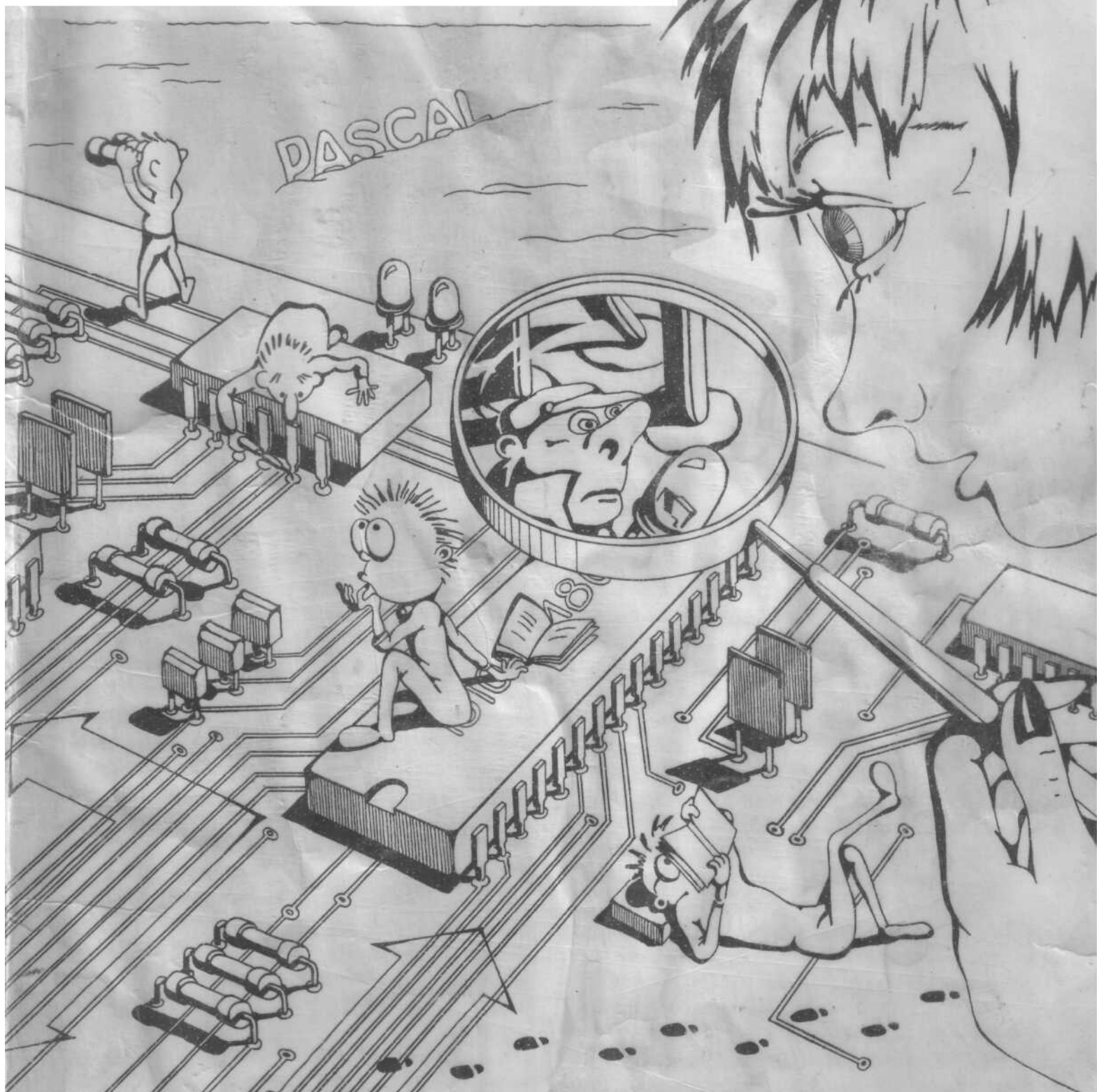


АППАРАТНО-ПРОГРАММНЫЕ СРЕДСТВА УСТРОЙСТВ НА БАЗЕ МИКРОПРОЦЕССОРА КМ1810ВМ86

МЕТОДИЧЕСКОЕ ПОСОБИЕ
ДЛЯ НАЧИНАЮЩИХ ЧАСТЬ 1

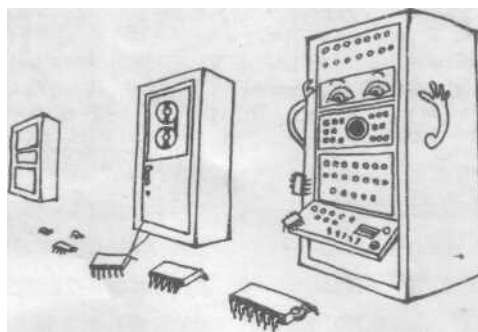


радиоизмерительная
техника

ВВЕДЕНИЕ. МИКРОПРОЦЕССОРЫ И МИКРО-ЭВМ

УРОК 1

МИКРОПРОЦЕССОР. ОБЩИЕ ПОНЯТИЯ



Создание микропроцессоров (МП) можно считать одним из важнейших достижений двух быстроразвивающихся отраслей промышленности: вычислительной техники и микроэлектроники.

Первые ЭВМ выполнялись с применением тысяч электронных ламп. Они были громоздкими и неудобными. Следующее поколение ЭВМ строилось на транзисторах, что сделало их значительно надежнее, уменьшило их размеры и стоимость. Совершенствование вычислительных машин происходило одновременно с развитием технологии производства полупроводниковых схем, при этом сокращалось разнообразие типов логических схем, используемых при конструировании ЭВМ, и возрастал спрос на эти схемы. Указанные факторы стали предпосылкой создания в середине 60-х годов семейства логических схем малого и среднего уровня интеграции. Технология производства интегральных схем развивалась под влиянием двух тенденций: стремления к выпуску дешевой продукции и расширения ее функциональных возможностей за счет усложнения схемных решений.

С конца 60-х - начала 70-х годов широко используются интегральные схемы большого уровня интеграции (БИС), в которых много разных функциональных возможностей реализовано в конструктивно единой схеме. Большинство пер-

вых БИС были ориентированы на выполнение определенных специальных функций, но некоторые типы этих схем были универсальны, например - запоминающие устройства.

Появление электронных калькуляторов свидетельствовало о существенных достижениях в технологии производства БИС. Аппаратная база первых калькуляторов - большое количество (70-100) отдельных интегральных схем. Создание специальных БИС позволило строить калькулятор на 5 или 6 модулях. К середине 70-х годов электронный калькулятор уже выполнялся на одной БИС.

Указанные достижения привели к следующему закономерному этапу в развитии вычислительной техники - реализации функции центрального процессора ЭВМ на одной интегральной схеме, получившей название микропроцессор.

МП, как и любая цифровая вычислительная машина, работает под управлением программы. Архитектура МП подобна архитектуре цифровой вычислительной машины.

С начала 70-х годов основные усилия ученых и инженеров были направлены на совершенствование архитектуры МП с целью увеличения их быстродействия и мощности. Первые МП обрабатывали одновременно 4 бита данных, представленных в цифровой форме, т.е. использовались

4-разрядные слова. Скорость работы таких МП была низкой, они значительно уступали мини-ЭВМ. Однако вскоре появилось новое поколение МП: на смену 4-разрядным пришли 8- и 16-разрядные микро-ЭВМ. Набор команд, выполняемых МП, расширился, а сами команды стали более сложными, с большими функциональными возможностями. Некоторые МП по своим рабочим характеристикам сравнялись с мини-ЭВМ, а в отдельных случаях даже превзошли их.

ФУНКЦИИ МИКРОПРОЦЕССОРА

Одна из главных функций МП - обработка данных, она включает в себя вычисления и манипулирование данными. Схемы, выполняющие вычисления, образуют арифметико-логическое устройство (АЛУ), в результате работы которого данные изменяют свои значения. АЛУ выполняет сложение, вычитание, сравнение, операции И и ИЛИ, положительное и отрицательное приращение. Но АЛУ не осуществляет перемещение данных ни до, ни после выполнения операции.

Как же в АЛУ поступают данные, подлежащие обработке? В МП, за пределами АЛУ, имеются другие схемы, манипулирующие данными и перемещающие их в места, доступные АЛУ. После того, как АЛУ выполнило требуемые операции, эти схемы пересылают данные другим адресатам.

Другая главная функция МП - управление системой. Схемы управления позволяют декодировать и выполнять программу - набор команд для обработки данных. Схемы управления записывают команды в память на хранение и извлекают их оттуда одну за другой. После извлечения команды из памяти МП декодирует ее. Схемы управления контролируют процесс выполнения декодированной команды. Поскольку команды хранятся в памяти, можно менять их, изменяя при этом характер обработки данных. Команды, которые записываются в память, определяют будущую работу МП.

Итак, МП выполняет две функции - обработку и управление. Обработка включает в себя перемещение данных с одного места на другое и выполнение операций над ними, управление определяет, как обрабатывать данные. Работа МП состоит из следующих шагов: сначала из памяти извлекается команда, затем команда декодируется, после этого осуществляется выполнение этой команды (рис. 1.1).

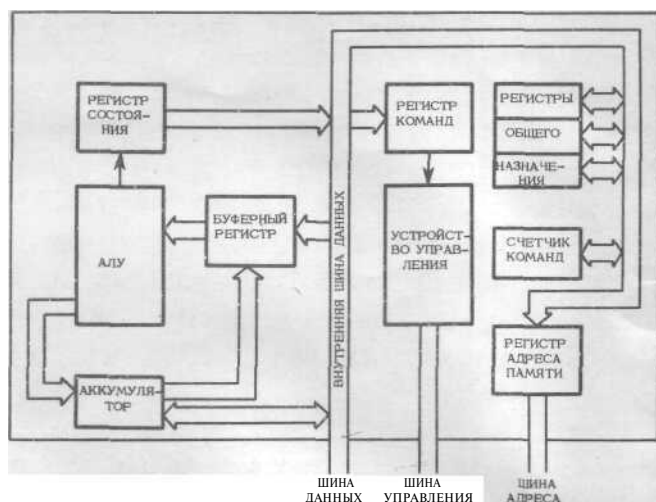


Рис. 1.1. Структурная схема МП

МИКРОПРОЦЕССОР И ВНЕШНИЕ УСТРОЙСТВА

Помимо извлечения команд из памяти и их выполнения схемы управления выполняют ряд других важных функций, например, обмен информацией с внешними устройствами, подсоединенными к МП.

МП можно сравнить с диспетчером на железнодорожной станции. К диспетчеру постоянно поступает информация о приближающихся и проходящих поездах, прибывающих на станцию и ждущих отправления. Диспетчер хранит необходимую ему информацию в своей собственной памяти, на бумаге, содержащей график движения поездов, и на табло, отображающем текущую информацию о перемещении поездов. МП тоже не обойтись без памяти различных типов.

Первый тип - это ПЗУ - постоянное запоминающее устройство. МП может только прочитать то, что записано в ПЗУ, изменить содержимое ПЗУ он не может, подобно тому, как диспетчер не может внести каких-либо изменений в график движения поездов, другой тип памяти - это ОЗУ - оперативное запоминающее устройство. МП может как считывать информацию из ОЗУ, так и записывать туда новые данные. ОЗУ подобно стенду-табло у диспетчера. И, наконец, у МП есть своя внутренняя память, называемая регистрами, подобная собственной памяти диспетчера. Данные, хранящиеся в этой регистровой памяти, МП может обрабатывать быстрее, чем данные, записанные, например, в ПЗУ.

Мы не назвали еще одно устройство, которое находится в диспетчерской. Это пульт управления. Именно отсюда диспетчер получает сообщения о движении поездов и передает указания

о приеме поезда или его отправлении. МП без пульта управления - это только устройство обработки данных, хранящихся в памяти. Чтобы МП работал еще устройством управления, ему нужны так называемые порты ввода-вывода. Через порты ввода МП получает информацию об объекте управления, затем он обрабатывает полученные данные в соответствии с программой, хранящейся в памяти, и снова через порты - теперь уже порты вывода - передает информацию на объект управления для соответствующего изменения его состояния. Таким образом, порты ввода-вывода являются связующим звеном между МП и внешними устройствами (управляемыми объектами).

Итак, МП является основой многих устройств. Отдельный МП не представляет собой законченное, самостоятельно функционирующее устройство. МП является центральной частью микро-ЭВМ, которая представляет собой законченную вычислительную систему, содержащую, кроме МП, память и схемы, реализующие операции ввода-вывода. На рис. 1.2 приведена структурная схема микро-ЭВМ.

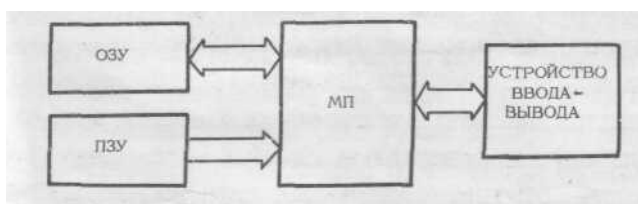


Рис. 1.2. Структурная схема типичной микро-ЭВМ

ХАРАКТЕРИСТИКИ МИКРОПРОЦЕССОРА

Способность МП обрабатывать данные принято оценивать тремя основными характеристиками: длиной слова данных, количеством адресуемых слов памяти и скоростью выполнения команд. Наиболее часто МП сравниваются по длине слова данных. Каждый МП оперирует данными, представленными словами фиксированной

длины, ибо в этом случае существенно упрощается построение МП. Широко используются слова длиной 4, 8, 16, 32 бит (разрядов). 8-битовое слово используется столь часто, что даже получило специальное название - байт.

Другой «мерой мощности» МП является количество слов памяти, к которым он может адресоваться. Каждому слову в памяти присваивается номер его местоположения - адрес. Для извлечения слова из памяти ЭВМ обращается по соответствующему адресу. Адреса памяти начинаются с нуля и представляются в двоичной форме. ЭВМ разного типа имеют различные значения максимального адреса памяти. Чем больше значение максимального адреса памяти, тем больше вычислительная мощность МП.

Типичный размер памяти 8-разрядных МП 64 К байт. Более современные 16-разрядные МП оперируют адресными пространствами до нескольких мегабайт.

Третьей характеристикой МП, используемой для оценки его мощности, является скорость, с которой он выполняет команды. Она оценивается продолжительностью во времени цикла "выборка - выполнение" для одного шага программы и зависит от максимальной частоты генератора тактовых импульсов.

Каждый МП способен понимать и выполнять только точно определенный для него набор команд. При составлении программы программист ограничен этими командами. Количество команд изменяется в зависимости от типа МП и характеризует его возможности и назначение. Таким образом, система команд также является важной характеристикой МП.

МП позволяет заменить логику последовательностью команд программы, которые хранятся в памяти. Поэтому их применение сводит к минимуму логическое проектирование, заменяя его разработкой программного обеспечения.

УРОК 2

ОСОБЕННОСТИ МП КМ1810ВМ86. ВОЗМОЖНОСТИ „МИКРОЛАБ 1810“

Для нужд вычислительной техники выпускаются разнообразные МП общего назначения, обеспечиваемые схемами памяти, контроллерами ввода-вывода и другими специализированными БИС.

Целесообразность применения в каждом конкретном случае того или иного МП оценивается по следующим признакам:

1. Средства программиро-

в а н и я - размер слова (данные/команда); емкость памяти; время выполнения команд; количество регистров; набор операций; режимы обращения и некоторые другие.

2. А п п а р а т н ы е с р е д с т в а - синхронизация; требуемые напряжения; рассеиваемые мощности; размеры корпусов; полнота набора БИС.

3. К о н ф и г у р а ц и я с и с т е м ы , которая влияет на согласование между аппаратной и программной частями, - организация прерываний; микропрограммируемость; количество кристаллов ЦП и т.д.

Наиболее широко используется 8-разрядный МП КР580ИК80.

Для изучения программных и аппаратных средств этого МП, методов диагностики и поиска неисправностей микропроцессорных систем нашей промышленностью выпускается портативная микро-ЭВМ на базе МП КР580ИК80 с встроенным блоком питания - „Микролаб КР580ИК80 908". Она используется для подготовки специалистов в области микропроцессорной техники на предприятиях, в вузах и техникумах для практических и лабораторных работ (рис. 2.1).



Ряс. 2.1. Микролаб „КР580ИК80 908'

Современные требования к средствам вычислительной техники обусловили необходимость разработки микропроцессорного комплекта, ориентированного на решение сложных задач в реальном масштабе времени, обладающего повышенным быстродействием, увеличенной длиной обрабатываемых слов, и в то же время архитектурно и программно совместимого с комплектом БИС серии КР580. Таким комплектом стал базовый микропроцессорный комплект серии К1810.

ОТЛИЧИТЕЛЬНЫЕ ОСОБЕННОСТИ

МП КМ1810ВМ86

Однокристалльный высокопроизводительный 16-разрядный МП КМ1810ВМ86, наряду с программной и аппаратной совместимостью с периферийными и контроллерными БИС серии КР580, обладает большим объемом адресуемой памяти и устройств ввода-вывода (объем адресуемой памяти 1М байт, число адресуемых устройств ввода-вывода 64К), возможностью обработки различных типов данных (биты, байты, слова, строки), разнообразными способами адресации памяти (16 основных типов адресации), гибкой и мощной системой команд (135 базовых команд), развитой структурой прерываний (до 256 типов прерываний), наличием программных и аппаратных средств для работы в многопроцессорных системах.

МП КМ1810ВМ86 реализован по традиционной архитектуре, при которой осуществляется последовательная выборка информации из памяти программ и памяти данных. Среди структурных решений прежде всего следует выделить:

- развитую регистровую структуру (14 внутренних 16-разрядных регистров), что существенно уменьшает количество обращений к памяти;

- конвейерный принцип выполнения команд с предварительной выборкой, обеспечивающий максимальную пропускную способность канала;

- распределенное аппаратно-микропрограммное устройство управления;

- обобщенный 16-разрядный канал адреса данных с разделением во времени;

- многофункциональное использование выводов схемы, позволяющее при существующих конструктивных ограничениях адаптировать МП к уровню сложности разрабатываемой системы.

МП КМ1810ВМ86 обладает возможностями как 8-, так и 16-разрядных МП, он реализует набор 8-разрядных команд МП КР580ИК80 и мощный новый набор 16-разрядных команд, в состав которого входят 16-разрядные арифметические операции, операции над 8- и 16-разрядными числами со знаком (включая умножение и деление), операции над байтовыми последовательностями с возможностью прерывания и усовершенствованная обработка битовых данных.

Одной из отличительных особенностей данного МП, позволяющей частично устранить противоречие между стремлением к увеличению функций МП и ограничением числа выводов корпуса СБИС, является возможность аппаратной перестройки

внутренней схемы управления и синхронизации. МП КМ1810ВМ86 может работать в одном из двух режимов (минимальном или максимальном) путем подключения вывода 33 (MN/MX) либо к источнику питания +5, либо к шине «Корпус». МП, работающий в минимальном режиме, используется в системах, имеющих несложную конфигурацию. При этом он сам вырабатывает все необходимые сигналы управления периферийными устройствами. Если вывод MN/MX подключается к корпусу, то МП переопределяет выводы 24-31 для работы в максимальном режиме (применяется в составе систем сложной конфигурации). При этом он вырабатывает сигналы СТО#, ST1#, ST2#, в которых закодирована информация о выполняемом ЦПУ цикле канала. Системный контроллер декодирует эти сигналы и вырабатывает все управляющие сигналы.

МП КМ181ВМ36 способен координировать взаимодействие нескольких МП, что упрощает построение на его базе мультипроцессорных систем. В этих системах применяются два типа МП: независимые, выполняющие собственный поток команд, и вспомогательные (сопроцессоры). Сопроцессор просматривает команды, выбираемые главным МП, признает некоторые из них «своими» и выполняет их. Микросхему КМ1810ВМ86 можно использовать в качестве независимого МП, а в роли сопроцессоров могут выступать специальные МП ввода-вывода, обработки данных и др.

МИКРО-ЭВМ „МИКРОЛАБ 1810“

Для изучения элементов аппаратного и программного обеспечения устройств, построенных на базе МП КМ1810ВМ86, предлагается микро-ЭВМ мМикролаб 1810" (рис. 2.2), которая, кроме того, может быть использована для разработки



Рис. 2.2. "Микролаб1810"

и сервисного обслуживания устройств со встроенными микро-ЭВМ на базе МП КМ1810ВМ86, а также для подготовки и обучения специалистов в области микропроцессорной техники.

Принцип работы «Микролаб 1810» заключается в том, что программы пользователя, записанные в машинных кодах МП КМ1810ВМ86, загружаются во внутреннюю память микро-ЭВМ, хранятся там и исполняются под управлением программы монитора в автоматическом режиме на рабочей частоте и в шаговом режиме, позволяющем проводить коррекцию программы в ходе ее выполнения на любом шаге.

В качестве центрального МП в микро-ЭВМ используется МП КМ1810ВМ86.

Память "Микролаб 1810" состоит из 8 микросхем ОЗУ общей емкостью 4К байт и 4 микросхем ПЗУ общей емкостью 8К байт.

Наряду с МП и элементами памяти микро-ЭВМ содержит вспомогательные схемы, реализующие те функции, выполнение которых МП невозможно или неэффективно. В современных микропроцессорных системах роль вспомогательных схем так велика, что популярность какого-либо комплекта микропроцессорных БИС зависит не только от технических данных процессорного элемента, но и от состава комплекта в целом. Для организации работы микропроцессорной системы требуется ряд дополнительных компонентов, управляющих работой процессорного элемента и формирующих управляющие и информационные сигналы. К таким компонентам относятся генераторы тактовых синхроимпульсов, системные контроллеры, организующие взаимодействие МП с шинами системы и работу системы в целом, дешифраторы сигналов адресных шин и т.д. Кроме того, в комплект микропроцессорных БИС входят схемы, подключаемые к шинам микропроцессорных систем в качестве периферийных. Их выбор определяется функциональным назначением системы.

Периферийные схемы могут быть выполнены как в виде ИС среднего уровня интеграции, так и в виде БИС. уровень интеграции которых может быть выше, чем у процессорных элементов комплекта.

В „Микролаб 1810“ используется микропроцессорный комплект серии 580. Так, работа клавиатуры осуществляется с помощью контроллера клавиатуры КР580ВВ79, в качестве портов ввода-вывода используются БИС КР580ВВ55А и КР580ВВ51А.

Одной из существенных особенностей перифе-

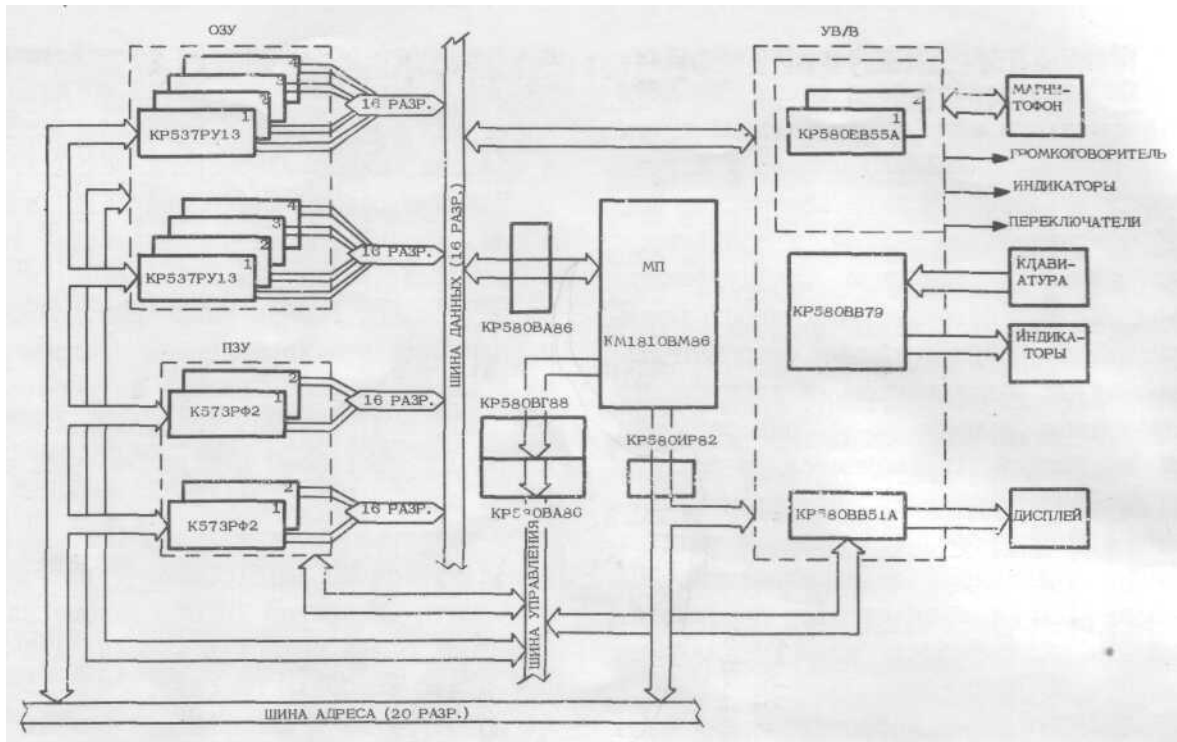


Рис. 2.3. Структурная схема „Микролаб 1810“

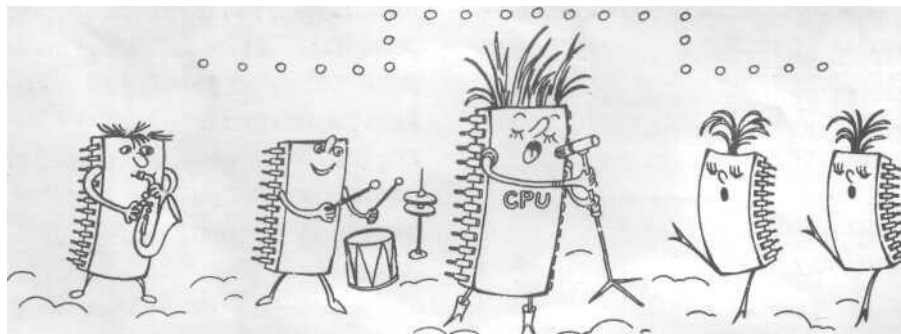
рийных схем является их программируемость. Режим их работы определяется управляющей информацией, которая в них заносится МП во время выполнения программы. Такие схемы позволяют программно варьировать структуру микропроцессорного устройства без каких-либо схемотехнических изменений.

В «Микролаб 1810» предусмотрена возможность подключения цветного графического дисплея и стандартной клавиатуры, а также накопителя на магнитной ленте (например, бытового магнито-

фона). При наличии дисплея, клавиатуры и внешнего накопителя «Микролаб 1810» можно использовать в качестве персональной ЭВМ небольшой мощности.

На рис. 2.3 приведена структурная схема "Микролаб 1810".

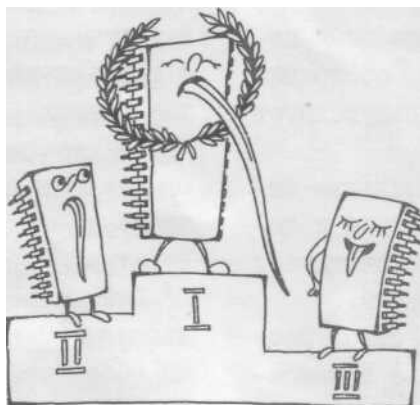
В следующих главах дано описание аппаратных и программных средств микро-ЭВМ „Микролаб 1810“. Описываются общие принципы построения устройств с использованием МП КМ1810ВМ86.



АППАРАТНЫЕ СРЕДСТВА „МИКРОЛАБ 1810’

УРОК 3

КАНАЛЫ СВЯЗИ



Как уже отмечалось, каждая ЭВМ содержит следующие основные функциональные блоки: МП, память и устройства ввода-вывода. Чтобы ЭВМ могла нормально функционировать между МП и памятью или устройствами ввода-вывода обязательно наличие канала связи. В вычислительной системе информация передается с помощью электрических сигналов, а канал связи представляет собой набор проводников, который называется шиной (рис. 3.1).

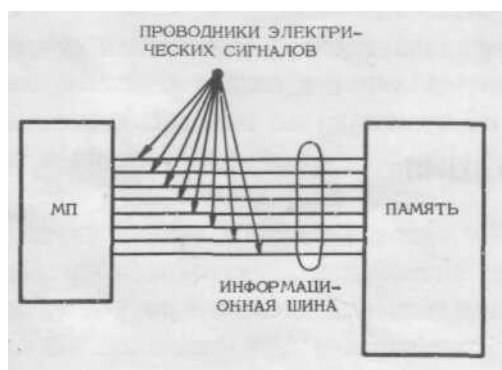


Рис. 3.1. 8-разрядная информационная шина

Память ЭВМ разбита на отдельные элементы - ячейки. Чтобы прочитать данные из ячейки, МП должен подключить к ней свой канал данных. Как переключить шину данных с одной ячейки на другую?

С помощью адреса. Для передачи адреса так-

же необходимо несколько проводников, которые составляют шину адреса. Чтобы вычислительная машина «знала», что ей нужно делать в каждый момент времени, нужны сигналы, управляющие работой. Для передачи этих управляющих сигналов к МП подключена еще одна шина - **шина управления**.

Итак, объединение функциональных блоков в ЭВМ осуществляется посредством следующей системы шин: шины данных, по которой осуществляется обмен информацией между блоками ЭВМ, шины адреса, используемой для передачи адресов, по которым осуществляется обращение к различным устройствам ЭВМ, и шины управления для передачи управляющих сигналов.

ШИНА МИКРОПРОЦЕССОРА КМ1810ВМ86

МП КМ1810ВМ86 связан с внешними устройствами через 20-разрядную мультиплексированную шину адреса (данных, состояния), а также линии управления, по которым передаются сигналы «Чтение», «Запись» и т.д. При работе шины на нее выдается, например, адрес ячейки памяти или устройства ввода-вывода, затем сигнал характера обращения («Чтение», «Запись»), а затем данные поступают в выбранное устройство. В конце цикла принимающее устройство фиксирует данные, а выдающее устройство снимает их.

Каждый цикл шины состоит, как минимум, из четырех машинных тактов (т). В такте Т1 МП выставляет адреса на 20-разрядную шину адреса. В такте т2 МП убирает адреса с шины и либо переводит младшие 16 разрядов шины в третье состояние, если выполняется цикл «Чтение», либо выставляет на шину данные для записи. Если выполняется цикл «Запись» данных во внешнее устройство, то выдача данных из МП происходит в тактах Т2-Т4. Данные для ввода в МП в цикле «Чтение» появляются в такте т3, а в такте Т4 сбрасываются.

В такте Т2 четыре старшие разряда шины переключаются с адреса (А19-А16) на состояние цикла канала и на этих выходах присутствует информация о состоянии (S6-S3) •

В такте Т3 МП поддерживает стабильную информацию на четырех старших линиях шины и либо выдает данные для записи, либо считывает входные данные с 16 младших линий шины.

Если выбранная память или устройство ввода-вывода не могут передавать данные с максимальной скоростью пересылки МП, то они должны послать МП сигнал «Не готов» и заставить его ввести после такта Т3 дополнительные такты (такты ожидания TW). Сигнал «Не готов» должен быть подан на МП в начале такта Т3. Состояние шин в такте ожидания то же, что и в такте т3. Когда выбранное устройство завершает передачу, оно выдает сигнал „Готов” и позволяет МП выйти из состояния ожидания. МП «защелкивает» данные на шине во время последнего такта ожидания или во время такта т3, если состояние ожидания не потребовалось.

Цикл канала завершается в такте Т4 (команды переходят в неактивное состояние, а выбранное внешнее устройство отключается от шины).

ШИНА „МИКРОЛАБ 1810”

В микро-ЭВМ „Микролаб 1810” линии адреса и данных демультиплексированы. «Микролаб 1810» имеет 3 шины: шина адреса - 20 разрядов; шина данных - 16 разрядов; шина управления - 8 разрядов.

Шина данных «Микролаб 1810» - двунаправленная 16-разрядная шина 3 состояний. МП, ОЗУ и порты ввода-вывода являются и приемниками, и передатчиками данных, ПЗУ - только передатчик данных.

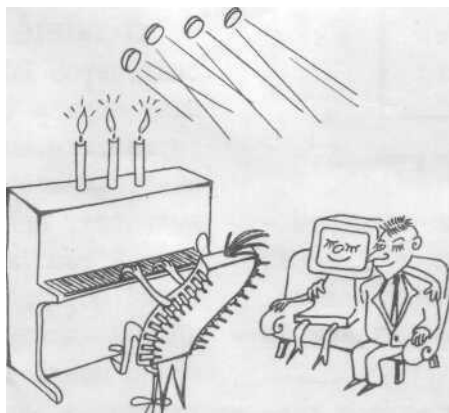
МП координирует все действия системы, он гарантирует, что в определенный момент времени шину использует только одно устройство.

Шина адреса «Микролаб 1810» - однонаправленная 20-разрядная. С помощью шины адреса МП выбирает определенное устройство, которое в данный момент должно быть соединено с шиной адреса.

Все процессы работы микро-ЭВМ координируются управляющей шиной, по которой передаются управляющие сигналы, вырабатываемые МП. Основное различие между управляющей шиной и шинами адреса (данных) заключается в том, что каждая линия управляющей шины выполняет отдельную функцию, в то время как у шин адреса и данных каждая линия передает одинаковый тип информации.

УРОК 4

КЛАВИАТУРА И ИНДИКАЦИЯ



Для ввода данных в МП и для их вывода используются устройства различного типа. Выбор устройства ввода-вывода зависит от источника информации. Например, если данные предполагается хранить на магнитном носителе, то в качестве устройства ввода-вывода может быть выбран накопитель либо на магнитной ленте, либо на магнитном диске. Для связи человека с ЭВМ, по-видимому, наиболее подходящими устройствами ввода-вывода данных являются клавиатура и дисплей.

Устройства ввода предназначены для приема информации из внешнего мира и передачи ее в МП или память микро-ЭВМ. Через эти устройства загружаются исходные данные программы. Устройства вывода используются для представления информации в наглядном для пользователя виде.

В микро-ЭВМ "Микролаб 1810" к устройствам ввода относятся клавиатура (переключатели S5-S16, S18-S27) и 8-разрядный переключатель S4, а устройствами вывода являются сегментные индикаторы D48-D51, D54-D57, светодиоды 7D3-VD10, динамик. Кроме того, с помощью двух микросхем программируемого периферийного интерфейса KP580BB55A (D38, D39) могут быть организованы и другие порты ввода-вывода.

В данном уроке в качестве основного средства диалога пользователя с микро-ЭВМ "Микролаб 1810" рассматриваются клавиатура и индикаторы. С помощью клавиатуры и индикаторов можно выполнять следующие действия:

- опрашивать и изменять содержимое регистров МП КМ1810ВМ86;
- опрашивать и изменять содержимое памяти;
- вводить и исполнять собственные программы;
- оценивать исполнение (отлаживать) программы с помощью шагового режима и точек прерывания;
- перемещать выбранные блоки памяти из одного места в другое;
- считывать данные из портов ввода;
- записывать данные в порты вывода.

Индикаторы используются в качестве устройства вывода и предназначены для представления информации в наглядном для пользователя виде. В зависимости от состояния, в котором находится МП КМ1810ВМ86, информация, высвечиваемая на индикаторах, будет означать:

- текущее содержимое регистра МП или ячейки памяти;
- „отзыв" на ввод шестнадцатеричного числа;
- дополнительные знаки (подсказки пользователю);

сообщение о состоянии МП.

Индикаторы разделены на две группы по 4 индикатора в каждой. Левую группу индикаторов будем называть "поле адреса", правую - „поле данных". Все числа, отображаемые на индикаторах, представлены в шестнадцатеричной системе счисления (см. табл. 4.1).

Т а б л и ц а 4.1

Шестнадцатеричное число	Изображение
0	0
1	1
2	2
3	3
4	4
5	5
6	6
7	7
8	8
9	9
A	A
B	B
C	C
D	D
E	E
F	F

Взаимодействие пользователя с микро-ЭВМ "Микролаб 1810" происходит под управлением программы монитора клавиатуры. Эта программа записана в ПЗУ емкостью 4К байт в адресном пространстве FF000H-FFFFFFH. Программа вызывается всякий раз при включении питания или после нажатия клавиши «Сброс».

