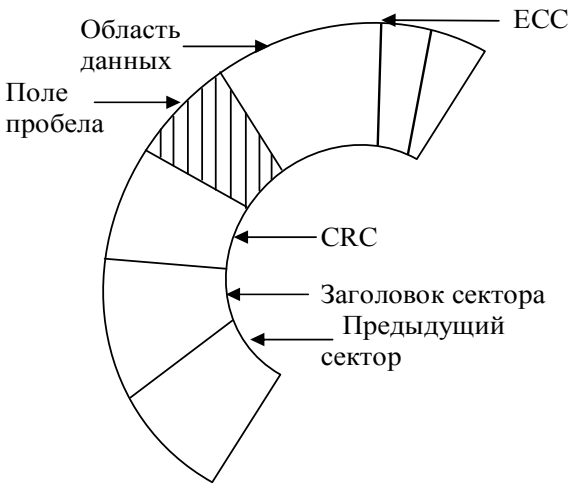


Рисунок 9.7- Структура сектора жесткого диска

В заголовок сектора входит номер цилиндра, головки, физический номер сектора и так называемый флажок качества. CRC служит для проверки контроллером правильности считывания заголовка сектора. Контроль проводится с помощью циклического кода и позволяет только обнаружить ошибку. Поле пробела используется для задания промежутка времени при переходе из режима чтения в режим записи (задержка, интервал успокоения).

Поле ECC используется для обнаружения и коррекции ошибок используются систематические разделимые помехоустойчивые коды. Если контрольным разрядам в комбинации для некоторого кода могут быть четко определено место (номера разрядов), тогда код называют систематическим или разделимым. В противном случае код является неразделимым.

Структура ECC зависит от вида данных и от кода коррекции.

Соседние секторы на дорожке разделены между собой межсекторными промежутками. Нумерация секторов на дорожке начинается с единицы, в отличие от головок цилиндров, отсчет которых ведется с нуля. Бит, находящийся вблизи центра вращающегося диска, проходит под головкой считывания/записи медленней, чем бит, расположенный на периферии диска. По этой причине нужен какой-либо механизм для компенсации этого различия в скоростях, с тем чтобы головка могла читать все биты с одинаковой скоростью.

Это может быть обеспечено путем увеличения расстояния между битами информации, записываемой в сегментах диска. Информация затем может сканироваться с той же скоростью путем вращения диска с фиксированной скоростью, известной как постоянная угловая скорость (ПУС). На рис. 9.8, а показана организация диска, использующего ПУС. Диск поделен на несколько секторов и на ряд концентрических дорожек. Преимущество использования ПУС в том, что к индивидуальным дорожкам данных можно прямо адресоваться, указав дорожку и сектор. Перемещение головки от текущего положения к определенному адресу требует лишь короткого движения головки к нужной дорожке и короткого ожидания, когда под ней окажется нужный сектор. Недостаток систем с ПУС в том, что количество данных, которое может храниться на длинных внешних дорожках, то же самое, что и на коротких внутренних дорожках, т.е. не рационально используется дисковое пространство.

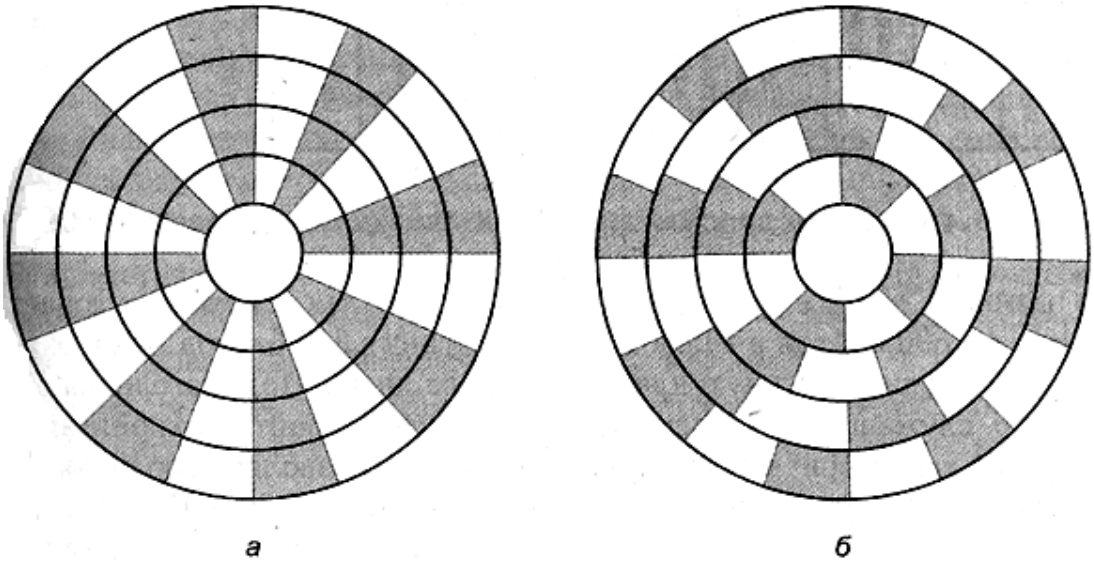


Рисунок 9.8- Методы организации диска: а - с постоянной угловой скоростью; б - с зонной записью

Ввиду того, что плотность информации возрастает по мере приближения к внутренним дорожкам, емкость диска ограничивается максимальной плотностью записи на самой внутренней дорожке. Для увеличения плотности в современных дисковых системах используется способ, известный как зонная запись, где поверхность разбивается на несколько зон (обычно 16). Внутри зоны число битов на дорожку постоянно. Зоны, более удаленные от центра, содержат больше битов (больше секторов), чем зоны, расположенные ближе к центру вращения. Это позволяет достичь большей емкости за счет более сложной аппаратной части. По мере движения головки диска от одной зоны к другой длина индивидуальных битов меняется, вызывая изменения во временных соотношениях чтения и записи. На рис. 9.8, б показана сущность зонной записи, каждая зона имеет ширину только одной дорожки.

При такой организации в ВЗУ МД должны быть заданы: точка отсчета секторов и способ определения начала и конца каждого сектора. Все это обеспечивается с помощью форматирования, в ходе которого на диск заносится служебная информация недоступная пользователю и используемая только аппаратурой дискового ЗУ.

Пример разметки МД показан на рис. 9.9. Здесь каждая дорожка включает в себя 30 секторов по 600 байтов в каждом.

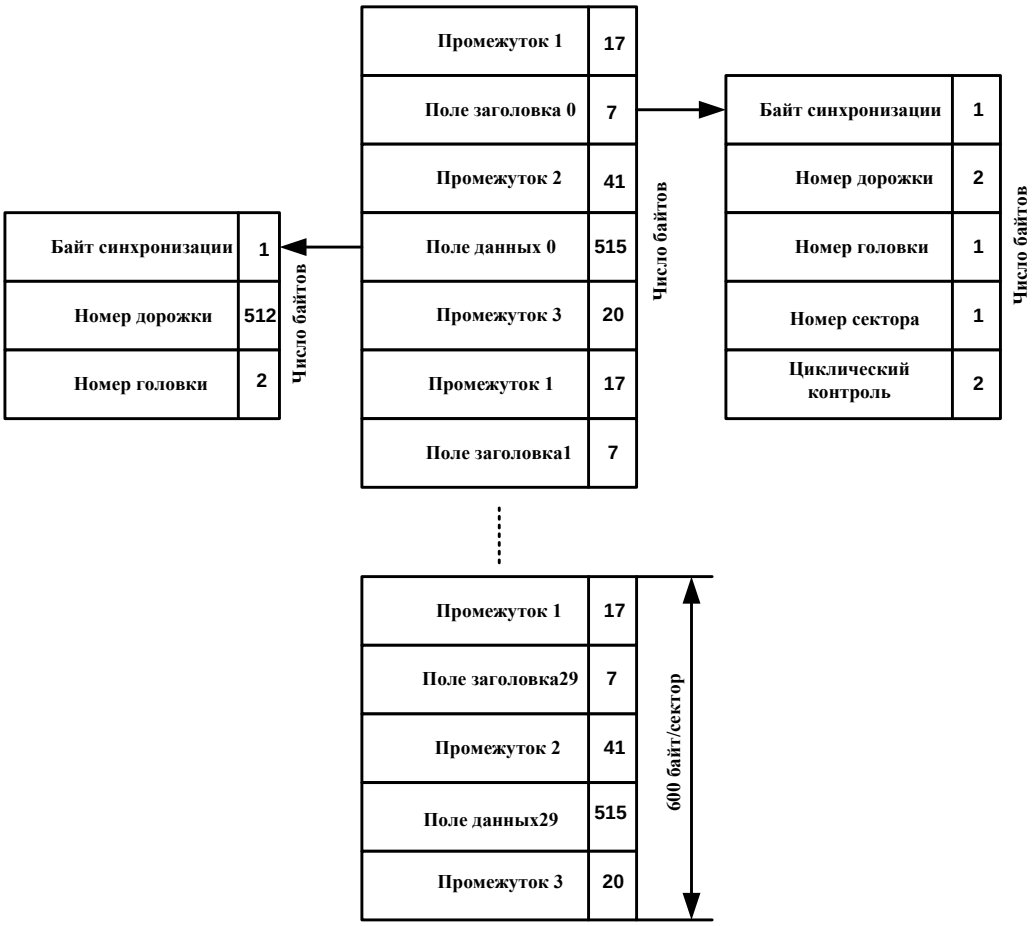


Рисунок 9.9- Формат дорожки диска типа «Винчестер» (Seagate ST506)

Сектор хранит 512 байтов данных и управляющую информацию, нужную для контроллера диска, Поле заголовка содержит информацию, служащую для идентификации сектора. Байт синхронизации представляет собой характерную двоичную комбинацию, позволяющую определить начало поля. Номер дорожки определяет дорожку на поверхности, если в накопителе используются несколько дисков, то номер головки определяет нужную поверхность. Поле заголовка и поле данных содержат также код циклического контроля позволяющий обнаружить и исправлять ошибки.

Своим названием эти коды обязаны такому факту, что для них часть комбинаций, либо все комбинации могут быть получены путем циклического сдвига одной или нескольких базовых комбинаций кода.

Построение такого кода основывается на использовании неприводимых многочленов в поле двоичных чисел. Такие многочлены не могут быть представлены в виде произведения многочленов низших степеней подобно тому, как простые числа не могут быть представлены произведением других чисел. Они делятся без остатка только на себя или на единицу.

Основной операцией при кодировании/декодировании циклического кода является операция деления кодовой последовательности на задающий полином $g(x)$. Деление полинома на полином заключается в последовательном сложении по модулю 2 делителя сначала со старшими разрядами (степенями) делимого, затем со старшими разрядами получающегося остатка (дополненного снесенными в него очередными разрядами делимого) до тех пор, пока степень остатка не станет меньше степени делителя.

Для проверки принадлежности воспроизведенной кодовой комбинации определенному циклическому коду, заданному полиному $g(x)$ необходимо разделить эту комбинацию на полином $g(x)$ и оценить остаток от деления: при нулевом остатке эта комбинация принадлежит коду, ненулевой остаток свидетельствует о наличии ошибки.

Полученный при указанном выше делении ненулевой остаток является синдромом ошибки. С другой стороны известно, что синдром равен остатку от деления полинома ошибки (разность по

модулю 2 ошибочной и истинной комбинации) на задающий полином. Таким образом, синдром ошибки содержит информацию об ошибочном разряде.

На практике операции деления заменяются операцией сложения по модулю 2 и сдвигом.

9.2.4 Чередование секторов

Ранее было объяснено, что время поиска данных в ВЗУ на МД состоит из двух частей:

1. Время обращения или доступа;
2. Время считывания данных после их нахождения.

Время доступа может быть снижено за счет разделения физического накопителя на несколько логических накопителей, а время считывания данных может быть уменьшено за счет оптимизации размещения секторов, которые называются чередованием секторов.

Если бы длина записываемого файла не превышала длину сектора, то говорить о чередовании секторов не было бы смысла, однако реально длина файла зачастую больше длины сектора. После чтения, записи каждого сектора контроллеру требуется определенное время (например, на обработку ECC) при этом, если следующий сектор располагается сразу за предыдущим у контроллера не будет времени на эту обработку. Начало следующего сектора пройдет мимо магнитной головки (МГ) и для его чтения или записи пришлось бы ждать целый оборот диска, пока начало сектора вновь не окажется под МГ. Для уменьшения потерь времени соседние по номеру сектора размещаются на расстоянии нескольких физических секторов друг от друга, для того чтобы время на выполнение вспомогательных операций по каждому сектору. Этот прием логического расположения секторов отличается от их физического размещения и называется чередованием секторов на диске. Наряду с нумерацией секторов подряд применяют нумерацию с чередованием секторов, при которой после нумерации каждого сектора происходит скачок через определенное число номеров (1, 3, 7 и т.д.). Такое чередование характеризуется коэффициентом чередования. Обычно для персональных компьютеров коэффициент изменяется от 1-6.

Если данные размещены в физически последовательных секторах, то коэффициент чередования равен 1 и сектора читаются за один оборот, но для этой реализации требуется более быстродействующий контроллер. Коэффициент чередования в накопителях устанавливается при форматировании низкого уровня, а именно при разбивке диска на секторы и может быть всегда изменено пользователем.

Если накопитель размечен для быстродействующего центрального процессора, то при работе с низко действующим накопителем будут потери оборотов диска. Если фактор чередования неоправданно велик, то контроллер и ЦП будут ждать следующий сектор неоправданно долго. Слишком малый фактор чередования более опасен, чем большой, т.к. создает большие потери времени. Определение оптимального значения фактора чередования становится важным по мере того, как аппаратура становится более разнообразной.

9.3 Массивы магнитных дисков

Несмотря на технологические успехи в области внешних ЗУ, построенных на самых различных физических принципах, подсистема памяти на базе магнитных дисков по-прежнему остается основой внешней памяти любой ВМ. По этой причине дисковой подсистеме предъявляются самые высокие требования в плане производительности и отказоустойчивости, и уже с самого начала использования подсистем на базе ЗУМД не прекращаются попытки улучшить их характеристики. Одно из наиболее универсальных усовершенствований было предложено в 1987 под аббревиатурой RAID (*Redundant Array of Independent (or Inexpensive) Disks)* - массив независимых (или недорогих) дисков с избыточностью. В основе концепции RAID лежит переход от одного физического магнитного диска (МД) большой емкости к массиву независимо и параллельно работающих физических дисковых ЗУ, рассматриваемых операционной системой как одно большое логическое дисковое запоминающее устройство. Такой подход позволяет повысить производительность дисковой памяти за счет возможности параллельного обслуживания запросов на считывание и запись, при условии, что данные находятся на разных дисках. Повышенная надежность достигается тем, что в массиве дисков хранится избыточная информация, позволяющая обнаружить и исправить возможные ошибки. В настоящее время, с развитием технологии производства МД, утверждение об экономичности массивов RAID становится проблематичным, что, однако, вполне компенсируется их повышенными быстродействием и отказоустойчивостью.

В настоящее время принята единая классификации RAID, включающая в себя восемь уровней. Хотя ни одна из схем массива МД не может быть признана идеальной для всех случаев, каждая из них позволяет существенно улучшить какой-то из показателей (производительность, отказоустойчивость) либо добиться наиболее подходящего сочетания этих показателей. Для всех уровней RAID характерны три общих свойства:

- RAID представляет собой набор физических дисковых ЗУ, управляемых операционной системой и рассматриваемых как один логический диск;
- данные распределены по физическим дискам массива;
- избыточное дисковое пространство используется для хранения дополнительной информации, гарантирующей восстановление данных в случае отказа диска.

9.3.1 Повышение производительности дисковой подсистемы

Повышение производительности дисковой подсистемы в RAID достигается с помощью приема, называемого расслоением или расщеплением (striping). В его основе лежит разбиение данных и дискового пространства на сегменты, так называемые полосы (strip - узкая полоса). Полосы распределяются по различным дискам массива, соответствии с определенной системой. Это позволяет производить параллельное считывание или запись сразу нескольких полос, если они расположены на разных дисках. В идеальном случае производительность дисковой подсистемы может быть увеличена в число раз, равное количеству дисков в массиве. Размер полосы выбирается исходя из особенностей каждого уровня RAID и может быть равен биту, байту, размеру физического сектора МД (обычно 512 байтов) или размеру дорожки.