С помощью клавиатуры в "Микролаб 1810" вводятся управляющие слова (команды) и данные. Как показано на рис. 4.1, клавиатура разделена

СБРОС	ПРЕР.	C /IP	D /FL	E	F
+	-	8 ВВ С/СS	9 ВВВ С/DS	A /SS	B /ES
:	РЕГ.	4 ВВ Б/SP	5 ВВВ Б/VP	6 ПРС/SI	7 СЛ/DI
,	.	0 БТ/АХ	1 /ВХ	2 ПУСК/СХ	3 ШАГ/DX

Рис. 4.1. Клавиатура „Микролаб 1810"

на на две логические группы: 8 функциональных клавиш слева и 16 шестнадцатеричных клавиш справа.

Шестнадцатеричные клавиши, кроме ввода числовых значений, используются и для других целей. Это отмечено в надписях на каждой клавише. Надписи под шестнадцатеричными числами обозначают команды монитора и имена регистров МП КМ1810ВМ86. Обозначение слева от косой черты - название команды монитора, справа - имя регистра МП. Функция шестнадцатеричной клавиши в каждый отдельный момент времени зависит от состояния „Микролаб 1810“.

НАЗНАЧЕНИЕ ФУНКЦИОНАЛЬНЫХ КЛАВИШ

Клавиша "Сброс" позволяет пользователю завершить любую текущую операцию МП и вернуть „Микролаб 1810“ в состояние инициализации. После нажатия этой клавиши на 8-сегментных индикаторах появляются сообщение "86" в двух младших разрядах поля адреса и символ ввода команды „-“ в старшем разряде поля адреса, а в младших разрядах поля данных - номер программной версии, т.е. „Микролаб 1810“ готов к приему команды.

Клавиша „Прер“ используется для немедленной генерации немаскируемого прерывания.

Клавиша „+“ дает возможность пользователю выполнять сложение двух шестнадцатеричных чисел.

Клавиша „-“ позволяет вычитать одно шестнадцатеричное число из другого.

Клавиша „:“ используется для введения адреса по частям.

Клавиша „Per“ позволяет использовать содержимое любого из регистров МП в качестве входного адреса или данных.

Клавиша „,“ используется для разделения вводов с клавиатуры и приращения содержимого шля адреса при просмотре ячеек памяти.

Клавиша „.“ - клавиша окончания ввода команды. После ее нажатия введенная команда монитора начинает выполняться.

При работе с „Микролаб 1810“ на индикаторы выводятся дополнительные знаки (подсказки), с помощью которых пользователю сообщается о том, какую информацию необходимо ввести в данный момент. Так, если ожидается ввод команды, то знак команды „-“ появляется в старшем разряде поля адреса. Нажатие одной из "командных" клавиш („0“-„9“) при отображении знака команды «-“ интерпретируется как

ввод команды. После нажатия клавиши знак команда исчезает и появляется символ „.“ в младшем индикаторе поля адреса, который сигнализирует о том, что последующий ввод будет направлен в поле адреса. Дальнейшая работа микро-ЭВМ с этой точкой определяется введенной командой. При ожидании ввода данных знак подсказки „.“ появляется в младшем индикаторе поля данных.

При нажатии клавиши „Сброс“ монитор немедленно прекращает текущую работу и переходит к программе инициализации. На индикаторах появляется сообщение монитора „86“ (в двух младших индикаторах поля адреса) и номер программной версии (в двух младших индикаторах поля данных), а также знак ожидания команды „-“ (в старшем индикаторе поля адреса). При этом регистры МП КМ1810ВМ86 устанавливаются монитором в соответствии со значениями, указанными в табл. 4.2.

Т а б л и ц а

Регистр	Значение
CS (сегмент кода)	0H
DS (сегмент данных)	0H
ES (сегмент дополнительный)	0H
SS (сегмент стека)	0H
IP (регистр адреса)	0H
PL (регистр признаков)	0H
SP (указатель стека)	0100H

Программа инициализации устанавливает векторы прерываний с 1 по 3 следующим образом:

прерывание 1 (шаг) - клавиша "Шаг";

прерывание 2 (немаскируемое прерывание) - клавиша „Прер“;

прерывание 3 (точка прерывания) - клавиша „Пуск“.

При любом прерывании в стек пользователя записывается содержимое всех регистров МП. Стек, зарезервированный за пользователем, имеет глубину 48 байт (ячейки 0000H-000FFH), из которых 26 байт должны быть оставлены для хранения содержимого регистров МП, если произойдет одно из указанных выше прерываний.

Первые 256 ячеек ОЗУ (00000H-000FFH) резервируются за программой монитора и областью стека пользователя (рис. 4.2). Первая ячейка ОЗУ, имеющаяся в распоряжении пользователя. 00100H.

Монитор всегда вычисляет 20-битный физи-

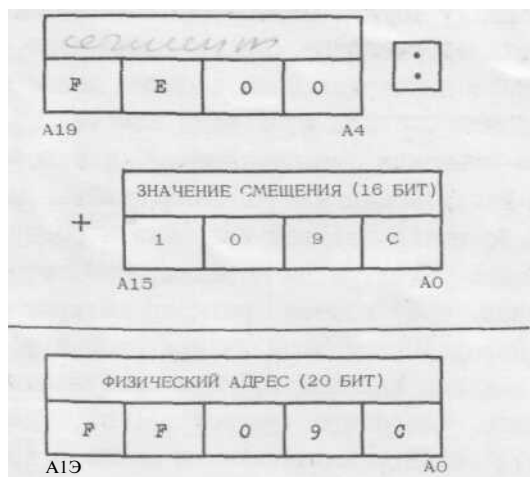


Рис. 4.2. Формирование физического адреса памяти

ческий адрес памяти из 16-битного значения адреса сегмента и 16-битного значения дополнительного адреса (смещения). Значение адреса сегмента вводится первым, затем нажимается клавиша „:” (для отделения второго ввода) и после этого вводится значение смещения. Два введенных адреса суммируются (см. рис. 4.2) для формирования 20-битного физического адреса памяти. Если вводится только одно адресное значение (двоеточие пропущено), то это интерпретируется монитором как ввод смещения и текущее содержимое регистра сегмента CS используется в качестве значения адреса сегмента. Содержимое регистра CS и введенное значение смещения суммируются, как показано на рис.4.3, для формирования 20-битного адреса памяти.

ВВОД 3В (В РЕГИСТРЕ CS-0010H)

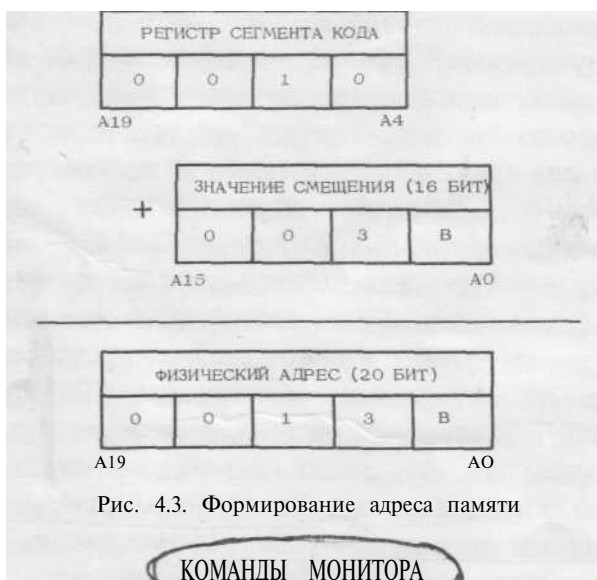


Рис. 4.3. Формирование адреса памяти

КОМАНДЫ МОНИТОРА

Команды „Байт” и „Слово”

Команды „Байт” и „Слово” (клавиши „БТ” и

„СЛ”) используются для просмотра содержимого ячеек памяти. Если ячейка памяти может быть модифицирована (например, ячейка в ОЗУ), то ее содержимое может быть изменено.

Для пользования командой „Байт” или „Слово” нажимают клавишу „БТ” или „СЛ” соответственно во время нахождения монитора в состоянии ожидания ввода команды (при отображении в левом индикаторе поля адреса символа ввода команды „-”). После этого в крайнем правом индикаторе адресного поля высветится символ „.” (остальные индикаторы будут погашены), указывающая на то, что последующий ввод с клавиатуры будет направлен в поле адреса. Затем пользователю необходимо с помощью шестнадцатеричных клавиш ввести адрес ячейки памяти для просмотра (ввод начинать со старшего символа).

Все адреса памяти включают значение сегмента и значение смещения. Если значение сегмента не указывается, оно определяется содержимым регистра CS: первый адресный ввод является значением сегмента, а двоеточие (:) вводится в качестве разделителя, второй адрес является значением смещения.

Разрядность поля адреса ограничена 4 индикаторами и поэтому, если вводится больше, чем 4 шестнадцатеричных числа в качестве значения сегмента или смещения, действительны только последние четыре (которые отображаются на индикаторах).

После того, как адрес введен, нажимают клавишу „.”. Байт (или слово данных), содержащийся в адресуемой ячейке памяти, будет отображаться в поле данных и символ „.” переместится в крайний правый индикатор поля данных, указывая, что последующий ввод будет направлен в поле данных. Когда используется команда „Слово” байт, содержащийся в ячейке памяти, отображаемой в адресном поле, появляется в двух младших разрядах поля данных, а байт, содержащийся в следующей ячейке памяти (Адрес +1), появится в двух старших разрядах поля данных.

Если содержимое адресуемой ячейки просмотрено, нажимают клавишу „.” для завершения команды или клавишу „,” для просмотра следующей ячейки памяти (команда „Байт”) или двух следующих ячеек памяти (команда „Слово”). Для изменения содержимого адресуемой ячейки памяти записывают новые данные с помощью шестнадцатеричных клавиш. Отображаемые данные не за-

писываются в память до тех пор, пока клавиша «.» или клавиша «,» не будет нажата. Если нажата клавиша «.», то команда заканчивается и символ "-" отображается в адресном поле. При нажатии клавиши «,» следующий адрес и данные, содержащиеся в адресной ячейке («БТ») или ячейках памяти («СЛ»), высвечиваются на индикаторах.

Состояние ошибки фиксируется при попытке модифицировать несуществующие ячейки или ячейки ПЗУ. Ошибка не выявляется до тех пор, пока не нажата клавиша „.” или «,». После выявления ошибки на индикаторах поля адреса появляется сообщение «ERR» (ошибка) и символ ввода команды.

Команда „Регистр”.

Команда «Регистр» используется для просмотра и изменения содержимого любого из регистров МПКМ18ЮВМ86.

Для просмотра содержимого регистра необходимо нажать клавишу «Per» при наличии на индикаторе символа ввода команды „-”. После нажатия клавиши в крайнем правом индикаторе адресного поля загорится символ „-”. Последующий ввод будет интерпретироваться как имя регистра (надпись справа от косой черты на клавише), а не как шестнадцатеричное число. После нажатия клавиши с именем регистра название соответствующего регистра будет отображаться в поле адреса, а 16-битное содержимое регистра высветится в поле данных и в крайнем правом индикаторе поля данных загорится символ «.”. В это время содержимое регистра может быть изменено, т.е. регистр «открыт» для ввода.

Т а б л и ц а |

Имя регистра	Обозначение
Аккумулятор	АХ
База	ВХ
Счетчик	СХ
Данные	ПХ
Указатель стека	SP
Указатель базы	BP
Индекс источника	SI
Индекс приемника	DI
Сегмент кода	CS
Сегмент данных	DS
Сегмент стека	SS
Сегмент дополнительный	ES
Регистр адреса команды	IP
Регистр признаков	PL

Значения, вводимые из шестнадцатеричного поля клавиатуры, будут отражаться в поле данных, и содержимое регистра будет изменено в соответствии с высвечиваемыми данными после нажатия клавиши «.» или «,». Если нажата клавиша „.”, то команда

символ ввода команды „-”. Если нажата клавиша „,», то высвечивается название и содержимое следующего регистра (в порядке, соответствующем табл. 4.3) и этот регистр открыт для ввода. Порядок регистров не круговой и поэтому после нажатия клавиши „,» при отображении содержимого флагового регистра (FL), команда «Регистр» будет закончена и монитор возвратится к ожиданию ввода команды.

Команды «Ввод байта» и «Ввод слова».

Команды "Ввод байта" и «Ввод слова» (клавиши "ВВБ" и "ВВС") используются для ввода (приема) байта или слова из порта ввода.

Для использования команды «Ввод байта» или «Ввод слова» нажимают клавишу "ВВБ" или „ВВС" соответственно при отображении в левом индикаторе поля адреса символа ввода команды «-”. После нажатия клавиши "ВВБ" или «ВВС" в крайнем правом индикаторе поля адреса загорится сигнал „.”, показывая, что запрашивается ввод адреса порта. Используя шестнадцатеричные клавиши, вводят адрес порта, из которого будет проводиться считывание. Адресация ввода-вывода ограничена 64К байтами (максимальный адрес FFFFH), значение сегмента для формирования адреса порта не используется.) После введения адреса порта нажимают клавишу «,». Байт или слово, вводимые из адресуемого порта, будут высвечиваться в поле данных. Повторное нажатие клавиши „,» изменит содержимое поля данных на слово или байт, вводимые из адресуемого порта. Нажатие клавиши „.” завершает исполнение команды, и монитор устанавливается в состояние ожидания следующей команды.

„Микролаб 1810" содержит две БИС КР580ВВ55А, которые могут быть использованы в соответствии с командами «Ввод байта» и «Ввод слова» для ввода данных из периферийных устройств. Этим БИС присвоены обозначения «ПОРТ1" и «ПОРТ2". Каждая микросхема, в свою очередь, содержит три отдельных 8-разрядных порта А, в и с. При работе с байтами каждый порт функционирует независимо. Во время работы со словом пара портов функционирует совместно (например, P1A и P2A) для формирования 16-разрядного слова данных (порт P2 соответ-

вует младшему байту). В табл. 4.4 приведены адреса отдельных портов.

Т а б л и ц а 4.4

Порт	Адрес
P1A	FFF8
P1B	FFFA
P1C	FFFC
P1 _{упр}	FFFE
P2A	FFF9
P2B	FFFB
P2C	FFFD
P2 _{упр}	FFFF

Во время работы со словами вводится только адрес младшего порта P2 (соответствующий старший порт адресуется автоматически).

Микросхемы параллельных портов ввода-вывода программируются на ввод при включении питания или при нажатии клавиши "Сброс". Если порт был запрограммирован на вывод, то нажимают клавишу „Сброс" перед нажатием клавиш команды или записывают соответствующее значение байта или слова в порт управления для программирования микросхем на ввод.

Команды "Вывод байта" и "Вывод слова"

Команды "Вывод байта" и „Вывод слова" (клавиши „ВЫВБ" и "ВЫВС") используются для вывода (передачи) байта или слова данных в порт вывода.

Для использования команды нажимают соответствующую шестнадцатеричную клавишу при указании монитора на ввод символом «-» в левом индикаторе поля адреса. После нажатия клавиши в правом индикаторе адресного поля загорается символ „.", индицируя, что необходим ввод адреса порта. Адресация ввода-вывода ограничена 64К байтами, и поэтому значения сегмента, как и в командах «Ввод байта" и "Ввод слова", не требуется. После того, как введен адрес порта, нажимают клавишу „.". В правом индикаторе поля данных загорается символ ".", указывая на то, что можно вводить выходной байт или слово данных. Используя шестнадцатеричную клавиатуру, записывают байт или слово, предназначенные для вывода. После введения данных нажимают клавишу "." для вывода байта (слова) в порт и завершения команды или клавишу ";", если дополнительные данные будут выводиться в адресуемый порт.

"Вывод байта" и "Вывод слова" могут быть

использованы

дельного ввода-вывода КР580ВВ55А на ввод или вывод, а также для вывода данных в любой порт БИС КР580ВВ55А программируются на ввод при включении питания или при сбросе и поэтому, прежде чем данные могут быть выведены через эти микросхемы, последние должны быть пере-программированы на вывод (записью соответствующего байта или слова в управляющий порт микросхем) .

Команда "Пуск" используется для передачи управления микро-ЭВМ от клавиатуры к программе пользователя, записанной в памяти.

Для вызова команды нажимают клавишу «Пуск" при указании монитором на ожидание командного ввода («-"). После этого в адресном поле высвечивается текущее содержимое регистра счетчика команд, в поле данных - содержимое ячейки памяти, адресуемой регистром IP, и в самом правом индикаторе адресного поля загорится символ ".", индицируя, что может быть осуществлен ввод нового стартового адреса. Если новый адрес необходим, то его вводят с помощью клавиатуры (при вводе адреса поле данных пусто). Для начала исполнения программы (с текущего или вновь введенного адреса программы) нажимают клавишу «.". После нажатия клавиши монитор высвечивает символ "Е" в старшем разряде поля адреса, а затем передает управление программе пользователя.

Для выхода из программы пользователя и возвращения управления монитору необходимо нажать клавишу "Сброс" или "Прер". Если нажимается клавиша "Сброс", то происходит инициализация микро-ЭВМ и регистры МП устанавливаются в значения, указанные в табл. 4.2. При нажатии клавиши "Прер" вызывается монитор, содержимое всех регистров МП сохраняется, на индикаторы выводится знак команды "-", и монитор переходит к ожиданию ввода новой команды. Если после этого вводится команда "Пуск", то на индикаторах высвечивается текущее содержимое регистра IP (с м е щ е н и е адреса команды, которая должна была быть выполнена перед нажатием клавиши "Прер") и байт, содержащийся в ячейке памяти, адресуемой регистрами IP и CS. Нажатие клавиши "." передает управление от монитора к программе пользователя по адресу, формируемому из содержимого регистров IP и SC, и выполнение программы продолжается.

В команде "Пуск" можно указать точку прерывания. Адрес точки прерывания аналогичен по своему действию нажатию клавиши „Прер" при выполнении программы. Для того, чтобы указать адрес точки прерывания необходимо после ввода стартового адреса нажать клавишу ";", а затем ввести значение адреса точки прерывания. Следует отметить, что при задании адреса точки прерывания используется значение сегмента, определенное при вводе стартового адреса, или, если оно было не задано, текущее содержимое регистра сс. После нажатия клавиши "." монитор сохраняет команду, находящуюся в ячейке с адресом точки прерывания, записывает по этому адресу команду прерывания, а затем передает управление программе пользователя. После того, как программа достигает точки прерывания управление передается монитору, команда точки прерывания возвращается в программу пользователя, все регистры МП сохраняются и монитор переходит в состояние ожидания команды (знак команды "-" загорится в левом индикаторе поля адреса). В двух старших индикаторах поля данных появится сообщение "BR" (прерывание).

Необходимо помнить, что команда в точке прерывания восстанавливается после того, как управление возвращается монитору. Поэтому адрес точки прерывания должен определяться каждый раз, когда программу необходимо выполнить с заданием точки прерывания.

Состояние ошибки возникает при попытке установить точку прерывания в ПЗУ.

Команда "Пересылка" позволяет перемещать блок данных в памяти. Для вызова команды "Пересылка" нажимают клавишу „ПРС" при указании монитором на ввод команда. После нажатия клавиши в поле адреса появляются три точки, индицируя, что необходимо ввести три адреса. Каждый раз после ввода адреса левая точка исчезает, так что количество точек, горящих в любой момент времени, указывает на необходимое еще число вводов. Адреса вводятся в таком порядке :

I - стартовый адрес памяти перемещаемого блока данных;

2 - конечный адрес памяти перемещаемого блока данных;

3 - начальный адрес памяти, куда перемещается блок (адрес места назначения).

Значение сегмента не используется при вводе конечного адреса и поэтому перемещаемый блок ограничен 64К байтами.

После нажатия клавиши „." данные пересылаются и высвечивается символ ввода команды „.-". При перемещении блока данных, данные, содержащиеся в блоке-источнике, не изменяются (за исключением случаев, когда адрес назначения попадает внутрь исходного блока данных и частично совпадающие ячейки памяти будут заполнены пересылаемыми данными).

Так как данные пересылаются побайтно, команда «Пересылка» может быть использована для заполнения памяти каким-либо значением. Это достигается заданием адреса назначения на единицу больше стартового адреса. После выполнения команда олок ячеек памяти будет заполнен значением, содержащимся в ячейке со стартовым адресом.

Частично совпадающие ячейки памяти будут заполнены пересылаемыми данными).

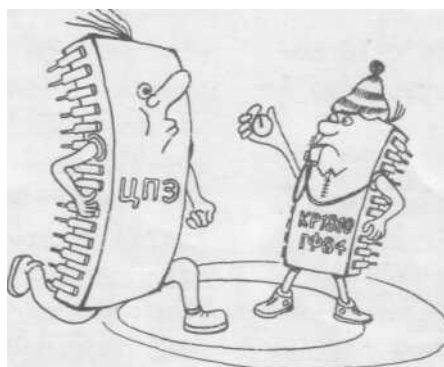
Состояние ошибки индицируется при попытке переслать данные в ПЗУ или несуществующие ячейки памяти.

Команда „Шаг".

Команда "Шаг" предназначена для индивидуального пополнения инструкций МП, записанных в памяти. После выполнения каждой инструкции управление из программы передается монитору.

Для использования команды «Шаг» нажимают клавишу "Шаг" при высвечивании в поле адреса символа ввода команды „.-". Если необходим стартовый адрес, отличный от адреса, высвечиваемого на индикаторах, то с помощью клавиатуры вводится новый адрес. После нажатия клавиши „." выполняется адресуемая инструкция и в адресном поле высвечивается адрес следующей инструкции, которую надо выполнить, а в поле данных появится соответствующий командный байт. Повторное нажатие клавиши ";" приведет к выполнению текущей инструкции и переведет программу для выполнения следующей инструкции.

ТАКТОВЫЙ ГЕНЕРАТОР



В микропроцессорной системе чрезвычайно важна роль схем управления, обеспечивающих последовательность функционирования всех звеньев системы.

Одной из важнейших входных линий управления является линия связи с генератором тактовых импульсов, синхронизирующим во времени работу МП.

Действия центрального МП цикличны. МП выбирает команду, выполняет требуемые операции, выбирает следующую команду и т.д. Регулярная последовательность событий, подобная указанной, требует синхронизации, поэтому центральному МП требуется тактовый генератор.

В „Микролаб 1810“ тактовый генератор выполнен на БИС КР1810ГФ84. Эта микросхема представляет собой биполярный генератор синхронимпульсов для МП КМ1810ВМ86, КМ1810ВМ88 и периферийных устройств. Времязадающим элементом является кварцевый резонатор, который имеет собственную частоту генерации 14,5476 МГц. БИС КР1810ГФ84 делит эту частоту на 3, вырабатывая на выходе CLK (вывод 8 D1) тактовые сигналы с частотой 4,9 МГц, а затем еще на 2, получая частоту 2,45 МГц на выходе сигнала PCLK (вывод 2 D1). Выходной сигнал CLK является тактовым сигналом для МП, выходной сигнал PCLK используется всеми остальными микросхемами.

Генератор также вырабатывает два выходных управляющих сигнала, которые выдаются синхронно частоте 4,9 МГц сигнала CLK: READY (вывод 5D1) - „Готовность“ и RESET (вывод 10 D1) - „Сброс“.

Сигнал RESET используется для установления микро-ЭВМ в состояние инициализации и вырабатывается, когда вход сигнала RES# (вывод 11 D1)

переходит в состояние лог. 0 (при включении источника питания или при нажатии клавиши „Сброс“).

Сигнал готовности READY активен, если на входе RDYI (вывод 4 D1) действует уровень лог. 1. Этот уровень обеспечивается генератором состояний ожидания.

Генератор состояний ожидания (микросхемы D4.3, D6, D7.1, D7.3) позволяет вводить состояния ожидания в цикл работы МП при работе с „медленными“ периферийными устройствами ввода-вывода или схемами памяти, которые могут быть дополнительно подключены к микро-ЭВМ. Кроме того, состояния ожидания необходимы при работе микро-ЭВМ на частоте 4,76 МГц (состояний ожидания не требуется при работе „Микролаб 1810“ на частоте 2,38 МГц).

Управление состояниями ожидания осуществляется переключателем S3.

Для разрешения работы генератора состояний ожидания необходимо установить контакт 8 переключателя S3 в положение „разомкнуто“. Выбор числа состояний ожидания осуществляется контактами 1-7 переключателя S3. Контакт 1 соответствует 1 состоянию ожидания, контакту 2 - 2 состояния, контакту 3 - 3 состояния, ..., контакту 7 - 7 состояний ожидания. Для установки необходимого числа состояний ожидания соответствующий контакт переводят в положение „замкнуто“. Для правильного выбора числа состояний ожидания только один из 7 контактов должен находиться в замкнутом положении, остальные должны быть разомкнуты.

Генерация состояний ожидания осуществляется микросхемой D6 (КМ555ИР8 - 8-разрядный регистр сдвига с параллельным выходом).

При установке контакта 8 переключателя S3

в положение «замкнуто» работа генератора состояний ожидания запрещается. В этом случае состояния ожидания отсутствуют.

ОПИСАНИЕ БИС КР1810ГФ84

БИС КР1810ГФ84 помещена в корпус с 18 выводами. Назначение выводов БИС приведено в табл. 5.1 и на рис. 5.1).

Таблица 5.1

Вывод	Назначение
1	Вход сигнала CSYNC - синхронимпульс
2	Выход сигнала PCLK - периферийный синхронимпульс (с частотой, равной 1/2 частоты CLK)
3	Вход сигнала AEN1 - разрешение адреса
4	Вход сигнала RDY1 - готовность шины
5	Выход сигнала READY
6	Вход сигнала RDY2 - готовность шины
7	Вход сигнала AEN2 - разрешение адреса
8	Выход сигнала CLK - синхронимпульс МП
9	Общий (\perp) вход
10	Выход сигнала RESET
11	Вход сигнала RES - сброс
12	Выход сигнала OSC кварцевого генератора
13	Вход сигнала F/E - выбор генератора синхронимпульса
14	Вход сигнала EPI - входная частота (в 3 раза больше частоты сигнала CLK)
15	Не соединяется
16	Вход сигнала кварцевого генератора X2
17	Вход сигнала кварцевого генератора X1
18	Напряжение питания ($U_n = 5 В$)

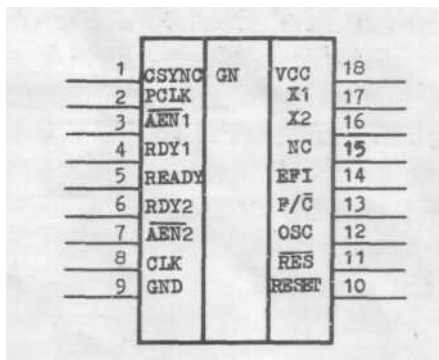


Рис. 5.1. Назначение выводов БИС КР1810ГФ84

Структурная схема БИС приведена на рис.5.2.

AEN1# и AEN2# - входы сигнала разрешения адреса, служат для получения соответствующих сигналов готовности шины RDY1 и RDY2. Сигналы AEN# используются в таких конфигурациях, где МП может выбирать 2 системные шины (Multi-Master, System Busses). В противном случае входы AEN1# и AEN2# заземляются. Активный уровень сигналов - лог. 0.

RDY1 и RDY2 - входы сигнала готовности шины, поступающего от устройства, подсоединенного к шине; извещает о том, что передача закончена или возможна. Активный уровень сигналов - лог. 1.

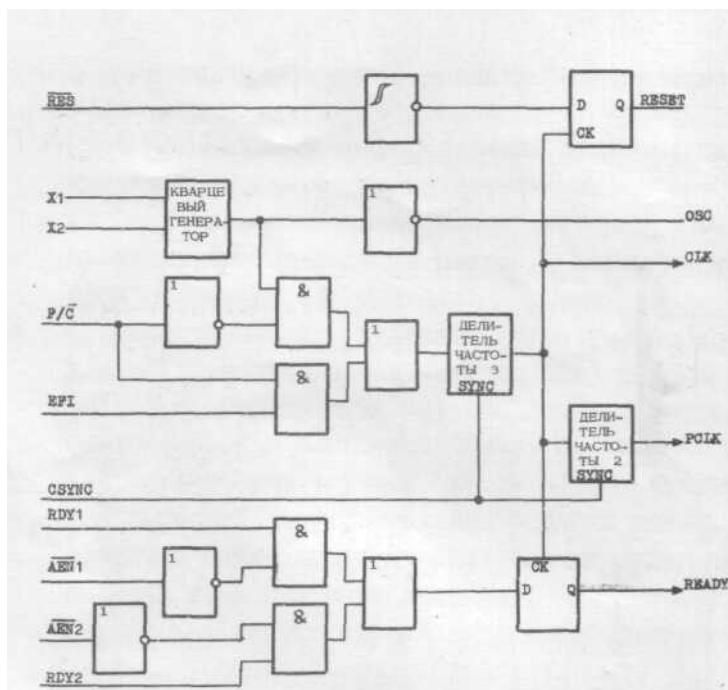


Рис. 5.2. Структурная схема БИС КР1810ГФ84

READY - выход сигнала готовности шины, формируется из входных сигналов RDY. Так как сигналы RDY появляются асинхронно по отношению к синхросигналу CLK, для получения выходного сигнала READY сигналы RDY нужно синхронизировать. Сигнал READY снимается через некоторое гарантированное время удержания. Активный уровень сигнала - лог. 1.

X1 и X2 - входы кварцевого генератора, к которым присоединяются выходы кристалла кварца, частота импульсов которого в 3 раза больше частоты, необходимой для МП.

F/C# - вход сигнала выбора источника синхроимпульса для МП. Если на входе F/C# сигнал уровня лог. 0, то разрешается генерация синхроимпульса от кристалла кварца, а если на входе F/C# сигнал уровня лог. 1, то синхроимпульс для МП поступает по входу EPI.

EPI - вход синхросигнала для МП, частота которого должна быть в 3 раза больше частоты синхроимпульса CLK.

CLK - выход синхроимпульса, используемый МП и внешними устройствами, которые непосредственно соединены с процессорной локальной шиной.

PCLK - выход периферийного синхроимпульса, частота которых в 2 раза меньше частоты сигнала CLK.

OSC - выход сигнала кварцевого генератора, частота которого равна частоте импульсов с кристалла кварца.

RES - вход сигнала сброса, используется для генерации выходного сигнала RESET. Сигнал

RES# поступает на вход триггера Шмидта, который используется для установления длительности импульса сброса, вырабатываемого при включении литания. Активный уровень сигнала - лог. 0.

RESET - выход сигнала сброса. МП, его временные характеристики определяются входом сигнала RES#. Активный уровень сигнала - лог.1.

CSYNC - вход синхросигналов, используется для синхронизации БИС КР1810ГФ84 (если используется несколько таких микросхем). Сигнал CSYNC должен быть синхронизован внешним сигналом EFI. При использовании кварца для генерации частоты вход сигнала CSYNC должен быть заземлен.

ФУНКЦИИ БИС КР1810ГФ84

Кварцевый генератор БИС КР1810ГФ84 через входы X1 и X2 подключен к внешнему кристаллу кварца и используется для генерации основной рабочей частоты для МП. Выходной сигнал кварцевого резонатора буферизуется и выводится на выход озс, так что другие временные сигналы для системы могут быть получены от этого стабильного источника.

Генератор синхроимпульса состоит из синхронного счетчика (делителя на 3) со специальным входом сигнала очистки, который запрещает счет. Сигнал очистки (CSYNC) разрешает синхронизацию выходного синхроимпульса с внешним синхроимпульсом. Сигнал CSYNG необходимо синхронизировать с сигналом EFI. Это достигается с помощью двух триггеров Шотки (рис. 5.3). Частота на выходе счетчика-делителя составляет 1/3 частоты входного сигнала.

Логика сброса обеспечивает генерацию временной диаграммы сброса. Сигнал сброса синхрс-

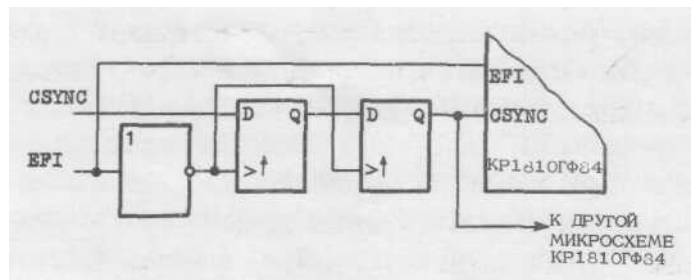


Рис. 5.3. Схема генератора синхроимпульсов

уизуруется с отрицательным фронтом сигнала CLK.

Сигнал READY формируется из 2 входных сигналов RDY1 И RDY2, которые ИСПОЛЬЗУЮТСЯ ДЛЯ обслуживания 2 системных шин. Каждый из этих входов имеет квалификатор (соответственно AEN1# и AEN 2#). Каждый входной сигнал RDY должен быть синхронизирован, чтобы гарантировать необходимое время установки и удержания его. Синхронизация может быть достигнута путем установки Д-триггера между асинхронным источником сигнала RDY и генератором КР1810ГФ84 и синхронизации его положительным фронтом сигнала CLK.

Временная диаграмма работы БИС КР1810ГФ84 приведена на рис. 5.4.

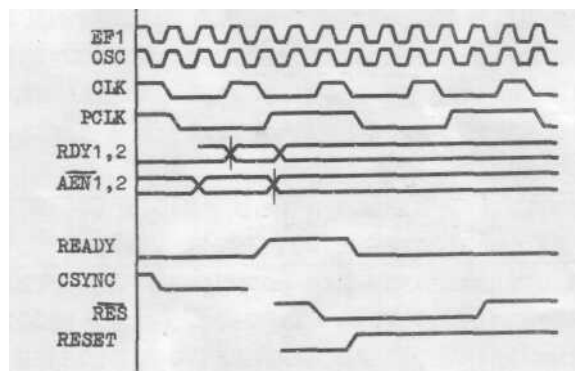
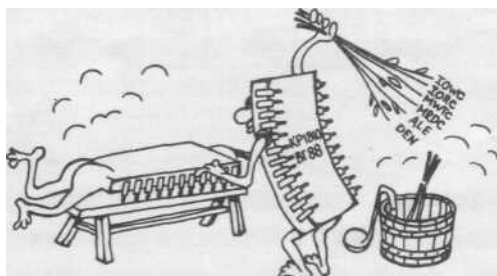


Рис. 5.4. Временная диаграмма работы БИС КР1810ГФ84

УРОК 6

СИСТЕМНЫЙ КОНТРОЛЛЕР



Контроллерканала

Выбор режима функционирования в МП КМ1810ВМ86 предоставляет пользователю уникальную возможность выбора комбинации выходных управляющих сигналов в соответствии со степенью сложности микропроцессорной системы.

В минимальном режиме, ориентированном на малые вычислительные система, МП выдает сиг-

налы управления каналом, необходимые для функционирования в системе ЗУ и УВВ. В максимальном режиме МП кодирует выходные управляющие сигналы так, что 8 сигналов оказываются закодированными в 3 сигнала состояния. В системах максимальной конфигурации предполагается применение контроллера канала, который декодирует сигналы состояния $STO\#-ST2\#$, поступающие из МП, и выдают расширенный набор сигналов управления для остальной части системы. А МП использует освобожденные 5 выводов для новых сигналов, необходимые для координации совместной работы с другими МП мультипроцессорной системы.

Если контроллеру канала необходим доступ к каналу, то он подает сигнал запроса на МП. Контроллер канала должен обеспечить все выходные управляющие сигналы вместо МП. Он использует код состояния цикла канала для генерирования всех выходных сигналов управления, необходимых для синхронизации цикла канала. Сигналы $STO\#-ST2\#$ идентифицируют тип цикла канала, который начинает выполнять МП.

В „Микролаб 1810“ МП может работать как в минимальном, так и в максимальном режимах; выбор осуществляется установкой переключки X2 в положение, соответствующее необходимому режиму. (Переключку X2 следует переставлять только тогда, когда микро-ЭВМ находится в выключенном состоянии.)

Контроллер канала в "Микролаб 1810" выполнен на микросхемах D3, D4.1, D4.2, D8, D10, D11 и предназначен для формирования сигналов управления работой микро-ЭВМ. При работе в максимальном режиме сигналы о состоянии поступают на системный контроллер КР1810ВГ88 (микросхема D3), который вырабатывает следующие управляющие сигналы: MRDC (чтение памяти), NWTC (запись в памяти), IORC (чтение ввода-вывода), IOWC (запись ввода-вывода), INTA (подтверждение прерывания), ALE (разрешение фиксации адреса), DT/R (выдача-прием данных), DEN (разрешение передачи данных). Все перечисленные сигналы поступают на формирователь D10.

В минимальном режиме сигналы INTA#, ALE, DT/R, DEN формируется МП, а сигналы MRDC#, NWTC#, IORC#, IOWC# формируются на вентилях D4.1 и D8 из сигналов M/IO# (память - ввод-вывод), RD (чтение), WR# (запись), вырабатываемых МП. Затем эти сигналы подаются на формирователь D11.

В зависимости от выбранного режима (минимального или максимального) разрешается работа только одного формирователя, сигналы с которого управляют работой микро-ЭВМ. Выходы другого в это время находятся в третьем состоянии.

ОПИСАНИЕ МИКРОСХЕМЫ КР1810ВГ88

Микросхема КР1810ВГ88 - биполярный контроллер шины, используемый в микропроцессорных системах на базе МП КР1810ВМ86. Контроллер декодирует информацию, заданную сигналами состояния МП, и вырабатывает необходимые управляющие сигналы.

Микросхема КР180ВГ88 помещена в корпус с 20 выводами. Назначение выводов микросхемы приведено в табл. 6.1 и на рис. 6.1.

Т а б л и ц а 6.1

Вывод	Назначение
1	Вход сигнала I/OB - режим входной-выходной шины
2	Вход сигнала CLK - синхроимпульс
3	Вход сигнала $S1$ - состояние
4	Выход сигнала DT/R - прием-передача данных
5	Выход сигнала ALE - разрешение адреса
6	Вход сигнала AEN - разрешение адреса
7	Выход сигнала $MRDC$ - чтение памяти
8	Выход сигнала $AMWC$ - расширенная команда записи памяти
9	Выход сигнала $MWTC$ - команда записи памяти
10	Общий вход
11	Выход сигнала $IOWC$ - команда записи ввода-вывода
12	Выход сигнала $AIOWC$ - расширенная команда записи ввода-вывода
13	Выход сигнала $IORC$ - команда чтения ввода-вывода
14	Выход сигнала $INTA$ - подтверждение прерывания
15	Вход сигнала CEN - разрешение команды
16	Выход сигнала DEN - разрешение данных
17	Выход сигнала $MCE/PDEN$ - разрешение основной функции или периферийных данных
18	Вход сигнала состояния $S2$
19	Вход сигнала состояния $S0$
20	Напряжение питания ($U_n = 5 В$)

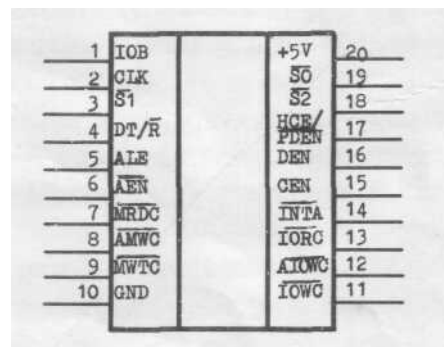


Рис. 6.1. Назначение выводов МП КР1810ВГ88

so, S1, S2 - входные сигналы состояния МП. Микросхема КР1810ВГ88 дешифрирует эти сигналы для того, чтобы генерировать команды и управ-

ляющие сигналы в соответствующие моменты времени.

CLK - синхроимпульс с тактового генератора KP1810GF84. Служит для определения моментов генерирования команд и управляющих сигналов.

ALE - сигнал разрешения адресного регистра. Служит для стробирования адреса в адресных регистрах. Активный уровень сигнала - лог. 1, фиксирование происходит на спаде (переход с уровня лог. 1 на уровень лог. 0).

DEN - сигнал разрешения данных. Служит для разрешения передачи данных по внутренней или системной шине данных. Активный уровень сигнала - лог. 1.

DT/R# - сигнал приема-передачи данных. Определяет направление потока данных. Уровень лог. 1 означает передачу (запись ввода-вывода или памяти), а уровень лог. 0 означает прием (считывание) данных.

AEN# - сигнал разрешения адреса. Разрешает выходы команд через 105 нс после того, как он станет активным (уровень лог. 0). Если сигнал AEN# переходит в неактивное состояние, то выходы команд немедленно переходят в третье состояние.

CEN - сигнал разрешения команды. Уровень лог. 0 означает, что все выходы команд KP1810BG88 и управляющие выходы сигналов DEN и PDEN переводятся в неактивное состояние; уровень лог. 1 - разрешает эти выходы.

IOB - сигнал режима входной-выходной шины. Уровень лог. 1 этого сигнала означает, что KP1810BG88 функционирует в режиме шины ввода-вывода, уровень лог. 0 означает, что микросхема работает в режиме системной шины.

AIOWC# - расширенная команда записи ввода-вывода. Выдает команду записи ввода-вывода в начале машинного цикла, чтобы предупредить устройство ввода-вывода о команде записи. Активный уровень сигнала - лог. 0.

ioWC# - команда записи ввода-вывода, по которой устройство ввода-вывода считывает данные с шины данных. Активный уровень сигнала - лог. 0.

IORC# - команда чтения ввода-вывода, по которой устройство ввода-вывода передает данные по шине данных. Активный уровень сигнала - лог. 0.

AMWC# - расширенная команда записи памяти. В начале машинного цикла предупреждает ЗУ о команде записи. Активный уровень сигнала - лог. 0.

MWTC# - команда записи памяти, по которой в память записываются данные, находящиеся на шине данных. Активный уровень сигнала - лог. 0.

MRDC# - команда считывания памяти, по которой данные из памяти передаются на шину данных. Активный уровень сигнала - лог. 0.

INTA# - сигнал подтверждения прерывания. Информировать прерывающее устройство о том, что оно должно передавать информацию на шину данных. Активный уровень сигнала - лог. 0.

MCE/PDEN# - вывод, имеющий двойную функцию:

MCE(10В на уровне лог. 0) - сигнал разрешения считывания адреса основного контроллера приоритета прерывания во время цикла подтверждения прерывания. Активный уровень сигнала - лог. 1.

PDEN# - сигнал разрешения передачи данных на периферийную шину данных во время команды ввода-вывода. Он выполняет ту же функцию для шины ввода-вывода, которую сигнал DEN выполняет для системной **ШИНЫ**. Активный уровень сигнала - лог. 0.

ЛОГИКА КОМАНД И УПРАВЛЕНИЯ

Логика команд дешифрирует 3 шины состояния $\overline{S_0}$, S1#, S2# МП, чтобы определить - какая команда должна выдаваться. В табл. 6.2 показаны значения каждого слова-состояния.

Таблица 6.2

$\overline{S_0}$, $\overline{S_1}$, $\overline{S_2}$	Состояние МП	Команды KP1810BG88
0 0 0	Разрешение прерывания	\overline{INTA}
0 0 1	Чтение ввода-вывода	\overline{IORC}
0 1 0	Запись ввода-вывода	\overline{IOWC} , \overline{AIOWC}
0 1 1	Останов	-
1 0 0	Выбор кода	\overline{MRDC}
1 0 1	Чтение памяти	\overline{MRDC}
1 1 0	Запись памяти	\overline{MWTC} , \overline{AMWTC}
1 1 1	Пассивное состояние	-

Команда выдается одним из 2 способов в зависимости от режима контрольной шины KP1810BG88.

Режим шины ввода-вывода. Микросхема KP1810BG88 находится в режиме шины ввода-вывода, если вывод 10В находится в состоянии лог. 1. В режиме шины ввода-вывода все выходы команд ввода-вывода (IORC#, IOWC#, AIOWC#, INTA#) всегда разрешены (т.е. не зависят от AEN#). Когда команда ввода-вывода инициируется, то микросхема KP1810BG88 немедленно возбуждаетли-

нии команд сигналами PDEN# и DT/R#, чтобы управлять передачей по шине ввода-вывода. Линии команд ввода-вывода не должны использоваться для управления системной шиной. Этот режим позволяет одному контроллеру KP1810BG88 управлять двумя внешними шинами. Когда МП необходимо получить доступ к шине ввода-вывода состояние ожидания отсутствует.

Для обычного доступа к памяти требуется сигнал «Шина готова» (уровень сигнала - лог. 0) до начала работы памяти. Если в многопроцессорной системе имеются ввод-вывод или периферийное устройство, назначенное одному МП, то лучше использовать режим 10В.

Режим системной шины. Микросхема KP1810BG88 находится в режиме системной шины, если выход ЮВ находится в состоянии лог. 0. В этом режиме не выдается ни одна команда до истечения 105 нс после того, как выход AEN# перейдет в состояние лог. 0. В этом режиме принимается, что логика управления шиной будет информировать контроллер шины (по выходу AEN#), когда шина свободна. Команды памяти и ввода-вывода ожидают управления шиной. Этот режим используется, когда имеется только одна шина. В данном случае ввод-вывод и память обслуживаются разными МП. Расширенная команда записи выдается для инициализации процедур записи в начале машинного цикла. Этот сигнал используется, чтобы запретить МП ненужное состояние ожидания (см. рис. 6.2).

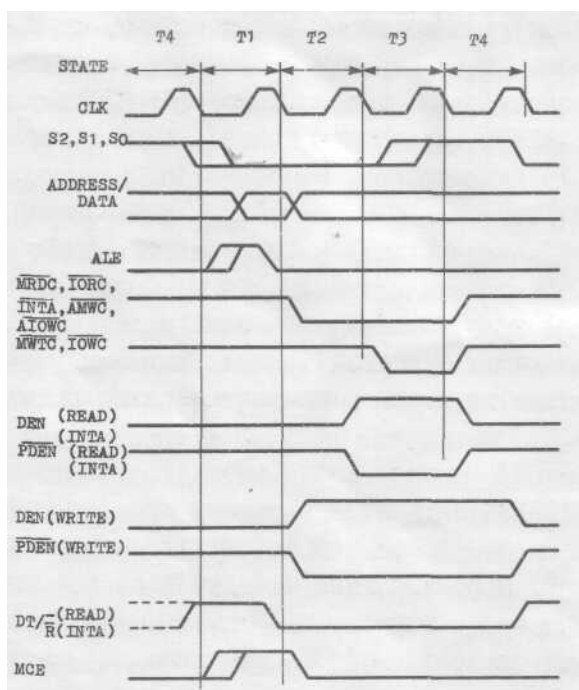


Рис. 6.2. Временная диаграмма работы МП KP1810BG88

Сигнал INTA# (подтверждение прерывания) разрешает чтение ввода-вывода во время цикла прерывания. Его назначением является информирование прерывающего устройства о том, что оно должно передать информацию на шину данных.

Сигнал DEN определяет когда разрешается обмен данными между внешней и внутренней шинами, а сигнал DT/R# определяет направление переноса данных. Эти два сигнала совместно подаются на выводы выбора кристалла и направления передачи приемника.

Выход MCE/PDEN# имеет двойную функцию.

Когда контроллер KP1810BG88 находится в режиме 10В (входной-выходной шины; уровень сигнала 10в - лог. 1), сигнал PDEN# служит как специальный сигнал разрешения данных для ввода-вывода или периферийной системной шины.

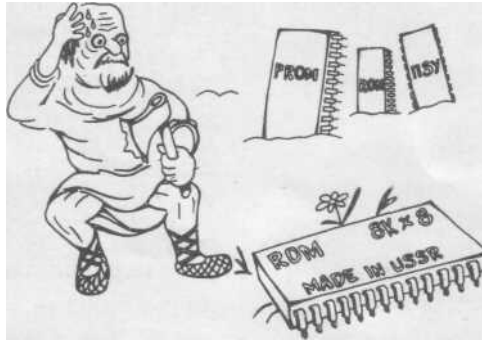
Сигнал MCE используется во время цикла подтверждения, когда контроллер KP1810BG88 находится в режиме системной шины (уровень сигнала 10В - лог. 0). Во время любой последовательности прерываний появляются 2 цикла подтверждения прерывания. Во время первого цикла прерывания не осуществляется передача, ни данных, ни адресов. Должна обеспечиваться логика, чтобы маскировать сигнал MCE во время этого цикла. Непосредственно перед началом второго цикла по сигналу MCE выдается адрес контроллера приоритета прерываний на внутреннюю шину МП, где по сигналу ALE заносится в адресные регистры. По переднему фронту второго цикла прерывания адресуемый контроллер приоритета прерываний выдает вектор прерывания на системную шину данных, где он считывается МП. Если система содержит только один контроллер приоритета прерываний, то сигнал MOE не используется. В этом случае второй сигнал подтверждения прерывания выдает вектор прерывания на системную шину.

Сигнал ALE возникает во время каждого машинного цикла и служит для разрешения передачи текущего адреса в регистр адреса. Сигнал ALE также служит для стробирования состояния (S0#, S1#, S2#) в регистре декодирования состояния останова.

Сигнал SEN действует как квалификатор команд для контроллера KP1810BG88. Если вывод SEN находится в состоянии лог. 1, то микросхема KP1810BG88 функционирует нормально. Если вывод SEN переходит в состояние лог. 0, то все выводы команд остаются в неактивном состоянии (но не в третьем).

УРОК 7

ОЗУ И ПЗУ



Для современных МП характерно наличие единого адресуемого пространства памяти, которое называется основной памятью. МП должен располагать возможностью чтения команд, составляющих определенную программу, и запоминания данных, вырабатываемых этой программой. Команды и данные необходимо хранить и выбирать по мере надобности в процессе работы ЭВМ. Для этой цели в микропроцессорных системах используются интегральные схемы памяти 2 типов - ПЗУ и ОЗУ.

ПЗУ - память, с которой можно только считывать информацию. Информация записывается в ПЗУ заранее до установки ее в схему. Содержимое ПЗУ не изменяется при отключении питания, поэтому ПЗУ используют для хранения постоянной программы и постоянных данных.

ОЗУ - память, в которой информация может храниться и изменяться, но она теряется при отключении питания и невозможно предсказать - какая информация будет записана в ОЗУ после нового включения.

ОЗУ делятся на 2 класса - статические и динамические. В статических ОЗУ для хранения 1 бита информации используется отдельный триггер, и эта информация сохраняется пока обеспечивается питание. В динамических ОЗУ информация хранится в виде электрических зарядов емкости затвор-подложка МОП-транзистора. Эта емкость через несколько миллисекунд разряжается, так что требуется ее периодическая подзарядка (регенерация). Динамические ОЗУ имеют важное значение, поскольку в памяти этого типа используется меньшее количество элементов на бит запоминаемой информации, чем в устройствах статического типа, что позволяет достичь более высокого уровня интеграции

микросхем. Динамическая память обладает также более высоким быстродействием и потребляет меньшую мощность. Однако для реализации цикла регенерации требуется дополнительное, часто внешнее схемное оборудование. Поэтому динамическая память эффективна лишь для ОЗУ относительно большого объема. Память же малого объема обычно реализуют на статических элементах.

При работе с памятью необходимо учитывать время доступа к памяти и время цикла памяти. Оба эти параметра характеризуют быстродействие памяти. Они тесно связаны между собой (чем меньше время доступа к памяти, тем короче время цикла памяти). Время доступа определяется организацией памяти и скоростью работы схем, на которых построена память. Время цикла - наименьший интервал времени, который может иметь место между двумя обращениями к памяти. Он зависит не только от характеристик, которые свойственны памяти, но и от других временных параметров микропроцессорной системы.

Память „Микролаб 1810“ включает в себя ОЗУ общей емкостью 4К байт и ПЗУ общей емкостью 8К байт.

ОЗУ микро-ЭВМ выполнено на 8 БИС КР537РУ13 (микросхемы D26-D33). БИС представляет собой статическое ОЗУ информационной емкостью 4096 бит (4К), с организацией 1024x4 бит. Линии данных являются двунаправленными, т.е. данные могут поступать в память и считываться из памяти. (На рис. 7.1 приведено назначение выводов БИС КР537РУ13). Микросхема ОЗУ имеет два управляющих входа: вход сигнала "Запись-считывание" (WR#/RD) и вход «Выбор микросхемы» (CS#). При записи данных в ОЗУ выбирается адрес (время выборки адреса не более 120 нс),

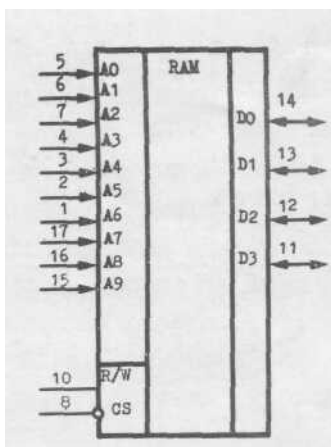


Рис. 7.1. Назначение выводов БИС КР537РУ13

по шине данных поступают данные и на входе "Запись-считывание" устанавливается уровень лог. 0. Если в данный момент и вход „Выбор микросхемы“ также в состоянии лог, 0, то информация записывается в память.

При считывании данных из ОЗУ вход „Выбор микросхемы“ в состоянии лог. 0, а вход „Запись-считывание“ в состоянии лог. 1.

ОЗУ присвоены адреса от 00000H до 00FFH. МП КР18ЮВМ86 обеспечивает одновременное считывание или запись полностью 16 бит (2 расположенных по соседству байта) или считывание (запись) либо старшего, либо младшего байта.

Микросхемы ОЗУ выбираются соответствующими выходами дешифратора ОЗУ (микросхема D36) • Дешифратор ОЗУ разрешается при уровне лог. 0 адресных битов A17-A19, которые поступают на его вход CS2 (вывод 14 микросхемы D36), и уровне лог. 0 на выводе П микросхемы D12.2, указывающем на работу с памятью. Микросхема D36 декодирует сигнал ВНЕ и адресные биты А0, А11-А16. Сигналы ВНЕ и А0 указывают будет ли обращение к слову или байту. Адресные биты А11-А16 определяют адресный блок ОЗУ.

Работа дешифратора поясняется в табл. 7.1.

Таблица 7.1

Входы дешифратора				Выходы дешифратора				Адресный блок
A7-A3	A2	A1	A0	D4	D3	D2	D1	
0	0	0	0	1	1	0	0	Два байта 0H-07FH
0	0	0	1	1	1	0	1	Старший байт 0H-07FH
0	0	1	0	1	1	1	0	Младший байт 0H-07FH
0	1	0	0	0	0	1	1	Два байта 0800H-0FFFH
0	1	0	1	0	1	1	1	Старший байт 0800H-0FFFH
0	1	1	0	1	0	1	1	Младший байт 0800H-0FFFH
Все другие состояния				1	1	1	1	

Уровень лог. 0 на выходе дешифратора разрешает выбор соответствующего байта в ОЗУ. Адресацию микросхем ОЗУ осуществляют адресные биты А1-А10, а управление (чтение или запись) – сигналы MWTC# и MRDC#.

Первые 256 байт ОЗУ (ячейки с он по OFFH резервируются за программой монитора, пользователю предоставляются в распоряжение ячейки ОЗУ начиная с адреса 0100H.

ПЗУ „Микролаб 1810“ построено на 4 микросхемах К573РФ2 с возможностью ультрафиолетового стирания информации. Бремя выборки адреса не более 450 нс. Информационная емкость 16384 бит, организация 2Кх8. Общая емкость ПЗУ составляет 8К байт, из которых 4К байт (2 микросхемы D22 и D23) содержат программу монитора клавиатуры, а остальные 4К байт (микросхемы D24 и D25) используются для хранения программ монитора дисплея (расположение выводов микросхемы К573РФ2 показано на рис. 7.2).

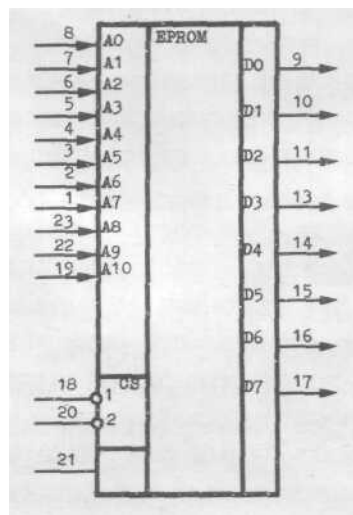


Рис. 7.2. Назначение выводов БИС К573РФ2

В „Микролаб 1810“ предусмотрена возможность увеличения емкости ПЗУ до 12К байт. Для этого на плате установлены 2 адаптера (микросхемы D18 и D19), в которые пользователь может вставлять микросхемы со своими собственными программами.

Управление выбором необходимых микросхем ПЗУ осуществляется дешифратором ПЗУ (микросхема D17) и сигналами ВНЕ и А0 МП. Дешифратор ПЗУ декодирует адресные биты А12-А18 для определения 4К байт (FC000H-FCFFFH, FD000H-FDFFFH, FE000H-FEFFFH или FF000H-FFFFFH), которые будут адресоваться, в то время как сигналы А17 и А19, разрешают вывод старшего (младшего) байта или слова (обеих байтов). Работа дешифратора поясняется в табл. 7.2.

Таблица 7.2

Входы дешифратора				Выходы дешифратора				Адресный блок
A7	A6-A2	A1	A0	D3	D2	D1	D0	
0	1	1	1	1	1	1	0	FFOOOH-FFFFFH
0	1	1	0	1	1	0	1	FE000H-FEFFFFH
0	1	0	1	1	0	1	1	FD000H-FDFFFFH
0	1	0	0	0	1	1	1	FC000H-FCFFFFH
1	1	1	1	1	1	0	1	FFOOOH-FFFFFH
1	1	1	0	1	1	1	0	FE000H-FEFFFFH
1	1	0	1	1	0	1	1	FD000H-FDFFFFH
1	1	0	0	0	1	1	1	FC000H-FCFFFFH
Все другие состояния				1	1	1	1	

По данным табл. 7.2, дешифратор ЕВУ определяет свои выходы D0 и D1 в зависимости от положения переключки X5. В положении „Клав” микросхемы D22 и D23 (в которых хранится программа монитора клавиатуры) отводится адресный блок FFOOOH-FFFFFH, а микросхемам D24, D25 с программой монитора дисплея - блок FE000H-FEFFFFH. При перестановке переключки X5

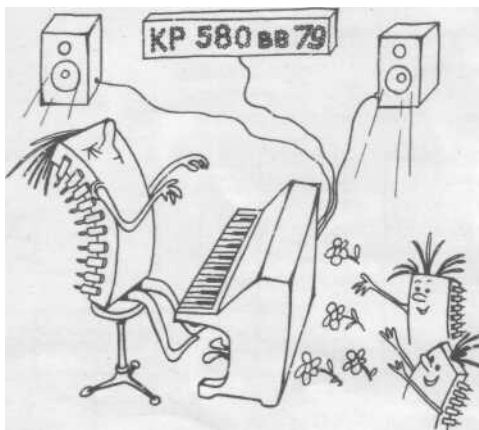
в положение „Дисп” микросхемам D22, D23 присваиваются адреса FFOOOH-FEFFFFH, а микросхемам D24, D25 - FE000H-FFFFFH. Такая лереадресация дает возможность пользователю по своему желанию передавать управление микро-ЭВМ после сброса либо монитору клавиатуры, либо монитору дисплея в зависимости от положения переключки X5.

Дешифратор ПЗУ обеспечивает сигнал выбора блока памяти FC000H-FCFFFFH (вывод 9 микросхемы D17). Этот сигнал выведен на разъем, что дает пользователю возможность добавить блок 4К байт памяти без использования дополнительной логики дешифрации адреса.

В качестве ПЗУ пользователя, устанавливаемых в адаптерах, допускается использование микросхем K573PФ2 или K556PT7. Выбор требуемого типа осуществляется переключкой X4. Одновременная установка микросхем разного типа не допускается.

УРОК 8

КОНТРОЛЛЕР КЛАВИАТУРЫ



В микро-ЭВМ „Микролаб 1810” логика клавиатуры и индикации сосредоточена вокруг контроллера клавиатуры и индикации КР580ВВ79 (микросхема D42). Микросхема представляет собой однокристальное программируемое интерфейсное устройство, предназначенное для ввода информации с клавиатуры и вывода ее на дисплей в системах, построенных на основе 8- и 16-разрядных МП КР580ВМ80 и КМ1810ВМ86.

Микросхема состоит из двух функционально-независимых частей: клавиатурной и дисплейной. Клавиатурная часть обеспечивает сканирование 64-контактной матрицы клавиатуры и матрицы датчиков, а также стробированный ввод.

Для этого имеются 8 выходных линий сканирования и 8 входных линий возврата. Для хранения вводимой информации в микросхеме предусмотрен обратный магазин - ОЗУ датчиков

(ОМ-ОЗУ) емкостью 8 байт. ОМ-ОЗУ работает по принципу «первый вошел - первый вышел». При наличии информации в ОМ-ОЗУ микросхема вырабатывает сигнал запроса прерывания IRQ.

В клавиатурной части микросхемы предусмотрен специальный режим обнаружения ошибок при замыкании двух и более клавиш, а также введена схема устранения дребезга клавиш.

Дисплейная часть микросхемы обеспечивает сканированный интерфейс с 8- и 16-символьными алфавитно-цифровыми дисплеями, представляющими собой светодиодную матрицу, матрицу ламп накаливания и т.д.

Для хранения информации, подлежащей отображению, в микросхеме имеется ОЗУ дисплея объемом 16 байт, которое может быть использовано как два ОЗУ объемом 16x4. Вывод информации на дисплей в виде двоичного кода осуществляется по двум 4-разрядным каналам OUTA0-OUTA3, OUTB0-OUTB3. Информация на дисплей может выводиться двумя способами: слева направо или справа налево со сдвигом.

Программирование режимов работы, запись информации в ОЗУ дисплея, чтение из ОМ-ОЗУ и ОЗУ отображения, чтение внутреннего состояния микросхемы осуществляются по двунаправленной шине данных при подаче соответствующих управляющих сигналов.

ОПИСАНИЕ БИС KP580BV79

На рис. 8.1 приведена структурная схема БИС KP580BV79.

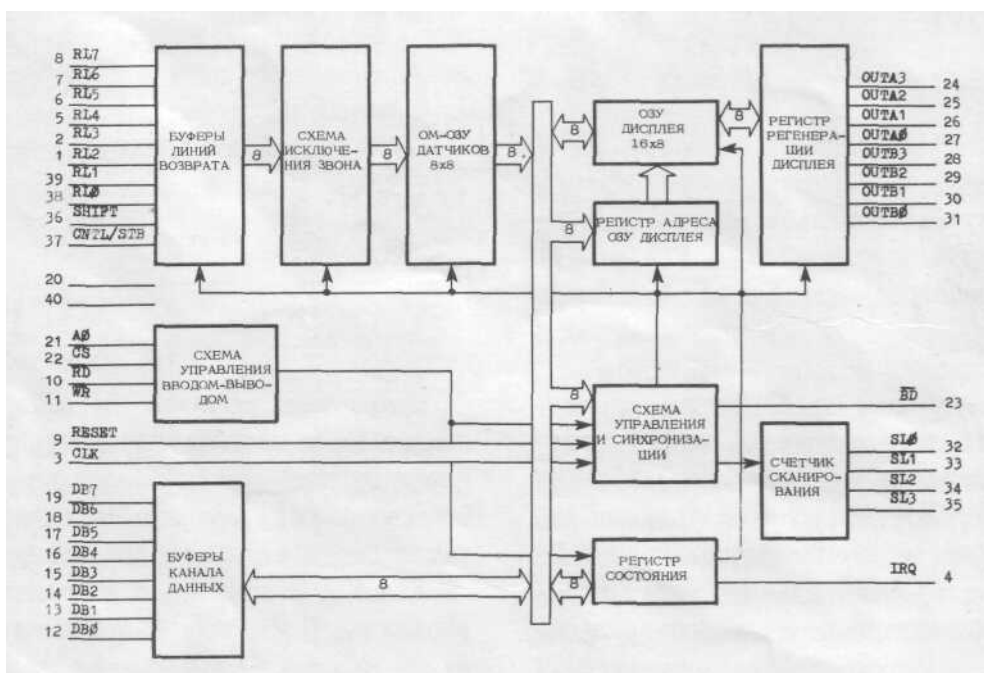


Рис. 8.1. Структурная схема БИС KP380BV79

На рис. 8.2 показано расположение выводов БИС KP580BV79.

ОПИСАНИЕ ФУНКЦИОНАЛЬНЫХ БЛОКОВ КОНТРОЛЛЕРА

Схема управления вводом-выводом и буферы канала данных.

Схема управления вводом-выводом использует входные сигналы \overline{CS} , AO, RD#, WR# для управления обменом информацией с МП, а также внутренними пересылками данных и команд к различным внутренним регистрам и буферам микросхемы. Все пересылки данных и команд разрешаются сигналами

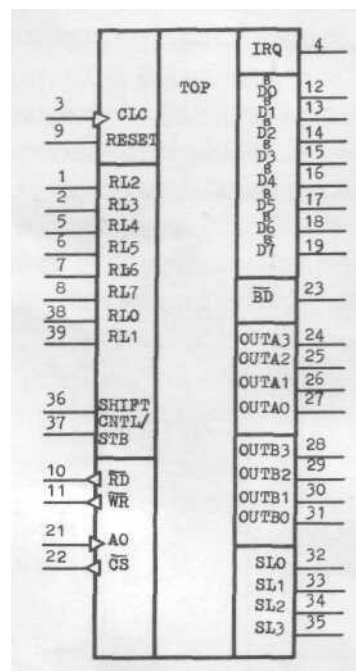


Рис. 8.2. Назначение выводов БИС KP580BV79

ОПИСАНИЕ ВЫВОДОВ

Номер вывода	Обозначение	Вход или выход	Функциональное описание
12-19	D\overline{B}0-D\overline{B}7	Вход(выход)	<u>Двухнаправленная шина данных.</u> Все данные и команды между ЦП к КР580ВВ79 передаются по этим линиям.
3	CLK	Вход	<u>Системный синхроимпульс.</u> Используется для генерации внутренней временной диаграммы.
9	RESET	Вход	<u>Сброс.</u> Уровень лог. 1 на этом входе устанавливает БИС КР580ВВ79 в исходное состояние.
22	\overline{CS}	Вход	<u>Выбор кристалла.</u> Уровень лог. 0 на этом входе разрешает интерфейсные функции приема или передачи.
21	AO	Вход	<u>Адрес буфера.</u> Уровень лог. 1 на этом входе означает, что входные сигналы с шины данных интерпретируются как команды, а выходные – как слово состояния. Уровень лог. 0 означает, что входные и выходные сигналы с шины данных интерпретируются как данные.
10 11	\overline{RD} \overline{WR}	Вход	<u>Чтение (запись).</u> Уровень лог. 0 на этих входах разрешает буферам данных либо принимать, либо посылать данные на внешнюю шину данных
4	IRQ	Выход	<u>Запрос прерывания.</u> В режиме сканирования клавиатуры выходной сигнал находится на уровне лог. 1 при наличии данных в ОМ-ОЗУ. При считывании данных выходной сигнал IRQ переходит на уровень лог. 0 и возвращается к уровню лог. 1 если еще есть информация в ОМ-ОЗУ. В режиме матрицы датчиков выходной сигнал IRQ переходит на уровень лог. 1 при обнаружении изменения в датчиках.
32-35	SIO-SL3	Выход	<u>Линии сканирования.</u> Используются для сканирования клавиатуры или матрицы датчиков, а также дисплея. Информация на этих линиях может быть кодированной или дешифрованной.
38, 39, 1, 2, 5, 6, 7, 8	RIO-RL7	Вход	<u>Линии возврата.</u> Замыкаются с линиями сканирования через клавиши или ключа датчиков. Входы RIO-RL7 имеют активные внутренние сопротивления для поддержки сигналов на уровне лог. 1 до тех пор, пока нажатие клавиши или ключа не переведет их на уровень лог. 0. В режиме стробируемого ввода эти выводы служат в качестве 8-разрядного входа данных.
36	SHIFT	Вход	<u>Входной сигнал в режиме сканирования клавиатуры.</u> Уровень сигнала SHIFT записывается в разряд D6 ОМ-ОЗУ вместе с кодом позиции нажатой клавиши.
37	CNTL/STB	Вход	<u>Входной сигнал в режимах сканирования клавиатуры и стробированного ввода.</u> В режиме сканирования клавиатуры состояние этого сигнала записывается в разряд D7 ОМ-ОЗУ вместе с кодом позиции нажатой клавиши. В режиме стробированного ввода этот вход используется в качестве строба, вводящего в ОМ данные, поступающие по линиям RIO-RL7
27-24 31-28	OUTA0- OUTA3 OUTB0- OUTB3	Выход Выход	<u>Выходы регистров регенерации дисплея.</u> (ОЗУ регенерации дисплея организовано как 2 матрицы объемом 16 регистров по 4 разряда). Для получения мультиплексированного режима работы алфавитно-цифровых дисплеев данные с этих выходов выдаются синхронно с сигналами на линиях сканирования SIO-SL3 . Гашение отображения в двух 4-разрядных каналах вывода может осуществляться независимо. Они могут также рассматриваться и как один 8-разрядный канал.
23	BD	Выход	<u>Гашение дисплея.</u> Сигнал, используемый для гашения отображения во время смены цифр или при поступлении команды гашения изображения.
20	\perp	-	
40	+5 В	-	

Уровень лог. 0 означает, что входные и выходные сигналы с шины данных интерпретируются как данные.

лом CS#. Характер информации, выдаваемой и принимаемой МП, определяется значением сигнала АО. При уровне сигнала АО-лог. I входная информация является командой, а выходная - словом состояния. При уровне лог. O входная и выходная информация является данными. Сигналы RD# и WR# определяют направление пересылки данных через буферы данных. Двухнаправленные буферы канала данных соединяют внутреннюю шину данных БИС с внешней. При подаче сигналов WR#=cs# с уровнем лог. O информация вводится в БИС, а при подаче сигналов RD#=CS# с уровнем лог. O - выводится из БИС. Если уровень сигнала CS#-лог. I, микросхема находится в высокоимпедансном состоянии и обмен информацией через буферы не происходит.

Схема управления и синхронизации.

Схема состоит из регистров хранения команд и счетчика синхронизации.

Регистры хранения команд используются для хранения кодов команд, управляющих работой клавиатурной и дисплейной частей БИС.

Коды команд подаются по шине данных при уровне сигнала АО лог. 1, а сигнала WR# - лог. O. Команда записывается в регистр по положительному фронту сигнала WR#, затем она декодируется, и устанавливается соответствующая функция

(Счетчик синхронизации состоит из предварительного делителя частоты и 10-разрядного двоичного счетчика. Делитель частоты предназначен для выработки базовой частоты синхронизации.

Коэффициент деления предварительного делителя программируется в пределах от 2 до 31. Этот коэффициент должен быть выбран таким образом, чтобы обеспечить базовую частоту синхронизации не более 100 кГц. Базовая частота делится 10-разрядным счетчиком для получения соответствующих периодов сканирования клавиш, ряда матрицы, дисплея. Счетчик устанавливается в исходное состояние программным и аппаратным способами. Схема управления и синхронизации вырабатывает также выходной сигнал ID, используемый для гашения изображения на дисплее во время смены знаков или при поступлении команды „Гашение отображения“.

Счетчик сканирования.

Счетчик сканирования вырабатывает сигналы сканирования клавиатуры, матрицы датчиков и дисплея (SLO-SL3). Счетчик сканирования работает в двух режимах.

1. В режиме кодированного сканирования он обеспечивает выдачу на выходы SLO-SL3 двоичного кода 4 младших разрядов счетчика синхронизации, который необходимо дешифровать внешним дешифратором для получения сигналов сканирования клавиатуры и дисплея.

2. В режиме дешифрованного сканирования он дешифрует внутри БИС 2 младших разряда счетчика синхронизации и обеспечивает выдачу дешифрованных сигналов на выходы SLO-SLO3.

Если запрограммирован режим дешифрованного сканирования клавиатуры, так же будет работать и дисплей. Это значит, что на дисплее воспроизводятся только первые 4 символа в 037 отображения.

Счетчик сканирования устанавливается в исходное состояние как программным, так и аппаратным способом.

Буферы линий возврата, схема исключения звена.

Содержимое 8 линий возврата RLO-RL7 поступает в буферы возврата для дальнейшей обработки.