Чаще всего логически последовательные полосы распределяются по последовательности дисков массива. Так, в *n*-дисковом массиве *n* первых логических полос физически расположены как первые полосы на каждом из *n* дисков, следующие *n* полос - как вторые полосы на каждом физическом диске и т. д. Набор логическая последовательных полос, одинаково расположенных на каждом ЗУ массива, называют лентой (stripe - широкая полоса).

9.3.2 Повышение отказоустойчивости дисковой подсистемы

Наиболее важным в ВЗУ является сохранение достоверности данных при записи и чтении. Их потери, имеющие высокую вероятность ведут к серьезным материальным ущербам. Чтобы этого избежать нужно знать физические и логические основы этих потерь, а также знать методику мер, предупреждающих данные потери.

Одной из целей концепции RAID была возможность обнаружения и коррекции ошибок, возникающих при отказах дисков или в результате сбоев. Достигается за счет избыточного дискового пространства для хранения дополнительной информации, позволяющей восстановить искаженные или утерянные данные. В RAID предусмотрены три возможности:

- дублирование (зеркалирование);
- код Хэмминга;
- биты паритета.

Первая возможность заключается в дублировании всех данных, при условии, экземпляры одних и тех же данных расположены на разных дисках массива. Это позволяет при отказе одного из дисков воспользоваться соответствующей информацией, хранящейся на исправных ВЗУ на МД. В принципе, распределение информации по дискам массива может быть произвольным, но для сокращения издержек, связанных с поиском копии, обычно применяется разбиение массива на пары ВЗУ на МД, где в каждой паре дисков информация идентична и одинаково расположена. Для управления парой дисков может использоваться общий или раздельные контроллеры. Избыточностъ такого дискового массива составляет 100%.

Вторая возможность основана на вычислении корректирующего кода Хэмминга для каждой группы полос, одинаково расположенных на всех дисках массива (для каждой ленты). Код Хемминга строится таким образом, что к основным информационным разрядам добавляется определенное число контрольных разрядов, которые формируются перед передачей информации путем подсчета четности суммы «1» для определенной группы информационных разрядов. Контролирующая аппаратура (приемник информации) образует из принятых информационных и контрольных разрядов, путем аналогичных подсчетов, корректирующее число, которое равно 0 при отсутствии ошибок или не равно 0, тогда оно показывает в двоичном коде номер разряда информация, в котором произошла ошибка. Ошибочный разряд автоматически, с помощью

корректирующего устройства, изменяет «ошибочный» разряд на противоположный и тем самым восстанавливается достоверность информации.

Корректирующие разряды (биты) хранятся на специально выделенных для этой цели дополнительных дисках (по одному диску на каждый бит). Так, для массива из десяти МД требуются четыре таких дополнительных диска, и избыточность в данном случае близка к 30%.

В третьем случае вместо кода Хэмминга для каждого набора полос, расположенных в идентичной позиции на всех дисках массива, вычисляется контрольная полоса, состоящая из битов паритета. В ней значение отдельного бита формируется как сумма по модулю 2 (определение четности) для одноименных битов во всех контролируемых полосах. Для хранения полос паритета требуется только один дополнительный диск. В случае отказа какого-либо из дисков массива производится обращение к диску паритета, и данные восстанавливаются по битам паритета и данным от остальных дисков массива. Избыточность при таком способе в среднем близка к 20%.

Различают несколько основных уровней организации RAID-массивов: RAID 0, 1, 2, 3, 4, 5, 6, 7. Также существуют комбинированные уровни, такие как RAID 10, 0+1, 30, 50, 53 и др.

Рассмотрим некоторые основные концепции уровней RAID, их достоинства и недостатки.

RAID 0: Дисковый массив без отказоустойчивости (Striped Disk Array without Fault Tolerance)

Схема RAID 0 обеспечивает максимально возможную производительность дисковой подсистемы. При распределении информации по дискам массива используется техника расщепления. В качестве полос могут выступать физические блоки, секторы или какие-то иные информационные единицы, размер которых не менее размера физического сектора МД. Полосы распределены по последовательным дискам массива по циклической схеме (рис. 9.10). Благодаря технике расщепления запись или чтение нескольких логически последовательных полос (вплоть до *n*) может производиться параллельно, поскольку они располагаются на разных дисках. Это обеспечивает существенное снижение общего времени ввода/вывода.

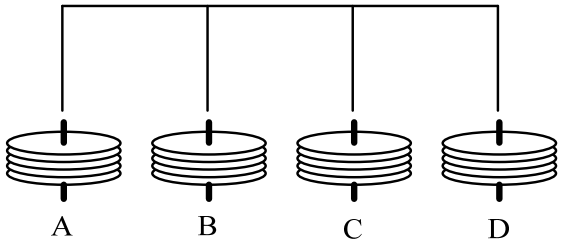


Рисунок 9.10- Схема RAID 0

В схеме отсутствует избыточность (не предусматривается хранение дополнительной информации для контроля и исправления ошибок), то есть пользователю предоставляется весь объем дискового массива. Однако выход из строя одного из физических дисков массива приводит к невосстановимой потере хранимых данных.

По этой причине схема RAID 0 применима лишь там, где производительность и емкость дисковой системы намного важнее частичной потери данных, например при хранении фото- и видеоизображений. Из-за отсутствия в RAID 0 средств по защите данных рекомендуется использовать в массиве высоконадежные диски и хранить дубликаты файлов на другом, более надежном носителе информации, например на оптических дисках или магнитной ленте.

Следует отметить, что применение RAID-массивов защищает от потерь данных только в случае физического отказа жестких дисков.

Преимущества:

- наивысшая производительность в приложениях, требующих интенсивной обработки запросов ввода/вывода и данных большого объема;
- простота реализации;
- низкая стоимость;
- максимальная эффективность использования дискового пространства — 100%.

Недостатки:

- не поддерживает отказоустойчивость;
- отказ одного диска влечет за собой потерю всех данных массива.

RAID1:Дисковый массив с зеркалированием (Mirroring & Duplexing)

Дисковый массив с дублированием информации (зеркалированием данных). В простейшем случае два накопителя содержат одинаковую информацию и являются одним логическим диском. При выходе из строя одного диска его функции выполняет другой. Для реализации массива требуется не меньше двух винчестеров.

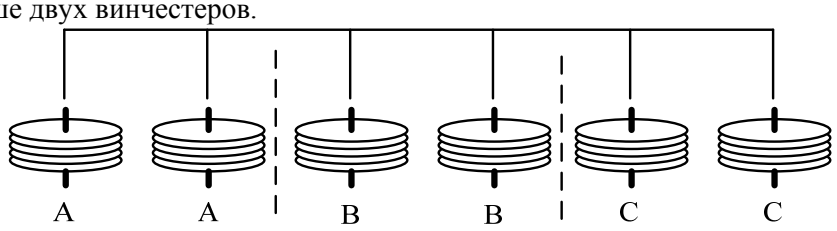


Рисунок 9.11- Схема RAID 1

RAID 1 – простейший отказоустойчивый массив

Преимущества:

- простота реализации;
- простота восстановления массива в случае отказа (копирование).

Недостатки:

- высокая стоимость — 100-процентная избыточность; невысокая скорость передачи данных.

RAID 2:Отказоустойчивый дисковый массив с использованием кода Хемминга (Hamming Code ECC)

Схема резервирования данных с использованием кода Хэмминга (Hamming code) для коррекции ошибок. Поток данных разбивается на слова — причем размер слова соответствует количеству дисков для записи данных. Для каждого слова вычисляется код коррекции ошибок, который записывается на диски, выделенные для хранения контрольной информации. Их число равно количеству бит в слове контрольной суммы.

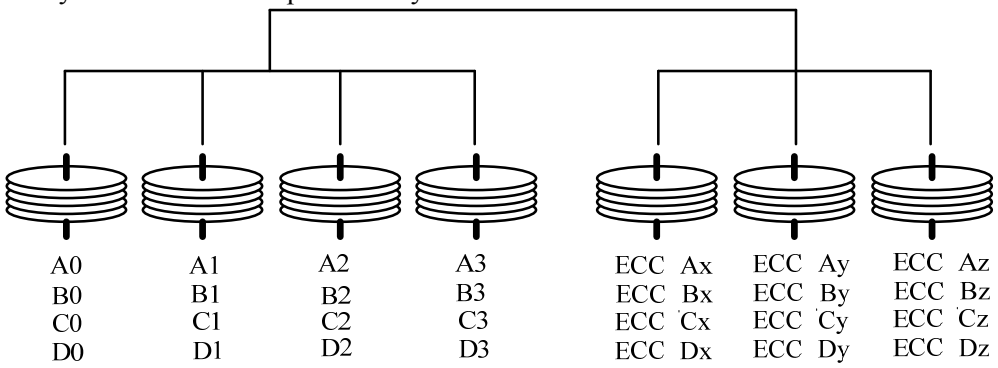


Рисунок 9.12- Схема RAID 2

Если слово состоит из четырех бит, то под контрольную информацию отводится три диска. RAID 2 — один из немногих уровней, позволяющих обнаруживать двойные ошибки и исправлять одиночные. При этом он является самым избыточным среди всех уровней с контролем четности. Эта схема хранения данных не получила коммерческого применения, поскольку плохо справляется с большим количеством запросов.

Преимущества:

- достаточно простая реализация;
- постоянная коррекция одиночных ошибок;
- очень высокая скорость передачи данных;
- при увеличении количества дисков накладные расходы уменьшаются.

Недостатки:

- низкая скорость обработки запросов;
- высокая стоимость;
- большая избыточность.

RAID 3: Отказоустойчивый дисковый массив с параллельной передачей данных и четностью (Parallel Transfer Disks with Parity)

Отказоустойчивый массив с параллельным вводом/выводом данных и диском контроля четности. Поток данных разбивается на порции на уровне байт (хотя возможно и на уровне бит) и записывается одновременно на все диски массива, кроме одного. Один диск предназначен для хранения контрольных сумм, вычисляемых при записи данных. Поломка любого из дисков массива не приведет к потере информации.

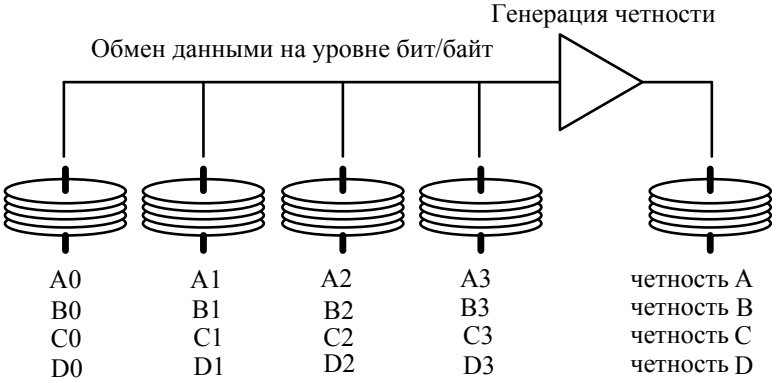


Рисунок 9.13- Схема RAID 3

Этот уровень имеет намного меньшую избыточность, чем RAID 2. Во втором RAID большинство дисков, хранящих контрольную информацию, нужны для определения ошибочного разряда. Как правило, RAID-контроллеры могут получить данные об ошибке с помощью механизмов отслеживания случайных сбоев. За счет разбиения данных на порции RAID 3 имеет высокую производительность. Поскольку при каждой операции ввода/вывода производится обращение практически ко всем дискам массива, то одновременная обработка нескольких запросов невозможна.

RAID 3 подходит для приложений с файлами большого объема и малой частотой обращений (в основном это сфера мультимедиа). Использование только одного диска для хранения контрольной информации объясняет тот факт, что коэффициент использования дискового пространства достаточно высок (как следствие этого — относительно низкая стоимость). Для реализации массива требуется не меньше трех винчестеров.

Преимущества:

- отказ диска мало влияет на скорость работы массива;
- высокая скорость передачи данных;
- высокий коэффициент использования дискового пространства.

Недостатки:

- сложность реализации;
- низкая производительность при большой интенсивности запросов данных небольшого объема.

RAID 4:Отказоустойчивый массив независимых дисков с общим диском четности (Independent Data Disks with Shared Parity Disk)

Этот массив очень похож на уровень RAID 3. Поток данных разделяется не на уровне байтов, а на уровне блоков информации, каждый из которых записывается на отдельный диск. После записи группы блоков вычисляется контрольная сумма, которая записывается на выделенный для этого диск. В RAID 4 возможно одновременное выполнение нескольких операций чтения. Этот массив повышает производительность передачи файлов малого объема (за счет распараллеливания операции считывания).

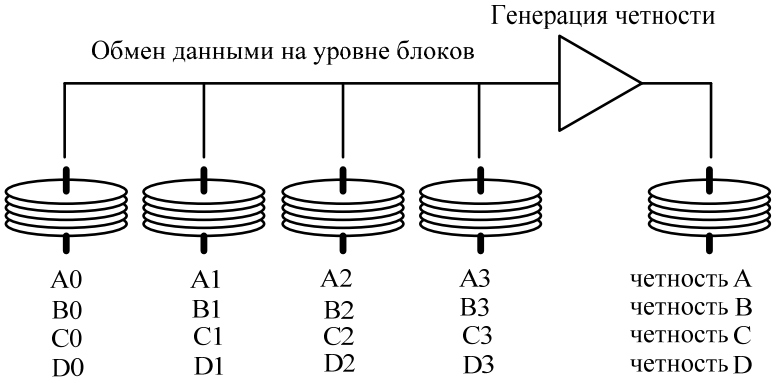


Рисунок 9.14- Схема RAID 4

Но поскольку при записи должна изменяться контрольная сумма на выделенном диске, одновременное выполнение операций невозможно (налицо асимметричность операций ввода и вывода). Этот уровень имеет почти все недостатки RAID 3 и не обеспечивает преимущества в скорости при передаче данных большого объема. Схема хранения разрабатывалась для приложений, в которых данные изначально разбиты на небольшие блоки, поэтому нет необходимости разбивать их дополнительно. Эта схема хранения данных имеет невысокую стоимость, но ее реализация достаточно сложна, как и восстановление данных при сбое.

Преимущества:

- высокая скорость передачи данных;
- отказ диска мало влияет на скорость работы массива;
- высокий коэффициент использования дискового пространства.

Недостатки:

- достаточно сложная реализация;
- очень низкая производительность при записи данных;
- сложное восстановление данных.

RAID 5: Отказоустойчивый массив независимых дисков с распределенной четностью (Independent Data Disks with Distributed Parity Blocks)

Самый распространенный уровень. Блоки данных и контрольные суммы циклически записываются на все диски массива, отсутствует выделенный диск для хранения информации о четности, нет асимметричности конфигурации дисков.

В случае RAID 5 все диски массива имеют одинаковый размер — но один из них невидим для операционной системы. Например, если массив состоит из пяти дисков емкостью 10 Гб каждый, то фактически размер массива будет равен 40 Гб — 10 Гб отводится на контрольные суммы. В общем случае полезная емкость массива из n дисков равна суммарной емкости n–1 диска.

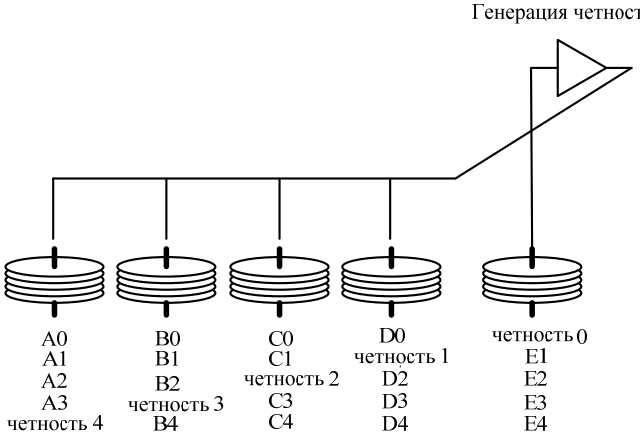


Рисунок 9.15- Схема RAID 5

В RAID 5 отсутствует выделенный диск для хранения информации о четности. Самый большой недостаток уровней RAID от 2-го до 4-го — это наличие отдельного диска (или дисков),

хранящего информацию о четности. Скорость выполнения операций считывания достаточно высока, так как не требует обращения к этому диску. Но при каждой операции записи на нем изменяется информация, поэтому схемы RAID 2-4 не позволяют проводить параллельные операции записи. RAID 5 не имеет этого недостатка, так как контрольные суммы записываются на все диски массива, что делает возможным выполнение нескольких операций чтения или записи одновременно. RAID 5 имеет достаточно высокую скорость записи/чтения и малую избыточность.