В режиме клавиатуры линий* RLO-RL7 сканируются, отыскивая нажатую клавишу. Когда найдена нажатая клавиша, схема исключения звена ждет не менее 10 мс, чтобы проверить остается ли еще нажатой клавиша. Если она нажата, то код сканирования вместе с состоянием разрядов SHIFT и CNTL передается в OM-O3Y.

В режиме сканирования матрицы датчиков содержимое линий возврата непосредственно передается в соответствующий регистр OM-O3Y каждый период сканирования.

В режиме стробированного ввода содержимое линий возврата передается в OM-O3Y по положительному фронту импульса STB.

OM-O3Y датчиков и регистр состояния.

OM-O3Y датчиков - двухфункциональное O3Y 8x8 бит. В режимах клавиатуры или стробированного ввода блок работает как OM («первый вошел - первый вышел»), в режиме сканирования матрицы датчиков эта память является O3Y датчиков. Каждый регистр O3Y датчиков загружается соответствующими датчиками в матрице.

Регистр состояния отражает наличие символов в OM-O3Y. При чтении или вводе в OM-O3Y более 8 символов в регистре состояния вырабатываются флаги переопустошения или переполнения.

Чтение регистра состояния происходит при уровне лог. O сигналов RD#=CS# и уровне сигнала АО-ЛОГ. I.

Логика состояния формирует также сигнал IRQ. В режимах клавиатуры и стробированного ввода сигнал IRQ с уровнем лог. 1 вырабатывается, когда он не пуст. В режиме сканирования матрицы датчиков сигнал IRQ с уровнем лог. 1 вырабатывается при обнаружении изменения в состоянии датчиков.

ОЗУ дисплея, регистр адреса ОЗУ дисплея, регистры регенерации дисплея.

ОЗУ дисплея хранит информацию, которая должна отображаться на дисплее. ОЗУ объемом 16x8 можно организовать как двойное ОЗУ объемом 16x4. Это ОЗУ загружается и опрашивается МП.

Регистр адреса ОЗУ дисплея хранит адрес данных, которые в данный момент записываются или считываются МП. Адреса чтения (записи) записываются в регистр командами МП. Адреса могут быть установлены автоинкрементом после каждого чтения или записи. Регистр устанавливается в состояние лог. 0 командой «Сброс».

Регистры регенерации дисплея хранят данные, которые в момент сканирования сигналами SLO-SL3 появляются на выходах OUTA0-OUTA3, OUTB0-OUTB3. Входная информация (D7-D0), записываемая в буферы канала данных, соответствует информации на выходах OUTA0-OUTA3, OUTB0-OUTB3:

D7	D6	D5	D4	D3	D2	D1	D0
OUTA3	OUTA2	OUTA1	OUTA0	OUTB3	OUTB2	OUTB1	OUTB0

ПРОГРАММНОЕ ОБЕСПЕЧЕНИЕ

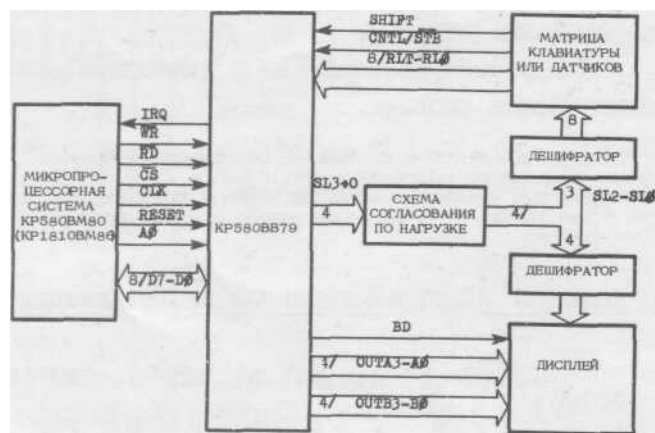
Описанные ниже команды программируют режим работы БИС KP580BB79. Команды посылаются на шину данных при уровне лог. 0 сигнала cs# и уровне лог. 1 сигнала AO и загружаются в БИС - по положительному фронту сигнала WR#.

Перед программированием режима работы необходимо установить БИС в исходное состояние. На вход RESET следует подать напряжение высокого уровня длительностью не менее 6 периодов тактового сигнала CLK. ОМ-ОЗУ датчиков и ОЗУ дисплея в состояние лог. 0 аппаратным способом не устанавливаются. На рис. 8.3 приведена схема подключения БИС KP580BB79.

Работа программируется с помощью 8 команд. Код команды определяется 3 старшими разрядами байта данных.

Команда „Установка режимов работы клавиатуры-дисплея“.

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	0	D	D	K	K	K



а

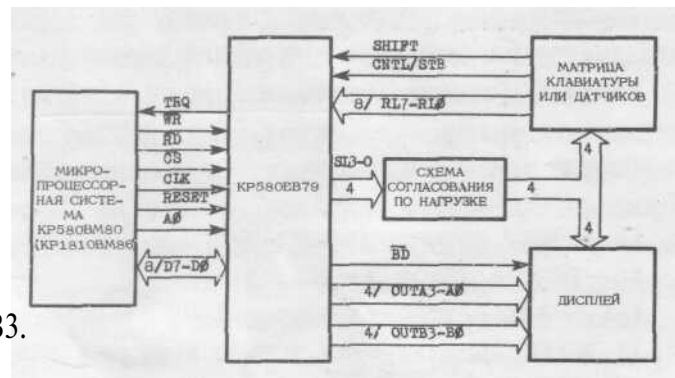


Рис. 8.3. Схема подключения БИС KP580BB79: * - на выходах SL3-SLO - кодированная информация; 6 - на выходах SL3-SLO - дешифрованная информация

- DD - режимы дисплея;
- 00 - 8-символьный, 8-битный дисплей, левый ввод;
- 01 - 16-символьный, 8-битный дисплей, левый ввод*;
- 10 - 8-символьный, 8-битный дисплей, правый ввод;
- 11 - 16-символьный, 8-битный дисплей, правый ввод;
- KKK - режим клавиатуры;
- 000 - кодированное сканирование клавиатуры, 2-клавишное сцепление*;
- 001 - декодированное сканирование клавиатуры, 2-клавишное сцепление;
- 010 - кодированное сканирование клавиатуры, N-клавишное сцепление;
- 011 - декодированное сканирование клавиатуры, N-клавишное сцепление;
- 100 - кодированное сканирование матрицы датчиков;
- 101 - декодированное сканирование матрицы датчиков;

110- стробированный ввод, кодированное сканирование дисплея;

111 - стробированный ввод, декодированное сканирование дисплея.

Примечания 1. Режим, обозначенный знаком *, устанавливается по умолчанию после сброса. 2. Если в режиме клавиатуры установлено декодированное сканирование, дисплей уменьшается до 4 символов независимо от установки режима дисплея.

Команда "Программирование синхронизации".

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	0	0	1	P	P	P	P	P

PPPPP - коэффициент деления внешнего синхроимпульса (изменяется от 2 до 31)

Внешний синхроимпульс делится на PPPP для получения базисной внутренней частоты.

Команда «Чтение ОМ-ОЗУ датчиков».

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	0	1	0	A1	X	A	A	A

A1 - флаг автоинкремента Для ОЗУ датчиков;

X - безразличное состояние;

AM - адрес строки, читаемой МП.

A1 и AAA используются только в режиме сканирования матрицы датчиков. В режимах сканирования клавиатуры и ввода по стробу достаточно подать только код команды. Приращение адреса в этих режимах происходит при чтении всегда, независимо от состояния A1. Эта команда используется для определения того, что источником данных, считываемых МП, является ОМ-ОЗУ датчиков.

Если в режиме сканирования матрицы датчиков флаг автоинкремента не установлен ($A1 \neq 0$), то перед каждым считыванием данных нужно подавать команду считывания. Если $A1=1$, то каждое считывание данных МП наращивает адрес на 1, так что следующее чтение происходит из следующей строки ОЗУ датчиков.

Команда „Чтение ОЗУ дисплея”.

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	0	1	1	A1	A	A	A	A

A1 - флаг автоинкремента для ОЗУ дисплея;

AAAA- адрес следующего знака, читаемого МП.

Так как центральный МП использует один и тот же счетчик адресов для чтения и записи в ОЗУ дисплея, эта команда устанавливает также ячейку следующей записи и режим автоинкремента при записи. Эта команда используется для

определения того, что источником данных, считываемых МП, является ОЗУ дисплея.

При $A1=1$ после каждого чтения (записи), адрес следующего считываемого знака будет приращиваться, так что следующее чтение (запись) будет происходить из (в) следующей ячейки.

Команда "Запись в ОЗУ дисплея".

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	1	0	0	A1	A	A	A	A

A1 - флаг автоинкремента;

AAAA- адрес следующего знака, записываемого в ОЗУ дисплея.

Эта команда работает так же, как предыдущая. Она так же изменяет адрес ячейки ОЗУ дисплея, из которой будет проводиться следующее считывание, но не влияет на источник считывания информации. Считывание будет проводиться из той памяти (ОЗУ дисплея или ОМ-ОЗУ датчиков), которая указана последней.

Команда »Запрет записи дисплея (гашение)".

Формат	D7	D6	D5	D4	D3	D2	D1	D0	
	1	0	1	x	iw	iw	BL	BL	
						A	B	A	B

iw - запрет записи в ОЗУ отображения по входам 07-04 для канала А и по входам D3-D6 для канала В;

BL - гашение выходов OUTA3-OUTA0 для канала А и выходов OITV3-OITV0 для канала В.

Эта команда используется для запрета записи в одну из половин ОЗУ дисплея или обе половины одновременно, а также гашение информации на выходах OUTA3-OUTA0, OITV3-OITV0 (в нули, единицы, шестнадцатеричное число 20).

Гашение отображения или запрет записи в ОЗУ дисплея осуществляется при записи единицы в соответствующий разряд D3-D0 команды. Выходной код на выводах OUTA, OITV, используемый в качестве гасящего, устанавливается командой «Сброс», описанной ниже.

Если каналы А и В гасятся одновременно, выход \overline{VD} во время действия команды переходит в состояние лог. 0. После аппаратного сброса выходы OUTA, OITV автоматически устанавливаются в состояние лог. 0.

Команда „Сброс”.

Команда «Сброс» предназначена для осуществления программного сброса (установления в

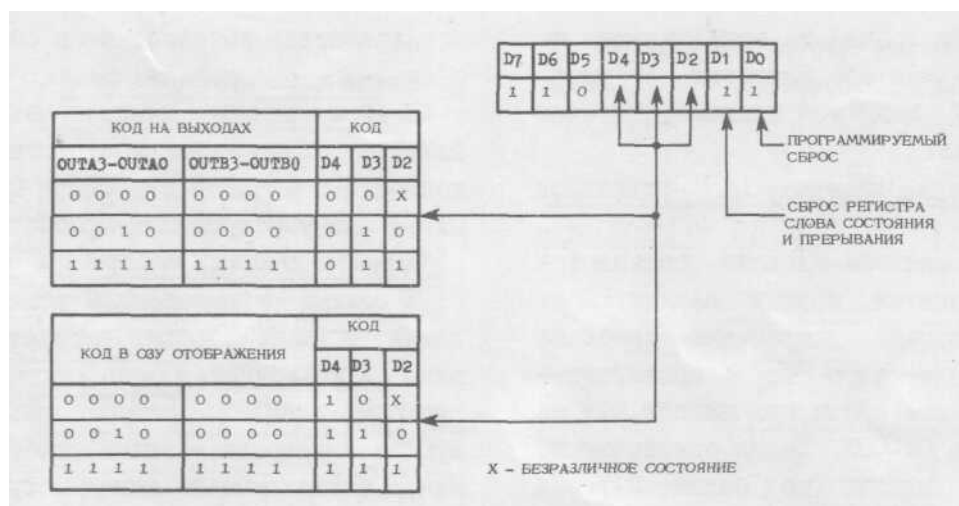


Рис. 8.4. Формат команды 'Сброс'

исходное состояние) БИС слова состояния и прерывания канала IRQ, а также для установления 037 отображения и выходов OUTA3-OUTA0, OUTB3-OUTB0 в код, который определяется разрядами D2, D3, D4 (см. рис. 8.4).

Разряд D0 предназначен для программного сброса БИС. Функции программного сброса аналогичны функциям аппаратного сброса (исключается сброс коэффициента деления и режима работы).

Разряд D1 устанавливает регистр слова состояния ОМ-ОЗУ датчиков в состояние, соответствующее опустошению ОМ-ОЗУ датчиков, а также сбрасывает выход IRQ в клавиатурном режиме и режиме ввода по стробу. В режиме сканирования матрицы датчиков этот разряд устанавливает вывод IRQ и регистр адреса ОЗУ датчиков в состояние лог. 0.

Разряды D2 и D3 определяют значение кода, в который будет устанавливаться ОЗУ отображения И ВЫХОДЫ OUTA3-OUTA0, OUTB3-OUTB0.

Разряд D4=1 используется только для установления ОЗУ отображения в нуль, единицу или шестнадцатеричное число.

Команда "Сброс" прерывания - установление режима обнаружения ошибок".

Формат	D7	D6	D5	D4	D3	D2	D1	D0
	1	1	1	1	X	X	X	X

D7-D5 - код команды «Сброс прерывания - установление ошибок»;

D4 - код установки режима обнаружения ошибок;

D3-D0 - безразличное состояние.

Команда «Сброс прерывания - установление режима обнаружения ошибок» выполняет 2 функции - сброс сигнала IRQ в режиме матрицы датчиков и установление специального режима об-

наружения ошибок в клавишном режиме. В режиме сканирования матрицы датчиков эта команда (разряды D7-D5) переводит вывод IRQ в состояние лог. 0, т.е. разрешает дальнейшую запись информации в ОЗУ датчиков. В клавишном режиме, если разряд D4 команды «Сброс прерывания - установление режима обнаружения ошибок» запрограммирован в единицу, микросхема будет работать в особом режиме обнаружения ошибок.

ФУНКЦИИ ИНТЕРФЕЙСА

Режим сканированной клавиатуры, 2-клавишное сцепление".

В этом режиме при нажатии клавиши включается логика устранения дребезга, и в течение 2 периодов сканирования (10,24 мс) происходит поиск другой нажатой клавиши.

Если за это время не произошло больше ни одного нажатия, схема устранения дребезга ожидает 10,24 мс и снова проверяет, нажата ли данная клавиша. Если нажата, значит данное нажатие было одиночным и код позиции клавиши вводится в ОМ-ОЗУ. Если ОМ-ОЗУ был до этого пуст, то выход IRQ устанавливается в состояние лог. 1, сообщая центральному МП, что произошел ввод в ОМ-ОЗУ. Если ОМ-ОЗУ был до этого уже заполнен, код позиции нажатой клавиши не вводится в ОМ-ОЗУ и устанавливается флаг ошибки.

Если за время поиска произойдет нажатие другой клавиши, то она вводится в ОМ-ОЗУ только в том случае, если все другие клавиши перед ней не нажаты. Если же она отпущена быстрее, чем другие, то ее нажатие полностью игнорируется.

Если за время поиска произойдет нажатие двух клавиш, это воспринимается как одновре-

менное нажатие. Ни одна из этих клавиш не распознается, пока они обе остаются нажатыми. Вводится в ОМ-ОЗУ код той клавиши, которая отпускается последней.

Режим сканирования клавиатуры, N-клавишное сцепление.

В этом режиме нажатие клавиши опознается независимо от нажатия других клавиш. Если клавиша нажата, логика устранения дребезга ждет 10,24 мс, после чего опять проверяет - нажата ли еще клавиша. Если она нажата, код ее позиции вводится в СМ-ОЗУ. Когда одновременно нажаты несколько клавиш (не более 8), они опознаются и вводятся в ОМ-ОЗУ в соответствии с тем порядком, в котором они обнаружены.

В режиме сканированной клавиатуры с N-клавишным сцеплением можно запрограммировать специальный режим обнаружения ошибок с помощью команды «Сброс прерывания - установление режима обнаружения ошибок». При этом цикл устранения дребезга и проверка нажатия клавиши происходит так же, как и в обычном режиме 1-клавишного сцепления. Однако если во время 1 цикла устранения дребезга обнаружены 2 нажатые клавиши, это рассматривается как множественное нажатие и в слове состояния устанавливается флаг ошибки. Этот флаг устанавливает на выходе IRQ напряжение высокого уровня и запрещает дальнейшую запись в СМ-ОЗУ. Флаг ошибки считывается при чтении олова состояния и сбрасывается командой „Сброс“, в которой DL=1.

Режим матрицы датчиков.

В этом режиме логика устранения дребезга не работает и состояние матрицы датчиков вводится непосредственно в ОЗУ датчиков. Таким образом, ОЗУ датчиков хранит отображение состояния ключей матрицы датчиков. Этот режим, в отличие от клавиатурного, только индицирующего происшедшее нажатие, имеет то преимущество, что МП «знает», как долго был замкнут ключ и когда он был разомкнут.

Для облегчения программного обеспечения разработчик должен сгруппировать датчики по функциональному назначению в строки по 8, так как считывает их в виде строк. Если обнаружено изменение в состоянии матрицы датчиков, то выход IRQ устанавливается в состояние лог. 1 в конце цикла сканирования. Выход сбрасывается в состояние лог. 0 первой же операцией чтения, если в команде „Чтение ОМ-ОЗУ“ флаг автоинкремента установлен в состояние лог. 0, или командой «Сброс прерывания», если флаг ав—

тоинкремента установлен в состояние лог. 1.

Режим стробируемого ввода.

В этом режиме данные вводятся с входов RL7-RLO по положительному фронту сигнала, приходящего с ввода CNTL/STB#, в ОМ-ОЗУ датчиков, минуя схему устранения дребезга.

Форматы данных, вводимых в БИС.

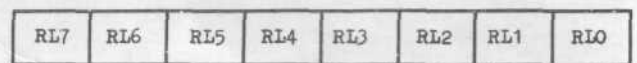
В режиме сканированной клавиатуры код, вводимый в ОМ-ОЗУ, соответствует позиции кла-

ВЫВОД +CNTL+SHIFT: CNTL SHIFT SCAN RETURN,

где CNTL и SHIFT - старшие разряды кода. Следующие 3 разряда - выход счетчика сканирования, индицирующий номер строки, в которой происходит поиск нажатой клавиши. Последние 3 разряда - выход счетчика линий возврата, показывающий номер столбца, в котором обнаружена нажатая клавиша.

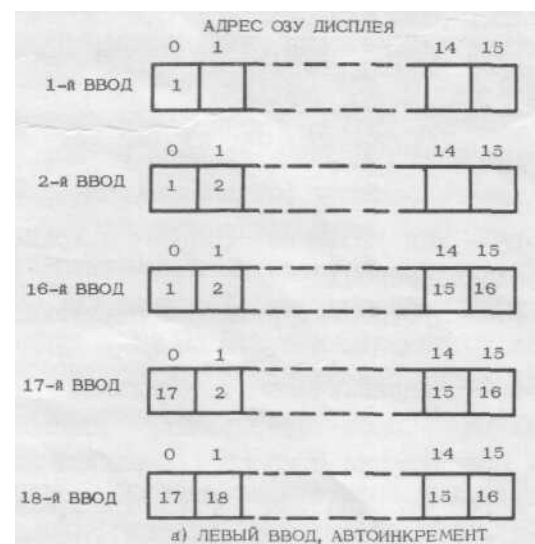
В режимах матрицы датчиков и стробируемого ввода данные с линий RL7-RLO вводятся непосредственно в регистр ОМ-ОЗУ датчиков, который соответствует сканируемой строке матрицы датчиков.

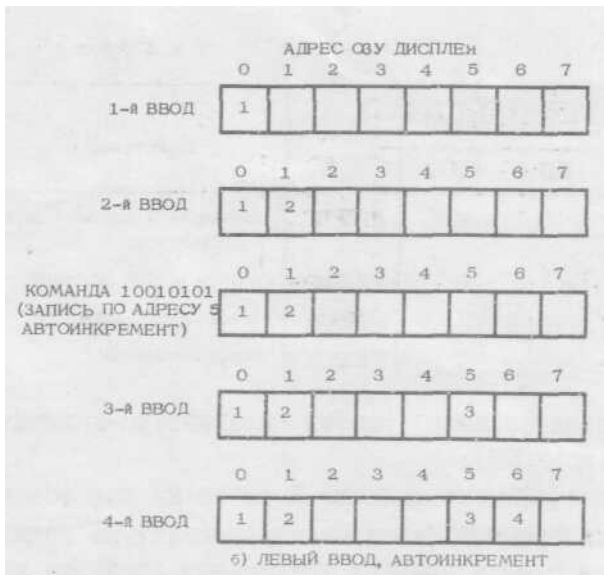
Формат вводимых данных в режимах матрицы датчиков и стробируемого ввода:



Вывод информации на дисплей.

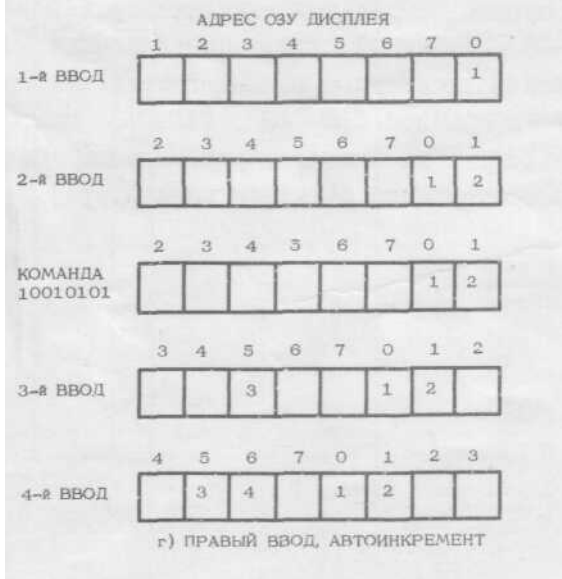
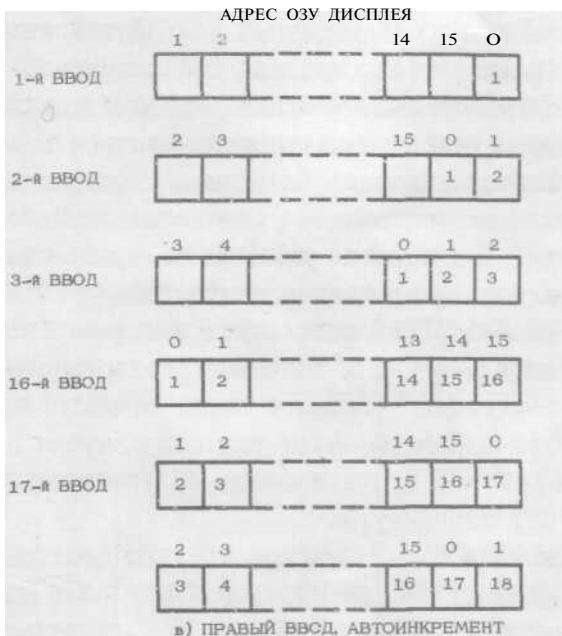
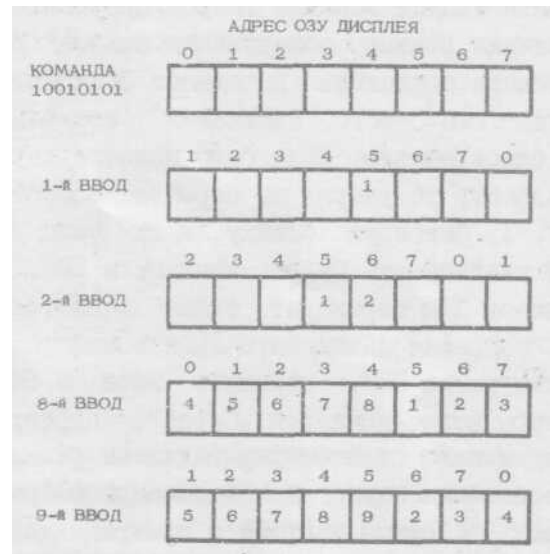
Левый ввод - простейший формат дисплея, в котором каждая позиция дисплея прямо соответствует адресу ОЗУ дисплея, т.е. адрес 0 в ОЗУ соответствует самому левому символу, адрес 15 (или адрес 7 в 8-символьном дисплее) - самому правому символу. Введение символов с нулевой позиции вызывает заполнение дисплея слева направо, 17-й (9-й) символ вводится вновь в самую левую позицию.





Правый ввод - используется в калькуляторах. Первый вводимый символ помещается в правую позицию. Следующий - снова в правую после сдвига предыдущего символа на разряд влево, и т.д.

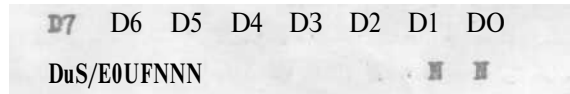
Заметьте, что при правом вводе номер позиции дисплея и адрес ОЗУ дисплея не совпадают (в, г).



Если начало ввода информации - не с нулевого адреса (д), то вводимый знак будет появляться в точке начала ввода.

Чтение регистра состояния.

Формат слова состояния



В регистре состояния содержится информация о состоянии ОМ-ОЗУ, ошибках и запрете доступа к ОЗУ отображения.

Все разряды слова состояния, кроме 06, используются только в режимах сканирования клавиатуры и стробированного ввода.

Разряды D3-D0 в двоичном коде определяют число символов, находящихся в ОМ-ОЗУ датчиков.

Разряд D4 - флаг переолуствования. Он устанавливается в состояние лог. 1, если имело место чтение из уже пустого QM-037.

Разряд D5 - флаг переполнения. Он устанавливается в состояние лог. 1, если имела место запись в уже заполненный ОМ-ОЗУ.

Разряд D6 в зависимости от режима работы выполняет двойную функцию. В режиме матрицы датчиков разряд D6 переходит в состояние лог. 1 при замыкании хотя бы одного ключа. При размыкании всех ключей он переходит в состояние лог. 0. Выход IRQ переходит в состояние лог. 1 при замыкании или размыкании хотя бы одного ключа. При этом запрещается дальнейшая запись информации в БИС из матрицы датчиков. Разрешить запись можно командой «Сброс пре—

рывания - установление режима обнаружения ошибок".

В режиме клавиатуры с N-клавишным сцеплением после записи команда «Сброс прерывания - установление режима обнаружения ошибок» разряд D6 слова состояния выполняет функцию обнаружения ошибочного замыкания нескольких клавиш одновременно. Если было нажатие двух и более клавиш, то разряд D6 перейдет в состояние лог. 1, фиксируя ошибку, и ни один код нажатой клавиши не будет введен в БИС. При этом вывод IRQ переходит также в состояние лог. 1, запрещая дальнейшую запись информации с клавиатуры в БИС. Разрешить ввод в БИС с клавиатуры можно командой «Сброс», запрограммировав в ней сброс прерывания и регистра слова состояния (D1=1, D0=1 в команде „Сброс“).

Разряд D7 сигнализирует о прекращении доступа к ОЗУ дисплея, так как команда «Сброс» не завершила свои операции по очистке дисплея. (Время доступа к ОЗУ дисплея равно $16 T_{clk} * P$, где T_{clk} - период синхронизации, P - коэффициент деления).

Регистр состояния сбрасывается как программным, так и аппаратным способом.

ИСПОЛЬЗОВАНИЕ БИС КР580ВВ79 В „МИКРОЛАБ 1810“

В „Микролаб 1810“ БИС КР580ВВ79 отвечает за сканирование клавиш, кодирование матрицы клавиатуры и обновление элементов индикации.

Контроллер клавиатуры и индикации занимает 2 порта ввода-вывода в адресном пространстве ввода-вывода МП. БИС КР580ВВ79 сопрягается с младшим байтом шины данных (D0-D7), поэтому оба порта имеют четные адреса FFE8 и FFEA. Функционирование БИС определяется адресным битом A1 и сигналами IORC, IOWC (табл. 8.1).

Программа монитора программирует БИС КР580ВВ79 следующим образом:

8-разрядных индикаторов с вводом слева и кодированное сканирование клавиатуры с обнаружением 2-клавишных сцеплений (команда ус-

Т а б л и ц а 8.1

Входы БИС КР580ВВ79			Адрес порта	Действие
AO	RD	WR		
0	0	1	FFE8H	Чтение ОЗУ дисплея или клавиатуры
0	1	0	FFE8H	Запись в ОЗУ дисплея
1	0	1	FFEAH	Чтение состояния
1	1	0	FFEAH	Запись команды

тановки режимов работы клавиатуры-индикатора, код 00H);

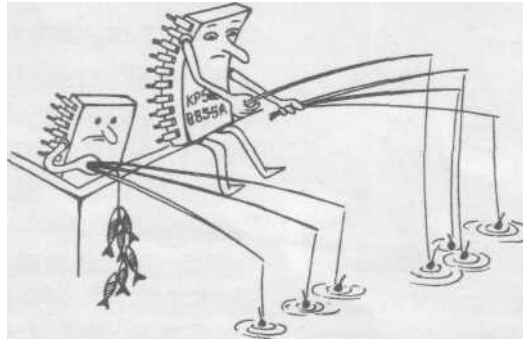
коэффициент деления K равен 25 для обеспечения времени сканирования клавиатуры индикаторов 5 мс и времени устранения дребезга контактов 10 мс (команда программирования синхронизации, код 39H).

Так как устанавливается режим кодированного сканирования клавиатуры, то выходы SLO-SL2 (выводы 32, 33, 34 микросхемы D42) представляют собой двоичный код. Дешифратор D47 преобразует этот код для обеспечения 3 входных сигналов сканирования ряда матрицы и клавиатуры. Выходы с матрицы клавиатуры поступают на входы RLO-RL7 (выводы 38, 39, 1, 2, 5, 6, 7, 8 микросхемы D42), представляя 8 колонок клавиш. При нажатии клавиши во время сканирования ряда разрешается соответствующая колонка. БИС КР580ВВ79 использует значение бита разрешенной колонки и значение сканируемого ряда для генерации 6-битного кода, представляемого нажатой клавишей. Этот ряд сохраняется в очереди ОЗУ БИС и считывается МП через порт чтения ОЗУ клавиатуры.

Сигналы SLO-SL2 дополнительно поступают на дешифратор D46, дешифратор возбуждает одну из 8 выходных линий, которая используется для разрешения отдельных индикаторов D48-D51, D54-D57. 8-битный код, определяющий - какие сегменты индикатора должны светиться, поступает с выходов DBO-DB3, DAO-DA3 микросхемы D42 (лог. 1 на выходе обуславливает свечение соответствующего сегмента индикатора).

УРОК 9

ПАРАЛЛЕЛЬНЫЙ ПОРТ ВВОДА-ВЫВОДА



В микропроцессорных системах под портами ввода-вывода подразумеваются схемы, осуществляющие сопряжение с внешними устройствами ввода-вывода. Они подключаются к шинам микропроцессорной системы с одной стороны и к внешнему устройству - с другой.

В "Микролаб 1810" в качестве параллельного порта ввода-вывода используется БИС КР580ВВ55А, которая позволяет подключить к микропроцессорной системе несколько внешних устройств, обеспечивая сопряжение с ними без применения дополнительных логических схем.

БИС КР580ВВ55А представляет собой однокристалльное программируемое устройство ввода-вы-

вода параллельной информации различного формата (программируемый периферийный интерфейс).

БИС КР580ВВ55А помещена в корпус с 40 выводами (рис. 9.1), из которых 24 образуют 3 8-разрядных порта ввода-вывода: КА, КВ, КС.

Обмен информацией между портами ввода-вывода и шиной данных микропроцессорной системы происходит через двунаправленную шину D0-D7 БИС КР580ВВ55А в соответствии с сигналами «Чтение», «Запись», «Выбор кристалла», а также А0 и А1, определяющими адрес порта. Эти управляющие сигналы вырабатываются МП и системным контроллером.

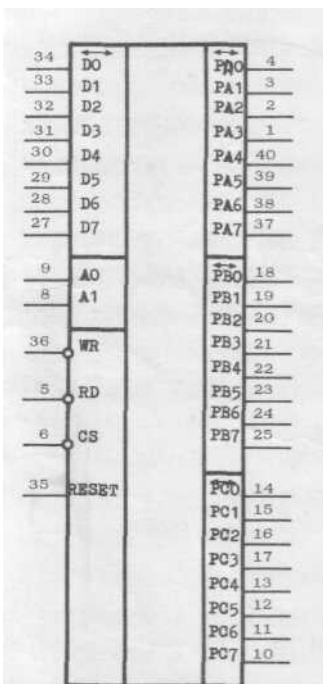


Рис. 9.1. Назначение выводов БИС КР580ВВ55А

Вывод	Назначение	Вывод	Назначение
1	Выход-вход 3-го разряда канала А	21	Выход-вход 3-го разряда канала В
2	Выход-вход 2-го разряда канала А	22	Выход-вход 4-го разряда канала В
3	Выход-вход 1-го разряда канала А	23	Выход-вход 5-го разряда канала В
4	Выход-вход 0-го разряда канала А	24	Выход-вход 6-го разряда канала В
5	Вход сигнала „Чтение“	25	Выход-вход 7-го разряда канала В
6	Вход выбора микросхемы	26	Напряжение питания $U_n = 5 В$
7	Общий (I)	27	Вход-выход данных D7
8	Вход 1-го разряда адреса	28	Вход-выход данных D6
9	Вход 0-го разряда адреса	29	Вход-выход данных D5
10	Выход-вход 7-го разряда канала С	30	Вход-выход данных D4
11	Выход-вход 6-го разряда канала С	31	Вход-выход данных D3
12	Выход-вход 5-го разряда канала С	32	Вход-выход данных D2
13	Выход-вход 4-го разряда канала С	33	Вход-выход данных D1
14	Выход-вход 0-го разряда канала С	34	Вход-выход данных D0
15	Выход-вход 1-го разряда канала С	35	Вход установки „0“
16	Выход-вход 2-го разряда канала С	36	Вход сигнала „Запись“
17	Выход-вход 3-го разряда канала С	37	Выход-вход 7-го разряда канала А
18	Выход-вход 0-го разряда канала В	38	Выход-вход 6-го разряда канала А
19	Выход-вход 1-го разряда канала В	39	Выход-вход 5-го разряда канала А
20	Выход-вход 2-го разряда канала В	40	Выход-вход 4-го разряда канала А

УСТРОЙСТВО И ПРИНЦИП РАБОТЫ

БИС состоит из следующих основных узлов:

- буферов канала данных,
- регистра управляющего слова,
- схемы выбора канала,
- буферов канала А (КА),
- буферов канала В (КВ),
- буферов канала О (КС),
- схемы управления каналом с,
- схемы вентилей канала С,
- схемы управления регистром с.

Буферы канала данных D7-DO представляют собой двунаправленные буферы, предназначенные для подключения внутренней магистрали данных интерфейса к 8-разрядной двунаправленной магистрали системы.

Каждый буфер канала данных состоит из входного формирователя, который предназначен для приема на внутреннюю магистраль интерфейса управляющих слов и данных, и выходного формирователя, который предназначен для передачи на магистраль системы информации о состоянии и данных от внешних устройств.

Регистр управляющего слова (РУС) предназначен для приема и хранения управляющего слова поступающего через буферы канала данных и внутреннюю магистраль данных. Управляющее слово записывается в РУС в том случае, если на входе канала данных D7 при остальных необходимых управляющих сигналах устанавливается лог. 1. Содержимое РУС определяет функцию и направление работы каждого из портов ввода-вывода (КА, КВ, КС).

Схема выбора канала предназначена для формирования сигналов управления внутренними и внешними передачами данных, управляющих слов и информации состояния. На схему поступают входные сигналы АО и А1 канала адреса и сигналы «Чтение», «Запись» и «Выбор кристалла».

Если уровень сигнала «Чтение» лог. 1, то

разрешается передача информации одного из каналов КА, КВ, КС на шину данных МП, т.е. проводится чтение информации из интерфейса. Если уровень сигнала «Запись» лог. 1, то разрешается передача информации с шины данных МП в РУС или один из каналов (КА, КВ, КС), т.е. проводится запись информации в интерфейс.

Разрешение работы МП с данным интерфейсом определяется подачей сигнала лог. 0 на вход CS БИС.

Входные сигналы АО и А1 вместе с управляющими сигналами «Чтение», «Запись» и «Выбор кристалла» обеспечивают выбор одного из каналов интерфейса и определяют направление передачи информации в соответствии с табл. 9.1.

Каждый из 3 портов имеет собственные отличительные черты и особенности, обеспечивающие гибкость БИС КР580ВВ55А.