Преимущества:

- высокая скорость записи данных;
- достаточно высокая скорость чтения данных;
- высокая производительность при большой интенсивности запросов чтения/записи данных;
- высокий коэффициент использования дискового пространства.

Недостатки:

- низкая скорость чтения/записи данных малого объема при единичных запросах;
- достаточно сложная реализация;
- сложное восстановление данных.

RAID 6: Отказоустойчивый массив независимых дисков с двумя независимыми распределенными схемами четности (Independent Data Disks with Two Independent Distributed Parity Schemes)

RAID 6 — это отказоустойчивый массив независимых дисков с распределением контрольных сумм, вычисленных двумя независимыми способами. Этот уровень во многом похож на RAID 5. Только в нем используется не одна, а две независимые схемы контроля четности, что позволяет сохранять работоспособность системы при одновременном выходе из строя двух накопителей. Для вычисления контрольных сумм в RAID 6 используется алгоритм, построенный на основе кода Рида-Соломона (Reed-Solomon).

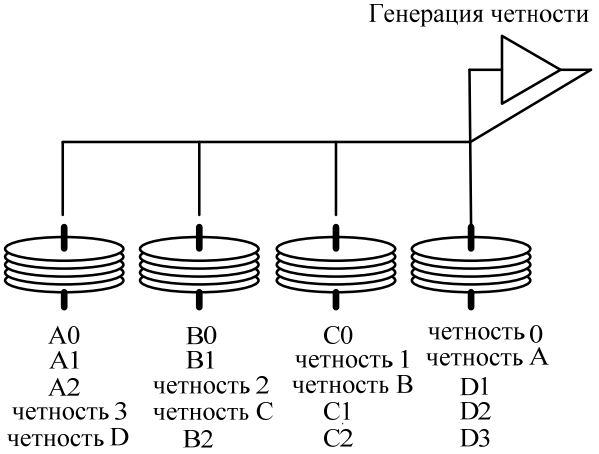


Рисунок 9.16- Схема RAID 6

RAID 6 использует две независимые схемы контроля четности. Этот уровень имеет очень высокую отказоустойчивость, большую скорость считывания (данные хранятся блоками, нет выделенных дисков для хранения контрольных сумм). В то же время из-за большого объема контрольной информации RAID 6 имеет низкую скорость записи. Он очень сложен в реализации, характеризуется низким коэффициентом использования дискового пространства: для массива из пяти дисков он составляет всего 60%, но с ростом числа дисков ситуация исправляется.

RAID 6 по многим характеристикам проигрывает другим уровням, поэтому на сегодня не получил коммерческого применения.

Преимущества:

- высокая отказоустойчивость;
- достаточно высокая скорость обработки запросов;

Недостатки:

- низкая скорость чтения/записи данных малого объема при единичных запросах;
- очень сложная реализация;
- сложное восстановление данных;
- низкая скорость записи данных.

RAID 7: Отказоустойчивый массив, оптимизированный для повышения производительности (Optimized Asynchrony for High I/O Rates as well as High Data Transfer Rates)

В отличие от других уровней, RAID 7 не является открытым индустриальным стандартом — это зарегистрированная торговая марка компании Storage Computer Corporation. Массив основывается на концепциях, использованных в третьем и четвертом уровнях. Добавилась возможность кэширования данных. В состав RAID 7 входит контроллер со встроенным микропроцессором под управлением операционной системы реального времени (real-time OS). Она позволяет обрабатывать все запросы на передачу данных асинхронно и независимо.

RAID 7 – зарегистрированная торговая марка компании Storage Computer Corporation. Блок вычисления контрольных сумм интегрирован с блоком буферизации; для хранения информации о четности используется отдельный диск, который может быть размещен на любом канале. RAID 7 имеет высокую скорость передачи данных и обработки запросов, хорошую масштабируемость. Самым большим недостатком этого уровня является стоимость его реализации.

Преимущества:

- очень высокая скорость передачи данных и высокая скорость обработки запросов (в 1,5…6 раз выше других стандартных уровней RAID);
- хорошая масштабируемость;
- значительно возросшая (благодаря наличию кэша) скорость чтения данных небольшого объема;
- отсутствие необходимости в дополнительной передаче данных для вычисления четности.

Недостатки:

- собственность одной компании;
- сложность реализации;
- очень высокая стоимость на единицу объема;
- не может обслуживаться пользователем;
- необходимость использования блока бесперебойного питания для предотвращения потери данных из кэш-памяти.

При программной реализации используются обычные дисковые контроллеры и стандартные команды ввода/вывода. Работа дисковых ВЗУ в соответствии с алгоритмами различных уровней RAID обеспечивается программами операционной системы ВМ. Программный режим RAID предусмотрен, например, начиная с Windows NT.

Это дает возможность программного изменения уровня RAID, в зависимости от особенностей решаемой задачи. Хотя программный способ является наиболее дешевым, он не позволяет добиться высокого уровня производительности, характерного для реализации RAID. Аппаратурная реализация RAID предполагает возложение всех или большей части функций по управлению массивом дисковых ЗУ на соответствующее оборудование, при этом возможны два подхода. Первый из них заключается в замене стандартных контроллеров дисковых ЗУ на специализированные, вместо стандартных. Базовая ЭВМ общается с контроллерами на уровне обычных команд ввода/вывода, а режим RAID обеспечивают контроллеры. При втором способе аппаратной реализации RAID - система выполняется как автономное устройство объединяющее в одном корпусе массив дисков и контроллер. Контроллер содержит микропроцессор и работает под управлением собственной операционной системы полностью реализующей различные RAID - режимы. Такая подсистема подключается к шине базовой ВМ или к ее каналу ввода/вывода как обычное дисковоё ЗУ При аппаратной реализации RAID - систем обычно предусматривается возможность замены неисправных дисков без потери информации и без остановки работы. Кроме того, многие из таких систем позволяют разбивать отдельные диски на разделы, причем разные разделы дисков могут объединяться в соответствии с разными уровнями RAID.

9.4 Запоминающие устройства на основе оптических дисков

Внешние запоминающие устройства на базе оптических дисков (ВЗУ на ОД) — это сравнительно «молодое» направление в области внешних запоминающих устройств. Накопители на оптических дисках появились в 80-х годах XX века и первоначально предназначались для звуковой звукозаписи, однако быстро стали широко использоваться в качестве устройств внешней памяти. Технология ОД с самого момента появления постоянно совершенствовалась, в результате чего к настоящему времени распространение получили несколько типов таких устройств: CD (Compact Disks — компакт- диски), DVD (Digital Versatile Disks — цифровые

универсальные диски) и пока еще в меньшей степени — BD (Blue-ray Disks — диски с голубым лучом). Каждый следующий тип в этом списке можно рассматривать как результат эволюции предшествующих типов, поскольку, несмотря на отличия, в основе всех упомянутых типов лежат некоторые общие принципы.

9.4.1 Общие принципы построения ВЗУ на ОД

В отличие от жестких магнитных дисков, где магнитные диски и привод представляют собой единый блок, ВЗУ на ОД включает два компонента: съемный оптический диск и привод, обеспечивающий вращение диска. На приводе также размещаются записывающий и считывающий лазеры и соответствующие электронные схемы. Носителем информации в ВЗУ на ОД служат круглые пластмассовые диски, с отражающим слоем с диаметром 12 или 8 см. Как чтение, так и запись (если диск предусматривает такую возможность) информации осуществляется лазерным лучом, направляемым на поверхность вращающегося диска. Для записи обычно используется более мощный лазер. Структурная схема CD-ROM представлена на рис.9.16.

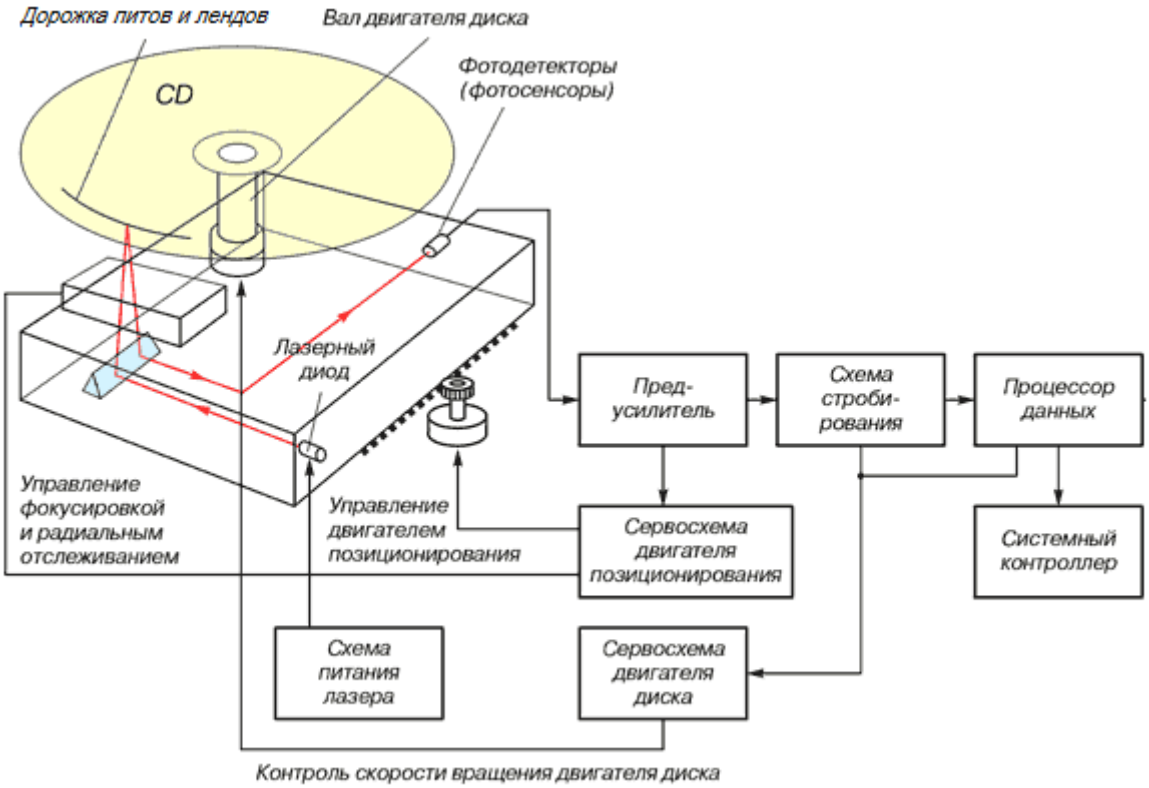


Рисунок 9.16- Структурная схема CD-ROM

В исходном состоянии вся поверхность диска обладает «идеальными» отражающими свойствами. В процессе записи информации определенные участки отражающего слоя диска, расположенные вдоль траектории перемещения луча записывающего лазера, нагреваются. В зависимости от типа отражающего слоя в облученном месте возникают углубления, либо материал отражающего слоя переходит в иное состояние. В обоих случаях изменяются показатели отражения участка отражающего слоя. Для обозначения углублений или участков с измененным состоянием применяют термин *«пит»*. Промежутки между питами, для которых сохраняется исходное «идеальное» отражение, называют *лендами*. Ленды отражают большую часть падающего на них света лазерного излучателя, а питы в силу своей удаленности от точки фокуса не отражают практически ничего.

Траектория перемещения луча лазера по поверхности вращающегося диска имеет вид единой спиральной дорожки, начинающейся от центра диска и идущей к его внешней границе. Питы формируются вдоль этой дорожки.

При чтении отраженный луч считывающего лазера фиксируется приемником, при этом в зависимости от того, является ли облучаемый участок питом или лендом, характеристики

отраженного луча (интенсивность или фаза) будут различными. Условно можно считать, что луч, отраженный от ленда, воспринимается приемником, а отраженный от пита в приемник не попадает. Такое различие можно трактовать как 1 и 0.

На самом деле питы и ленды не являются битами 0 или 1 в привычном понимании, а представляют сразу несколько битов информации. Связано это с тем, что для надежности хранения информацию перед записью подвергают помехоустойчивому кодированию. При таком кодировании каждый байт информации с помощью специальной таблицы заменяется 14-разрядным словом. К этому слову добавляются три так называемых соединительных бита. В полученном 17-разрядном коде между двумя единицами никогда не может быть менее двух и более десяти нулей. Именно такие последовательности и представлены питами и лендами, причем питы и ленды чередуются в моменты, соответствующие началу бита, равного 1. С позиций представления информации питы и ленды равнозначны и характеризуют лишь временные интервалы между двумя единицами в последовательности битов. Сенсор тестирует поверхность через равные временные интервалы. Так как временные интервалы отсчитываются в количестве периодов тактовой частоты, то протяженность каждого пита или ленда лежит в пределах от 3 (интервал 3T) до 11 (интервал 11T) периодов тактовой частоты (учитывается и бит, равный 1). Переход от пита к ленду (или наоборот) соответствует логической 1, а логический 0 представляется отсутствием переходов в данном месте (рис. 9.17).

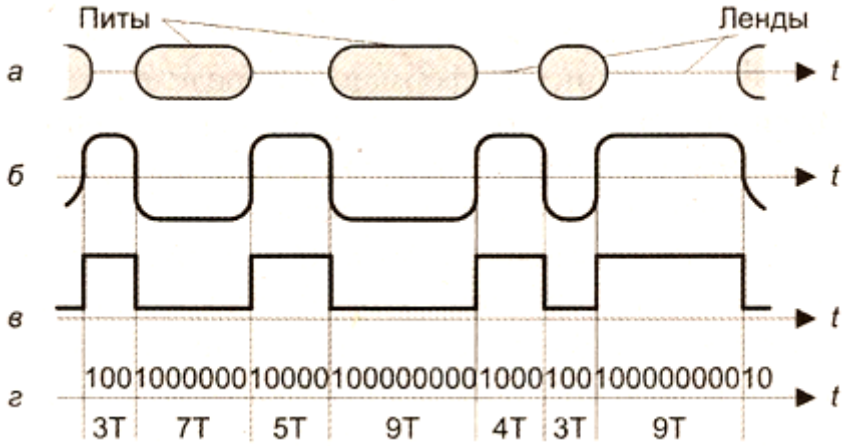


Рисунок 9.17 - Представление информации на оптическом диске: а — питы и ленды (вид сверху); б — информационный сигнал с фотоприемника; в — преобразованный информационный сигнал; г — интервалы

Информационный сигнал с приемника (рис. 9.17, б) преобразуется в последовательность прямоугольных импульсов, длительность которых кратна периоду следования битов последовательного кода (рис. 9.17, е). Каждые 14 последовательных периодов декодируются в один байт информации, соединительные разряды просто отбрасываются.

Таким образом, информация на оптическом диске представлена цепочкой питов и лендов, расположенной вдоль спиралевидной дорожки. Одним из кардинальных отличий между разными типами ОД служит длина волны используемых лазеров. Чем она меньше, тем меньше размер сфокусированного луча лазера, а значит, размер питов и лендов, ширина дорожки и расстояние между витками спирали. Как следствие, возрастает количество информации, которую можно разместить на диске. На рис. 9.18 приведены увеличенные фотографии поверхности дисков типа CD и DVD.

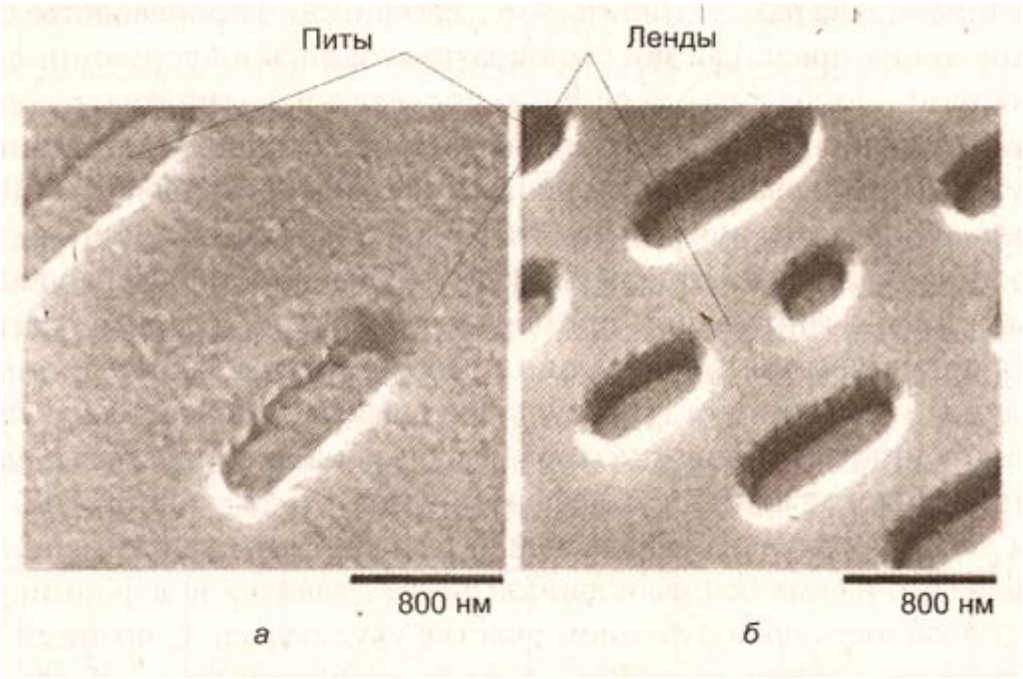


Рисунок 9.18 - Размещение информации на поверхности дисков типа: а — CD; б — DVD

9.4.2 Способы записи информации на оптические диски

В рамках каждого из упоминавшихся типов ВЗУ на ОД существуют оптические диски, различающиеся способом занесения на них информации и возможностями ее смены. Обычно выделяют три группы дисков, которые можно отличить по аббревиатуре, отделяемой от названия типа ОД символом «минус» или «плюс», например CD-ROM, DVD-R или DVD+RW.

Первую группу образуют *прессованные* диски, информация в которые заносится в процессе изготовления. В их названии используется добавка ROM по аналогии с масочными постоянными запоминающими устройствами. Сначала изготавливается так называемый мастер-диск, запись информации на который производится с помощью сильно сфокусированного луча лазера высокой интенсивности или механически. На основе мастер-диска создается матрица для штамповки копий, которые и являются оптическими дисками. Копия представляет собой диск из пластмассы, например поликарбоната. Цифровая информация в виде углублений на подложке переносится на диск с матрицы путем штамповки. Поверхность копий с углублениями покрывается алюминием или золотом, то есть материалом с высокой отражающей способностью. Далее углубления на копии защищаются от пыли и повреждений путем покрытия поверхности диска прозрачным лаком. Рабочая поверхность таких дисков обычно белого цвета.

Вторая группа оптических дисков — это диски *с однократной записью.* Они обозначаются добавкой R (от Recordable — записываемые), например CD-R (CD+R) или DVD-R (DVD+R). Технология дисков с однократной записью и многократным считыванием была разработана для мелкосерийного производства оптических дисков. Такие диски предполагают однократную запись информации лучом относительно мощного лазера с возможностью последующего многократного считывания. В структуре CD-R диска можно выделить четыре основных слоя (пятый - изображение, нанесенное на поверхность диска), наносимых поэтапно (рис.9.19).

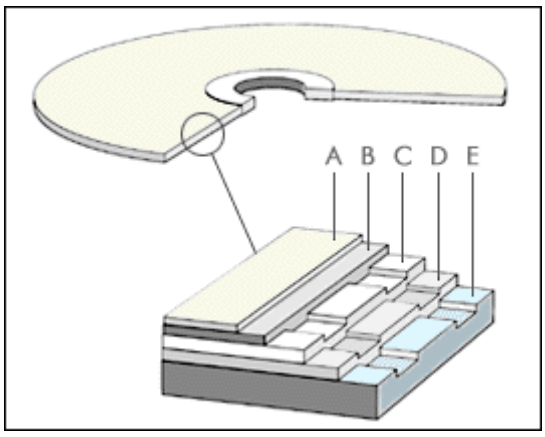


Рисунок 9.19- Структура CD-R диска

Изначально изготавливается пластмассовая основа диска - поликарбонат (E), которая составляет основную часть CD-R и придает ему необходимую прочность и форму. Далее, на готовую пластмассовую форму наносится активный слой (D) /dye/. Именно этот слой позволяет осуществлять запись на диск и определяет его надежность и качество считывания информации в дальнейшем. На сегодняшний день широко используется два типа активного слоя: цианин и фталоцианин.

В исходном состоянии на его поверхности уже сформирована направляющая спиральная канавка, вдоль которой в дальнейшем перемещается луч лазера. Эта канавка заполнена органическим красителем. В процессе записи на отдельных участках мощность лазера увеличивается относительно уровня считывании примерно на порядок. Энергия лазерного луча поглощается органическим красителем и преобразуется в тепло. Иногда этот процесс называется «прожигание». В результате нагрева краситель обугливается и в нем появляются микроскопические газовые пузырьки. В процессе выделения газов увеличивается объем красителя и деформируется отражающий слой. Краситель нагревается до температуры, превышающей температуру плавления поликарбоната, вследствие чего и сама основа в данной точке плавится и деформируется, а отражающая способность на облученном участке ухудшается. С позиций отражения такой участок эквивалентен питу. Роль лендов выполняет не вся поверхность диска, а лишь недеформированные участки канавки. В зависимости от применяемого красителя диски имеют цвета зеленоватых или синих оттенков. Данный тип носителя привлекателен для архивного хранения документов и файлов.

Третью группу образуют *перезаписываемые* диски. Их обозначают добавкой RW (Rewritable), например CD-RW (CD+RW) или DVD-RW (DVD+RW). Информация в оптических дисках данного типа может быть многократно перезаписана, как это имеет место с магнитными дисками. В дисках используется эффект *фазового перехода,* для чего используется материал, который может быть в одном из двух фазовых состояний, причем отражающая способность его в этих состояниях существенно отличается. Одно из этих состояний — аморфное, когда молекулы материала имеют случайную ориентацию и материал отражает свет плохо. Второе состояние — кристаллическое, характеризующееся высокой отражающей способностью. Под воздействием луча лазера активный слой оптического диска может менять свое состояние с кристаллического на аморфное, и наоборот. До записи поверхность диска находится в кристаллическом состоянии. Для перевода участка активного слоя в аморфное состояние (записи пита) он облучается коротким лазерным импульсом высокой мощности. Импульс нагревает участок до температуры T, превышающей температуру плавления $T_{плав}(T > T_{плав})$, и расплавляет материал активного слоя в этом месте. Затем следует охлаждение ниже температуры кристаллизации $T_{крист}$, при этом центры кристаллизации не образуются, и вещество остается в аморфном состоянии. Для стирания информации надо вернуть вещество в кристаллическое состояние. Это достигается опять же с помощью лазера, но работающего в другом режиме. Аморфное вещество нагревают до температуры, меньшей, чем $T_{плав}$ но большей, чем $T_{крист}$ $(T_{крист} < T < T_{плав})$. Нагрев (его называют обжигом) продолжается в течение времени, достаточного для восстановления кристаллического состояния вещества.

Главный недостаток оптических дисков с фазовым переходом заключается в том, что материал со временем теряет желательные свойства. Современные материалы могут быть

использованы для 500 000-1 000 000 циклов стирания. Возможность перезаписи позволяет использовать диски типа RW в качестве вторичной памяти как дополнение к магнитным дискам.

9.4.3 Оптические диски типа CD

Компакт-диск (CD — compact disk) — это односторонний диск, способный хранить двоичную информацию. Выпускаются CD различной емкости. В типовом варианте расстояние между витками спиралевидной дорожки составляет 1,6 микрон, что позволяет обеспечить 20 344 витка, при этом длина спиралевидной дорожки равна 5,27 км. Диск вращается с постоянной линейной скоростью 1,2 м/с, то есть для «прохождения» спирали требуется 4391 с или 73,2 мин. Так как данные считываются с диска со скоростью 176,4 Кбайт/с, емкость CD равна 774,57 Мбайт.

Данные на CD-ROM организованы как последовательность блоков. Типичный формат блока показан на рис. 6.57. Блок включает в себя следующие поля:

- *Синхронизация.* Это поле идентифицирует начало блока и состоит из байта со всеми нулями, десяти байтов, содержащих только единичные разряды, и вновь байта из всех нулей.

- *Идентификатор.* Заголовок, содержащий адрес блока и байт режима. Режим 0 определяет пустое поле данных; режим 1 отвечает за использование кода, корректирующего ошибки, и наличие 2048 байтов данных; режим 2 определяет наличие 2336 байтов данных и отсутствие корректирующего кода.

Рисунок 9.20-Формат блока CD-ROM

- *Данные.* Данные пользователя.

- *Корректирующий код (КК).* Поле предназначено для хранения дополнительных данных пользователя в режиме 2, а в режиме 1 содержит 288 байтов кода с исправлением ошибок. Рисунок 9.20 иллюстрирует организацию информации на CD-ROM.

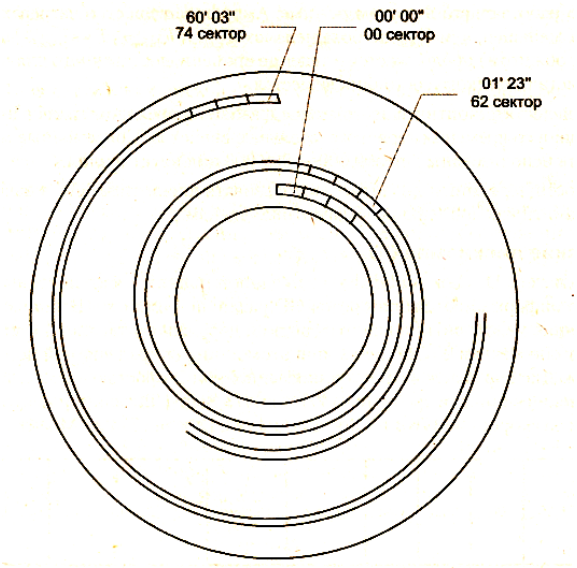


Рисунок 9.20- Организация CD диска с постоянной линейной скоростью

Ввиду вращения диска с постоянной линейной скоростью (ПЛС), а не с постоянной угловой скоростью (как это имеет место в случае магнитных дисков) произвольный доступ к информации

становится более сложным. Позиционирование по указанному адресу включает перемещение головки к общей области, подбирая скорость вращения, и чтение адреса, а затем небольшую регулировку для нахождения и доступа к нужному сектору.

Для обеспечения более быстрого доступа в современных ВЗУ на ОД на базе CD поддерживается метод постоянной угловой скорости при относительном снижении емкости (до 682 Мбайт).

9.4.4 Оптические диски типа DVD

Последние годы характеризуются повсеместным переходом от CD к DVD-дискам (Digital Versatile Disk — цифровой универсальный диск). По сравнению с CD в DVD существенно увеличена емкость диска, что достигается благодаря трем моментам:

1. Биты на DVD упакованы более плотно. Расстояние между витками спирали на CD равно 1,6 мкм, и минимальное расстояние между питами составляет 0,834 мкм. В DVD-технологии используется лазер с меньшей длиной волны (650 нм против 780 нм для стандартных CD), что позволяет сократить расстояние между витками до 0,74 мкм, а между питами — до 0,4 мкм. Результатом этих двух улучшений становится почти семикратное увеличение емкости, то есть вместо 682 Мбайт до 4,7 Гбайт.

2. В DVD реализован второй слой питов и лендов поверх первого слоя. В двухслойном DVD поверх отражающего слоя расположен полуотражающий слой, и путем изменения фокусировки накопители DVD могут читать каждый слой по отдельности. Эта техника почти удваивает емкость диска, увеличивая ее до величины порядка 8,5 Гбайт. Более низкие отражающие свойства второго слоя ограничивают емкость диска, поэтому полного удвоения не происходит. Вместимость можно определить на глаз - нужно посмотреть, сколько рабочих (отражающих) сторон у диска, и обратить внимание на их цвет: двухслойные стороны обычно имеют золотой цвет, а однослойные — серебряный, как у компакт-диска.

3. DVD-ROM могут быть двусторонними, в то время как на CD данные записываются только на одной стороне. При использовании двустороннего DVD его нужно переворачивать.

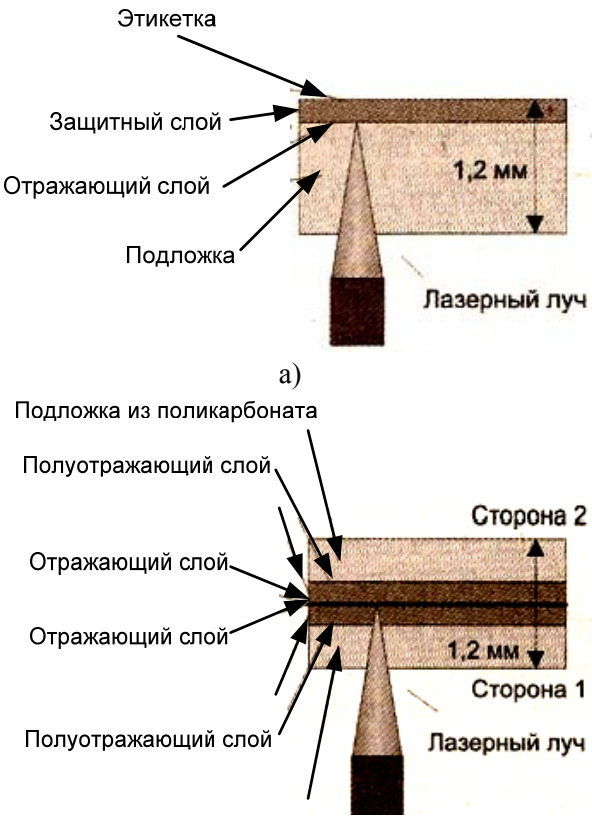


Рисунок 9.21-Структура оптического диска: CD емкостью 682 Мбайт (а); двустороннего двухслойного DVD (б)

9.4.5 Оптические диски типа Blue-ray

Blue-ray (голубой луч) диски или сокращенно BD представляют собой очередное поколение оптических дисков. В технологии Blue-ray для чтения и записи используется сине-фиолетовый лазер с длиной волны 405 нм. Напомним, что обычные DVD и CD используют красный и инфракрасный лазеры с длиной волны 650 нм и 780 нм соответственно. Такое уменьшение позволило сузить дорожку вдвое (до 0,32 микрон). В сочетании с другими изменениями (высококачественной системой фокусировки луча с двумя линзами и уменьшением толщины защитного слоя на носителе) это позволило записывать информацию в меньшие точки на диске, а значит, при сохранении стандартных размеров диска хранить на нем больше информации, а также увеличить скорость считывания до 430-435 Мбит/с.

В настоящее время выпускаются BD-диски с диаметром 12 и 8 см, причем оба могут быть однослойными и двухслойными. Емкость 12-сантиметровых дисков может достигать 27 Гбайт (в однослойном варианте) или 54 Гбайт (в двухслойном варианте). Аналогичные показатели для 8-сантиметровых дисков — 7,8 Гбайт и 15,6 Гбайт соответственно.

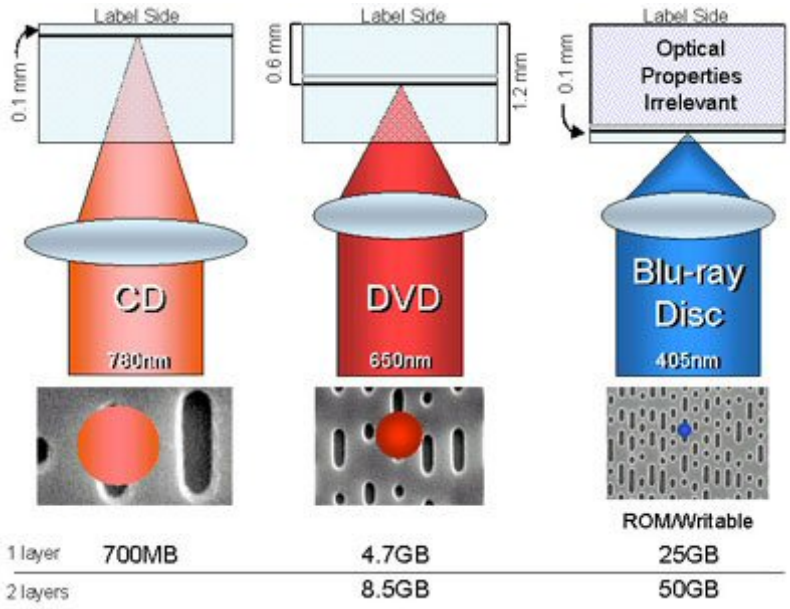


Рисунок 9.22- Сравнение технологий CD, DVD, Blue-ray

В разработке находятся диски вместимостью 100 Гбайт и 200 Гбайт с использованием соответственно четырех и шести слоев.