Порт А состоит из 8-разрядного выходного регистра и выходных формирователей, из входных формирователей и 8-разрядного входного регистра и может работать на ввод или вывод 8-разрядных слов в 3 режимах работы: 0, 1, 2.

Порт В состоит из 8-разрядного регистра ввода-вывода, входных и выходных формирователей и может работать на ввод или вывод 8-разрядной информации в 2 режимах работы - 0 и 1.

Порт С состоит из 2 4 разрядных регистров, включающих разряды 0-3 и 4-7. Каждому регистру соответствует своя группа выходных и входных формирователей, используемых для ввода и вывода 4-разрядных слов в режиме 0.

Схемы управления портом С предназначены для управления входными и выходными формирователями, а также для формирования управляющих сигналов, если порт используется для передачи управляющих сигналов.

Схемы вентилей порта с предназначены для передачи информации со входа или с регистров порта о на внутреннюю магистраль данных.

Т а б л и ц а Э.1

Управляющие сигналы					Вид передаваемой информации	Направление передачи информации
АО	А1	Запись	Чтение	Выбор кристалла		
0	0	0	1	0	Данные	Порт А → шина данных
1	0	0	1	0	—	Порт В → шина данных
0	1	0	1	0	—	Порт С → шина данных
0	0	1	0	0	—	Шина данных → порт А
1	0	1	0	0	—	Шина данных → порт В
0	1	1	0	0	—	Шина данных → порт С
1	1	1	0	0	Управляющее слово	Шина данных → РУС
-	-	-	-	1	-	Нет передачи

Схемы управления регистром с позволяют программным способом устанавливать в состояние лог. 1 и сбрасывать в состояние лог. 0 любой из разрядов регистра с.

Центральный МП управляет выбором режимов работы портов при помощи засылки в интерфейс 2 различных типов управляющих слов.

1. Управляющее слово определения режима работы интерфейса (разряд 7 - в состоянии лог. 1). Путем записи этого слова в РУС можно организовать заданную структуру ввода-вывода, определяемую комбинацией из 3 основных режимов. Запись управляющего слова проводится всегда в начале работы микросхемы, а также в случае необходимости программного изменения структуры ввода-вывода в процессе работы (рис. 9.2).

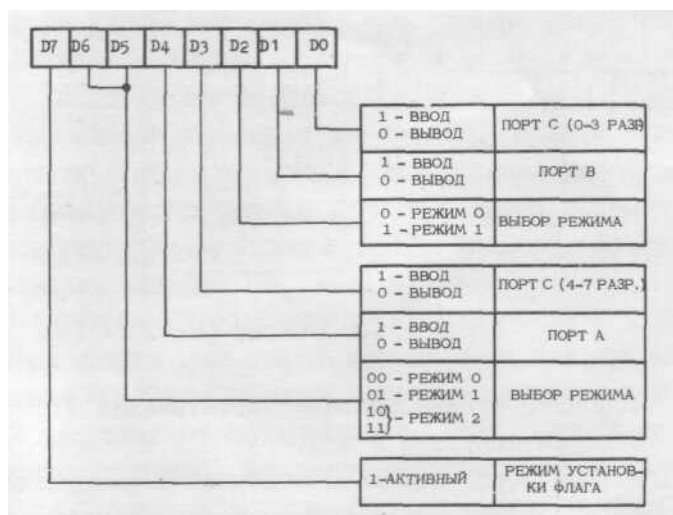


Рис. 9.2. Формат управляющего слова, задающего режим работы интерфейса

2. Управляющее слово установления разрядов регистра с в состояние лог. 1 или сброса в состояние лог. 0 (разряд 7 - в состоянии лог. 0). Любой из 8 разрядов регистра порта С может быть программным способом установлен в состояние лог. 1 или 0. Для этого необходимо провести запись управляющего слова, формат которого показан на рис. 9.3, в интерфейс.

ВЫБОР РЕЖИМА

БИС КР580ВВ55А имеет 3 основных режима работы:

- режим 0 - режим основного ввода-вывода;
- режим 1 - режим стробируемого ввода-вывода;
- режим 2 - режим двунаправленной шины.

При установлении на входе „RESET” уровня лог. 1 все порты настраиваются на режим 0. После того, как сигнал "RESET" снимается, БИС

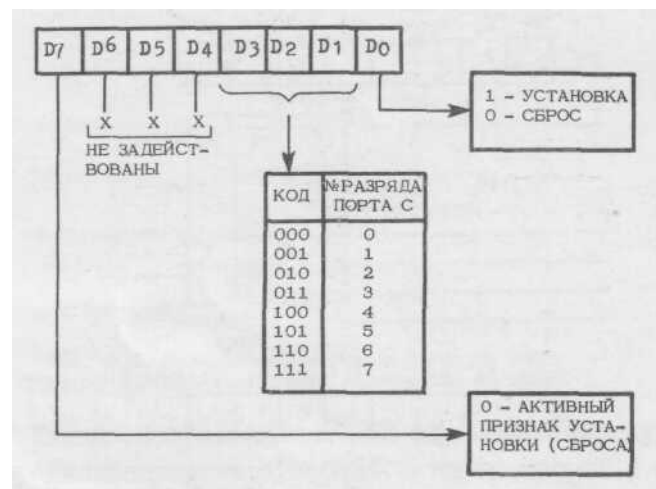


Рис. 9.3. Формат управляющего слова установки (сброса) разрядов регистра порта С

КР580ВВ55А может оставаться в том же режиме без дополнительных установок в исходное состояние.

Режимы для порта А. и порта В могут определяться отдельно, в то время как режим порта С, разделенный на 2 части, определяется в соответствии с режимами портов А и В.

Режимы можно объединить так, что они могут быть приспособлены почти к любой структуре ввода-вывода. Например, порт В может быть установлен в режим 0 для управления простым переключателем или для вывода результатов вычислений, а порт А может быть установлен в режим 1 для управления клавиатурой или чтения с ленты.

РАБОЧИЕ РЕЖИМЫ

Режим 0 (режим основного ввода-вывода).

При работе в режиме 0 возможна организация обмена данными между внешними устройствами и микропроцессорной системой через 3 8-разрядных порта, А, В и С, причем порт С, в свою очередь, может быть представлен в виде 2 отдельных 4-разрядных портов, образованных его младшими и старшими разрядами. Наличие 4 портов с возможностью работы каждого в режиме приема или передачи обеспечивает программную реализацию 16 вариантов работы БИС КР580ВВ55А в режиме 0. На рис. 9.4 показаны 2 возможные конфигурации.

В этом режиме обмен данными между БИС КР580ВВ55А и внешними устройствами проводится без управляющих сигналов готовности к работе и подтверждения приема. Он осуществляется по инициативе МП, причем характер изменения данных во времени должен позволять ему опрашивать порты в произвольные моменты.

Основные функциональные черты режима 0:

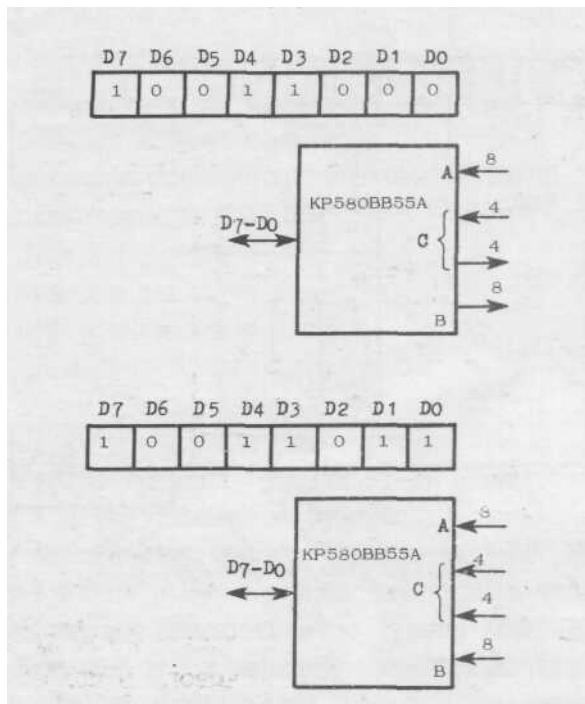


Рис. 9.4. Примеры: конфигурации портов Б режиме 0

2 8-разрядных и 2 4-разрядных порта; любой порт служит для ввода и вывода; вывод фиксируется; ввод не фиксируется.

Обычно режим 0 используется для ввода и вывода медленно изменяющихся данных или каких-либо постоянных значений, например, номеров обслуживаемых устройств.

Временная диаграмма режима 0 представлена на рис. 9.5.

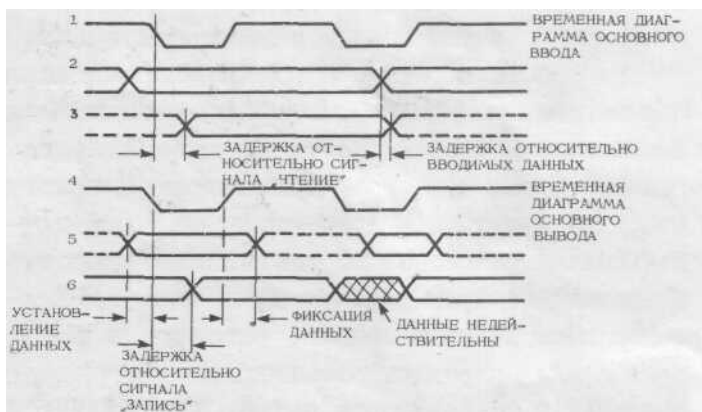


Рис. 9.5. Временная диаграмма работы БИС KP580BB55A в режиме 0: 1 - сигнал "Чтение"; 2 - ввод; 3 - шина данных; 4 - сигнал "Запись"; 5 - шина данных; 6 - вывод

В тех случаях, когда интерфейс настроен на работу в режимах 1 или 2, сигналы управления могут быть использованы в качестве запросов прерывания в МП. Источником сигнала «Запрос прерывания» при обмене данными служат соответствующие разряды регистра С, используемые как триггеры разрешения прерывания по каналу

А или В (ПРа и ПРв). Они формируют этот сигнал в момент приема и выдачи информации внешними устройствами. В БИС KP580BB55A предусмотрена возможность установления в состояние лог, 0 или лог. 1 любого разряда буферного регистра порта С с помощью управляющего слова, передаваемого МП в порт С. С помощью команды установки (сброса) сигналы «Запрос прерывания», полученные из порта С, могут быть разрешены или запрещены при помощи триггера ПР. Если триггер ПР установлен - прерывание разрешено, сброшен - прерывание запрещено.

Режим 1 (режим стробируемого ввода-вывода).

При работе в режиме 1 происходит обмен данными между 8-разрядными портами А и В и внешними устройствами под управлением сигналов, поступающих по линиям порта с (рис. 9.6).

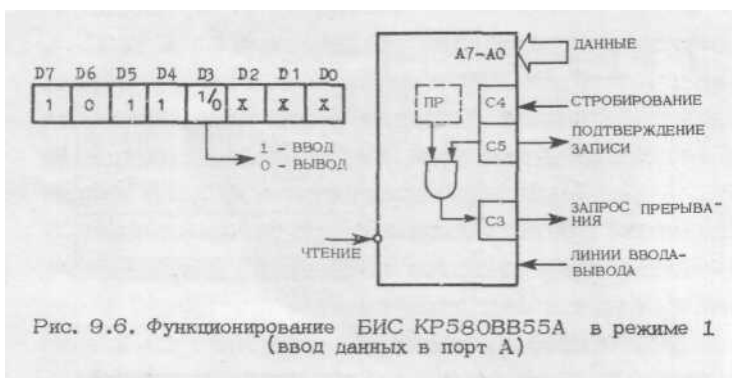


Рис. 9.6. Функционирование БИС KP580BB55A в режиме 1 (ввод данных в порт А)

Основные функциональные черты режима 1:

2 порта передачи А и В;

каждый порт передачи содержит 8-разрядный порт данных и 4-разрядный порт управления;

8-разрядный порт данных может служить как для ввода, так и для вывода, при этом ввод и вывод фиксируются.

После того, как внешнее устройство установило данные на линиях А7-А0 порта А, оно инициирует их ввод в интерфейс путем подачи на вход С4 порта с сигнала "Стробирование". По этому сигналу данные с входных линий переписываются в буферный регистр порта А, о чем свидетельствует сигнал «Подтверждение записи», передаваемый из БИС KP580BB55A во внешнее устройство через линию С5 порта С. Этот сигнал вызывает сброс сигнала "Стробирование" и появление сигнала «Запрос прерывания», передаваемого в МП. Через период времени, необходимый для обслуживания прерывания, информация из буферных регистров порта А будет прочитана процессорным элементом системы путем подачи сигнала «Чтение» на вход. Сигнал «Запрос прерывания» сбрасывается по переднему

фронту", а по заднему фронту - сигнал "Подтверждение записи" (рис. 9.7).

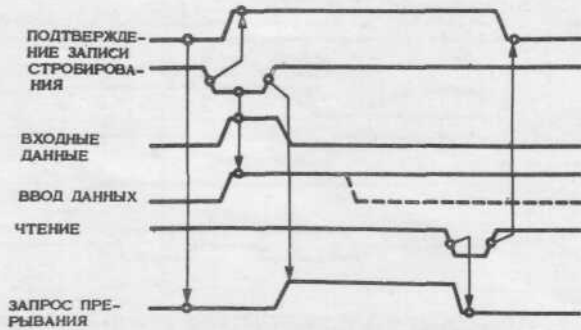


Рис. 9.7. Временная диаграмма сигналов БИС KP580BV55A при вводе информации в режиме 1

Определение управляющих сигналов ввода в режиме I:

"Стробирование" - уровень лог. 0 на входе указывает на то, что данные загружены в фиксатор ввода;

"Подтверждение записи" - уровень лог. 1 на этом выходе указывает на то, что данные загружены в фиксатор ввода. Этот сигнал устанавливается по переднему фронту сигнала "Стробирование", а сбрасывается по заднему фронту сигнала «Чтение».

"Запрос прерывания" - уровень лог. 1 на этом выходе может быть использован для прерывания МП в тех случаях, когда устройство ввода запрашивает обслуживание. Этот сигнал устанавливается по заднему фронту сигнала "Стробирование", если «Подтверждение записи» и триггеры ПРа, ПРв находятся в состоянии лог. 1. Сигнал «Запрос прерывания» сбрасывается по переднему фронту сигнала «Чтение». ПРа - управляется разрядом установки (сброса) С4, ПРв - управляется разрядом установки (сброса) С2.

На рис. 9.8 показана схема вывода данных через порт в из микропроцессорной системы во внешнее устройство в режиме I.а на рис. 9.9 - соответствующие временные диаграммы.

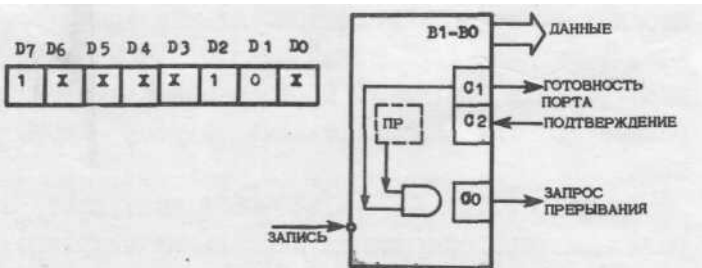


Рис. 9.8. Функционирование БИС KP580BV55A в режиме 1 (вывод данных из порта В)

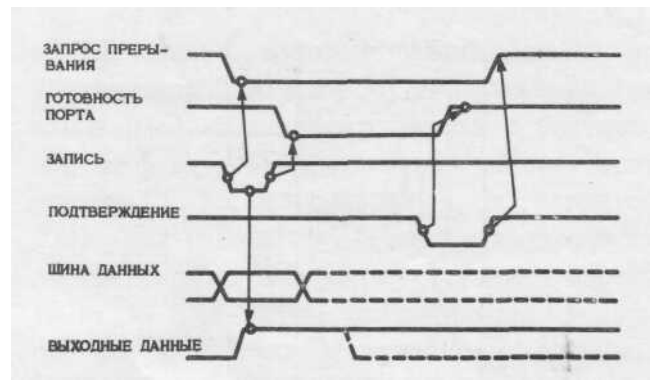


Рис. 9.9. Временная диаграмма сигналов БИС KP580BV55A при выводе информации в режиме 1

Данные, подготовленные МП системы к выводу, переписываются в буферный регистр порта в с помощью сигнала "Запись", что подтверждается сигналом «Готовность порта», передаваемым через линию С1 порта с во внешнее устройство. Уровень лог. 0 сигнала «Готовность порта» указывает, что МП имеет данные, предназначенные для вывода в указанный порт. Он устанавливается по заднему фронту сигнала «Запись», а сбрасывается по переднему фронту сигнала «Подтверждение». По получении этого сигнала внешнее устройство считывает информация с выходных линий порта В, по окончании операции она выдает сигнал «Подтверждение» по линии С2 порта с. Уровень лог. 0 на входе «Подтверждение» информирует БИС KP580BV55A о получении данных из порта А или в. Задний фронт этого сигнала инициирует «Запрос прерывания» для МП. Уровень лог. 1 на выходе «Запрос прерывания» может быть использован для прерывания МП при получении устройством вывода данных из МП. Сигнал «Запрос прерывания» устанавливается по заднему фронту сигнала «Подтверждение», если на выходе - сигнал «Готовность порта» и на ПРв - уровень лог. 1. Сброс проводится по переднему фронту сигнала «Запись»» ПРа-управляется разрядом установки (сброса) Об, ПРв - управляется разрядом установки (сброса) С2.

Как при вводе, так и при выводе информации в режиме I две линии порта о свободны и могут быть использованы для ввода или вывода какой-либо информации в режиме 0.

Режим 2 (стробируемый ввод-вывод по двунаправленной шине).

Режим 2 осуществляется с помощью одной двунаправленной 8-разрядной шины обмена с внешними устройствами или другой двунаправленной шиной. Обмен данными с внешним устройством происходит по сигналам, передаваемым через порт о, аналогичным управляющим сигналам в режиме I (рис. 9.10).

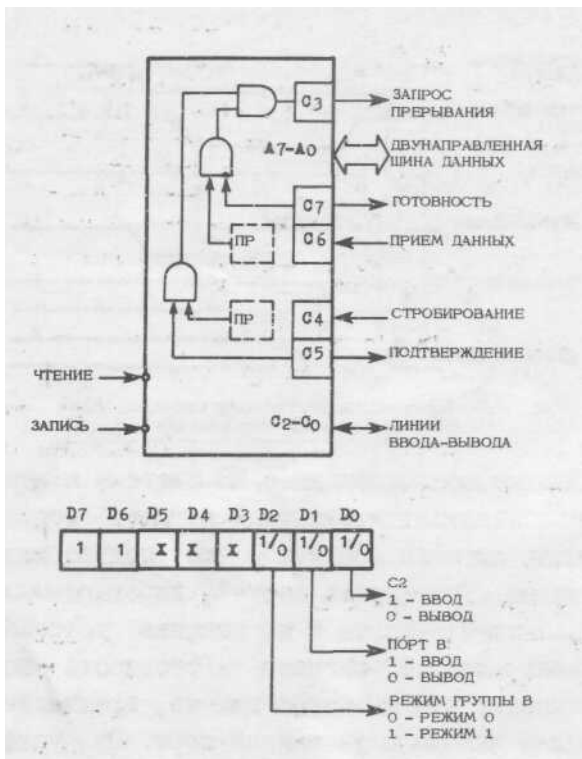


Рис. 9.10. Функционирование БИС KP580BV55A в режиме 2

Режим 2 используется только для порта А.

В режиме вывода информации, после записи данных по сигналу «Запись» в порт А, через линию С7 порта С выдается сигнал „Готовность” (уровень лог. 0 на этом выходе указывает, что МП имеет данные для передачи в порт А), при этом линии ввода-вывода порта А. находятся в высокоимпедансном состоянии. В момент готовности внешнего устройства к приему информации оно передает на линию С6 порта с сигнал «Прием данных», по которому на линиях А7-А0 порта А появляются данные для передачи во внешнее устройство. (Уровень лог. 0 на этом входе переводит буфер вывода порта А в состояние, при котором он может выдавать данные. В противном случае буфер вывода находится в высокоимпедансном состоянии.)

Ввод данных в порт микропроцессорной системы инициируется внешним устройством, которое подает информацию на линии А7-А0 и сигналом «Стробирование» через линию С4 порта С, уведомляет об этом БИС KP580BV55A. По сигналу "Стробирование" данные переписываются в порт А. По окончании процесса внешнему устройству передается сигнал «Подтверждение». (Уровень лог. 1 на этом выходе показывает, что данные загружены в фиксатор ввода.) Сигнал «Запрос прерывания» при работе в режиме 2 генерируется по окончании ввода или вывода информации из порта А (рис. 9.11).

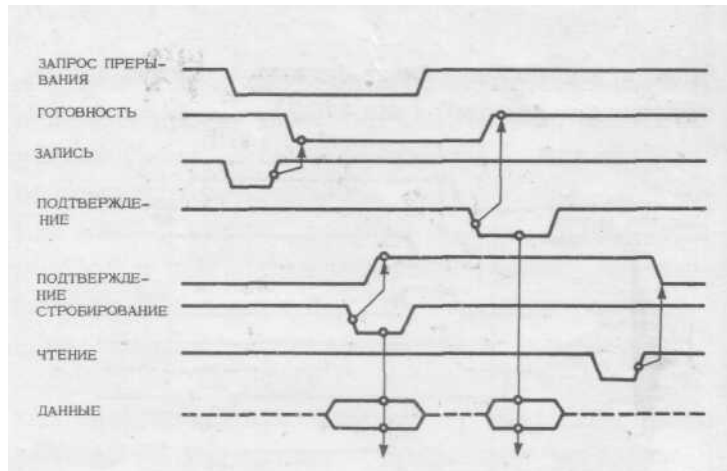


Рис. 9.11. Временная диаграмма сигналов БИС KP580BV55A в режиме 2

В режимах 1 и 2 свободные от управляющих сигналов линии порта с могут быть использованы для ввода-вывода информации в режиме 0.

БИС KP580BV55A позволяет осуществлять разнообразные комбинации режимов работы портов, например, подключение клавиатуры терминала к микропроцессорной системе (рис. 9.12). При

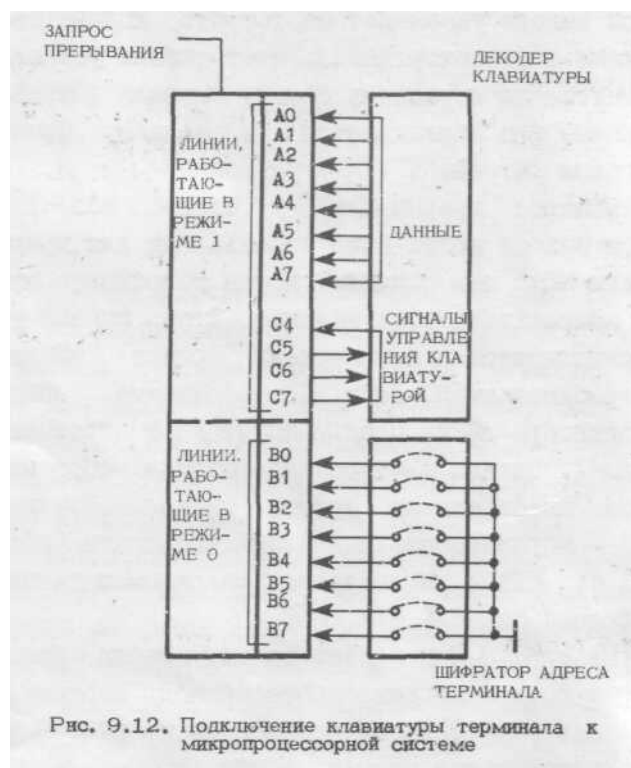


Рис. 9.12. Подключение клавиатуры терминала к микропроцессорной системе

этом на порт А возлагается передача информационных сигналов с декодера клавиатуры в режиме 1, сигналы управления следуют через старшие разряды порта с. Порт в используется в режиме 0 для идентификации адреса терминала.

На рис. 9.13 БИС KP580BV55A выступает в роли адаптера сопряжения нескольких микропроцессорных систем, работающих на общую системную шину. В этом случае реализуется режим 2

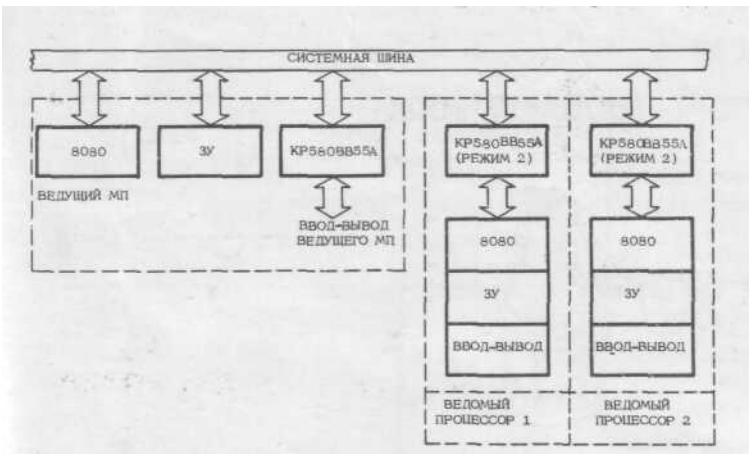


Рис. 9.13. Интерфейс распределенной многопроцессорной системы

БИС KP580BB55A, работающей на двунаправленную шину.

ИСПОЛЬЗОВАНИЕ БИС KP580BB55A. В „МИКРЛАБ 1810“

Параллельный порт ввода-вывода состоит из двух программируемых периферийных интерфейсов KP580BB55A (микросхемы D38 и D39). Каждая из этих БИС содержит 3 8-разрядных порта данных (обозначаемых буквами А, в, с), которые могут быть как входными, так и выходными, и 1 порт управления. Микросхема D38 соединена с младшим байтом данных (D0-D7), а D39 обеспечивает связь со старшим байтом данных (D8-D15). Все порты могут быть адресованы индивидуально (например, P1A, P2C, P18), или соответствующая пара портов (P1A и P2A, P1B и P2B) может быть адресована одновременно для образования 16-разрядного порта данных.

Адреса портов ввода-вывода для 2 БИС параллельных интерфейсов приведены в табл. 9.2.

Т а б л и ц а 9.2

Порт	Адрес
P1A	FFF8
P1B	FFFA <i>вывод P1</i>
P1C	FFFC
P1 _{упр}	FFFE
P2A	FFF9
P2B	FFFB <i>вывод P2</i>
P2C	FFFD
P2 _{упр}	FFFF

Управление выбором необходимых портов обеспечивает дешифратор ввода-вывода (микросхема D34). Во время операций с байтами дешифратор вырабатывает сигнал выбора соответствующего порта (выводы 11 и 12 микросхемы D34). При работе со словами для адресации желаемой пары портов задается только адрес порта P1, а дешифратор ввода-вывода генерирует сигналы выбора обоих портов одновременно.

Работа дешифратора ввода-вывода разрешается уровнем лог. 0 на входе CS1 микросхемы D34 (сигнал, указывающий на то, что МП выполняет операцию ввода-вывода) и уровнем лог. 1 адресных битов A9-A15 (поступающих на микросхему D35 и указывающих на то, что адрес ввода-вывода находится в пределах FEO0H-FFFFH).

Микросхема D34 декодирует адресные биты A0, A3-A8 и сигнал ВНЕ для генерации сигнала выбора соответствующего порта ввода-вывода. Лог. 0 на выходе дешифратора разрешает выбор соответствующего порта ввода-вывода.

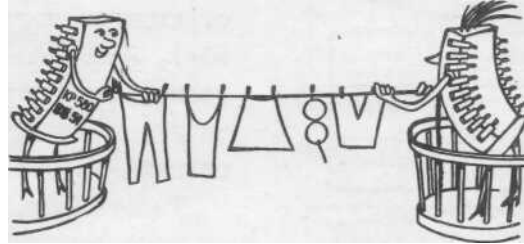
Дешифратор ввода-вывода обеспечивает выбор еще двух БИС KP580BB51A и KP580BB79, Назначение и работа этих микросхем рассматриваются в уроках 10 и 8 соответственно.

На плате микро-ЭВМ установлены устройства, имитирующие внешние порты ввода-вывода. Это 8 светодиодов VD3-VD10, которые подключены к порту P18 и служат приемниками информации (порт вывода с адресом FFFAH) и 8 переключателей S4, соединенных с портом P2B, информация о состоянии которых может быть считана МП (порт ввода с адресом FFF8H. Разряд во порта P1 соединен с громкоговорителем, используемым в качестве приемника информации» Дня связи микро-ЭВМ с бытовым кассетным магнитофоном используются разряды c0 и c1 порта P2.

При работе с имитаторами внешних портов ввода-вывода следует помнить, что микросхемы D38 и D39 программируются на ввод после включения питания или после нажатия клавиши «Сброс». Установка необходимого режима работы портов P1 и P2 может быть осуществлена с помощью команд „Вывод байта" или «Вывод слова".

УРОК 10

ПОСЛЕДОВАТЕЛЬНЫЙ ПОРЯДОК ВВОДА-ВЫВОДА



Среди периферийных БИС особую группу составляют микросхемы, ориентированные на передачу данных по линиям связи. Их основными функциями являются, во-первых, организация последовательного обмена информацией между микропроцессорной системой и аппаратурой передачи данных, во-вторых, реализация протокола передачи данных, под которым понимается ряд правил, регулирующих упорядоченный обмен данными между двумя или большим числом источников информации.

Промышленностью выпускаются периферийные БИС для систем передачи данных, ориентированные на различные протоколы, причем, в зависимости от сложности схемы, протокол может быть реализован полностью или частично. В последнем случае часть функций по реализации протокола связи передается программному обеспечению. Так, БИС KP580BB5I предназначена для организации программируемого последовательного обмена информацией между микропроцессорной системой и аппаратурой передачи данных.

Режим работы и параметры интерфейса устанавливаются и изменяются программно путем выдачи МП управляющих слов и специальных команд. Возможны синхронный режим обмена со скоростью до 56 К бит/с и асинхронный (старт-стопный) режим со скоростью до 9600 бит/с, с программированием формата передаваемого символа.

ОПИСАНИЕ БИС KP580BB5I

БИС KP580BB5I - универсальный синхронно-асинхронный приемник (передатчик), предназначенный для обеспечения связи между МП KM1810BM86, KP580ИК80) и периферийными устройствами. Интерфейс воспринимает данные в параллельном формате с шины данных МП и преобразует их в последовательный формат и, наоборот, входной последовательный поток данных преобразует в параллельный формат.

БИС помещена в 28-выводной корпус. Функции выводов БИС приведены в табл. 10.1.

БУФЕР ПИНЫ ДАННЫХ

8-разрядный двунаправленный буфер с 3 состояниями обеспечивает связь БИС KP580BB5I с системной шиной данных. Данные передаются или принимаются буфером при выполнении команд ввода IN или команды вывода OUT с МП. Слова управления, слова команды и информация о состоянии также передаются через буфер шины данных. Команда, состояние, входные и выходные данные передаются через отдельные 8-разрядные регистры для обеспечения двойного буферирования.

ЛОГИКА ЧТЕНИЯ (ЗАПИСИ)

Этот блок воспринимает входные сигналы с системной шины управления и генерирует сигналы управления работой схемы. Он содержит регистр слова управления и регистр слова команды, в которых хранится формат выбранного режима работы.

RESET - уровень лог. 1 этого сигнала переводит БИС KP580BB5I в исходное состояние. БИС остается в этом состоянии до тех пор, пока не будет запрограммирована для работы в нужном режиме. Минимальная длительность импульса должна быть не менее 6 периодов тактового импульса.

CLK - сигнал синхронизации, используемый для внутренней синхронизации работы БИС.

WR# - уровень лог. 0 этого сигнала информирует БИС о том, что МП записывает данные или слово управления в БИС KP580BB5I..

RD# (чтение) - уровень лог. 0 этого сигнала информирует БИС о том, что МП считывает данные или информацию о состоянии.

S/D# (управление/данные) - входной сигнал в сочетании с WR# и RD# информирует БИС о том, что информация на шине данных является словом уп-

Номер вывода	Обозначение	Функция
1,2,5-8,27,28	D7-DO	Канал данных БИС (двунаправленный трехстабильный) для передачи данных, управляющих слов и информации состояния из БИС в МП и из МП в БИС. DO - младший разряд канала данных
21	RESET	Входной сигнал установки БИС в исходное состояние при включении питания или при сбоях системы
20	CLK	Входной сигнал внутренней синхронизации БИС. К этому выводу обычно подсоединяется фаза Ф2 тактового генератора микропроцессорной системы
13	\overline{RD}	Входной сигнал разрешения передачи информации состояния и данных из БИС в канал данных МП
10	\overline{WR}	Входной сигнал разрешения передачи данных и управляющего слова из канала данных МП в БИС
11	\overline{CS}	Входной сигнал выбора БИС
12	C/D	Входной сигнал записи (чтения) данных или управляющих сигналов. Если уровень сигнала - лог. 0, то данные записываются или читаются, если - лог. 1, то записываются управляющие сигналы или читается информация о состоянии БИС
24	\overline{DTR}	Выходной сигнал запроса о готовности передатчика терминала к передаче данных
22	\overline{DSR}	Входной сигнал готовности установки данных. Может быть ответным на сигнал \overline{DTR} . Контролируется сигналом "Чтение состояния".
23	\overline{RTS}	Выходной сигнал запроса о готовности приемника терминала к приему данных
17	\overline{CTS}	Входной сигнал готовности приемника терминала к приему данных
15	TxRDY	Выходной сигнал готовности передатчика БИС к приему данных из МП
18	TxEMPTY	Выходной сигнал, фиксирующий окончание передачи данных. Контролируется сигналом "Чтение состояния"
19	TxD	Выходной сигнал передатчика БИС
9	TxC	Входной сигнал синхронизации передачи. Управляет скоростью передачи символов
14	RxRDY	Выходной сигнал готовности приемника БИС к передаче данных в канал данных МП. Может быть использован для прерывания ЦП или при операциях опроса
3	RxD	Входной сигнал приемника БИС
25	\overline{RxC}	Входной сигнал синхронизации работы приемника. Управляет скоростью приема символа
16	SYNDET/BD	Двунаправленный программируемый вход-выход. Используется только в синхронном режиме приема. Контролируется сигналом "Чтение состояния"
4	\perp	Корпус
26	Vcc	+5 В

равления или информацией о состоянии (лог. 1) или данными (лог. 0).

CS# (выбор кристалла) - активный уровень сигнала - лог. 0, уровень лог. 1 переводит шину данных в третье состояние.

УПРАВЛЕНИЕ МОДЕМОМ

Интерфейс KP580BB5I имеет набор входных и выходных сигналов, который может быть использован для упрощения связи с любым периферий-

ным устройством. Сигналы управления модемом являются сигналами общего назначения и могут быть использованы в случае необходимости для других целей.

DSR# (готовность установки данных) - входной сигнал, 1-битный инвертирующий входной порт. Его состояние может быть проверено МП с помощью операции чтения состояния.

DTR# (данные терминала готовы) - выходной сигнал, 1-битный инверсный выходной порт, мо-

жет быть установлен в состояние лог. О программированием соответствующего бита в слове команды.

RTS# (запрос на посылку) - выходной сигнал, 1-битный инвертирующий порт, может быть установлен в состояние лог. О программированием соответствующего бита в слове команды.

CTS# (очистка посылки) - уровень лог. О на этом входе разрешает БИС KP580BB5I передавать последовательные данные, если бит TXEN (разрешение передачи) в байте команды установлен в уровень лог. I. Если во время передачи вход CTS# будет выключен или бит TxEN сброшен, БИС KP580BB5I передаст все данные, записанные в порт до команды запрещения. Если сигналы TXEN и CTS будут сброшены раньше, чем последний записанный символ появится в последовательном потоке данных, то этот символ будет передан снова при установке сигналов стз или TXEN.

БУФЕР ПЕРЕДАТЧИКА

Буфер передатчика воспринимает данные с буфера шины данных, преобразует их в последовательность битов, добавляет соответствующие разряды (в соответствии с требованием протокола) и выдает сформированную последовательность на вывод TXD по отрицательному фронту сигнала TxC#. Передача будет происходить, если есть разрешение, т.е. уровень сигнала CTS# - лог. 0.