9.4.6 Перспективные типы оптических дисков

Среди перспективных типов оптических дисков, необходимо отметить ВЗУ на ОД типа HVD. HVD (Holographic Versatile Disk — голографический универсальный диск) — новая технология хранения данных на оптических носителях, которая значительно увеличит объем информации, записываемой на один диск. Она основывается на явлении, называемом коллинеарная голография. В работе используются два лазера — красного и сине-зеленого цветов. Сине-зеленый лазер считывает информацию, закодированную в голографическом слое, а красный лазер — читает информацию для правильного позиционирования механизма из обычного алюминиевого слоя в самой глубине диска. На CD и DVD эта информация записывается вместе с основными данными. В HVD же есть особый пограничный слой, отражающий сине-зеленый лазер, но пропускающий красный. Эта особенность позволяет разделять два потока информации (служебную и сами данные), что было недостижимо в более ранних вариантах подобных устройств. Новые носители могут хранить до 3,9 Тбайт данных, что примерно в 160 раз превышает емкость обычного BD-диска. Планируемая скорость передачи данных — 1 Гбит/с. В 2009 году австралийские ученые представили новый тип ЗУОД, в котором оптический диск способен хранить до 1,6 терабайт информации. Информация записывается в несколько уровней, и если плотность записи на DVD равна 51 мегабайт на квадратный сантиметр, то здесь плотность составляет 1,1 терабита на кубический сантиметр. В основе технологии лежит эффект оптической поляризации светового излучения. Данные записываются в несколько слоев, отделенных друг от друга тончайшей прослойкой на основе молекул золота. У каждого слоя своя способность к светоотражению, каждый слой способен воспринимать только свет определенной длины волны. Когда луч лазера заданной длины достигает искомого слоя, то происходит эффект поляризации и слой начинает предоставлять данные, записанные на нем. Это позволяет хранить множество битов на одном и том же месте. Сейчас созданы считывающие системы, способные работать с 6-9 слоями. В перспективе предполагается довести количество слоев до 10. Появление подобных ЗУОД на рынке ожидается к 2015-2017 году, поскольку необходимо разрешить несколько проблем, в частности проблему разогрева дисков, так как лазерные лучи, считывающие данные, сравнительно мощны. Кроме того, предстоит доработать считывание таким образом, чтобы слои ни в коем случае не читались одновременно.

9.5 Магнитооптические накопители

Магнитооптические накопители (МОН) представляет собой накопитель информации, в основу которого положен магнитный носитель с оптическим (лазерным) управлением.

Поверхность МОН покрыта слоем, свойства которого меняются как под действием тепла, так и под действием магнитных полей. Если нагреть диск сверх некоторой температуры, то

становится возможным изменение магнитной поляризации посредством небольшого магнитного поля. На этом основаны принципы чтение и запись.

Установленный в устройство диск подвергается воздействию магнитного поля с одной поверхности и лазерного луча — с противоположной (рис. 9.22). Диски покрыты слоем специального сплава, который обладает свойством отражать излучения лазера под различающимися углами в зависимости от направления намагниченности, и данные могут записываться как «северные» и «южные» магнитные полюса, как и в случае жесткого диска.

В то время как жесткий диск может перемагничиваться при любой температуре, магнитное покрытие, используемое на МО-носителях, чрезвычайно устойчиво к намагничиванию при комнатной температуре, сохраняя данные неизменными, пока записывающий слой не будет нагрет выше уровня температуры, называемого точкой Кюри (около 200 °С). Магнитооптические накопители используют лазер для нагревания определенных областей магнитных частиц. После разогрева магнитных частиц направление их магнитных полей может быть легко изменено полем магнитной головки.

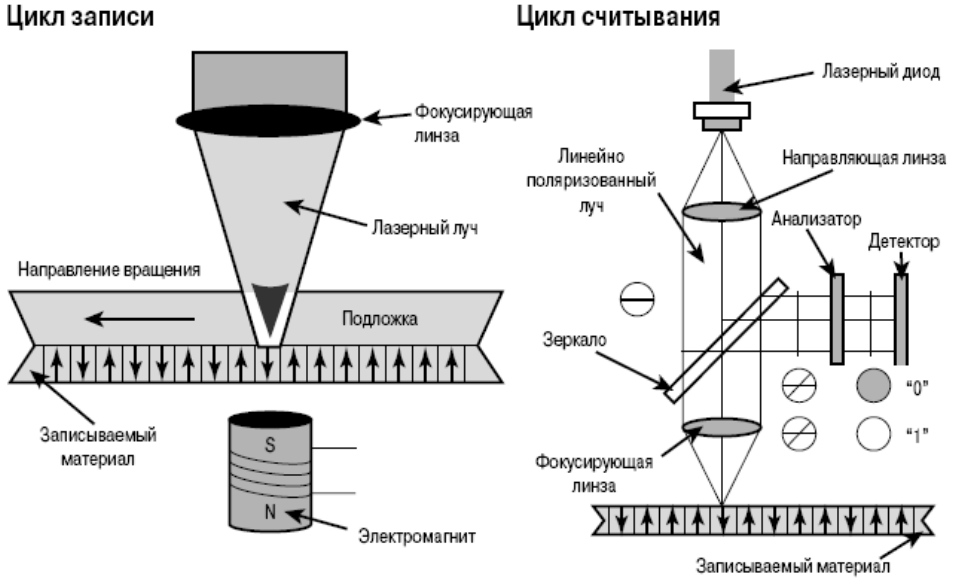


Рисунок 9.22 –Магнитооптическая технология

При считывании информации используется менее мощный лазер и эффект Керра, состоящий в том, что поляризация отраженного света меняется в зависимости от ориентации магнитного поля. В тех точках, где поверхность не была подвергнута лазерно-магнитному воздействию, участок представляется «0», а там, где точка была нагрета и намагничена, будет считана «1».

Таким образом, слой диска подвергается воздействию 2-х энергетических факторов: магнитного поля и лазерного луча. Взаимодействие магнитного поля не меняется во времени, а интенсивность лазерного луча модулируется сигналом данных, записываемых на диск. Возможен и обратный процесс, но при этом будет резко увеличено время переключения, и выполнить это будет технически сложно.

МО диски бывают четырех размеров:

- 5,25" (двусторонние) емкостью 650 Мбайт, 1,3, 2,6, 4,6 Гбайт;
- 3,5" (односторонние) емкостью 128, 230, 540, 640 Мбайт и 1,3 Гбайт;
- минидиски 2,5 дюйма емкостью 140 Мбайт;
- 12-дюймовые диски – используются для односторонней и однократной записи. Емкость для односторонних – 3,5 Гбайт, при двойной плотности записи – до 7 Гбайт.

Длина волны лазера 680 нм. Среднее время доступа от 30 до 60 мкс.

Существует ряд градаций объемов МОД, обозначаемых через кратность объему начальной модели диска. Диски различаются плотностью размещения треков; методами модуляции – PWM (Pulse

Widht Modulation - широтно-импульсная модуляция) для 1x и 2x, PPM (Pulse Position Modulation - позиционно-импульсная модуляция) для 4x; методами кодирования - RLL 2.7 для 1x и 2x, RLL 1.7 для 4x и 5x; а также организацией зонного формата записи. Устройства и диски имеют обратную совместимость: устройства большей емкости могут работать и с дисками меньшей емкости. Устройства 4x (5x) обычно могут полноценно работать с дисками 2x и читать диски 1x. Емкость магнитооптических дисков представлена в таблице 9.1.

Таблица 9.1 – Емкость магнитооптических дисков

Плотность	Диск 130 мм (5")	Диск 90 мм (3,5")
1x	560/650 Мбайт	128 Мбайт
2x	1,2/1,3 Гбайт	230 Мбайт
4x	2,3/2,6 Гбайт	-
5x	-	540/640 Мбайт
10x	4,6 Гбайт	1,3 Гбайт

Данный тип дисков в настоящее время практически не используется, но авторы приводят данный подраздел в качестве ознакомительного, с заслуживающей внимание комбинированной технологией магнитной и оптической одновременно.

9.6 Запоминающие устройства на основе твердотельных дисков

Последние достижения в области полупроводниковых микросхем памяти сделали такие микросхемы привлекательными в плане создания на их базе альтернат жестким магнитным дискам. Подобные «заменители» магнитных дисков получили название «твердотельных дисков» (SSD — Solid State Disk). Запоминающую среду в SSD образуют микросхемы энергозависимой (DRAM SSD) либо энергонезависимой (FLASH SSD) памяти. Твердотельные диски полностью взаимозаменяемы со стандартными накопителями на жестких магнитных дисках как по конструктивному исполнению, так и по интерфейсу.

DRAM SSD и FLASH SSD

Твердотельные диски на базе энергозависимой динамической памяти (DRAM SSD) характеризуются сверхбыстрыми чтением, записью и поиском информации.

Рисунок 9.23 – Пример твердотельного диска

В то же время DRAM — это энергозависимая память, поэтому DRAM SSD, как правило оснащены аккумуляторами для сохранения данных при потере питания, а более дорогие модели — системами резервного и/или оперативного копирования. Система питания утяжеляет накопитель и делает его менее надежным. В настоящее время DRAM SSD используются для ускорения работы крупных систем управления базами данных и мощных графических станций. Накопители, построенные на основе энергонезависимой NAND флэш-памяти (FLASH SSD) появились относительно недавно. Пока стоимость FLASH SSD 3-10$/ ГБ) несколько выше, чем у магнитных дисков, однако с каждым годом она уменьшается на 20-30%. Будучи сопоставимыми по скорости чтения и записи (порядка 70 Мбит/с), SSD-устройства значительно быстрее при поиске информации. Так, время у них составляет около 0,5 мс против 11 мс у жестких магнитных дисков. По этой причине FLASH SSD предпочтительнее при работе с приложениями, для которых характерно частое чтение и запись небольших, случайно расположенных данных.

FLASH SSD характеризуются относительно небольшими размерами и низким потреблением. Они практически полностью завоевали рынок ускорителей баз данных среднего уровня и начинают теснить традиционные диски в мобильных приложениях.

Преимущества и недостатки по сравнению с жесткими дисками.

По сравнению с жесткими магнитными дисками SSD обладает рядом преимуществ, большинство из которых являются следствием того, что SSD не содержит движущихся частей. Три основных преимущества:

- меньшая потребляемая мощность;
- быстрый доступ к данным;
- высокая надежность.

Первое из них связано с отсутствием основного потребителя энергии — двигателя, вращающего диск. В среднем потребляемая мощность сокращается на порядок, быстрый доступ связан с исключением затрат времени на перемещение головки считывания/записи. Наконец, отсутствие механически перемещаемых элементов, склонных к поломкам или способных стать причиной повреждений других в ВМ, положительно сказывается на надежности устройства в целом.

К упомянутым преимуществам SSD можно добавить:

- полное отсутствие шума от движущихся частей и охлаждающих вентиляторов;
- высокую механическую стойкость;
- широкий диапазон рабочих температур;
- практически устойчивое время считывания файлов вне зависимости от их расположения или фрагментации;
- малый размер и вес;
- совместимость со стандартными интерфейсами, используемыми для подключения дисковых ЗУ - SATA-1, SATA-2.

Основными недостатками (на момент написания раздела) можно считать:

- высокую стоимость (от 8 $/Гбайт);
- относительно малую емкость (в продаже доступны Flash SSD до 250 Гбайт);
-более высокую чувствительность к внезапной потере питания, магнитным и электрическим полям;
- ограниченное количество циклов перезаписи (несколько миллионов раз).

9.7 Дисковая КЭШ-память

Концепция кэш-памяти применима и к дисковым ЗУ. Принцип кэширования дисков во многом схож с принципом кэширования основной памяти, хотя способы доступа к диску и ОП существенно разнятся. Если время обращения к любой ячейке ОП одинаково, то для диска оно зависит от целого ряда факторов. Во-первых, нужно затратить некоторое время для установки головки считывания/записи на нужную дорожку. Во-вторых, поскольку при движении головка вибрирует, необходимо подождать, чтобы она успокоилась. В-третьих, искомый сектор может оказаться под головкой также лишь спустя некоторое время.

Дисковая кэш-память представляет собой память с произвольным доступом, «размещенную» между дисками и ОП. Емкость такой памяти обычно достаточно велика — от 8 Мбайт и более. Пересылка информации между дисками и основной памятью организуется контроллером дисковой КЭШ-памяти. Изготавливается дисковая кэш-память на базе таких же полупроводниковых запоминающих устройств, что и основная память, поэтому в ряде случаев с ней обращаются как с дополнительной основной памятью.

В дисковой кэш-памяти хранятся блоки информации, которые с большой вероятностью будут востребованы в ближайшем будущем. Принцип локальности, обеспечивающий эффективность обычной КЭШ-памяти, справедлив и для дисковой, приводя к сокращению времени ввода/вывода данных от величин 20-30 мс до значений порядка 2-5 мс, в зависимости объема передаваемой информации. В качестве единицы пересылки может выступать сектор, несколько секторов, а также одна или несколько дорожек диска. Кроме того, иногда применяется пересылка информации, начиная с выбранного сектора на дорожке и до ее конца. В случае пересылки секторов кэш-память заполняется не только требуемым сектором, но секторами, непосредственно следующими за ним, так как известно, что в большинстве случаев взаимосвязанные данные хранятся в соседних секторах. Этот метол известен также как опережающее чтение (read ahead).

Для дисковой кэш-памяти наиболее характерно полностью ассоциативное отображение, замещение информации по алгоритму LRU и согласование информации методом сквозной записи.

Специфика дисковой кэш-памяти состоит в том, что далеко не всю информацию, перемещаемую между дисками и основной памятью, выгодно помещать в дисковый КЭШ. В ряде случаев определенные данные и команды целесообразно пересылать напрямую между ОП и диском. По этой причине в системах с дисковой КЭШ-памятью предусматривают специальный динамический механизм, позволяющий переключать тракт пересылки информации: через кэш или минуя его.

9.7.1 Стратегия управления дискового КЭШ

Дисковый КЭШ может быть аппаратным и программным.

Аппаратный дисковый КЭШ выполняется в виде блока памяти контроллера ВЗУ. Функции управления памяти такого КЭШ реализуются с помощью компьютера через контроллер.

Программный дисковый КЭШ использует программу, которая использует часть ОП для имитации аппаратного дискового КЭШ.

Для управления памятью такого дискового КЭШ привлекается резидентная программа, хранящаяся в ОП. Аппаратный дисковый КЭШ более дорогой, но более быстродействующий. Эффективность программного дискового КЭШ может быть доведена до уровня аппаратного. Причина успешного конкурирования программного с аппаратным КЭШ состоит в том, что эффективность зависит от быстродействия шины вводы вывода.

Программный дисковый КЭШ использует для своей работы ЦП, а аппаратный имеет свой собственный ЦП.

Управление дисковым КЭШ берет на себя изготовитель контроллера.

Один из способов управления- это буферизация дорожки. Когда ОС запрашивает сектор диска, контроллер считывает не только сектор, но и остальную часть дорожки, при этом сокращается время доступа к этим дорожкам или секторам. Наличие буфера дорожки в контроллере может смягчить влияние недостатков при чередовании секторов.

Второй способ – КЭШ-считывание. Емкость такого дискового КЭШ достаточна для хранения данных нескольких дорожек от 32 Кбайт до 16 Мбайт. При запросе сектора данных уточняется нет ли запроса сектора, нет ли обращения к системному диску и требуется ли сохранить копию данных. Обнаружение данных дискового КЭШ называется коэффициентом попадания, достигает 90%. При этом дисковый накопитель работает в 10 раз быстрее обычного режима. Но эффективность КЭШ не постоянна. Достаточным уровнем попадания считается 50%.

Третий способ - дисковый КЭШ со сквозной записью. Он ускоряет процесс записи за счет выделения части памяти. Дисковый КЭШ для хранения последней записанной информации, остальная часть для кэширования считывается. При необходимости записи такой КЭШ перехватывает запросы и выясняет не записаны ли эти данные на диске. Если результат положительный, то контроллер сообщает ЦП об окончании режима записи.

Четвертый способ - дисковый КЭШ с отложенной записью. Информация не записывается на МД при ее получении, а записывается позже при готовности контроллера дискового КЭШ (но запись еще не произведена). В таком случае имеется опасность потери данных, например, при отключении питания компьютера. Чтобы этого избежать надо использовать источник бесперебойного питания. В конечном итоге этот вид КЭШ обеспечивает максимальную производительность по сравнению с другими.

Одна из привлекательных сторон дисковой КЭШ-памяти в том, что связанные с ней преимущества могут быть получены без изменений в имеющемся аппаратном и программном обеспечении. Многие серийно выпускаемые дисковые КЭШ интегрированы в состав дисковых ЗУ. Дисковая КЭШ-память применяется и в персональных ВМ

9.8 Внешние запоминающие устройства на основе магнитных лент

ВЗУ на базе магнитных лент (ВЗУ на МЛ) в иерархии запоминающих устройств занимают нижнюю позицию и используются в основном в качестве устройств резервного копирования информации. В ленточных системах используется та же техника чтения и записи, что и в дисковых системах. Носителем служит тонкая полистироловая лента, покрытая намагничивающимся материалом. Покрытие может состоять из частиц технически чистого металла и связующего материала либо специально напылённой металлической пленки. В различных типах ВЗУ на МЛ лента может находиться в кассетах или картриджах. Кассета представляет собой корпус с двумя бобинами, на которые намотана магнитная лента. В картридже имеется лишь одна бобина с магнитной лентой, а приемная бобина находится в накопителе, куда устанавливается картридж.

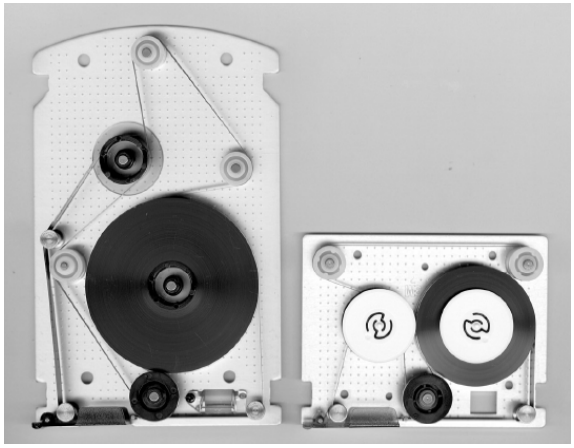


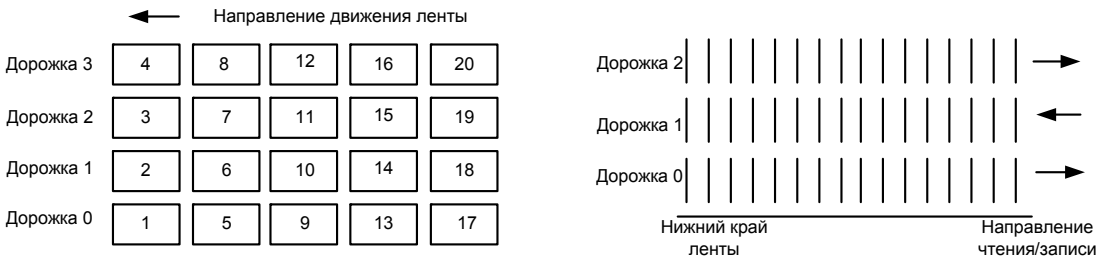
Рисунок 9.23- Ленточные картриджи Verbatim DC2120

В момент установки картриджа свободный конец ленты из картриджа захватывается специальным устройством накопителя и наматывается на приемную бобину. В зависимости от типа ВЗУ на МЛ ширина ленты варьируется от 0,38 см (0,15 дюйма) до 1, 27 см (0,5 дюйма). Толщина ленты порядка 0,025 мм.

При записи информации в ВЗУ на МЛ применяется один из двух подходов: линейная запись или наклонная запись.

Линейный метод записи. В первых ВЗУ на МЛ, реализующих этот метод, использовалась единственная дорожка, протяженностью во всю длину ленты. Данные представляются как последовательность битов, расположенных вдоль дорожки, аналогично тому, как это имеет место в случае дисков. Биты информации сгруппированы в байты, содержащие 9 битов информации (8 информационных и один контрольный). Так же, как и в дисках, данные читаются и записываются в виде смежных блоков, называемых *физическими записями.* Каждая запись отделялась от соседней *межблочным промежутком,* дающим возможность позиционирования головки считывания/записи на начало любого блока. Конец каждого блока помечается специальным маркером «конец блока», а конец ленты — маркером «конец ленты». Как и диски, лента форматируется для обеспечения доступа к конкретной физической записи.

Позже число дорожек стали увеличивать, а сами дорожки стали логически соединять в виде серпантина. Это позволило выполнять операции чтения/записи при двустороннем движении; увеличилась «логическая» длина ленты, которая составляла сумму длин всех дорожек. Такая техника записи представляет собой подвид линейной записи и называется *серпантинной записью.* Первое множество битов записывается вдоль полной длины ленты. При достижении конца ленты головки позиционируются на новую дорожку, и запись опять продолжается вдоль всей длины ленты, но уже при движении ленты в противоположном направлении. Процесс продолжается, пока не будет заполнена вся лента (рис. 9.24, *а).* Для увеличения скорости головка способна читать или писать несколько смежных дорожек одновременно (обычно от 2 до 8). Данные также записываются последовательно вдоль индивидуальных дорожек, но последовательные блоки распределяются по смежным дорожкам, как это показано на рис, 9.24, б.



а) б)

Рисунок 9.24-Типовая организация хранения данных на магнитной ленте при линейном методе записи: а — расположение блоков в системе, где одновременно читаются 4 дорожки; б — серпантинное чтение и запись

Наклонный-строчный метод записи. В ВЗУ на МЛ с наклонно-строчной записью несколько головок считывания/записи размещают на вращающемся барабане, установленном под углом к вертикальной оси (рис. 9.23). Последний охватывается лентой, прилегающей к нему рабочим слоем. Количество головок может быть различным; чаще всего применяют две головки, сдвинутые относительно друг друга на 180°. Ось барабана наклонена к плоскости движения ленты, благодаря чему и получается наклонное расположение дорожек. Лента транспортируется медленно, а головки вращаются с большой скоростью. Движение ленты при записи и чтении может производиться только в одном направлении. Сочетание быстрого вращения барабана и медленного перемещения ленты обеспечивает высокую плотность и скорость записи.

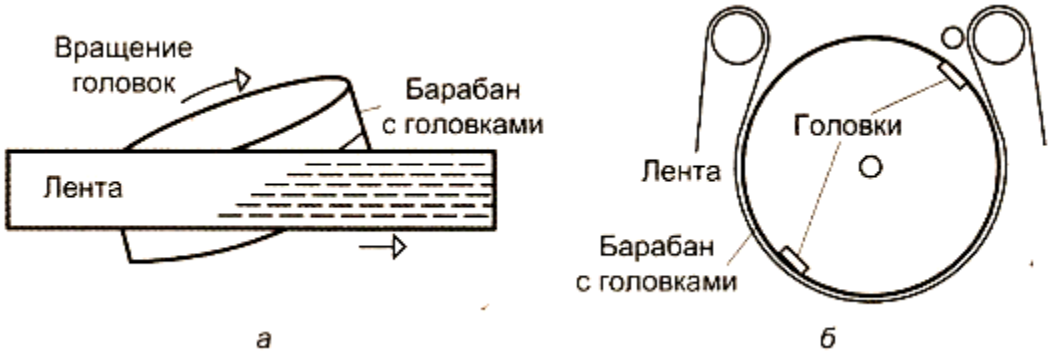


Рисунок 9.25-Наклонно-строчный принцип записи: а — принцип записи; б — охват лентой барабана

9.9 Устройства резервного копирования

В процессе эволюции вычислительной техники характер задач, возлагавшихся на ВЗУ на МЛ, неоднократно менялись. К настоящему моменту такие ЗУ используются главным образом в качестве устройств резервного копирования (УРК) информации. Такие устройства обеспечивают создание копий данных, хранящихся на магнитных дисках, с целью их восстановления в случае потери или искажения. Исходя из этого, рассмотрим лишь те типы ВЗУ на МЛ, которые используются именно для этой цели. В основе всех типов систем резервного копирования на базе магнитных лент всегда лежит *стример.*

Стримером называют накопитель, который записывает информацию на установленный в этот накопитель картридж или кассету.

Следующим типом УРК можно считать *стекер,* представляющий собой стример, которому придаются несколько картриджей, собранных в специальный лоток. Картриджи подаются в накопитель с помощью роботизированного механизма, причем последовательность подачи картриджей строго определена.

Разновидностью стекера можно считать *автозагрузчик.* Отличие состоит в том, что картриджи, размещенные в специальном магазине, могут подаваться в стример в произвольном порядке.

Значительно большие возможности для резервного копирования предоставляет *ленточная библиотека.* Она содержит несколько стримеров, так называемые отсеки, в которых хранятся картриджи, и механизм смены картриджей в стримерах. Возможны две схемы смены картриджей в стримерах: в первой схеме любой картридж может быть загружен в любой стример; во второй — каждый стример работает лишь с определенными картриджами. Ленточную библиотеку отличает высокая скорость передачи данных, огромная емкость и большая надежность. В рамках рассматриваемой группы УРК имеется также система, по структуре аналогичная RAID и полностью повторяющая спецификации RAID. Такой структурой являются массивы стримеров с избыточностью (RAIT — Redundant Array of Independent Tape). Задачи, решаемые с помощью RAIT, те же, что и в случае RAID — обеспечение высокой скорости и надежности. Физически RAIT-массив представляет собой корпус, где расположены несколько одновременно работающих стримеров. К сожалению, RAIT-массивы имеют малую емкость и не позволяют проводить автоматическую смену (ротацию) картриджей.

9.10 Контрольные вопросы

1. Перечислите основные характеристики ЗУ внешней памяти ЭВМ

2.Объясните процедуру записи на магнитный носитель.

3. Объясните процедуру чтения на магнитный носитель.

4. В чем заключается принцип продольной записи на магнитный носитель?

5. В чем заключается принцип перпендикулярной записи на магнитный носитель?

6.Перечислите основные характеристики дисковых систем.

7.Как размещается информация на магнитном диске?

8. Какова структура сектора жесткого диска?

9. В чем заключается метод организации диска с постоянной угловой скоростью?

10. В чем заключается метод организации диска с зонной записью?

11. Что такое технология RAID?

12. Перечислите методы повышения производительности дисковой подсистемы.

13.Какая идея лежит в основе систем обнаружения и коррекции ошибок?

14.Какие ошибки может обнаруживать схема контроля по паритету?

15.От чего зависят возможности выявления и коррекции ошибок с использованием кода Хэмминга?

16.Поясните назначение и принцип формирования кода синдрома в системе коррекции ошибок.

17.Какие типы запоминающих устройств на оптических дисках вы знаете?

18.Объясните общие принципы построения ВЗУ на ОД.

19.Объясните принцип записи информации на CD-ROM.

20.Как располагаются данные на CD-ROM?

21.Какова структура DVD дисков?

22. Перечислите основные характеристики дисков Blue-ray.

23. Что такое технология HVD?

24. Объясните технологии записи и чтения данных MO дисков.

25. Перечислите преимущества и недостатки SSD дисков по сравнению с магнитными дисками.

26. Какое назначение имеет дисковый КЭШ?

27. Чем отличается наклонный-строчный от линейного метода записи в ВЗУ на МЛ.

28.Назовите типы устройств резервного копирования. Перечислите их характеристики.

8.2 Схема системной платы ПК на основе процессора Pentium

Компьютеры на основе процессора Pentium используют высокоскоростную процессорную шину (ША, ШД, ШУ) и две периферийные шины: PCI (Peripheral Component Interconnect-соединение периферийных компонентов), предназначеная для обмена с высокоскоростными устройствами ввода-вывода; ISA(Industry Standard Architecture- стандартная промышленная архитектура).

В PCI используется мультиплексированная шина адреса- данных и шина управления. Проблема организации высокоскоростного обмена между ОП и процессором решена благодаря подключению ОП непосредственно к шине процессора. Все некритичные к скорости обмена периферийные устройства взаимодействуют с процессором, используя шину ISA.

ISA относится к демультиплексированным (то есть имеющим раздельные шины адреса и данных) 16-разрядным шинам. Обмен осуществляется 8-или 16-разрядными данными. На шине реализован раздельный доступ к памяти и к УВВ. Максимальный объем адресуемой памяти составляет 16 Мбайт (24 адресные линии). Максимальное адресное пространство для устройств ввода- вывода - 64 Кбайта (16 адресных линий), хотя практически все выпускаемые платы расширения используют только 10 адресных линий (1 Кбайт).

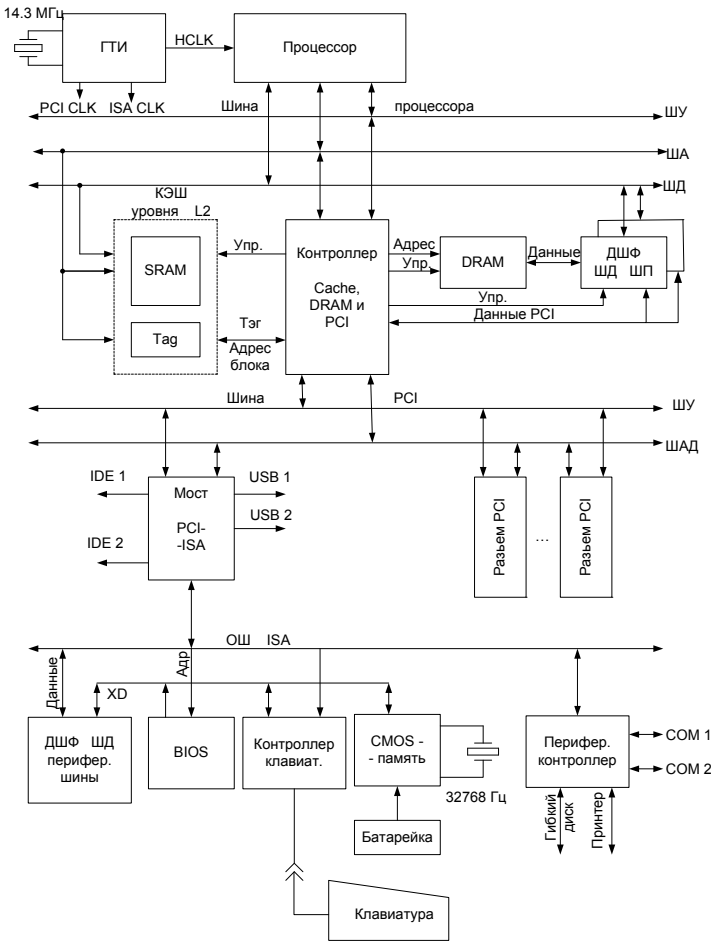


Рисунок 8.5 - Структура системной платы ЭВМ типа IBM PC/AT- Pentium

Формирование необходимых для работы большинства устройств системной платы тактовых частот осуществляет ГТИ с подключенным к нему кварцевым резонатором на 14,3 МГц (точнее-14.318 МГц). Он вырабатывает сигналы тактовой частоты, подаваемые на процессор (HCLK), шины PCI (PCICLK) и ISA (ISACLK). Тактовые сигналы, необходимые для работы ядра процессора и его шинного интерфейса формируются внутри процессора путем умножения сигнала тактовой частоты HCLK.

Повышение степени интеграции чипсетов, используемых в ЭВМ IBM PC/AT Pentium, позволило совместить в одной микросхеме (северном мосте) функции контроллеров шины PCI, динамической памяти DRAM и КЭШ- памяти. Контроллер динамической памяти обеспечивает формирование сигналов ША и ШУ шины памяти. Данные с DRAM пересылаются на шину процессора и обратно через ДШФ ШД ШП.

Контроллер КЭШ- памяти осуществляет формирование сигналов управления как собственно статической памятью SRAM КЭШ- памяти, так и памятью ТЭГов (Tag). В начале циклов записи или чтения содержимое Tag- памяти по линиям "Адрес блока" подается в контроллер КЭШ- памяти для сравнения его со старшей частью адреса строки ОП, подлежащей записи или чтению. При записи нового содержимого в строку КЭШ- памяти по этим линиям осуществляется передача старшей части адреса кэшируемой памяти для записи в соответствующую ячейку Tag.

Мост PCI- ISA (южный мост) обеспечивает организацию взаимодействия соответствующих шин, формируя используемые в системе сигналы обеих шин. Дополнительными его функциями являются организация взаимодействия с ВЗУ через интерфейсы IDE и с периферийными устройствами, использующими последовательный интерфейс USB (Universal Serial Bus- универсальная последовательная шина). В этой- же микросхеме интегрированы большинство стандартных подсистем - прерывания, ПДП, ПИТ и др.

Подключение низкоскоростных стандартных УВВ в архитектуре ЭВМ IBM PC/AT Pentium аналогично ЭВМ IBM PC/AT 286.

8.3 Основные сигналы шинного интерфейса процессора Pentium

Упрощенное условное графическое обозначение процессора Pentium приведено на рисунке 8.6.