УПРАВЛЕНИЕ ПЕРЕДАТЧИКОМ

Этот блок управляет всеми операциями, связанными с передачей последовательных данных.

Сигнал TXRDY (передатчик готов) - информирует МП, что передатчик готов принять символ данных. Сигнал TxRDY может быть использован для прерывания, так как он маскируется командой TxDisabled, или при прерываниях по опросу, когда МП сможет проверить состояние вывода с помощью операции чтения состояния. Сигнал TXRDY автоматически сбрасывается по переднему фронту сигнала WR#, когда данные записываются с МП.

Сигнал TxEMP (передатчик пуст) - уровень лог. I этого сигнала указывает на то, что БИС KP580BB5I не имеет символа для передачи. Сигнал автоматически сбрасывается при получении символа с МП, если передатчик разрешен. Сигнал TxEMP может быть использован для индикации конца режима передачи, чтобы МП «знал» когда можно переключать линию в обратном направлении в полудуплексном режиме.

Сигнал Txc (синхронизация передачи) - управляет скоростью передачи символов. В синхронном режиме передачи скорость передачи в бодах (бит/с) равна частоте сигнала TxC. В режиме асинхронной передачи скорость в бодах кратна реальной частоте. Коэффициент деления частоты программируется: 1, 16 или 64.

БУФЕР ПРИЕМНИКА

Приемник принимает последовательность данных, убирает биты протокола (старт- и стоп-биты), формирует параллельный формат и посылает на МП. Данные вводятся через вывод RxD и синхронизируются по положительному фронту сигнала RxC#.

УПРАВЛЕНИЕ ПРИЕМНИКОМ

Этот блок управляет всеми операциями, связанными с приемом.

Схема инициализации (RXD) предотвращает ошибки при приеме данных, имеющих уровень лог. 0. Перед началом приема последовательного символа, после сброса схемы должно быть определено наличие лог. I. После определения данного условия разрешается прием старт-бита (лог. 0). Это характерно только для асинхронного режима и выполняется только после общего сброса.

Схема определения ложного старт-бита предотвращает ложный запуск из-за возможных помех путем стробирования отрицательного фронта старт-бита и повторного стробирования в номинальном центре старт-бита ($VRxD=0$).

Схема переключения триггера четности и схема ошибки четности используются для определения ошибки четности при приеме и установки соответствующего бита состояния.

Триггер флага ошибки кадра устанавливается если в конце байта данных отсутствует стоп-бит (асинхронный режим); он также устанавливает соответствующий бит состояния.

Сигнал RXRDY (синхронизация приемника) - указывает, что БИС KP580BB5I содержит символ, готовый к вводу в МП. Сигнал RXRDY может быть использован для прерывания МП или при операциях опроса.

В асинхронном режиме для установки сигнала RXRDY приемнику должен быть разрешен прием старт-бита и полный символ должен быть сформирован и переслан в регистр вывода данных. В синхронном режиме для установки сигнала RXRDY приемник должен быть разрешен и символ

должен быть сформирован и переслан в регистр вывода данных.

Если полученный символ не будет считан до момента формирования нового символа, то будет установлена ошибка переполнения, на место предыдущего символа будет записан новый и символ будет потерян.

Сигнал $RxC\#$ (синхронизация приема) - управляет скоростью приема символа. В синхронном режиме скорость в бодах равна действительной частоте сигнала $RxC\#$. В асинхронном режиме скорость в бодах равна реальной частоте, поделенной на коэффициент 1, 16 или 64 (программируется).

(В большинстве случаев БИС KP580BB5I работает одновременно на прием и на передачу. Следовательно, скорости приема и передачи должны быть равны. Сигналы $TxC\#$ и $RxC\#$ объединяются и подаются к одному генератору.)

ОПИСАНИЕ РАБОТЫ

Режим работы интерфейса программируется с помощью управляющих слов. Слова управления определяют скорость работы в бодах, длину символа, число стоп-битов, синхронный или асинхронный режим работы, установку и контроль четности и т.д. В синхронном режиме также определяется выбор внешнего или внутреннего символа синхронизации.

После программирования БИС KP580BB5I готова к работе, вывод $TXRDY$ переходит в состояние лог. 1 указывая МП, что БИС готова к приему символа с МП. Этот выходной сигнал автоматически сбрасывается, когда МП записывает символ в интерфейс. С другой стороны, интерфейс воспринимает последовательные данные с периферийного устройства. После приема симво-

ла вывод $RXRDY$ переходит в состояние лог. 1, сигнализируя МП, что имеется символ для ввода. Сигнал $RXRDY$ автоматически сбрасывается при считывании данных МП.

БИС KP580BB5I не может начать передачу до тех пор, пока в слове команда не будет установлен бит $TXEN$ (разрешение передачи) и пока активен сигнал $CTS\#$. По сигналу „сброс“ вывод TXD переходит в неактивное состояние.

ПРОГРАММИРОВАНИЕ БИС KP580BB5I

Перед приемом или передачей данных в БИС KP580BB5I должен быть загружен набор управляющих слов, которые определяют режим работы и должны следовать сразу же после сброса БИС (внутреннего или внешнего).

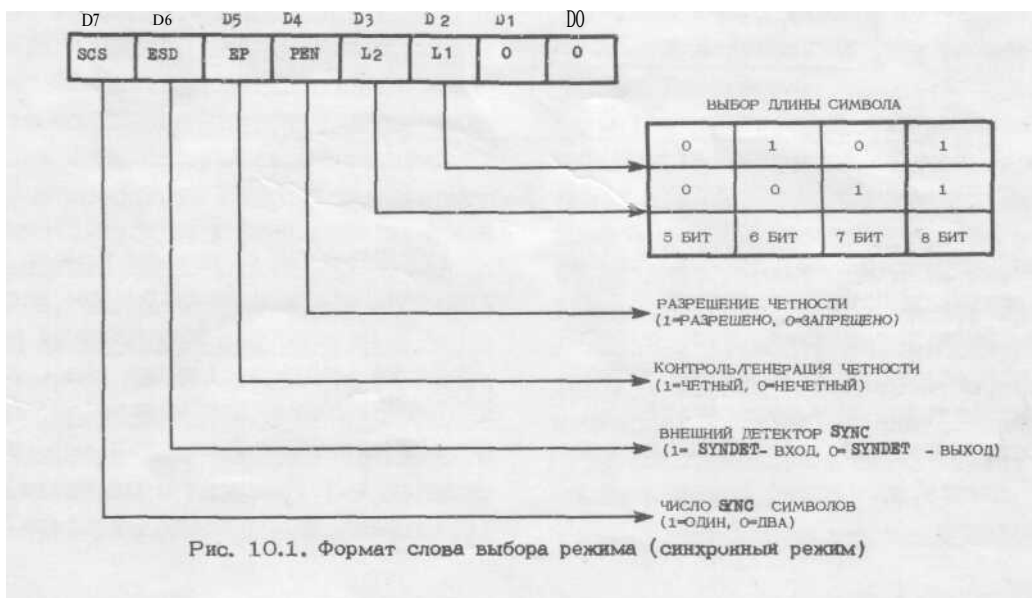
Управляющие слова делятся на 2 группы: слова выбора режима и слова команды.

Выбор режима - определяет общие рабочие характеристики интерфейса. Слово выбора режима должно следовать сразу же за операцией сброса. После того, как слово выбора режима записано в БИС могут идти символы SYNC (синхронизации) и слова команды (см. рис. 10.1, 10.2).

Слово команды определяет слово состояния, которое используется для управления конкретной операцией БИС KP580BB5I (рис. 10.3).

Слова выбора режима и команды ДОЛЖНЫ образовывать определенную последовательность для обеспечения нужной операции интерфейса (рис. 10.4). Слово выбора режима должно следовать непосредственно за сбросом БИС до любых операций по приему и передаче данных.

Все слова управления, записываемые в БИС после слова выбора режима, интерпретируются как слова команды. Команда может быть записа-



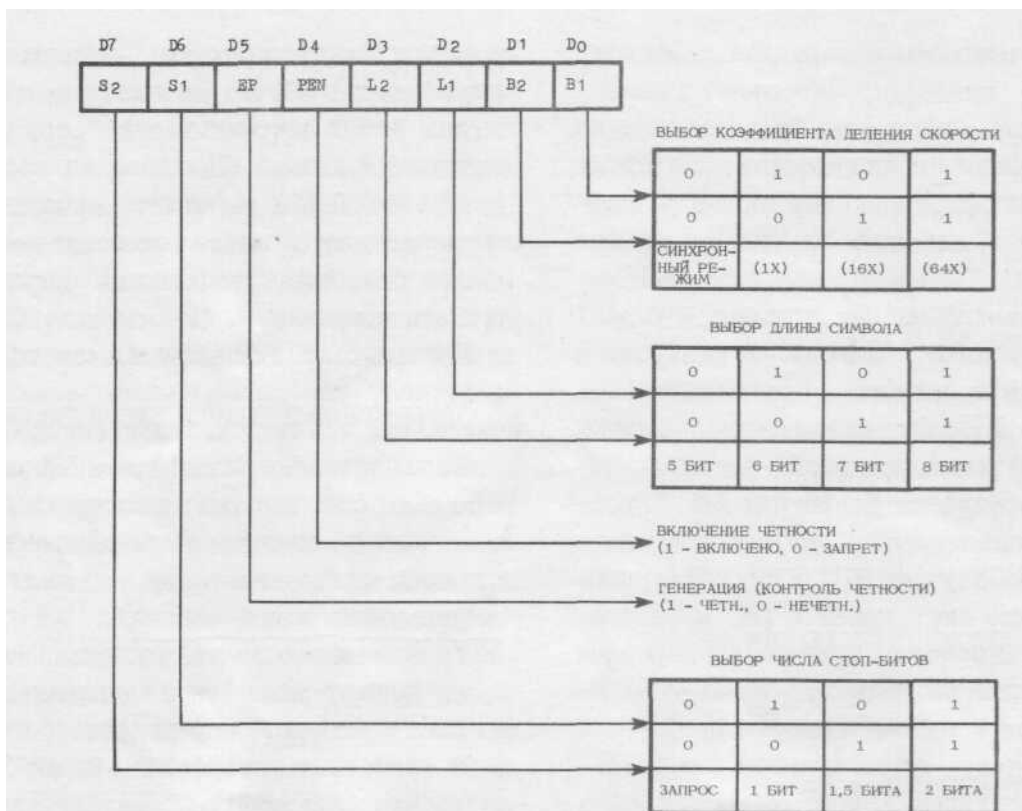


Рис. 10.2. Формат слова выбора режима (асинхронный режим)

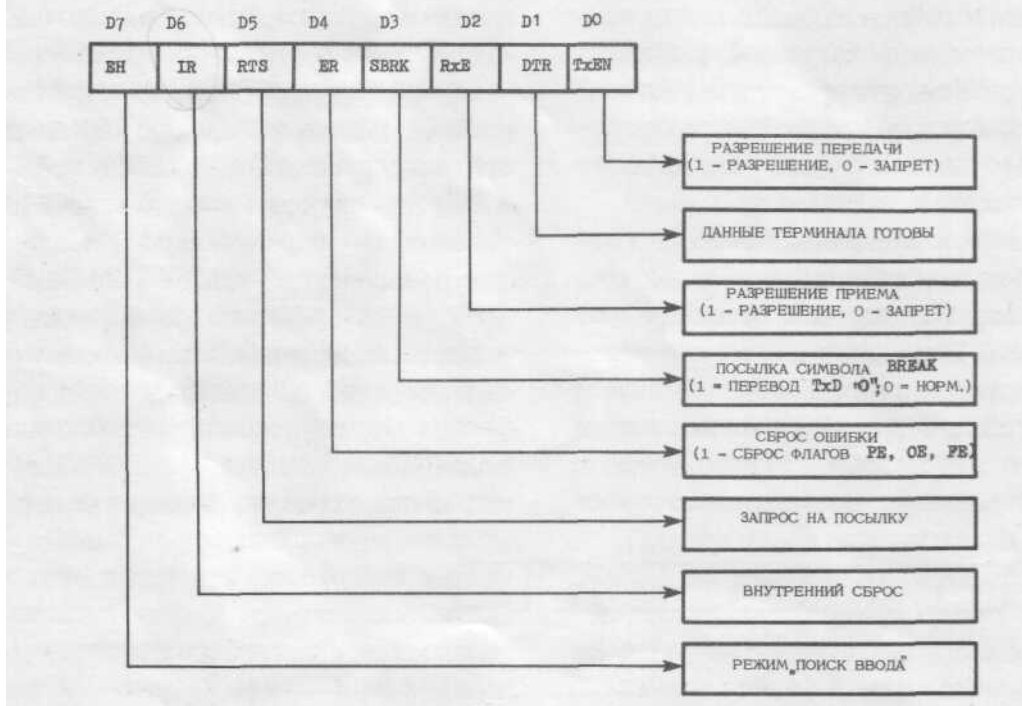


Рис. 10.3. Формат слова команды

на в БИС в любом месте блока данных. Чтобы вернуться к формату выбора режима, в слове команды должен быть установлен бит общего сброса для запуска операции внутреннего сброса, которая автоматически возвращает БИС KP580BB5I к выбору режима. За словом выбора режима должно следовать слово команды или символы синхронизации.

БИС KP580BB5I может работать как в синхронном, так и в асинхронном режиме.

Передача в асинхронном режиме,

Когда символ данных послан МП в БИС KP580BB5I схема автоматически добавляет стартовый бит (имеющий уровень лог. 0), затем следуют биты данных (начиная с младшего), бит четности (если указано в слове команды) и запрограм-



Рис. 10.4. Типичная последовательность управляющих слов

мированное число столбцов-битов. Затем символ передается на вывод TXD по отрицательному фронту сигнала txC со скоростью, равной 1, 1/16 или 1/64 частоты сигнала txC. На вывод TXD могут непрерывно поступать символы BREAK.

Если БИС KP580BB5I не содержит символа для передачи, то вывод TxD остается в состоянии лог. 1, если не была запрограммирована посылка символа BREAK (непрерывно уровень лог. 0).

Прием в асинхронном режиме.

Линия RxD обычно находится в состоянии лог. 1. Отрицательный фронт на этой линии означает появление старт-бита. Истинность старт-бита проверяется снова стробированием этого бита в его номинальном центре. Если снова обнаружен уровень лог. 0, то это означает действительно наличие старт-бита, и счетчик битов начинает считать. Счетчик битов, таким образом, определяет центр битов данных, бита четности (если он есть) и стоп-битов.

Если обнаруживается ошибка четности, то устанавливается соответствующий флаг. Данные и биты четности «защелкиваются» на выводе по положительному фронту сигнала RxC#. Если уровень лог. 0 определяется как стоп-бит, будет установлен флаг ошибки кадра. Стоп-биты означают конец символа. Заметим, что приемнику требуется только один стоп-бит независимо от того, сколько столбцов-битов запрограммировано. Полученный символ загружается в параллельный буфер ввода-вывода. Вывод RXRDY переходит в состояние лог. 1, сигнализируя МП, что символ готов к считыванию. Если предыдущий символ не был считан МП, то вновь полученный символ замешает его в буфере ввода-вывода и устанавливается флаг ошибки переполнения (таким образом предыдущий символ теряется). Все флаги ошибок могут быть сброшены командой сброса

ошибок. Наличие любой из этих ошибок не влияет на работу БИС KP580BB5I.

Передача в синхронном режиме.

Вывод TXD находится в состоянии лог. 1 до тех пор, пока МП не пошлет свой первый символ в БИС KP580BB5I (обычно это символ синхронизации). Когда линия CTS# переходит в состояние лог. 0, начинается последовательная передача первого символа. Все символы передаются по отрицательному фронту сигнала TxC#. Скорость передачи равна частоте сигнала TxC#.

Когда передача началась, данные на вывод TXD непрерывно должны поступать со скоростью сигнала TxC#. Если же МП не обеспечил БИС очередным символом данных к тому времени, когда буфер передатчика освободился, в поток данных автоматически вставляются символы SYNC. В этом случае вывод TxEMP переходит в состояние лог. 1, сигнализируя, что БИС свободна, и посылаются символы синхронизации. Вывод TxEMP не переходит в состояние лог. 0, если завершается пересылка символа SYNC. Сигнал TxEMP сбрасывается, когда в БИС записывается символ данных.

Прием в синхронном режиме.

В этом режиме может быть выбран внешний или внутренний символ синхронизации. Если выбран синхронный режим, то в первое записываемое слово команды должна быть включена команда «Поиск ввода». Данные на выводе RxD затем записываются по положительному фронту сигнала RxC#. Содержимое буфера R приемника сравнивается на границе каждого бита с первым символом SYNC, пока не будет обнаружено совпадение. Если БИС запрограммирована для работы с двумя символами SYNC, то последующие записываемые символы также сравниваются. Когда будут обнаружены оба символа SYNC схема завершит режим «Поиск». Вывод SYNDET переходит в состояние лог. 1 и автоматически сбрасывается операцией чтения состояния.

В режиме внешней синхронизации синхронизация обеспечивается подачей уровня лог. 1 на вывод SYNDET, что заставляет БИС выйти из режима «Поиск». Высокий уровень может быть снят после одного такта сигнала RxC. Команда «Поиск ввода» не используется в асинхронном режиме.

Ошибки четности и переполнения, проверяются так же, как и в асинхронном режиме. Четность проверяется, когда схема не находится в режиме поиска, независимо от того, разрешен приемник или нет.

МП может заставить (скомандовать) приемник

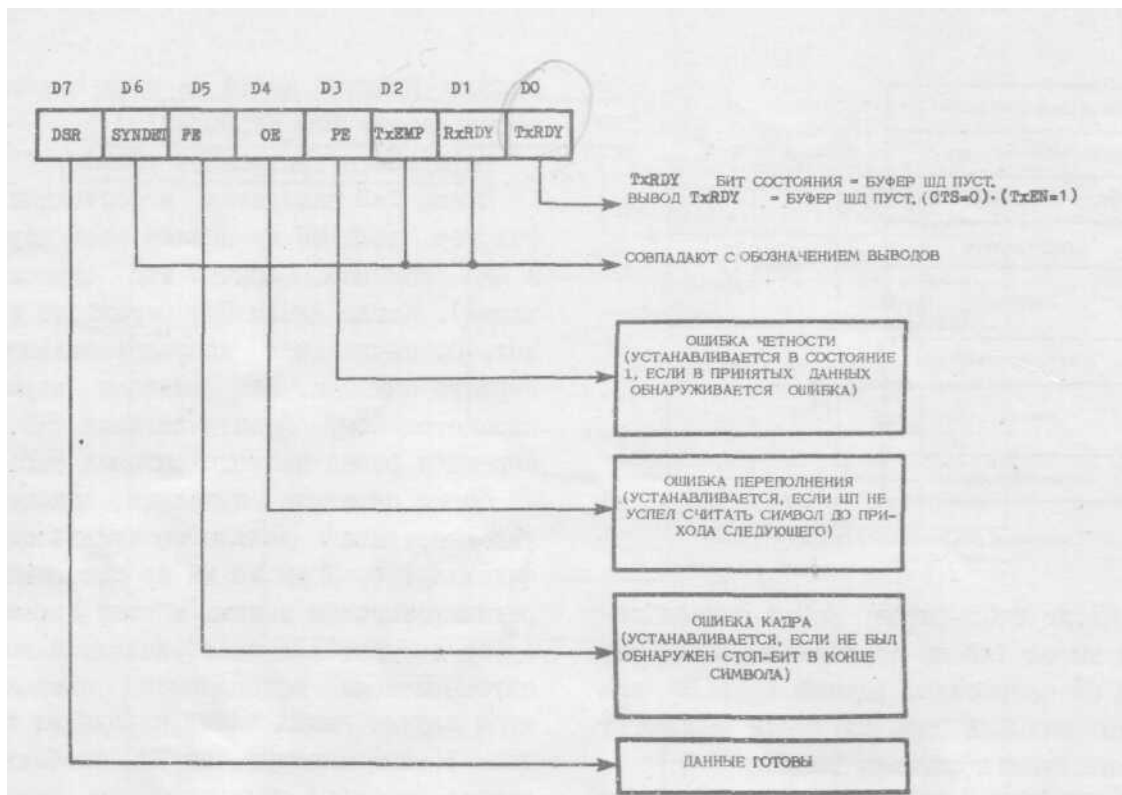


Рис. 10.5. Формат регистра состояний

вести режим «Поиск», если синхронизация потеряна. Это также установит все использованные биты символа в буфере на уровне лог, 1, предотвращая таким образом возможность ложного сигнала SYNDET, вызванного данными, попавший в буфер в режиме "Поиск ввода".

Заметим, что сигнал SYNDET сбрасывается яри каждой операции чтения независимо от того, внутренняя или внешняя синхронизация запрограммирована. Это не возвращает ШС КР580BB51 в режим поиска.

£ система? передачи данных часто необходимо контролировать состояние БИС, которое устанавливается в процессе работы, сбоев, ошибок ЕЛЕ других ситуаций. БИС КР580BB51 содержит регистр состояний, позволяющий программисту читать состояние БИС в любой момент времени в процессе выполнения операции. Формат регистра состояний представлен на рис. 10.5.

Каждая из ошибок не прерывает работу БИС. Триггеры ошибок устанавливаются в исходное состояние инструкцией команды.

Режим «Чтение состояния» позволяет использовать данную схему в системах с прерыванием г г системах с последовательным опросом внешних устройств.

На рис. 10.6 приведен пример подключения БИС КР580BB51 к шине данных и периферийному устройству.

В „Микролаб 1810" интерфейс последовательного ввода-вывода выполнен на БИС программи-

руемого интерфейса связи в последовательном коде БИС КР580BB51А (микросхема D43), которая работает в асинхронном режиме.

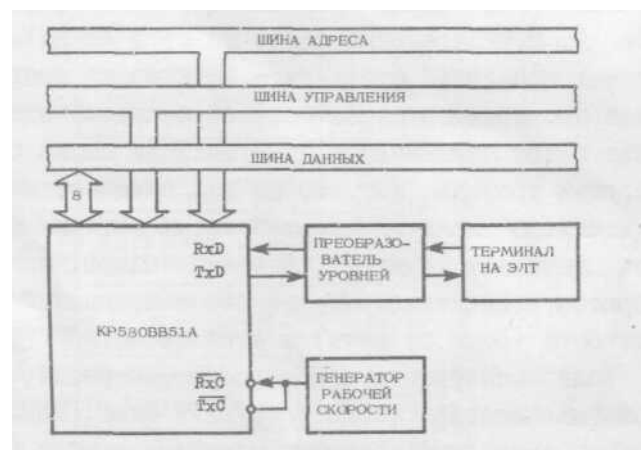


Рис. 10.6. Схема подключения БИС КР580BB51А к шине данных и периферийному устройству

БИС КР580BB51А занимает два порта ввода-вывода в адресном пространстве ввода-вывода МП. Так как она соединена с младшим байтом данных (D0-D7), то оба порта имеют четные адреса FFF0H и FFF2H. Функционирование микросхемы определяется адресным битом А0 и сигналами IORC и IOWC (см. табл. 10.2).

Программа монитора с помощью записи кода 0CFH в управляющий порт микросхемы D43 определяет работу БИС следующим образом: длина символа 8 бит; контроль на четность отсутствует; имеется два стоповых бита; коэффициент пересчета скорости 64.

Т а б л и ц а 10.2

Входы БИС КР580ВВ51А			Адрес порта	Действие
G/D	RD	WR		
0	0	1	FFF0H	Чтение данных
0	1	0	FFF0H	Запись данных
1	0	1	FFF2H	Чтение состояния
1	1	0	FFF2H	Запись команды

Генератор скорости передачи, выполненный на двух двоично-десятичных счетчиках (микросхемы D52, D53), использует сигнал PCLK для формирования различных скоростей передачи данных. Для выбора скорости передачи необходимо установить соответствующий контакт переключателя s17 в положение «замкнуто», а осталь-

Т а б л и ц а 10.3

Скорость передачи, бод	Контакт переключателя
19200	1
9600	2
4800	3
2400	4
1200	5
600	6
300	7
150	8

ные контакты должны при этом быть разомкнуты. Соответствие контактов переключателя s17 скоростям передачи представлено в табл. 10.3.

ПРОГРАММНЫЕ СРЕДСТВА „МИКРОЛАБ 1810’

УРОК 11

ОРГАНИЗАЦИЯ ПАМЯТИ



МП КМ1810ВМ86 обеспечивает адресацию памяти емкостью 1 М байт (один миллион байт). Адресовое пространство памяти представляет собой одномерный массив байтов, каждый из которых имеет 20-битный физический адрес, простирающийся от 0 до $2^{20}-1$ (от 0000 0000 0000 0000 0000 до ПИ ШШ ШШ ШШ в двоичном коде, от 00000 до FFFFF в шестнадцатеричном коде (рис. 11.1).

ШЕСТИНАДЦАТЕРИЧНЫЙ АДРЕС	ДВОИЧНЫЙ АДРЕС	ПАМЯТЬ
00000	0000 0000 0000 0000 0000	[Memory cell diagram]
00001	0000 0000 0000 0000 0001	
00002	0000 0000 0000 0000 0010	
00003	0000 0000 0000 0000 0011	
00004	0000 0000 0000 0000 0100	
...		
FFFFE	1111 1111 1111 1111 1110	[Memory cell diagram]
FFFFF	1111 1111 1111 1111 1111	

Рис. 11.1. Адреса памяти

Про такую память говорят, что она имеет байтовую организацию. Можно рассматривать память иначе. Любые два последовательных байта в памяти называются словом. Каждый байт в слове имеет свой адрес и меньший из этих двух адресов используется в качестве адреса слова. Примеры слов показаны на рис. 11.2.

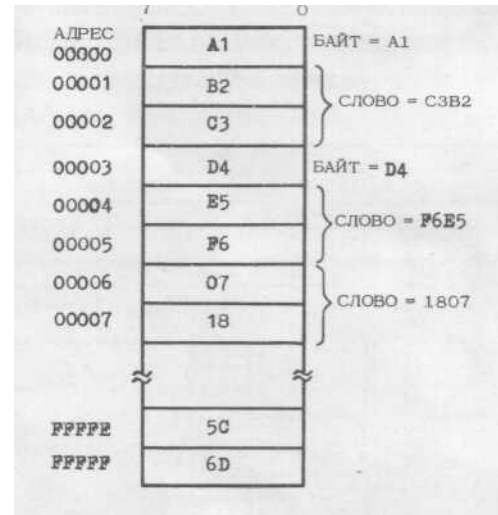


Рис. 11.2. Адресное пространство в памяти

В соответствии с рис. 11.2, адрес 00000 может обозначать и байт с этим адресом, что условно обозначается в виде $([00000])=A1$, и слово с таким же адресом, что записывается в виде $([00000])=B241$. Круглые скобки здесь заменяют слово "исодержимое", а в квадратные скобки заключается ячейка памяти, адрес которой находится в этих скобках.

Слово содержит 16 бит. Байт с большим адресом памяти содержит 8 старших бит слова и байт с меньшим адресом памяти содержит 8 младших бит, т.е. МП запоминает свои слова „наоборот“.

МП КМ1810ВМ86 имеет несколько команд, которые выбирают (записывают или считывают) байты,

и команды, которые выбирают слова. Однако информацию МП передает з память или из памяти всегда по 16 бит, поэтому в случае байтовых команд используются только 8 из этих бит, остальные 8 игнорируются.

Команды, байты и слова данных можно свободно размещать по любому адресу байта. Слова целесообразнее размещать в памяти до четным адресам, так как МП может передавать такие слова за один цикл канала. Слова с нечетными адресами также допустимы, но для их передачи требуются 2 цикла канала, что снижает производительность МП.

СЕКМЕНТАЦИЯ ПАМЯТИ

Так как МП КМ1810ВМ86 обеспечивает адресацию до 2^{20} байтов памяти, то оказывается, что внутри МП адреса байтов и слов должны быть представлены 20-битовыми величинами. Но МП был разработан для выполнения арифметических действий над 16-битовыми числами и, таким образом, адреса могут быть длиной только 16 бит. Поэтому необходим какой-то дополнительный механизм для формирования адресов.

Программы "видят" пространство памяти 1М байт как группу сегментов, определенных самой программой. Сегмент представляет собой логическую единицу памяти размером 64К байт. Он состоит из сложных ячеек памяти и является независимой и отдельно адресуемой единицей памяти. Каждому сегменту программой назначается базовый (начальный) адрес, являющийся адресом его первого байта в адресном пространстве памяти. Все сегменты начинаются на 16-битных границах памяти (адрес нацело делится на 16). В любой данный момент времени программа может непосредственно выбрать содержимое 4 таких сегментов: текущего кодового сегмента CS, текущего сегмента данных DS, текущего стекового сегмента SS и текущего экстракодового сегмента ES (текущий экстракодовый сегмент - область общего назначения, часто рассматриваемая в качестве дополнительного сегмента данных). Сегменты могут быть соседними неперекрывающимися, частично или полностью перекрывающимися (рис. П.3).

Предположим, что 16-битовый регистр кодового сегмента содержит шестнадцатеричную величину C018. Это приводит к тому, что кодовый сегмент начинается с адреса байта C0180 и простирается на целых 2^{16} байт. Поэтому последний байт кодового сегмента имеет адрес D017F

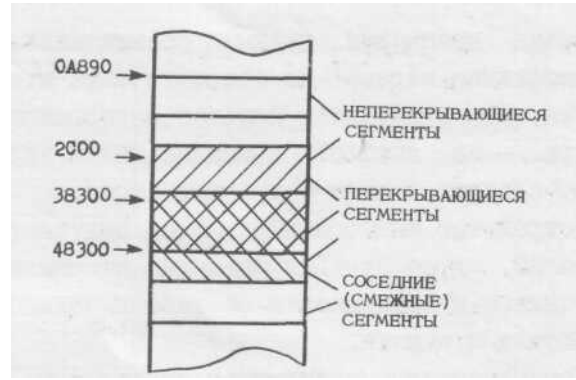


Рис. 11.3. Размещение сегментов в памяти

Удобно считать, что каждая ячейка памяти имеет два адреса: физический и логический. Физический адрес представляет собой 20-битное значение в диапазоне от 0 до FFFFF, которое однозначно идентифицирует положение каждого байта в пространстве памяти 1М байт. Физический адрес выдается на шину адреса в начале каждого цикла канала, связанного с обращением к памяти.

Программы оперируют не с физическими, а с логическими адресами, что позволяет разрабатывать программы не заботясь о том, как они размещаются в памяти. Логический адрес состоит из двух 16-битных беззнаковых значений: базового (начального) адреса сегмента, который называется также просто базой или сегментом, и внутрисегментного адреса или смещения. Для любой ячейки памяти база идентифицирует начало сегмента (т.е. первый байт), а смещение определяет расстояние в байтах от начала сегмента до этой ячейки. Нулевое смещение имеет байт с наименьшим внутрисегментным адресом, а максимальное смещение равно FFFF.

ДЕЛ образования физического адреса база сегмента сдвигается влево на 4 бита и суммируется со смещением (рис. 11.4), т.е. принцип формирования физического адреса предполагает, что абсолютные адреса сегментов оканчиваются

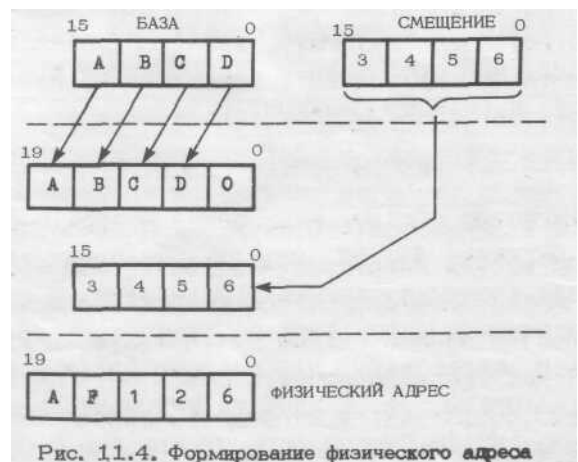


Рис. 11.4. Формирование физического адреса

четырьмя двоичными нулями. Возникающий при суммировании перенос из старшего бита игнорируется. Это приводит к кольцевой организации памяти, - за ячейкой с максимальным адресом FFFFF следует ячейка с нулевым адресом.

Устройство шинного интерфейса получает логический адрес ячейки памяти из различных источников в зависимости от типа выполняемого обращения к памяти.

Команды всегда выбираются из текущего сегмента кода - базовый адрес сегмента находится в регистре CS, а смещение в регистре IP. Стековые команды всегда обращаются к текущему сегменту стека: базовый адрес находится в регистре SS, а смещение в регистре SP. Считается, что большинство переменных (операндов) находится в другом сегменте (т.е. базовый адрес находится не в регистре DS, а в другом сегментном регистре).

ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА

МП КМ1810ВМ86 имеет достаточный набор

средств ввода-вывода: большое адресное пространство ввода-вывода, изолированное от пространства памяти, и специальные команды, которые передают данные между регистрами МП, и портами в пространстве ввода-вывода.

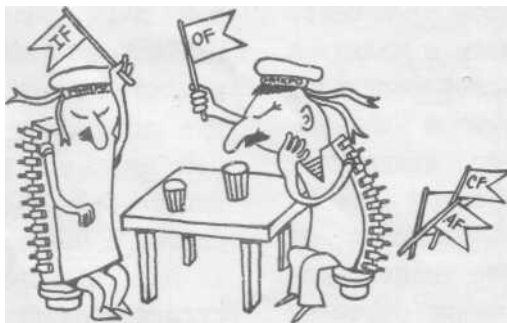
Адресное пространство ввода-вывода содержит до 64К 8-битных портов или до 32К 16-битных портов. Команды IN (ввод) и OUT (вывод) передают данные между аккумуляторами AL (байты) или AX (слова) и адресуемыми портами.

Пространство ввода-вывода не сегментируется, т.е. все порты считаются находящимися в I сегменте. 16-битовый порт (2 последовательных 8-битовых порта) с нечетным адресом будет требовать 2 выборки вместо 1.

Адрес битного устройства может быть четным или нечетным, но всем внутренним регистрам устройства должны быть поставлены в соответствие только четные или только нечетные адреса.

УРОК 12

ОРГАНИЗАЦИЯ РЕГИСТРОВ



МП КМ1810ВМ86 содержит 14 16-битовых регистров. Для наглядности регистры подразделяются на 4 группы (рис. 12.1). 3 группы содержат по 4 регистра, а четвертая объединяет указатель команд, непосредственно недоступный программисту, и регистр флажков,

РЕГИСТРЫ ОБЩЕГО НАЗНАЧЕНИЯ

Регистры общего назначения - это 16-битовые регистры AX, BX, CX, DX. Они предназначены для хранения данных и находятся в полном распоряжении программиста. Допускается раздельно адресовать их старшую (H) и младшую (L) половины, т.е. каждый из них можно использовать как 16-битный регистр или два 8-битных

регистры. Поэтому каждой половине любого регистра общего назначения присвоено свое собственное имя. Младшие половины имеют имена AL, BL, CL, DL, а старшие - AH, BH, CH и DH. Двойственный характер регистров позволяет им оперировать с одинаковой легкостью с байтами и словами.