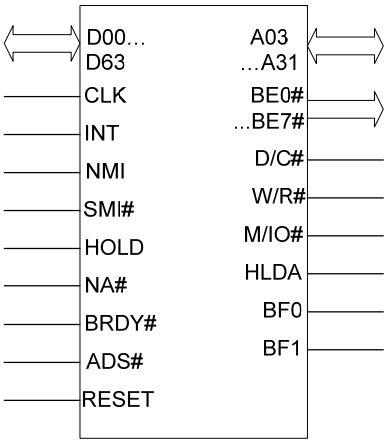


Рисунок 8.6 - Упрощенное графическое обозначение процессора Pentium

Ширина внешней двунаправленной ШД этого процессора равна 64 битам (разряды D0…D63), разделяемых на 8 байт (байт ноль B0- разряды D0…D7, B1- разряды D8…D15, …, B7- разряды D56…D63)

ША процессора Pentium эквивалентна 32- разрядной двоичной параллельной шине, но имеет более сложную организацию, состоящую из основной 29- ти разрядной параллельной шины A03…A31 и дополнительной шины BE0#…BE7#. Основная 29- ти разрядная шина позволяет адресовать 512 миллионов учетверенных слов полной ширины ШД процессора. Унитарный код, выдаваемый дополнительной шиной BE0…BE7, используется взамен трех пропущенных разрядов A0…A2 основной шины. Он позволяет объявлять действительность или не действительность любого байта данных B0#…B7#, выдаваемых на ШД. Признаком действительности какого- либо байта данных является установка в состояние лог. 0 соответствующего сигнала BE# (Bait Enable- разрешение байта).

Информация о типе шинного цикла, выполняемого в данный момент процессором, выдается им с помощью трех сигналов D/C#, W/R# и M/IO#. Первый из них извещает систему о том, что передается по ШД- данные (D) или команда (C), второй – о записи (W) или чтении (R) информации, а третий сигнал сообщает, с чем процессор взаимодействует- с памятью (M) или УВВ (IO). Кодировка типов шинных циклов процессора приведена в таблице 8.1.

Таблица 8.1 - Кодировка типов шинных циклов процессора Pentium

Уровень сигнала			Тип шинного цикла
M/IO#	D/C#	W/R#	
1	1	1	Запись данных в память
1	1	0	Чтение данных из памяти
1	0	1	Остановка процессора
1	0	0	Чтение кода программы из памяти
0	1	1	Запись данных в УВВ
0	1	0	Чтение данных из УВВ
0	0	0	Подтверждение прерывания

Как видно из таблицы, определение типа шинного цикла при заданных значениях уровней сигналов D/C#, W/R# и M/IO# довольно проста. Высокий уровень сигнала M/IO# означает, что текущий шинный цикл осуществляет обращение к памяти, низкий- к УВВ. Высокий уровень сигнала W/R# означает, что текущий шинный цикл осуществляет запись в память или УВВ, низкий- их чтение. Однако два шинных цикла- остановка процессора и подтверждение прерывания не вписываются в это простое правило и их коды надо просто запомнить.

Первостепенным по важности среди оставшихся сигналов процессора является сигнал ADS# (Address Status), активный уровень которого информирует о начале очередного шинного цикла.

NMI (Non Mascable Interrupt)- сигнал немаскируемого прерывания, на которое процессор реагирует вне зависимости от каких- либо флагов. Это прерывание используется для сообщения о серьезных ошибках в работе системы, приводящих к не достоверным результатам. В ЭВМ IBM PC сигнал на вход NMI поступает от схем контроля по нечетности памяти (на рисунке 8.1 не показаны) и ошибки работы УВВ через ключ, управляемым битом 7 периферийного порта B.

SMI (System Management Interrupt)- сигнал прерывания для входа в режим SMM, который в основном используется для организации снижения энергопотребления при не использовании компьютера пользователем.

NA# (Next Addres)- сигнал запроса адреса следующего шинного цикла при “мягкой” конвейеризации шины.

BRDY# (Burst RDY)- вход готовности, по которому завершается текущий пакетный цикл передачи данных. Используется в процессоре Pentium вместо сигнала RDY#.

BF0, BF1 (Bus Frequency)- входы управления коэффициентом умножения частоты CLK, подаваемой на процессор с внешнего ГТИ, внутри процессора. Умноженное значение частоты используется процессором для синхронизации работы его ядра и шинного интерфейса.

Назначение остальных сигналов процессора Pentium не отличается от назначения одноименных сигналов классического процессора.

8.4 Организация шины PCI

8.4.1 Общая характеристика шины PCI

PCI (Peripheral Component Interconnect) local bus- шина соединения периферийных компонентов. Она разрабатывалась в расчете на шинный интерфейс процессоров Pentium. В архитектуре ЭВМ эта шина стала центральной, через которую процессор взаимодействует со всеми остальными шинами (см. рисунок 8.7). Первая версия стандарта шины PCI 1.0 появилась в 1992 г., PCI 2.1 - в 1995 г.

Шина PCI является синхронной шиной, в которой фиксация всех сигналов выполняется по нарастающему фронту сигнала синхронизации CLK (см. рисунок 8.4.3). Номинальное значение частоты синхронизации CLK равно 33 МГц. Начиная с версии 2.1 допускается повышение частоты CLK до 66 МГц.

Шина PCI относится к мультиплексированным шинам, в которой для передачи адреса и данных (последовательно во времени) используются одни и те же линии. Номинальная разрядность ШД и ША- 32бита, возможно увеличение их разрядности до 64 бит. При частоте шины 33 МГц пропускная способность шины равна 132 Мбайт/с для 32 разрядной шины и 264 Мбайт/с для 64- битной шины.

Подключенные к шине устройства (функции) представляются процессору непосредственно подключенными к его шине. Им назначаются адреса из адресного пространства памяти или УВВ. Дешифрирование адреса на шине PCI распределено, т.е. выполняется в каждом устройстве. Каждое устройство отвечает только на свой адрес. Спецификация PCI требует перемещаемости всех занимаемых ресурсов в пределах доступного пространства адресации, что обеспечивает их бесконфликтное распределение для многих устройств.

С устройствами PCI процессор может взаимодействовать командами обращения к памяти и портам ввода-вывода, адресованным к областям, выделенным данному устройству при конфигурировании системы. Устройства могут вырабатывать запросы маскируемых и немаскируемых прерываний. Понятие каналов ПДП для шины PCI не вводится, но устройство может выступать в роли задатчика, поддерживая высокопроизводительный обмен с памятью без привлечения процессора. Так может быть реализован обмен в режиме ПДП с устройствами IDE, подключенными к мосту PCI- ISA (см. рисунок 8.5).

8.4.2 Основные сигналы шины

CLK - сигнал синхронизации работы устройств. Является входным сигналом для каждого PCI - устройства. Все сигналы PCI, за исключением RST#, IRQA#, IRQB#, IRQC# и IRQD# фиксируются по нарастающему фронту сигнала CLK.

RST# - сигнал сброса устройств в исходное состояние.

AD[31::00]- сигналы адреса данных. Адрес и данные последовательно во времени выдаются (мультиплексированы) на одни и те же линии шины PCI. Транзакция (обмен) шины состоит из фазы адреса, сопровождаемой одним или большим количеством фаз данных. В течение фазы адреса на линии AD[31::00] выдается физический адрес (32 бита) устройства. В фазе данных на лини AD[31::00] выдаются данные, при этом разряды AD[07::00] содержат младший значащий байт, а AD[31::24] содержат старший значащий байт.

C/BE[3::0]# (Bus Command и Byte Enables)- команды шины и разрешение байта. Сигналы мультиплексированы на одних и тех же линиях шины. Во время фазы адреса транзакции, сигналы C/BE[3::0]# определяет команду шины (смотри раздел 3.1). В течение фазы данных сигналы C/BE[3::0]# используется в качестве сигналов Byte Enable т.е определяют, какие байты действительны, а какие не используются.

FRAME# (Кадр). Активный уровень сигнала означает начало транзакции (с фазы адреса). Снятие сигнала указывает, что последующий цикл передачи данных является последним в транзакции.

IRDY# (Initiator Ready)- готовность инициатора к обмену. Сигнал показывает, что на линиях AD[31::00] присутствуют достоверные данные. При чтении данных сигнал означает готовность мастера к приему данных.

TRDY#(Target Ready)- целевое устройство готово. Показывает способность целевого агента (выбранного устройства) завершить текущую фазу данных транзакции. Используется совместно с сигналом IRDY#. При чтении TRDY# указывает, что на линиях AD[31::00] присутствуют достоверные данные. Во время записи это означает готовность целевого устройства к принятию данных. Циклы ожидания вставляются до тех пор, пока активны оба IRDY# и TRDY#.

DEVSEL# (Device Select)- устройство выбрано. Сигнал показывает, что ЦУ дешифрировало адрес, выданный на шину AD. Используется в качестве ответа ЦУ инициатору обмену на адресованную к нему транзакцию.

INTA#, INTB#, INTC# и INTB# (Interrupt A…D) – входы запросов прерываний. Активным уровнем сигнала прерывания является лог. 0, при этом для устройств используется выход с открытым коллектором. Переход сигналов INTx# в активное состояние и обратно асинхронно по отношению к сигналу CLK.

REQ# (Request) – запрос. Сигнал показывает арбитру, что данному агенту требуется поработать с шиной. Каждый мастер имеет свой индивидуальный вывод REQ#.

GNT# (Grant)- разрешение. Сигнал показывает агенту, что разрешен доступ к шине. Каждый мастер имеет свой индивидуальный вывод GNT#.

8.4.3 Протокол шины PCI

В каждой транзакции (обмене по шине) участвуют два устройства- инициатор (initiator) обмена, он же мастер (master) или ведущее устройство и целевое (target) устройство (ЦУ), оно же ведомое (slave). Шина PCI все транзакции трактует как пакетные. Каждая транзакция начинается фазой адреса, за которой могут следовать одна или несколько фаз данных (см. рисунок 8.7).

В каждый момент времени шиной может управлять только один мастер, получивший на это право от арбитра шины. Каждый мастер имеет пару сигналов- REQ# для запроса управления шиной и GNT# для подтверждения предоставления управления шиной. Устройство может начинать транзакцию (устанавливать сигнал FRAME#) только при активном полученном сигнале GNT#. Арбитраж запроса на использование шины выполняет арбитр шины, входящий в состав контроллера шины PCI, в свою очередь входящего в состав чипсета северного моста системной платы ЭВМ.

Для передачи адреса и данных используются общие мультиплексированные линии AD[31::00]. Четыре мультиплексированные линии C/BE[3::0]# обеспечивают кодирование команд в фазе адреса транзакции и разрешение байт в фазе данных. Для начала транзакции инициатор обмена должен активизировать сигнал FRAME# (т.е установить его в лог. 0), вслед за чем выставить на шину AD[31::00] адрес ЦУ, а на линии C/BE[3::0]# - информацию о типе транзакции (команду).

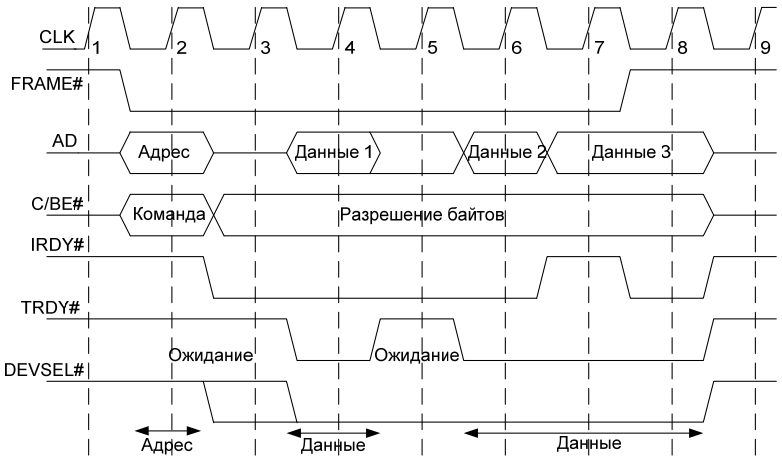


Рисунок 8.7 - Цикл обмена шины PCI

Адресованное ЦУ после дешифрации адреса должно заявить о своем присутствии сигналом DEVSEL#. Для продолжения обмена инициатор должен указать на свою готовность к обмену данными активизацией сигнала IRDY#. Когда к обмену данными будет готово и ЦУ оно должно активизировать сигнал TRDY#. Данные по шине AD могут передаваться только при активных уровнях обеих сигналов IRDY# и TRDY#. С помощью этих сигналов инициатор и ЦУ согласуют свои скорости обмена, вводя такты ожидания.

Транзакции заканчиваются переводом сигналов FRAME# и IRDY# в пассивное состояние (состояние лог.1).

8.1.3 Организация шины ISA

8.1.3.1 Особенности шины ISA

Шина ISA была разработана специально для персональных компьютеров типа IBM PC AT (начиная с процессора i80286) и являлась фактическим стандартом для всех изготовителей этих компьютеров. В то же время отсутствие официального международного статуса шины ISA (она не утверждена в качестве стандарта ни одним международным комитетом по стандартизации) приводило к тому, что многие производители допускали некоторые отличия от фирменного стандарта.

Наиболее распространенное конструктивное исполнение шины - разъемы (слоты), установленные на материнской плате компьютера, все одноименные контакты которых соединены между собой, то есть все разъемы абсолютно равноправны. На шине присутствуют четыре напряжения питания: +5 В, -5 В, +12 В и -12 В, которые используются платами расширения.

8.1.3.2 Основные сигналы шины ISA

Рассмотрим назначение сигналов магистрали ISA и их особенности.

SA0...SAI9 - фиксируемые адресные разряды (они действительны в течение всего цикла обмена). Используются для выдачи 20 младших разрядов адреса памяти и для адресации УВВ. При обращении к устройствам ввода/вывода действительны только сигналы SAO...SAI5 (но практически все платы расширения работают только с SA0...SA9). При регенерации памяти действительны только сигналы SA0...SA7.

LAI7...LA23 - нефиксируемые адресные разряды. Используются для адресации памяти и выработки сигнала -MEM CS16. Действительны только в начале цикла обмена. Исполнитель должен фиксировать их по отрицательному фронту сигнала BALE. Для фиксации необходимо использовать регистр типа "Защелка", стробируемый сигналом BALE.

BALE (Bus Address Latch Enable- разрешение защелкивания адреса)- сигнал стробирования адресных разрядов. Его отрицательный фронт соответствует действительности адреса на линиях SAO...SAI9 и LA17...LA23.

SBHE (System Bus High Enable - разрешение старшего байта) - определяет разрядность передаваемых данных (8- или 16- разрядные). Становится активным при передаче старшего байта или 16-разрядного слова (определяется сигналом SA0), пассивен при передаче младшего байта. В режиме MASTER источником этого сигнала является устройство, которое захватило магистраль.

SD0...SD15 - разряды данных.. Обмен данными с 8-разрядными платами расширения осуществляется только по линиям SD0...SD7

-SMEMR, -MEMR (Memory Read - чтение памяти) – сигналы чтения данных из памяти. Память должна выставлять данные на ШД при активизации этих сигналов. Сигнал -SMEMR вырабатывается только при обращении к адресам, не превышающим FFFFFH (в пределах 1 Мбайта), сигнал MEMR- при обращении ко всем адресам.

-SMEMW, -MEMW (Memory Write - запись памяти) - стробы записи данных в память. Память должна принимать данные с ШД по положительному (заднему) фронту этих сигналов. Сигнал -SMEMW вырабатывается только при обращении к адресам, не превышающим FFFFF (в пределах 1 Мбайта), сигнал -MEMW - при обращении ко всем адресам.

-IOR (I/0 Read) - сигнал чтения данных из устройств ввода- вывода. Устройство ввода-вывода должно выставлять свои данные при активизации сигнала IOR и снимать их при снятии IOR.

-IOW (I/O Write) - сигнал записи данных в устройства ввода- вывода. Устройство ввода-вывода должно принимать данные по положительному (заднему) фронту сигнала -IOW.

MEM CSI6 (Memory Cycle Select - выбор цикла для памяти) - сигнал выставляется памятью для сообщения процессору (задатчику) о том, что она имеет 16-разрядную организацию. При отсутствии этого сигнала выполняется 8-разрядный обмен.

-I/O CS16 (I/O Cycle Select - выбор цикла для устройства ввода/вывода) - сигнал выставляется устройством ввода- вывода для сообщения задатчику о том, что оно имеет 16-разрядную организацию. При отсутствии этого сигнала выполняется 8 -разрядный обмен. Сигнал вырабатывается при распознавании устройством ввода/вывода своего адреса на линиях SAO...SAI5.

I/0 CH RDY (I/0 Channel Ready - готовность канала ввода- вывода)- сигнал снимается (делается низким) исполнителем (устройством ввода/вывода или памятью) по переднему фронту сигналов IOR и IOW в случае, если он не успевает выполнить требуемую операцию в темпе задатчика. При этом реализуется асинхронный обмен. Если исполнитель успевает работать в темпе задатчика, сигнал не снимается (фактически не устанавливается в низкий уровень). Шинный цикл процевссора в ответ на снятие этого сигнала продлевается на целое число периодов сигнала SYSCLK.

-I/O CH CK (I/O Channel Check - проверка канала ввода- вывода). Сигнал вырабатывается любым исполнителем (устройством ввода- вывода или памятью) для информирования задатчика о фатальной ошибке работы компьютера (например - об ошибке четности при доступе к памяти). Сигнал вызывает немаскируемое прерывание.

-REFRESH (Refresh - регенерация) - сигнал выставляется контроллером регенерации для информирования всех устройств на магистрали о выполнении циклов регенерации динамического ОЗУ компьютера (каждые 15,6 мкс). При регенерации выполняется псевдочтение по одному из 256 адресов ОЗУ (активизируются только разряды адреса SA0...SA7). Полный цикл регенерации всех строк DRAM - 4 мс.

RESET DRV (Reset of Driver - сброс устройства) - сигнал сброса в начальное состояние всех устройств на магистрали ISA. Вырабатывается при включении или сбое питания, а также при нажатии на кнопку RESET компьютера. Внешние платы должны в ответ на этот сигнал (длительностью не менее 1 мс) перевести все свои выходы в высокоимпедансное состояние.

SYSCLK (System Clock - системная частота) - сигнал тактовой частоты шины ISA. В большинстве компьютеров его частота равна 8 МГц независимо от тактовой частоты процессора. Если в программе SETUP предусмотрена возможность изменения тактовой частоты магистрали, пользователь может задавать ее в широких пределах. Но для обеспечения наибольшей совместимости со всеми имеющимися платами расширения ISA не рекомендует поднимать эту частоту выше 8 МГц. К тому же на производительность новых компьютеров в целом она влияет незначительно.

OSC - не синхронизированный с SYSCLK сигнал кварцевого генератора с частотой 14,31818 МГц. Может использоваться платами расширения в качестве тактового сигнала, так как его частота одинакова для всех компьютеров с магистралью ISA.

IRQ (Interrupt Request - запрос прерывания) - сигналы запроса радиальных прерываний. Запросом является нарастающий фронт на соответствующей линии IRQ. Сигнал должен удерживаться до начала обработки процессором запрошенного прерывания. На каждой линии IRQ должен быть один выход. Многие входы IRQ заняты системными ресурсами компьютера. Сигналы IRQO...IRQ2, IRQ8 и IRQI3 задействованы на системной плате и недоступны платам расширения

DRQ (DMA Request - запрос ПДП) - сигналы запросов прямого доступа к памяти. Запросом является положительный переход на соответствующей линии DRQ. Сигнал должен удерживаться до получения ответного сигнала - DACK с тем же номером.

DACK (DMA Acknowledge - подтверждение ПДП) - сигналы подтверждения предоставления прямого доступа. Вырабатываются в ответ на соответствующий сигнал DRQ в случае, если прямой доступ предоставлен данному каналу. Удерживаются до окончания прямого доступа.

-MASTER (Master- хозяин, задатчик) - используется платой расширения, желающей стать задатчиком магистрали. В этом случае она выставляет сигнал DRQ и, получив в ответ сигнал - DACK, устанавливает сигнал –MASTER.

8.1.3.3 Шинные циклы магистрали ISA

В режиме программного обмена информацией на магистрали ISA выполняются четыре типа циклов: цикл записи в память, цикл чтения из памяти; цикл записи в устройство ввода-вывода; цикл чтения из устройства ввода- вывода. Временные диаграммы циклов программного обмена с устройствами ввода- вывода изображены на рисунке 8.3.

Циклы начинаются с выставления задатчиком адресных сигналов на линиях SA0...SAI5 и сигнала SBHE. В случае чтения устройства ввода- вывода задатчик выставляет сигнал IOR, в ответ на который исполнитель должен выдать данные на шину данных. Эти данные должны быть сняты исполнителем после окончания сигнала IOR. В цикле записи задатчик выставляет записываемые данные и сопровождает их стробом записи IOW. Здесь надо отметить, что хотя в соответствии со стандартом установка записываемых данных предшествует выставлению IOW, в некоторых компьютерах реализуется обратный порядок: сначала выставляется IOW, а затем появляются данные.

В случае, когда исполнитель не успевает выполнить требуемую от него действия в темпе магистрали, оно может приостановить их на целое число периодов сигнала SYSCLK с помощью снятия (перевода уровня сигнала в состояние лог. 0) сигнала I/O CH RDY (так называемый удлиненный цикл). Это производится в ответ на получение сигнала IOR или IOW.

На рисунке 8.4 приведены временные диаграммы циклов обмена с памятью. Для асинхронного режима обмена (удлиненного цикла) здесь также используется сигнал I/O CH RDY. Отметим, что память, должна обрабатывать все адресные разряды, включая LAI7...LA23.

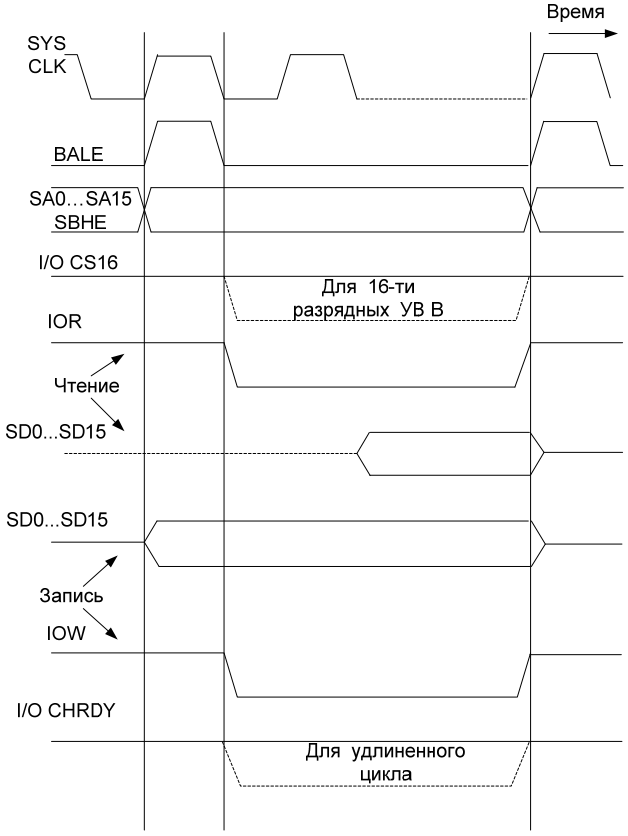


Рисунок 8.3 - Временные диаграммы циклов программного обмена с устройствами ввода-вывода

Одной из особенностей магистрали ISA является поддержка проведения регенерации динамической памяти компьютера с помощью специальных циклов регенерации на магистрали. Эти циклы выполняет входящий в состав

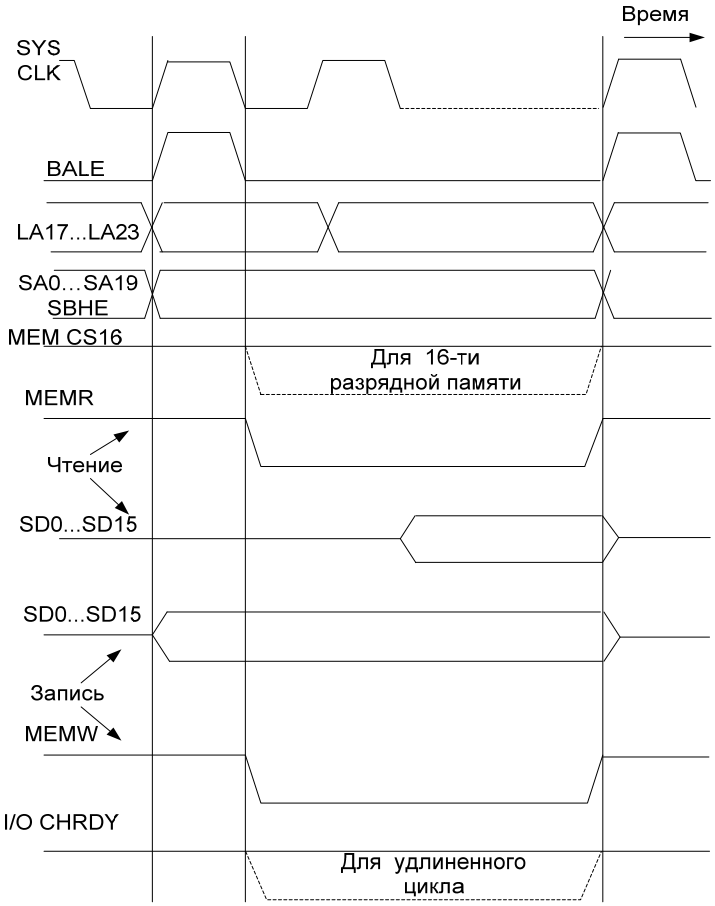


Рисунок 8.4 - Временные диаграммы циклов программного обмена с памятью

материнской платы компьютера контроллер регенерации, который должен для этого получать управление магистралью каждые 15,6 микросекунд. Во время цикла регенерации производится чтение одной из 256 строк DRAM. При этом читаемая информация нигде не используется, то есть используется цикл псевдочтения. Проведение 256 циклов регенерации обеспечивает непрерывное сохранение информации в ОЗУ. Цикл регенерации включает в себя:

- установление активного уровня сигнала REFRESH;
- формирование адреса SA0...SA7;
- формирование сигнала MEMR.

В случае необходимости может использоваться сигнал I/O CH RDY.

8.1.3.4 Электрические и конструктивные характеристики шины ISA

При проектировании систем с использованием шины ISA помимо протоколов обмена по магистрали надо учитывать также электрические характеристики сигналов.

Стандарт магистрали определяет требования к входным и выходным токам приемников и передатчиков каждой из плат расширения. Несоблюдение этих требований может нарушить функционирование всего компьютера и даже вывести его из строя. Выходные каскады передатчиков магистральных сигналов шины должны выдавать ток низкого уровня не меньше 24 мА, а ток высокого уровня - не меньше 3 мА. Входные каскады приемников сигналов должны потреблять входной ток низкого уровня не больше 0,8мА, а входной ток высокого уровня - не больше 0,04 мА. Кроме этого необходимо учитывать, что максимальная длина печатного проводника от контакта разъема до вывода микросхемы не должна превышать 65 миллиметров, а максимальная емкость относительно земли по каждому проводнику шины не должна быть больше 20 пФ.

К некоторым линиям шины подключены нагрузочные резисторы, идущие на шину питания +5В, а в некоторые линии включены последовательные резисторы номиналом 22 или 27 Ом.

Системный интерфейс ISA является развитием интерфейса XT-Bus, используемого в ЭВМ IBM PC/XT и характеризуется большей разрядностью ША (24 вместо 20 в XT-Bus), ШД (16

вместо 8) и почти вдвое большем количестве сигналов запроса прерываний и подсистемы ПДП. Для электрической и конструктивной совместимости УВВ, разработанных для интерфейса XT-Bus с интерфейсом ISA, основной 62-контактный соединитель (разъем) XT-Bus был дополнен 36-контактным. Ряды контактов 62-контактного соединителя имеют обозначения A и B, а 36-контакного - C и D. На выводах A1-A31, B1-B31 основного соединителя были оставлены сигналы шины XT-bus, на контакты C1-C18, D1-D18 второго соединителя были выведены дополнительные сигналы шины AT-bus.

8.1.3.5 Конвейеризация шины

Анализ временных диаграмм обмена по шине ISA показывает, что сигналы ША шины ISA используются только в начальной фазе любого шинного цикла. После дешифрации адреса памятью или УВВ, т.е после использования информации об адресе УВВ или ячейки памяти, удержание адреса, используемого в текущем шинном цикле становится ненужным. В то же время, после выдачи адреса на ША в начале любого шинного цикла требуется определенная временная пауза до начала дешифрации адреса из-за задержки распространения сигналов в логических схемах и переходных процессов в линиях шины. Логичным решением, приводящим к повышению скорости обмена по шине, является конвейеризация шины. Под конвейеризацией шины понимается прием установки адреса для следующего шинного цикла в предыдущем шинном цикле. Первоначально (в ЭВМ IBM PC/AT 286) адрес для следующего шинного цикла устанавливался с середины предыдущего. В более поздних версиях используется так называемая “мягкая конвейеризация”, в которой устройства сообщают процессору о своей готовности к приему следующего адреса с помощью сигнала NA# (Next Address- следующий адрес), после появления которого процессор выдает на ША адрес для следующего шинного цикла. Конвейеризация позволяет повысить скорость обмена по шине и производительность ЭВМ в целом благодаря уменьшению количества тактов ожидания, вырабатываемых процессором при неготовности к обмену памяти или УВВ (см. назначение сигнала I/0 CH RDY шины ISA).

8.5 Контрольные вопросы

1. Перечислите названия шин ПК на основе процессора Pentium.
2. Назовите основные сигналы шинного интерфейса процессора Pentium и назначение?
3. К какому типу шин относится шина PCI?
4. Каково назначение сигналов FRAME# и IRDY#?
5. Значения тактовых частот шины PCI?
6. Назовите основные сигналы шины ISA?

Библиографический список

1. Организация ЭВМ. 5-е изд./К. Хамахер, З. Вранешич, С. Заки.- СПб.: Питер; Киев: Издательская группа BHV, 2003. – 848 с.: ил.
2. Цилькер Б. Я., Орлов С. А. Организация ЭВМ и систем: Учебник для вузов. – СПб.: Питер, 2004. – 668 с.: ил.
3. Гук М. Аппаратные средства IBM PC. Энциклопедия. 2-е изд. – СПб.: Питер. 2001. – 928 с.: ил.
4. Сергеев Н.П., Вашкевич Н.П. Основы вычислительной техники: Учеб. пособие для электротехн. специальн. вузов. – 2-е изд. перераб. и доп. – М.: Высш. шк., 1988. – 311с., ил.
5. Балашов Е.П. и др. Микро- и мини ЭВМ: Учеб. пособие для вузов. - Л.: Энергоатомиздат, Ленингр. отд-ние, 1984. – 376с., ил.
6. Балашов Е.П., Пузанков Д.В. Микропроцессоры и микропроцессорные системы: Учеб. пособие для вузов/Под ред. В.Б. Смолова.- М., "Радио и связь",1981. – 328с., ил.
7. Бикташев Р.А., Князьков В.С. Многопроцессорные системы. Архитектура, топология, анализ производительности: Учеб. пособие.- Пенза: Пенз. гос. ун-т, 2003. - 217 с.
8. Компьютеры на СБИС: В 2-х кн. Кн. 2: Пер. с япон./ Мотоока Т., Сакаути М., Харикоси Х. И др.- М.: Мир, 1988.- 336с.
9. Сетевые операционные системы/ В.Г. Олифер, Н.А. Олифер.- СПб.: Питер. 2001. –544 с.
10. Шагурин И.И., Бердышев Е.М. Процессоры семейства Intel P6. Архитектура, программирование, интерфейс.- М.: Горячая линия - Телеком, 2000.- 248 с.
11. Корнев В.В., Кисилев А.В. Современные микропроцессоры.- М.: Нолидж, 1988. - 240 с.
12. Карасев В.В., Кисилев А.В. Современные микропроцессоры- 3-е изд., перераб и доп. СПб.: БХВ- Петербург, 2003. -448 с.
13. Таненбаум Э. Архитектура компьютера. 4-е изд. СПб.: Питер, 2003.-704 с.
14. Айден К., Колесниченко О. и др. Аппаратные средства PC - 2-е изд., перераб и доп. СПб.: БХВ- Санкт-Петербург, 1998. -608 с.
15. Каган Б.М. Электронные вычислительные машины и системы: Учеб. пособие для вузов. – 2-е изд. перераб. и доп. – М.: Энергоатомиздат, 1985. – 552с., ил.
16. Партыка Т.Л., Попов И.И. Вычислительная техника: Учеб. пособие. – М.: Форум: ИНФРА-М, 2007. – 608с.

СОДЕРЖАНИЕ