В большинстве случаев регистры общего назначения могут взаимозаменяемо участвовать в арифметических и логических операциях. Например, с помощью команды ADD можно складывать содержимое любого 8- или 16-битового регистра общего назначения с любым другим регистром общего назначения такого же размера и запоминать результат в любом из регистров. Однако

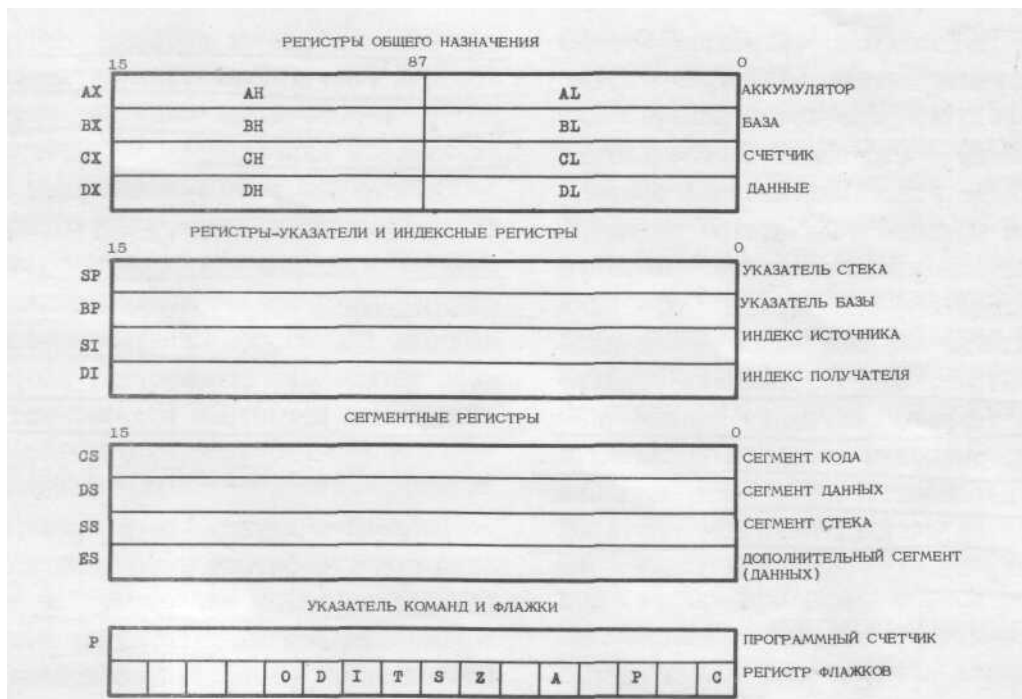


Рис. 12.1. Регистры МП КМ1810ВМ86

имеется много команд, которые специализируют регистры общего назначения на определенные функции, что отражено в их названиях (аккумулятор А, база в, счетчик о, данные D). Например, строчная команда требует, чтобы регистр cx (счетчик) содержал результат подсчета числа элементов в строке. Для этой цели не могут быть использованы регистры AX, vx, DX.

Недостаток специализированного применения регистров общего назначения состоит в том, что МП труднее программировать из-за более специфических правил запоминания.

Программы получаются наиболее компактными, если в командах арифметических и логических операций и в командах пересылки данных использовать аккумулятор АХ. Регистр AL в основном соответствует аккумулятору А МП КР580ИК80, а регистр АН называется расширением регистра AL. Регистр vx почти эквивалентен основному указателю памяти МП КР580ИК80 - регистровой паре HL и участвует в командах как источник базового адреса.

РЕГИСТРЫ-УКАЗАТЕЛИ И ИНДЕКСНЫЕ РЕГИСТРЫ

Группа регистров-указателей и индексных регистров состоит из 16-битных регистров SP, BP, si и DI. Они предназначены для хранения 16-битных адресов (внутрисегментных смещений) и обеспечивают при этом косвенную адресацию и динамическое вычисление эффективного адреса памяти.

Наряду с регистрами общего назначения они

дают возможность командам оперировать с ячейками, адреса смещения которых являются результатом предыдущих вычислений, выполненных во время прохождения программы. Это особенно важно при трансляции программ с языков высокого уровня. Регистры-указатели и индексные регистры могут участвовать в арифметических и логических операциях.

Между регистрами этой группы существуют некоторые различия, что приводит к делению группы на регистры-указатели SP и BP и индексные регистры si и DI.

Регистры-указатели предназначены для упрощения выборки данных из текущего сегмента стека, а не из сегмента данных. Такое применение стекового сегмента (как "Области данных") имеет определенные преимущества при использовании языков высокого уровня. Индексные регистры содержат смещения, относящиеся к сегменту данных. Таким образом, если сегмент специально не указан, то предполагается, что смещения, содержащиеся в регистрах-указателях, относятся к текущему стековому сегменту, тогда как смещения, содержащиеся в индексных регистрах, относятся к текущему сегменту данных.

В свою очередь, имеются особенности в использовании регистров-указателей внутри пары. Так команды PUSH и POP получают смещение для ячейки верхушки стека только из регистра SP. Регистр BP не может быть использован для этой цели. Он остается свободным для того, чтобы иметь возможность содержать смещение «базы»

области данных в стековом сегменте. Отсюда вытекает и название этих регистров STACK POINTER и BASE POINTER соответственно.

Строчные команды - это область использования индексных регистров. Смещение для операнда источника эти команды получают из регистра SI, а регистр DI содержит смещение операнда назначения. Называются они (исходя из этих функций) SOURCEINDEX и DESTINATION INDEX. Команда сдвига строки будет сдвигать строку, расположенную в текущем сегменте данных, начиная с ячейки, смещение которой хранится в регистре si, и переносить ее в экстракодовый сегмент, начиная с ячейки, смещение которой указывает регистр DI.

СЕГМЕНТНЫЕ РЕГИСТРЫ

Наличие 4 сегментных регистров объясняется способом адресации памяти. Хотя МП КМ1810ВМ86 имеет 20-битную шину физического адреса и адресует таким образом один мегабайт памяти, он оперирует 16-битными логическими адресами, состоящими из базового адреса сегмента и внутрисегментного смещения. Внутреннее устройство преобразования адресов превращает 2 логических адреса в 20-битный физический адрес.

1М байт памяти МП разделен на логические сегменты емкостью 64К байт (т.е. внутри сегмента для адресации достаточно 16-битового регистра). В ходе выполнения программы МП может одновременно обращаться к 4 сегментам. Их базовые (начальные) адреса и содержатся в сегментных регистрах CS (кода), DS (данных), SS (стека) и ES (экстракода).

Регистр CS указывает на текущий сегмент кода (программы), откуда выбираются команды. Регистр SS адресует текущий сегмент стека, в этом сегменте реализуются все стековые операции. Регистр DS указывает на текущий сегмент данных, в котором содержатся переменные. Регистр ES определяет текущий экстракодовый сегмент, использующийся для хранения данных.

В большинстве команд содержится только одна часть логического адреса - внутрисегментное смещение. Оно может быть вычислено в соответствии с указанным режимом адресации, может находиться в команде или содержаться в общем регистре.

Физический 20-битный адрес формируется путем суммирования смещения и содержимого одного из сегментных регистров. В системах, об-

щая емкость памяти которых составляет не более 64К байт, сегментные регистры можно привести к лог. 0, и они не будут влиять на физические адреса.

Сегментные регистры доступны программе, их содержимым манипулирует несколько команд. При программировании необходимо очень внимательно следить за упорядоченным использованием сегментных регистров. Следует отметить, что размер сегментов фиксирован, а средства защиты сегментных регистров отсутствуют.

УКАЗАТЕЛЬ КОМАНД И ФЛАЖКИ

Последнюю группу регистров МП КМ1810ВМ86 образуют 16-битный указатель команд IP и регистр флажков.

Все команды вызываются из текущего кодового сегмента. Поэтому необходим регистр, содержащий смещение в кодовом сегменте следующей команды, которая должна быть выполнена. Это функция регистра IP (INSTRUCTION POINTER)-аналога стандартного программного счетчика. Шинный интерфейс осуществляет модификацию IP таким образом, что он содержит смещение следующей команды от начала текущего сегмента кода, т.е. указывает на следующую по порядку команду. Непосредственный доступ к IP имеют команды передачи управления.

Формат 16-битного регистра флажков приведен на рис. 12.2.



Рис. 12.2. Формат регистра флажков МП КМ1810ВМ86

Младший байт этого регистра полностью соответствует регистру флажков МП КР580ИК80, а старший байт содержит 4 флажка, аналога которых нет в МП КР580ИК80.

6 арифметических флажков фиксируют определенные признаки результата арифметической или логической операции. Группа команд условных переходов позволяет изменить ход программы в зависимости от состояний флажков. Флажки отражают следующие условия:

AF (флажок вспомогательного переноса) - фиксирует перенос (заем) из младшей тетрады в старшую 8- или 16-битного результата. Он необходим только для команд десятичной арифметики;

CP - фиксирует значение переноса (заема), возникающего при сложении (вычитании) байт или слов, а также значение выдвигаемого бита при сдвиге операнда;

OF (флажок переполнения) - сигнализирует о потере старшего бита результата сложения или вычитания. Имеется специальная команда прерывания при переполнении, которая в данной ситуации генерирует программное прерывание;

SP (флажок знака) - повторяет значение старшего бита результата, который при использовании дополнительного кода соответствует знаку числа;

PF (флажок паритета, или четности) - фиксирует наличие четного числа единиц в младших 8 бит результата операции. Этот флажок предназначен для контроля правильности передачи данных;

ZF (флажок нуля) - сигнализирует о получении нулевого результата операции;

3 дополнительных флажка предназначены для управления некоторыми действиями МП:

DP (флажок направления) - определяет порядок сканирования цепочки в соответствующих

командах: от меньших адресов к большим (DF=0) или наоборот (DF=1);

IP (флажок прерывания) - определяет реакцию МП на запросы внешних прерываний по входу INT. Если уровень IP - лог. 0, запросы прерываний игнорируются (прерывания замаскированы), а если уровень IP - лог. 1 МП распознает и соответственно реагирует на запрос прерывания. Однако, состояние флажка IP не влияет на восприятие внешних немаскируемых прерываний по входу NMI, а также внутренних (программных) прерываний;

TF (флажок прослеживания (трассировки)) - установка его в лог. 1 переводит МП в одношаговый режим работы, который применяется при отладке программ. В этом режиме МП автоматически генерирует внутреннее прерывание после выполнения каждой команды и переходит к соответствующей подпрограмме обработки, которая может, например, индентифицировать содержимое внутренних регистров.

Предусмотрены специальные команды, с помощью которых программист может задавать требующиеся ему состояния любого из этих флажков (кроме TF).

УРОК 13

РЕЖИМЫ АДРЕСАЦИИ



При выполнении любой программы МП обращается к памяти, в которой хранятся команды и данные. В командах преобразований данных определяются адреса, которые указывают местоположение необходимых данных, а в командах передачи управления определяются адреса команд, которым передается управление, т.е. адреса переходов. Способ (или метод) определения в команде адреса операнда или адреса перехода, называется

режимом адресации или ПРОСТО

В наиболее простом режиме адресации, называемом прямой адресацией, адрес находится в самой команде. Однако использование этого режима, хотя и предусмотрено в МП, приводит к чрезмерной длине команд, особенно в условиях постоянно увеличивающейся емкости памяти. Поэтому в МП применяется много других режимов адресации.

Назначением режима адресации является указание способа формирования эффективного (или исполнительного) адреса ЕА. Этот адрес является либо адресом операнда (в командах, оперирующих данными), либо адресом перехода (в командах передачи управления). В МП ЕА памяти представляет собой 16-битовое беззнаковое целое, являющееся смещением относительно базы некоторого сегмента. Полный (физический) адрес памяти формируется привлечением одного из сегментных регистров.

Режимы адресации подразделяются на прямые и косвенные. При прямой адресации ЕА либо содержится в команде, либо вычисляется с использованием значения, находящегося в команде, и содержимого указанного в команде регистра (или двух регистров). При косвенной, адресации ЕА в команде определяет регистр или ячейку памяти, содержащую окончательный эффективный адрес.

/ ПРЯМАЯ АДРЕСАЦИЯ

Как уже говорилось, существует несколько режимов адресации. Один из них - абсолютная адресация, при которой задается непосредственно номер области памяти. Для тех случаев, когда требуется обеспечить эффективное управление памятью и минимизировать длину команды, этот способ не подходит. Среди вычислительных машин с МП КМ18IOBM86 имеются различные модели с разным объемом памяти (максимальный 1М байт). Эти байты пронумерованы последовательно: первый имеет номер 0, второй - номер 1 и т.д. до последнего байта.

Мы информируем ЭВМ о том, что она должна делать, с помощью машинных команд. Обычно команда содержит адрес, данных, над которыми должны быть выполнены действия, ИЛИ адрес, по КОТОРОМУ хранится следующая команда. Следовательно, для обращения к любой из возможных областей памяти с использованием двоичного адреса команда должна располагать 20 битами для формирования любого адреса. Так как память ЭВМ довольно дорогая и, кроме того, имеет ограниченный объем, был разработан способ адресации с использованием всего 16 бит. Этот способ известен как относительная адресация.

ОТНОСИТЕЛЬНАЯ АДРЕСАЦИЯ

Относительная адресация - это способ обращения к любой области памяти, в соответствии с которым определенные фиксированные адреса

используются в качестве точек отсчета. В МП КМ18IOBM86 применяются 2 метода вычисления относительных адресов; 1) с использованием базы и смещения, или 2) использованием индекса, базы и смещения.

1) Адресация с использованием базы и смещения.

Данный способ адресации состоит в следующем. Программист выбирает адрес, называемый базовым, который он помещает в регистр общего назначения (используемый в этом случае регистр называется регистром базы). Тем самым программист вводит точку отсчета, называемую базой; эта точка является началом отсчета при вычислении в машине действительного, или исполнительного адреса программы. Таким образом,

$$\text{Исполнительный адрес} = \text{Содержимое регистра базы} + \text{Смещение,}$$

где смещение - это количество байтов памяти, на которое адрес смещен (удален) по отношению к номеру, находящемуся в регистре базы. Если, например, регистр базы содержит число 1027, а обратиться необходимо по адресу 4127, то этот адрес можно описать как смещенный на 3100 байт по отношению к области памяти с номером 1027 (рис. 13.1).

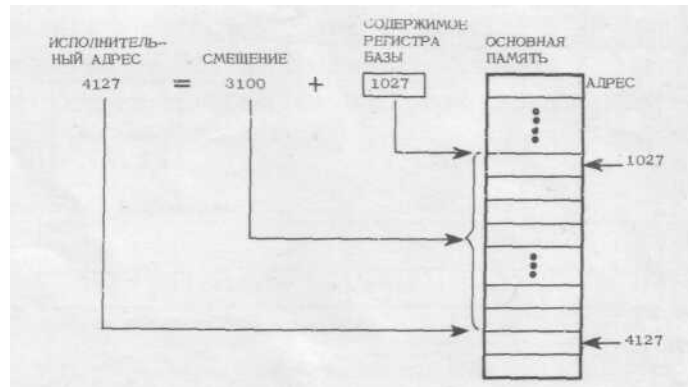


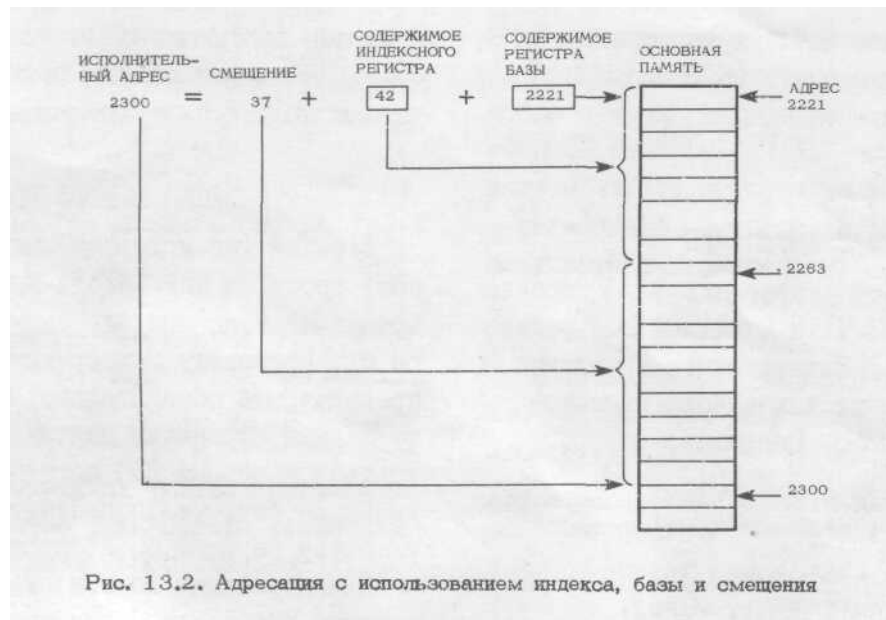
Рис. 13.1. Способ адресации с использованием базы и смещения

Адресация с использованием индекса, базы и смещения.

В соответствии с этим методом программист выбирает два регистра общего назначения: один из них - это описанный выше регистр базы, а второй - индексный регистр. В индексном регистре так же, как и в регистре базы, содержится число, которое используется для вычисления исполнительного адреса (рис. 13.2).

$$\text{Исполнительный адрес} = \text{Содержимое регистра базы} + \text{Содержимое индексного регистра}$$

После загрузки регистра базы (т.е. после



записи в него числа) его содержимое не может быть изменено в процессе выполнения программы. Что же касается индексного регистра, то его содержимое как раз можно изменять. Это обстоятельство оказывается полезным при разработке программ. Решение вопроса о том, какой из способов адресации предпочтительнее, зависит от типа используемой команды.

В командах, адресующихся к памяти посредством базы-смещения или индекса-базы-смещения для смещения отводится 12 бит. Смещение указывает, на какое количество байтов адресуемая область памяти смещена по отношению к базовому адресу (числу, находящемуся в регистре базы), и является, таким образом, всегда положительной величиной.

В команде для описания номера регистра отведено 4 бита. Регистры нумеруются от 0-го до 15-го, и с помощью 4 бит может быть задан любой из этих номеров. Смещение (12 бит) и номер регистра (4 бит) совместно определяют номер любого байта памяти.

Достоинством относительной адресации является то, что она позволяет перемещать программу. Перед очередным выполнением программы ЭВМ может разместить ее в другой области памяти. От ЭВМ не требуется, чтобы она модифицировала адреса команд и данных в перемещаемой программе, корректировке подлежит только значение базового адреса. Относительные расстояния других программных адресов от базового остаются неизменными: значения смещений сохраняются прежними. Если бы применялась абсолютная адресация, то ситуация была бы иной, так как при использовании определен-

ных чисел, присвоенных ячейкам памяти, подлежащую выполнению программу каждый раз приходилось бы загружать в одну и ту же область памяти.

СКОЛЬКО ОПЕРАНДОВ В КОМАНДЕ

Для понимания дальнейшего необходимо понять какую роль играют операнды в командах МП.

Для выполнения различных команд требуется различное количество данных. Например, вычисление по сложной формуле может потребовать несколько десятков, а то и сотен различных значений. К тому же необходимо иметь место в памяти, куда поместить результат. С другой стороны, для остановки работы МП, например, не нужно никаких данных. Таким образом, очевидно, что возможностей бесконечно много. МП не может обеспечить все случаи, поэтому существует ряд ограничений. Вот они:

у команды может быть не больше двух операндов;

один из операндов (если они есть) является приемником для результата операции, источниками данных могут быть оба операнда.

В случае двух операндов приемник информации называется операндом-приемником и при записи команды располагается левее другого операнда. Второй операнд называется операндом-источником и при записи команды располагается правее операнда-источника, отделяется запятой.

РЕГИСТРОВАЯ АДРЕСАЦИЯ

Используется в том случае, если оба операнда расположены в регистрах. Например:

· CM r7 (57)

При записи команды базовую адресацию можно указать и другим способом:

ADD [BX+TEMP], CX ; то же самое, что и предыдущий ; пример

ИНДЕКСНАЯ АДРЕСАЦИЯ

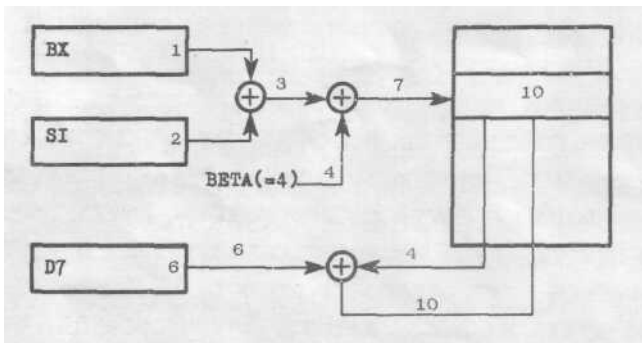
В режиме индексной адресации EA операнда вычисляется МП аналогично вычислению в режиме базовой адресации. Единственное различие - использование значения индексного регистра или DI вместо базового. Например:

MOV ARRAY [Si], AX ; передать значение из регистра AX в элемент массива

БАЗОВАЯ ИНДЕКСНАЯ АДРЕСАЦИЯ

Этот режим является наиболее гибкой адресацией. EA операнда вычисляется суммированием значений базового, индексного регистров и смещения. Например:

ADD [BX] BETA [Si], DI ; прибавить к слову в памяти содержимое регистра DI



АДРЕСАЦИЯ ОТНОСИТЕЛЬНО ТЕКУЩЕГО АДРЕСА

При таком режиме адресации EA вычисляется как сумма непосредственного смещения и содержимого программного счетчика PC, т.е., если рассматривать PC в роли базового регистра, описываемый режим аналогичен базовой адресации, но может применяться только в командах переходов, вызовов подпрограмм и управления циклами.

АДРЕСАЦИЯ ЦЕПОЧЕК

Режим адресации цепочек данных в памяти аналогичен индексной адресации по механизму вычисления EA операнда. Отличием является то, что для получения EA операнда-источника обязательно используется индексный регистр si, а для вычисления EA операнда-приемника - только индексный регистр DI. В повторяющихся опера-

циях над цепочками МП автоматически корректирует содержимое регистров si и DI по мере перехода к следующим элементам цепочек.

АДРЕСАЦИЯ ПОРТОВ ВВОДА-ВЫВОДА

Для доступа к портам, находящимся в адресном пространстве ввода-вывода, используются прямая адресация и адресация с помощью регистра DX. Поскольку непосредственный адрес может представлять собой только 8-битовую константу, то прямая адресация портов обеспечивает доступ только к первым 256 портам. Регистр DX имеет длину 16 бит, что позволяет с его помощью обращаться ко всем 65536 портам ввода-вывода МП.

При адресации портов не используются сегментные регистры, поэтому здесь адресация аналогична прямой.

КАК ПОСТРОИТЬ КОД КОМАНДЫ

Сначала подведем некоторые итоги.

1. Команда может адресовать не более 2 операндов.
2. Операнд-приемник может быть адресом памяти, регистром или портом ввода-вывода.
3. Операнд-источник может быть адресом памяти, регистром, портом ввода-вывода или непосредственно числом.
4. Приемник и источник одновременно не МОГУТ быть адресами памяти или портов ввода-вы-

МП должен определить:

- что он хочет сделать с адресуемым операндом - взять из него данные или записать их;
- с каким количеством информации (байт, слово=2 байта) он собирается работать;
- какой режим адресация используется.

Посмотрим, как все это определяется кодом команды.

Наиболее общий формат команды приведен на рис. 13.3. 3-й и (или) 4-й байты команды могут отсутствовать, если их наличие не требуется выбранным режимом адресации.

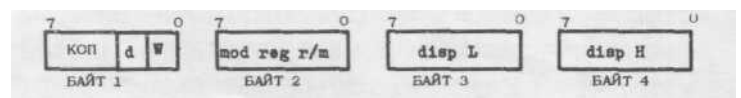
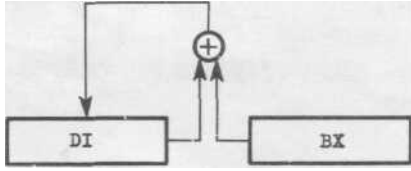


Рис. 13.3. Наиболее общий формат команды

Байт I содержит код команды, для указания МП, что, собственно, от него требуют. Здесь могут быть коды самых разных действий: сложить, вычесть, умножить, разделить и т.д.

См. стр 58

ADD DI, BX ; прибавить содержимое BX к содержащему DI и поместить результат в DI



Значение BX при этом не изменяется.
Еще пример:

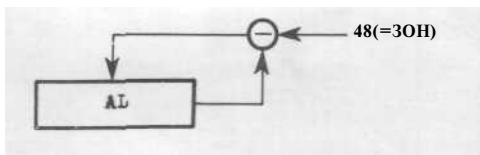
MOV AX, SI ; передать содержимое SI в AX

В приведенных примерах AX и DI называются операндами-приемниками (или просто приемниками), а BX и SI - операндами-источниками (просто источниками).

НЕПОСРЕДСТВЕННАЯ АДРЕСАЦИЯ

Используется в том случае, если источником является непосредственное значение (называется литералом). Стандартный для МП литерал имеет длину 16 бит, а короткий - 8 бит (при необходимости он расширяется до 16 бит). Например:

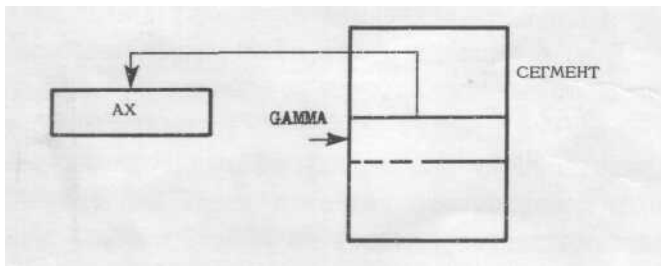
SUB AL, 30H ; вычесть из содержимого AL число 48



ПРЯМАЯ АДРЕСАЦИЯ

Прямая адресация (не забывайте о сегментации!) является простейшим способом определения эффективного адреса. ЕА непосредственно указывается в команде, никакие регистры для его вычисления не привлекаются. Например:

MOV AX, GAMMA ; загрузить в AX переменную GAMMA



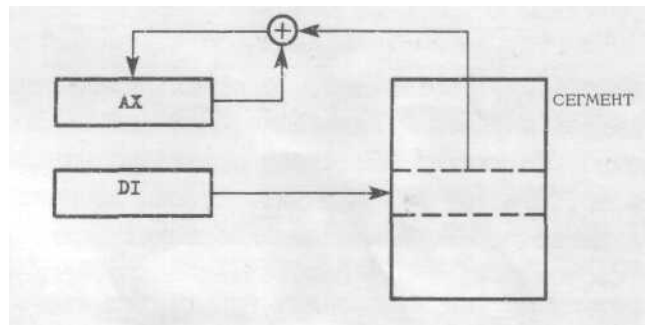
GAMMA фактически является конкретным числом (например 296), указывающим смещение ячейки с исходными данными относительно начала сегмента.

В командах межсегментного перехода и вызова подпрограмм может использоваться разновидность прямой адресации - длинная прямая адресация. В этом случае непосредственно указываются число, которое должно быть загружено в сегментный регистр, и смещение от начала получившегося сегмента.

КОСВЕННАЯ РЕГИСТРОВАЯ АДРЕСАЦИЯ

В этом режиме ЕА операнда находится в одном из базовых или индексных регистров. Например:

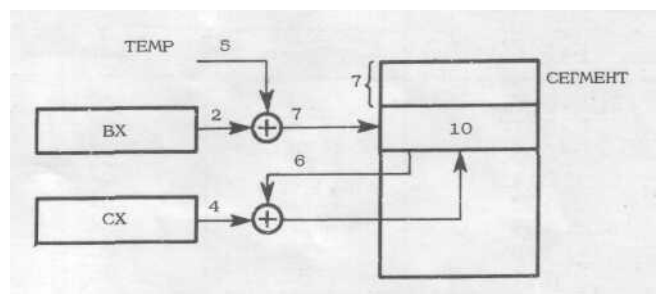
DD AX, [DI] ; прибавить к содержимому AX содержимое ячейки памяти, адресуемой смещением от начала сегмента. Величину смещения взять из регистра DI. Результат сложения поместить в AX.



БАЗОВАЯ АДРЕСАЦИЯ

При базовой адресации ЕА операнда вычисляется МП суммированием значения смещения и содержимого регистров BX или BP используемых как базовые. Например:

ADD TEMP, CX ; прибавить к содержимому слова в памяти, адресуемому с помощью BX и заданного смещения TEMP, содержимое регистра CX



Предположим, что TEMP=5, в BX записано число 2, в CX записано число 4, в ячейку памяти со смещением 7 от начала сегмента записано число 6. Тогда команда исполнится следующим образом:

$$EA = TEMP + BX = 5 + 2 = 7$$

$$\text{результат} = [7] = [7] + CX = 6 + 4 = 10$$

см стр 56

Второе поле *d* - это один бит, указывавший откуда нужно взять данные и куда их поместить. При помощи этого бита МП может точно указать, что он хочет сделать с каждым операндом: взять из него данные или записать их.

Третье однобитовое поле *w* определяет размер операндов. Если *w*=0, МП работает с байтами, а если *w*=1, - со словами.

Таким образом, в зависимости от значений *d* и *w*, имеются 4 возможности (табл. 13.1):

Т а б л и ц а 13.1

<i>d</i>	<i>w</i>	Передача или операция
0	0	Байт из <u>регистра</u> в память или регистр
0	1	Слово из <u>регистра</u> в память или регистр
1	0	Байт из памяти или регистра в <u>регистр</u>
1	1	Слово из памяти или регистра в <u>регистр</u>

Байт 1, не объясняет, о каких конкретно регистрах и режимах адресации идет речь. Его конечно не хватит МП, чтобы правильно выполнить то, что от него хотят. Чтобы до конца объясниться с МП, в команду добавляют байт 2. Он состоит из трех полей: *mod*-режим, *reg* - регистр, *r/m* - регистр/память.

Поле *reg* состоит из 3 бит и говорит о том операнде, который расположен в регистре. Собственно говоря, из байта 1 команды уже известно, как используется регистр (как операнд-источник или операнд-приемник), поэтому на долю поля *reg* остается показать какой конкретно регистр МП должен быть использован. Как это делается, видно в табл. 13.2.

Т а б л и ц а 13.2

Поле <i>reg</i> (<i>r/m</i>)	Адресация регистров	
	8-битных	16-битных
000	AL	AX
001	CL	CX
010	DL	DX
011	BL	BX
100	AH	SP
101	CH	BP
110	DH	SI
111	BH	DI

Само по себе значение поля *reg* не определило конкретно регистра. Например, *reg*=111 может показать и на DI, и на BH. Но если вспомнить о поле *w*, то можно легко ответить на этот вопрос: если *w*=0, то регистр 8-битный, т.е. это BH; если *w*=1, то - 16-битный (DI). Говоря о поле *reg* очень важно помнить, что оно описывает тот операнд, который всегда рас-

положен в регистре. В табл. 13.1 эти операнды подчеркнуты.

Поле *mod* определяет используемый режим адресации (впрочем вместе с полем *r/m*) и присутствие или отсутствие байтов 3 и 4 (табл. 13.3).

Т а б л и ц а 13.3

Поле <i>mod</i>	Режим адресации
00	Нет байта 3 и 4
01	Нет байта 4 (8-битное смещение)
10	Есть байты 3 и 4 (16-битное смещение)
11	Регистровый режим адресации

Итак, при *mod*=11 сразу определяется тип адресации. В этом случае поле *r/m* по своему смыслу похоже на поле *reg*. Только определяет оно тот регистр, который не подчеркнут в табл. 13.1.

В остальных случаях для определения режима адресации привлекается поле *r/m* (табл. 13.4).

Т а б л и ц а 13.4

Поле <i>r/m</i>	EA
000	BX+SI+disp
001	BX+DI+disp
010	BP+SI+disp
011	BP+DI+disp
100	SI+disp
101	DI+disp
110	BP+disp
111	BX+disp

Здесь *disp* - это смещение, которое уже понятно. Чтобы узнать его длину (байт или слово), загляните в предыдущую таблицу: отношение у полей *mod* и *r/m* такое же, как у полей *reg* и *w*.

Из всего сказанного есть только одно исключение: *mod*=00, *r/m*=110. В этом случае у команды есть байты 3 и 4 смещения и EA=*disp*. Это случай непосредственной адресации.

Если в команде есть непосредственный операнд, то он добавляется к команде байтами 5 и 6 (сколько нужно). Поле *reg* в этом случае не нужно, поэтому оно используется как расширение кода команды. Это делается так: код операции ограничивается длиной в 8 бит. 5 бит помещаются там, где им и положено быть, а 3 бита отрезаются и помещаются, как указано на рис. 13.4.

Очевидно, что поле *d* изменило название на *a*. Это произошло потому, что теперь вмес-

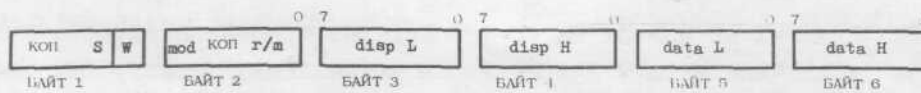


Рис. 13.4. Формат команды при наличии в ней непосредственного операнда

те с полем w оно определяет байты 5 и 6 (табл. 13.5).

Таблица 13.5

Поле		Режим адресации
S	W	
Любое	0	Отсутствует байт 6
0	1	Присутствуют оба байта
1	0	Байт 6 отсутствует, но при исполнении команды он расширится до 16 бит

Все остальные поля своих значений не меняли.

Если в команде один операнд, то единственным отличием является то, что поле кода операции стало еще длинней, захватив в себя поле S из предыдущего пункта (рис. 13.5).

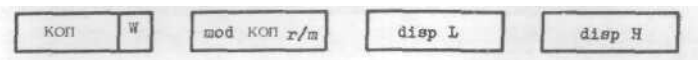


Рис. 13.5. Формат команды с одним операндом

Все исключения в форматах команд МП

КМ1810ВМ86 избыточные дублирующие форматы, единственная цель которых - сократить длину команда. Эти форматы есть не для всех команд, а преимущественно для наиболее часто употребляемых операций - например с аккумуляторами AL и AX. На рис. 13.6 приведены нормальный (рис. 13.6,а) и сокращенный (рис. 13.6,б) форматы команды передачи непосредственного операнда в регистр.

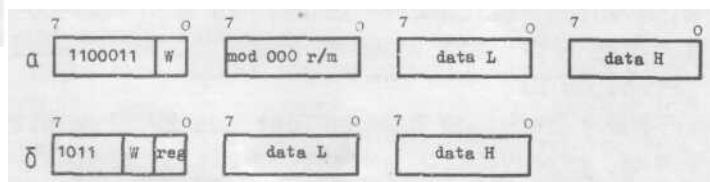


Рис. 13.6. Нормальный (а) и сокращенный (б) форматы команды передачи непосредственного операнда в регистр

ручном конструировании команд этих форматов следует избегать, так как они нерегулярны, их трудно запомнить, а ошибиться очень легко. При трансляции программы ассемблер сам выберет нужный формат и помнить о нем программисту нет никакой необходимости.

УРОК 14

ОРГАНИЗАЦИЯ СИСТЕМЫ КОМАНД



Система команд МП содержит 135 базовых команд, которые можно разделить на 6 функциональных групп:

- 1)- пересылка данных;
- 2)- арифметика;
- 3)- логика;
- 4)- операции со строками;
- 5)- передача управления;
- 6)- управление.

КОМАНДЫ ПЕРЕСЫЛКИ ДАННЫХ

Команды пересылки данных предназначены для пересылки операнда-источника на место операнда-приемника. Существуют 14 команд для пересылки данных, которые делятся на 4 группы:

- общего назначения;
- указывающие аккумулятор;
- адресных объектов;
- регистров флагов.

Команда пересылки данных не влияют на установку флагов, за исключением SAHF и POPF.

Команда общего назначения. Имеются 4 операции пересылки данных общего назначения. Они могут применяться для большинства операндов, хотя есть некоторые исключения. Это операции (кроме XCHG), в которых (и только в них) разрешается указывать в качестве операнда регистр сегмента.

MOV - проводит пересылку байта или слова из операнда-источника (правого) в операнд-приемник (левый);

PUSH - уменьшает значение регистра SP на 2, а затем пересылает слово из операнда-источника в текущий элемент стека, адресованный регистром SP;

POP - передает словный операнд из элемента стека, адресуемого регистром SP, в операнд-приемник, а затем увеличивает содержимое SP на 2;

XCHG - обменивает содержимое (байт или слово) операнда-источника с содержимым операнда-приемника. Сегментные регистры не могут быть операндами XCHG.

Команды, указывающие аккумулятор.

IN - передает байт (или слово) из входного порта в регистр AL (или AH). Либо порт указывает как непосредственный байт данных, разрешающий фиксированный доступ к портам 0...255, либо номер порта размещается в регистре DX, позволяя переменный доступ к 64К входных портов;

OUT - аналогично IN, за исключением того, что передача происходит из аккумулятора в выходной порт;

XLAT - проводит табличную перекодировку байтов. Регистр используется как индекс в 256-байтной таблице, адресуемой регистром BX. Байтный операнд, выбранный таким образом, передается в регистр AL.

Команды адресных объектов.

LEA (загрузка эффективного адреса) - передает смещение адреса операнда-источника в операнд-приемник. Операнд-источник должен быть операндом-памятью, а операнд-приемник должен быть 16-битным общим регистром, регистром-указателем или индексным регистром;

LDS(загрузка указателя в регистр DS)-передает «объект-указатель» (т.е. 32-битный объект, содержащий смещение адреса и адрес сегмента) из операнда-источника (который должен быть двойным словом памяти) в пары ре-

гистров приемника. Адрес сегмента направляется в сегментный регистр DS. Смещение адреса передается, в 16-битный общий регистр, регистр-указатель или индексный регистр, который закодирован;

LES (загрузка указателя в регистр ES) - аналогично команде LDS, за исключением того, что адрес сегмента передается в регистр сегмента ES.

Команды регистров флагов.

LAHF (загрузка регистра AH флагами) - пересылает регистры флагов SF, ZF, AF, PF, CF (флаги KP580ИК80) в определенные биты регистра AH;

SAHF (запоминание регистра AH во флагах) - передает определенные биты регистра AH в регистры флагов SF, ZF, AF, PF, CF;

PUSHF(передача флагов в стек) - уменьшает регистр SP на 2, а затем передает все регистры флагов в определенные биты элемента стека, адресуемого регистру SP;

POPF(передача флагов из стека) - передает определенные биты элемента стека, адресуемого регистром SP, в регистры флагов, а затем увеличивает регистр SP на 2.

2) АРИФМЕТИЧЕСКИЕ КОМАНДЫ

МП КМ18IOBM86 обеспечивает 4 основных математических операции над числами различных типов. Выполняются операции над 8- и 16-битными операндами знаковой и беззнаковой арифметики. К знаковым значениям применяется стандартное дополнение до 2. Операции сложения и вычитания имеются для знаковых и беззнаковых операндов. В этих случаях установки флагов различны для беззнаковых и знаковых операций. Для обеспечения арифметических операций над упакованными десятичными цифрами или над упакованными десятичными представлениями (Имеется в виду упакованный и неупакованный двоично-десятичный код (BCD), в котором каждая десятичная цифра представляется 4-битным двоичным эквивалентом. Для неупакованного формата в 1 байте 4 младших бита содержат десятичную цифру, а 4 старших - равны 0. Для упакованного формата в 1 байте закодированы 2 десятичные цифры.) имеются операции коррекции.

Установки регистров флагов. 6 регистров флагов устанавливаются в состояние лог. 1 или очищаются при арифметических операциях для того, чтобы отразить некоторые особенности результата операции. Установка или сброс ре-

гистров флагов выполняются по следующим правилам:

CF - устанавливается в состояние лог. 1, если результат операции приводит к переносу (при сложении), или к заему (при вычитании) в старший бит результата, иначе регистр очищается;

AF - устанавливается в состояние лог. 1, если результаты операции вызывают перенос (при сложении), или заем (при вычитании) из 4-го бита в младшем байте результата; иначе регистр очищается;

ZF - устанавливается в состояние лог. 1, если результат операции равен 0, иначе регистр очищается;

SF - устанавливается в состояние лог. 1, если старший бит результата равен 1, иначе регистр очищается;

Pf - устанавливается в состояние лог. 1, если сумма по модулю 2, 8 битов младшего байта результата операции равна 0 (четность), иначе регистр очищается (нечетность);

OF - устанавливается в состояние лог. 1, если результаты операции вызывают перенос в старший бит результата или наоборот; иначе регистр очищается.

Сложение. Имеются 5 операций сложения:

ADD - выполняет сложение операндов (источника и приемника) и возвращает результат в операнд-приемник;

ADC (сложение с переносом) - выполняет сложение операндов (источника и приемника), а затем добавляет 1, если флаг CF был предварительно равен 1; результат возвращается в операнд-приемник;

INC (приращение) - выполняет прибавление 1 к операнду-источнику и возвращает результат в операнд;

AAA (коррекция неупакованных значений кода КОИ-7 для сложения) - выполняет коррекцию результата в регистре AL, полученного от сложения 2 неупакованных десятичных чисел, устанавливая неупакованную десятичную сумму;

DAA (десятичная коррекция для сложения) - выполняет коррекцию результата в регистре AL, полученного от сложения 2 упакованных десятичных чисел, устанавливая упакованную десятичную сумму.

Вычитание. Имеются 7 операций вычитания:

SUB - выполняет вычитание операнда-источника из операнда-приемника и возвращает результат в операнд-приемник;

SBB (вычитание с заемом) - выполняет вычитание операнда-источника из операнда-приемника, а затем вычитает 1, если флаг CF был предварительно равен 1; результат возвращается в операнд-приемник;

DEC (уменьшение) - выполняет вычитание 1 из операнда-источника и возвращает результат в операнд;

NEG (отрицание) - выполняет вычитание операнда источника из нуля и возвращает результат в операнд;

CMR (сравнение) - выполняет вычитание операнда-источника из операнда-приемника, устанавливая соответствующие флаги, но не возвращает результата;

AAS (коррекция неупакованных значений кода КОИ-7 для вычитания) - выполняет коррекцию результата в регистре AL, полученного от вычитания неупакованных десятичных операндов, устанавливая неупакованную десятичную разность;

DAS (десятичная коррекция для вычитания) - выполняет коррекцию результата в регистре AL, полученного от вычитания 2 упакованных десятичных операндов, устанавливая упакованную десятичную разность.

Умножение. Имеются 3 операции умножения:

MUL - выполняет беззнаковое умножение аккумулятора (AL или AX) и операнда-источника, возвращая результат двойной длины в аккумулятор и его расширение (AL и AH - для 8-битных операндов, AX и DX - для 16-битных операндов). Регистры CF и OF устанавливаются в состояние лог. 1, если старшая половина результата ненулевая;

IMUL (целое умножение) - аналогично MUL, за исключением того, что выполняется знаковое умножение. Регистры CF и OF устанавливаются в состояние лог. 1, если старшая половина результата не является знаковым расширением младшей части результата;

AAM (коррекция неупакованных значений кода КОИ-7 для умножения) - выполняет коррекцию результата в AX, полученного от умножения 2 неупакованных десятичных операндов, устанавливая неупакованное десятичное произведение.

Деление. Имеются 3 операции деления и 2 операции со знаковым расширением для выполнения знакового деления:

DIV - выполняет беззнаковое деление аккумулятора и его расширения (AL и AH для 8-битных операндов, AX и DX для 16-битных операндов);

дов) на операнд-источник и возвращает частное одинарной длины в аккумулятор (AL или AX, соответственно) и остаток одинарной длины в расширение аккумулятора (AH или DX). Флаги - неопределенные, деление на ноль вызывает прерывание нулевого типа;

IOIV (целое деление) - аналогично DIV, кроме того выполняется знаковое деление;

AAD (коррекция неупакованных значений кода КОИ-7 для деления) - выполняет коррекцию делимого в регистре AL перед делением 2 неупакованных десятичных операндов, результат - неупакованное десятичное частное;

CBW (преобразование байта в слово) - выполняет знаковое расширение регистра AL в регистр AH;

CWD (преобразует слово в двойное слово) - выполняет знаковое расширение регистра AX в регистр DX.

3) ЛОГИЧЕСКИЕ КОМАНДЫ

KM1810BM86 выполняет логические операции над 8- и 16-битными операндами.

Однооперандные операции. Имеются 3 логические операции с одним операндом.

NOT - формирует обратный код от операнда-источника и возвращает результат в операнд. Флаги не устанавливаются.

Операции сдвига четырех разновидностей для операндов типа память и регистровых операндов:

SHL (логический сдвиг влево); SHR (логический сдвиг вправо); SAL (арифметический сдвиг влево); SAR (арифметический сдвиг вправо).

Возможны сдвиги как на один бит, так и на переменное число битов; счетчик сдвигов находится в регистре CL. Флаг CF равен значению последнего сдвигаемого бита. Регистр OF определяется только для сдвигов со счетчиком, равным I, и устанавливается в состояние лог. I, если только результирующее значение знакового бита отличается от его предыдущего значения; флаги PF, SF, ZF устанавливаются в зависимости от значения результата;

Циклические сдвиги 4 разновидностей для операндов типа „память” и регистровых операндов: ROL (циклический сдвиг влево); ROR (циклический сдвиг вправо); RCL (циклический сдвиг влево через CF); RCR (циклический сдвиг вправо через CF).

Возможны циклические сдвиги как на I бит,

так и на переменное число битов, указываемое счетчиком сдвигов в регистре CL. Флаг CF получает значение последнего передаваемого бита; OF определяется только для сдвигов со счетчиком, равным I, и устанавливается в состояние лог. I только в том случае, если результирующее значение знакового бита отличается от его предварительного значения.

Двухоперандные операции. Имеются 4 двухоперандные логические операции:

AND - выполняет побитное логическое умножение операнда-источника и операнда-приемника и возвращает результат в операнд-приемник;

TEST - выполняет операцию, аналогичную AND, устанавливая соответствующие флаги, но не возвращает результата;

OR - выполняет побитное логическое сложение операнда-источника и операнда-приемника и возвращает результат в операнд-приемник;

XOR - выполняет побитное логическое исключение операнда-источника и операнда-приемника и возвращает результат в операнд-приемник.

Флаги CF и OF устанавливаются в состояние лог. 0 во всех операциях; флаги SF, PF, ZF зависят от результата.

4) КОМАНДЫ ОПЕРАЦИЙ СО СТРОКАМИ

Команды работы со строками обеспечивают обработку (пересылку, сравнение, запись в память, загрузку в аккумулятор) одного элемента строки за одно обращение к памяти. Эти команды совместно с префиксом повторения позволяют организовать аппаратные циклы для обработки строк длиной до 64 К байт, выполняемые намного быстрее программных циклов.

Аппаратное управление операциями. Все простые строковые операции используют регистр SI для адресации операндов-источников. Регистр DI используется для адресации операндов-приемников, которые располагаются в текущем дополнительном сегменте. Если значение флага назначения (DP) равно 0, указатели операндов увеличиваются после каждой операции на I для байтных операций и на 2 для словных операций. Если он установлен в состояние лог. I, указатели операндов уменьшаются после каждой операции.

Любая из команд простых строковых операций может иметь однобайтный префикс, указывающий, что операция будет повторена до тех пор, пока счетчик операций в регистре CX не будет ис-

черпан. Проверка завершения счетчика делается перед каждым повторением операции; т.е. если начальное значение *sx* равно 0, то простая операция выполняться не будет.

Байт префикса повторений также указывает значение для сравнения с флагом *ZF*. Если простая операция влияет на флаг *ZF* и флаг *ZF* не равен указанному значению после каждого выполнения простой операции, то повторение прекращается. Это позволяет выполнять операцию сканирования (например, сканировать до выполнения некоторого условия).

Во время повторения простой операции индексные регистры операндов (*si* и *DI*) и регистр счетчика операций (*CX*) изменяются после каждого повторения, причем указатель команды будет сохранять смещение адреса байта префикса повторений (который непосредственно предшествует команде строковой операции). Это значит, что прерванная повторяющаяся операция будет корректно возобновлена, когда управление вернется из прерывающей программы.

Не следует использовать 2 других префиксных байта для строковых команд с префиксом повторения (сегментный префикс или префикс *LOCK*). Выполнение повторяющейся строковой операции будет возобновлено неправильно после прерывания, если более чем 1 префикс активен и предшествует простой строковой операции. Выполнение будет восстанавливать только 1 байт перед операцией (предположительно, где находится повторение), т.е. игнорируя дополнительные префиксы.

Простые строковые операции.

MOVW (или *MOVW*) - передает байт (или слово) из операнда-источника (правого) в операнд-приемник (левый). Если операция повторяющаяся, она проводит передачу строки из одного места в памяти в другое;

CMPB (или *CMPW*) - вычитает правый байтный (или словный) операнд из левого операнда и устанавливает флаги, но не возвращает результата. Повторяющаяся операция используется для сравнения 2 строк.

С предварительным префиксом повторений операция применяется для того, чтобы определить, после какого элемента две строки становятся неравными;

SCAB (или *SCAW*) - вычитает байтный (или словный) операнд-приемник из регистра *AL* (или *AX*) и устанавливает флаги, но не возвращает результат. Повторяющаяся команда применяется

для сканирования строки с целью обнаружения заданного значения в строке;

LODB (или *LODW*) - передает байтный (или словный) операнд из операнда-источника в регистр *AI* (или *AX*). Эта операция обычно не повторяется;

STOW (или *STOW*) - передает байтный (или словный) операнд из регистра *AL* (или *AH*) в операнд-приемник. Повторяющаяся операция используется для заполнения строки заданной величиной.

Во всех случаях, указанных выше, операнд-источник адресуется через *si*, а операнд-приемник адресуется через *DI*. Только в *CMPB* (*CMPW*) операнд, индексруемый *DI*, появляется как правый операнд.

Управление программными операциями. Префикс повторений создается для быстрого выполнения повторений в строковых операциях с аппаратным способом повторений, операции повторения управления (см. *LOOP*) создают такое же управление для выполнения программных циклов для того, чтобы осуществлять сложные строковые операции. Эти операции повторений проводят аналогичное изменение счетчика операций, операцию проверки исчерпывания счетчика и проверку флага *ZF* аналогично префиксу повторений,

При комбинировании простых строковых операций и операций управления повторениями и другими операциями возможно построить сложные, но эффективные программы, манипулирующие со строками.

Особенно полезна операция *XLAT*; она позволяет перекодировать байт, полученный из одной строки, перед запоминанием его в другую строку, или перед тем, как он будет обработан каким-либо другим способом. Для перекодировки используется значение в регистре *AL* в качестве индекса таблицы, указываемой регистром *BX*. Новое значение извлекается из таблицы и замещает индекс в регистре *AL* (см. *XLAT*).

5) КОМАНДЫ ПЕРЕДАЧИ УПРАВЛЕНИЯ

Различают 4 группы операций передачи управления:

- безусловные переходы, вызовы подпрограмм, возвраты из подпрограмм;
- условные переходы;
- управление циклами;
- прерывания.

Все операции передачи управления вызывают продолжение выполнения программы с некоторого

нового адреса памяти, возможно, в новом кодовом сегменте. Условные передачи управления обеспечивают переходы в точки, находящиеся в диапазоне -128...127 байт от команды перехода.

Безусловные переходы, вызовы подпрограмм, возвраты из подпрограмм. Имеются 2 основных типа безусловных переходов, вызовов и возвратов:

- обеспечивающие передачу управления в текущем кодовом сегменте (внутрисегментный переход);

- обеспечивающие передачу управления в произвольный кодовый сегмент, который становится при этом текущим (межсегментный переход).

Ниже описаны 3 операции передачи управления:

CALL - запоминает адресное смещение следующей команды в стеке (в случае межсегментной передачи сегментный регистр CS загружается в стек первым), а затем передает управление команде назначения;

JMP - передает управление команде назначения;

RET - передает управление на адрес возврата, предварительно занесенный в стек командой CALL, дополнительно может скорректировать регистр SP таким образом, чтобы исключить zapomненные в стеке параметры.

Внутрисегментные прямые вызовы и переходы указывают относительное прямое смещение, что позволяет программисту организовывать пози-

ционно независимые программы. Команда короткого перехода справедлива только для переходов в диапазоне -128...127 байт от команды перехода.

Условные переходы. Условные передачи управления создают совокупность переходов при различных значениях булевых функций регистров флагов. Операнд-приемник должен находиться внутри диапазона -128...127 байт от команды (см. таблицу).

Управление циклами. Команды управления циклами позволяют эффективно выполнять многократно повторяющиеся операции внутри диапазона -127...128 байт от команды управления циклом. Выход из цикла осуществляется, если содержимое счетчика равно 0 или по условию.

LOOP - уменьшает регистр CX («счетчик») на 1 и передает управление, если CX не равен 0; LOOPZ (также называемый LOOPE) - уменьшает регистр CX на 1 и передает управление, если CX не равен 0 и флаг ZF равен 1 (цикл, пока "ноль" или пока «равно»);

LOOPNZ (также называемый LOOPNE) - уменьшает регистр CX на 1 и передает управление. Если CX не равно 0 и/или флаг ZF равен 0 (цикл, пока «не ноль» или пока не равно»);

2CX2 - передает управление, если регистр CX равен 0.

Прерывания. Управление выполнением программы может состоять в передаче управления аналогично внешним прерываниям. Все прерывания позволяют осуществить передачу управления

Команда	Условие	Смысл операции
JE или JZ	ZF=1	„Равно” или „Ноль”
JL или JNGE	(SF XOR OF)=1	„Меньше” или „Не больше или равно”
JLE или JNG	((SF XOR OF) OR ZF)=1	„Меньше или равно” или „Не больше”
JB или JNAE, или JC	CF=1	„Ниже” или „Не выше или равно”, или „Перенос”
JBE или JNA		„Ниже или равно” или „Не выше”
JP или JPE	PF=1	„Четность” или „Четность четная”
JO	OF=1	„Переполнение”
JS	SF=1	„Знак”
JNE или JNZ	ZF=0	„Не равно” или „Не ноль”
JNL или JGE	(SF XOR OF)=0	„Не меньше” или „Больше или равно”
JNLE или JG	((SF XOR OF) OR ZF)=0	„Не меньше или равно” или „Больше”
JNB или JAE, или JNC	CF=0	„Не ниже” или „Выше или равно”, или „Нет переноса”
JNBE или JA	(CF OR ZF)=0	„Не ниже или равно” или „Выше”
JNP или JPO	PF=0	„Нечетность” или „Четность нечетная”
JNO	OF=0	„Нет переполнения”
JNS	SF=0	„Нет знака”

посредством занесения регистров флагов в стек (как в PUSHF), а затем выполнения косвенного междусегментного вызова с помощью элементов вектора прерываний, размещенного на абсолютных адресах 0...3FFH. Этот вектор содержит 4-байтный элемент для каждого из 256 различных уровней прерываний.

INT - загружает регистры флагов (как в PUSHF), очищает флаги TF и IF и передает управление с помощью косвенного вызова через любой из 256 элементов вектора прерываний. Однобайтный формат этой команды используется только для прерывания уровня 3;

INTO - загружает регистры флагов (как в PUSHF), очищает флаги TF и IF, а затем передает управление посредством косвенного вызова через 4-й элемент вектора прерываний, если флаг OF равен 1 (ловушка переполнения). Если флаг OF равен 0, операция не выполняется;

IRET - осуществляет передачу управления на адрес возврата, запомненный в стеке предыдущей операцией прерывания, а затем восстанавливает запомненные регистры флагов (как в POPF).

б) КОМАНДЫ УПРАВЛЕНИЯ МП

Операции с флагами. Имеются 7 команд, которые оперируют непосредственно отдельными регистрами флагов:

CLC - очищает флаг CF;

CMC - проводит дополнение флага CF;

STC - устанавливает флаг CF;

CLD - очищает флаг DF, указывая строковым операциям автоматическое увеличение указателей операнда;

STD - устанавливает флаг DF, указывая строковым операциям автоматическое уменьшение указателей операнда;

CLI - очищает флаг IF, запрещая внешние прерывания (исключением является немаскируемое внешнее прерывание);

STI - устанавливает флаг IF, разрешая внешние прерывания после выполнения следующей команды.

Останов МП. Команда HLT переводит МП КМ1810ВМ86 в состояние останова. Состояние останова снимается разрешенным внешним прерыванием или сигналом RESET.

Состояние ожидания. Команда WAIT вызывает перевод МП в состояние ожидания, если сигнал TEST на его выводе не подтвержден. Состояние ожидания может быть прервано разрешенным внешним прерыванием. В этом случае сохраняется

адрес команды WAIT, так что при возврате из прерывающей задачи состояние ожидания возобновляется. Состояние ожидания прекращается и работа МП продолжается, когда сигнал TEST подтвердится.

Работа МП возобновляется с запрещенными внешними прерываниями до конца следующей команды.

Эта команда позволяет МП синхронизировать свою работу с работой внешнего оборудования.

Отстранение МП (передача управления сопроцессору). Команда ESC создает механизм, при котором другие МП могут принимать свои команды из потока команд МП КМ1810ВМ86 и использовать типы адресации МП КМ1810ВМ86. МП не выполняет никаких действий при команде ESC, кроме доступа к операнду типа память.

Блокировка магистрали. Специальный однобайтный префикс может предшествовать любой команде, вынуждая МП подтверждать свой сигнал блокировки магистрали на время операции, вызываемой этой командой. Это свойство используется в мультипроцессорных вариантах (см. LOCK).

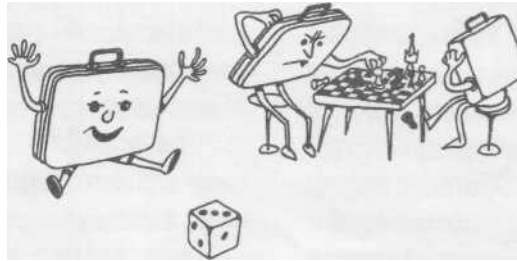
Пошаговый режим. Когда флаг TF устанавливается в состояние лог. 1, МП генерирует прерывание I после выполнения каждой команды. Во время передачи управления по прерыванию, вызванной любым уровнем прерывания, флаг TF сбрасывается после загрузки флагов в стек. Последующая операция выхода из подпрограммы обработки прерывания (команда IRET) восстанавливает содержимое TF (TF становится равным 1, если флаг был установлен до поступления прерывания). Это позволяет выполнять диагностическую работу в пошаговом режиме для отлаживаемой программы.

Если команда в пошаговом режиме сама очищает флаг TF, прерывание типа I будет появляться после завершения этой команды. Если эта команда генерирует прерывание, или, если разрешенное внешнее прерывание возникает до завершения этой команды, последовательность прерываний типа I будет возникать после последовательности сгенерированного или внешнего прерывания, но перед исполнением первой команды прерывающей программы.

Не существует команд для непосредственной установки или сброса флага TF; значение флага можно изменить в стеке.

УРОК 15

ПРИМЕР ПРОГРАММЫ



Наглядным примером для обучения работе на микро-ЭВМ „Микролаб 1810“ может служить приведенная ниже программа. Она имитирует так называемую игру в кости, в которой бросается всего одна кость. Напомним, что „кость“ - это кубик, все 6 граней которого помечены цифрами от 1 до 6.

Программа работает следующим образом: после того, как коды программы введены с клавиатуры и управление передано на начало исполнения, нажатием любой клавиши (кроме „Сброс“ и Прер“) можно заставить „кость“ начать „катиться“. Еще одно нажатие любой клавиши, кроме указанных, останавливает „кость“, а на дисплее в самой левой позиции появляется ее значение, т.е. цифра от 1 до 6. Вывод значений на дисплей и опрос клавиатуры осуществ-

ляется самой программой, поэтому много места в ней отводится управлению контроллером клавиатуры и дисплея КР580ВВ79 (см. урок 8).

В табл. 15.1 приведен текст на языке Ассемблера и коды программы „катящаяся кость“.

После ознакомления с программой можно попытаться поработать с ней практически. Для этого в момент запроса монитором команды (в левой позиции поля адреса дисплея высветится знак „-“) введите в память микро-ЭВМ коды программ. Используйте при этом команду „Байт“ (клавиша „БТ/АХ“). Всего программа занимает 4ЕН (или 78D) байтов. Адреса указаны в колонке „Адреса памяти“ (табл. 15.1), а колонка „Данные“ указывает содержимое соответствующих байтов памяти. Проконтролируйте правильность ввода с помощью этой же команда монитора

Т а б л и ц а 15.1

Адреса памяти	Данные	Команды	Комментарии
100	8C	MOV CX, CS	Установка значений сегментных регистров: ; значение регистра сегмента данных ; равно значению регистра сегмента кода
101	C9		
102	8E	MOV DS, CX	
103	D9		
104	BA	MOV DX, OFFEАН	Контроллер КР580ВВ79 - в исходное состояние: ; команда программного сброса
105	EA		
106	FF		; и сброса регистра состояния ; контроллера КР580ВВ79
107	BO	MOV AL, OD3H	
108	D3		
109	EE	OUT DX	
10A	BA	READKEY: MOV DX, OFFEАН	Ожидание нажатия клавиши, т.е. запуска „катящейся кости“: ; адрес порта команд КР580ВВ79 ; считывание регистра состояния ; счетчик ОМ-ОЗУ=0 (клавиша нажата)? ; если нет, продолжать ожидание ; считать код клавиши, чтобы освободить ОМ-ОЗУ
10B	EA		
10C	FF		
10D	EC	IN DX	
10E	24	AND AL, OFH	
10F	OF		
110	74	JZ READKEY+3	
111	FB		
112	E8	CALL READATA	
113	28		
114	00		

Адреса памяти	Данные	Команды	Комментарии
		ZERO:	Запуск:
115	BB	MOV BX, 0000H	; начальное значение счетчика „кости“
116	00		
117	00		
		START:	
118	43	INC BX	; увеличить счетчик на 1
119	80	CMP BL, 07H	; если счетчик = 7
11A	FB		
11B	07		
11C	74	JZ ZERO	; на сброс в 0
11D	F7		
11E	8B	MOV DI, BX	; преобразование значения счетчика в 7-сегментный
11F	FB		
120	8A	MOV CL, CDTBL [DI-1]	; код для индикации в поле регистра DI
121	4D		
122	47		
123	90		
124	BA	MOV DX, OFFEAH	; команда „запись в ОЗУ
125	EA		
126	FF		
127	BO	MOV AL, 087H	дисплея“ в 7-ю позицию
128	87		
129	EE	OUT DX	
12A	BA	MOV DX, OFFE8H	; вывод значения счетчика „кости“
12B	E8		
12C	FF		
12D	8A	MOV AL, CL	; на дисплей (в порт KP580BB79)
12E	C1		
12F	EE	OUT DX	
			Ожидание нажатия клавиши (остановки „катящейся кости“)
130	BA	MOV DX, OFFEAH	; считывание регистра состояния
131	EA		
132	FF		
133	EC	IN DX	; контроллера KP580BB79
134	24	AND AL, 0FH	; счетчик OM-ОЗУ=0 (клавиша нажата)?
135	0F		
136	74	JZ START	; если нет, продолжать просчет значений
137	EO		
138	E8	CALL READATA	; считать код клавиши, чтобы освободить OM-ОЗУ
139	02		
13A	00		
13B	EB	JMP READKEY	; на ожидание запуска
13C	CD		
		READATA:	Процедура считывания кода клавиши из OM-ОЗУ KP580BB79:
13D	BA	MOV DX, OFFEAH	; команда контроллера
13E	EA		
13F	FF		
140	BO	MOV AL, 040H	; KP580BB79 „считывание данных“
141	40		
142	EE	OUT DX	
143	BA	MOV DX, OFFE8H	; считать код клавиши AL
144	E8		
145	FF		
146	EC	IN DX	; результат в регистре AL
147	C3	RET	; возврат
			Таблица преобразования чисел от 1 до 6 в код для 7-сегментной индикации:
148	06	DB 06H	; 7-сегментный код „1“
149	5B	DB 05BH	; 7-сегментный код „2“
14A	4F	DB 04FH	; 7-сегментный код „3“
14B	66	DB 066H	; 7-сегментный код „4“
14C	6D	DB 06DH	; 7-сегментный код „5“
14D	7D	DB 07DH	; 7-сегментный код „6“

«Байт" (см. урок 4). После того, как программа введена, надо передать ей управление. Для этого используйте команду «Пуск»: нажмите клавишу "Пуск/СХ" в момент запроса монитором команды и, когда на дисплее появится текущий адрес памяти, его содержимое, а в самом правом индикаторе адресного поля высветится сигнал «.", введите стартовый адрес программы, используя значение адреса сегмента, например 10, а смещение 0 (весь адрес 10:0). Закончите команду нажатием клавиши ".". При этом дисплей должен погаснуть, даже буква „Е" в самом левом индикаторе поля адреса, которая говорит о том, что исполняется программа пользователя, не будет высвечиваться, так как в самом начале программы подается команда программного сброса контроллера клавиатуры и дисплея КР580ВВ79, а затем в крайней левой позиции индицируется значение счетчика. Теперь можете "поиграть" с программой, как описано выше.

Для того, чтобы остановить исполнение программы и вернуть управление монитору, можно нажать клавишу „Сброс" или "Прер". При нажатии клавиши «Сброс» все регистры МП КМ1810ВМ86 устанавливаются в исходное состояние, так как программа в памяти сохраняется до выключения питания. Можно запустить ее снова, однако необходимо повторить всю процедуру запуска, описанную выше. После нажатия клавиши «Прер» содержимое регистров сохраняется, и программа может быть запущена с точки, в которой была прервана, с помощью команды "Пуск" без ввода начального адреса. Надо только после клавиши «Пуск" нажать сразу клавишу ".". Попробуйте осуществить тот и другой варианты.

Затем попробуйте поработать с командой "Пуск" с указанием точки прерывания (см. описание команды "Пуск" в уроке 4).

УРОК 16

СТРУКТУРА МП КМ1810ВМ86



Назначение выводов МП в минимальном и максимальном режимах приведено в табл. 16.1.

Функции выводов, характерные только для максимального режима, приведены в табл. 16.2, для минимального - в табл. 16.3.

Структура.

МП состоит из двух основных функциональных блоков, работающих асинхронно (рис. 16.1): устройства обработки (УО) и устройства сопряжения канала (УСК).

УО предназначено для декодирования и выполнения команд, состоит из АЛУ на базе 16-разрядного сумматора с последовательно-параллельным переносом, 3 регистров для промежуточного хранения операндов и результата операций и регистра признаков. Обмен данными происходит через регистры общего назначения (РОН). Функции управления выполнением команд возложены на микропрограммное устройство управления, которое декодирует команды и выра-

Номер вывода	Обозначение вывода	Тип вывода	Наименование и назначение
2-16, 39	ADO-AD15	Вход-выход с тремя состояниями	<u>Канал адреса (данных)</u> . Эти выводы образуют мультиплексированный канал адреса памяти или ввода-вывода (в такте T1) и данных (в тактах T2, T3 и Tw, T4). Сигнал AO похож на сигнал BHE , но служит для выбора младшего байта канала данных (выводы D7-DO). Когда при операциях с памятью или устройствами ввода-вывода (УВВ) пересылается байт данных по младшей половине канала данных, то 8-разрядные УВВ используют сигнал AO как условие выбора устройства. Эти сигналы при подтверждении прерывания и подтверждении захвата находятся в третьем состоянии.
35-38	A19/ST6, A18/ST5, A17/ST4, A16/ST3	Выход с тремя состояниями	<u>Канал адреса (состояния)</u> . В такте T1 - это 4 самых старших разряда адреса памяти. При работе с УВВ на эти выходы подаются сигналы уровня лог. 0. При работе с ЗУ и УВВ в тактах T2, T3, Tw и T4 на этих выходах присутствует информация о состоянии. Состояние триггера подтверждения прерывания (ST5) обновляется в каждом такте. Сигналы ST4 и ST3 указывают, какой регистр используется для обращения к данным. В третьем состоянии выводы переводятся при подтверждении захвата.
34	BHE/ST7	Выход с тремя состояниями	<u>Разрешение передачи по старшим тактам канала адреса (состояния)</u> . В такте T1 сигнал BHE должен использоваться для разрешения передачи данных по старшим разрядам канала данных (D15-D8); 8-разрядные устройства, подключенные к старшим тактам канала, обычно используют этот сигнал как условие выбора устройства. Когда байт данных должен передаваться по старшим тактам канала, то в циклах чтения, записи и подтверждения прерывания в такте T1 выдается сигнал BHE уровня лог. 0. В тактах T2, T3 и T4 выдается сигнал состояния ST7 . При захвате сигнал находится в третьем состоянии.
32	RD	Выход с тремя состояниями	<u>Чтение</u> . Строб чтения указывает, что МП выполняет цикл чтения ЗУ или УВВ в зависимости от состояния вывода ST2 . Сигнал используется для чтения устройств, подключенных к локальному каналу МП. Активный сигнал RD уровня лог. 0 выдается в тактах T2, T3 и Tw каждого цикла чтения. Если локальный канал находится в третьем состоянии, то в такте T2 выдается сигнал RD уровня лог. 0. Сигнал RD при подтверждении захвата находится в третьем состоянии.
22	RDY	Вход	<u>Готовность</u> . Сигнал служит подтверждением того, что адресуемое ЗУ или УВВ закончит передачу данных в текущем цикле. Активный уровень сигнала - лог. 1. Сигнал RDY не синхронизирован.
18	INT	Вход	<u>Запрос прерывания</u> . Потенциальный вход с запоминанием запроса на внутреннем триггере, который опрашивается в последнем такте каждой команды с целью определения необходимости выполнения цикла подтверждения прерывания. Адрес подпрограммы обслуживания прерывания определяется по вектору из таблицы, расположенной в области системной памяти. Прерывание может быть маскировано программной установкой триггера разрешения прерывания. Сигнал INT внутренне синхронизирован.
23	TEST/	Вход	<u>Проверка</u> . Состояние входного сигнала опрашивается при команде ожидания WAIT . Если на входе TEST/ сигнал уровня лог. 0, то МП продолжает обработку, в противном случае МП ожидает в "холостом" состоянии. Сигнал внутренне синхронизирован.
21	CLR	Вход	<u>Установка</u> . Сигнал служит для установки внутренних схем МП в начальное состояние. При подаче на вход CLR сигнала уровня лог. 1 МП немедленно прекращает свои действия. Сигнал должен поддерживаться на входе не менее 4 тактов. Работа МП возобновляется, когда сигнал снимается со входа CLR . Сигнал CLR внутренне синхронизирован.
17	ЮН	Вход	Немаскируемый запрос прерывания. Импульсный вход, по которому вызывается прерывание типа 2. Адрес подпрограммы обслуживания прерывания определяется по вектору из таблицы векторов в системной области памяти. Положительный перепад напряжения на входе вызывает прерывание в конце текущей команды. Сигнал внутренне синхронизирован.

Номер вывода	Обозначение вывода	Тип вывода	Наименование и назначение
19	CLC	Вход	<u>Тактовый импульс.</u> Сигнал обеспечивает синхронизацию МП.
40	Ucc		Напряжение источника питания (+5 В)
1, 20	GND		Корпус
33	MN/M\bar{X}	Вход	Режим управления минимальный (максимальный).

Т а б л и ц а 16.2

Номер вывода	Обозначение вывода	Тип вывода	Наименование и назначение вывода в максимальном режиме
26-28	ST2/, ST1/, ST0	Выход с тремя состояниями	<u>Состояние цикла канала.</u> Сигнал выдается в тактах T4, T1 и T2 и в такте T3 или Tw . При наличии сигнала готовности (уровень сигнала RDY лог. 1) эти сигналы переходят в пассивное состояние (1,1,1). Всякое изменение сигналов состояния в такте T4 используется как указание на начало цикла канала, а возврат к пассивному состоянию указывает на окончание цикла канала. Сигналы при подтверждении захвата находятся в третьем состоянии.
30, 31	R\bar{Q}/EO RQ/E1	Вход-выход	<u>Запрос (разрешение) доступа к магистрали.</u> Выводы используются другими устройствами, чтобы принудить МП освободить локальный канал в конце текущего процессорного цикла канала. Оба вывода двунаправленные, причем вывод RQ/EO имеет более высокий приоритет, чем вывод RQ/E1 .
29	LOCK/	Выход с тремя состояниями	<u>Канал занят.</u> Сигнал на этом выходе запрещает внешним „владельцам“ системного канала захватывать управление системным каналом, пока сигнал находится в состоянии уровня лог. 0. Сигнал LOCK/ выдается по команде LOCK (префикс блокировки) и поддерживается на выходе до следующей команды. При подтверждении захвата сигнал находится в третьем состоянии.
24, 25	QS0, QS1	Выход	<u>Состояние очереди команд.</u> Код состояния указывает на состояние очереди команд в том такте, после которого было выполнено действие.

Т а б л и ц а 16.3

Номер вывода	Обозначение вывода	Тип вывода	Наименование и назначение вывода в максимальном режиме
29	W/	Выход с тремя состояниями	<u>Запись.</u> Указывает, что МП выполняет запись в ЗУ или УВВ, в зависимости от состояния сигнала M/$\bar{I}O$. Сигнал уровня лог. 0 выдается в тактах T2, T3 и Tw каждого цикла записи. При подтверждении захвата сигнал W/ находится в третьем состоянии.
28	M/$\bar{I}O$	То же	<u>Признак обращения к ЗУ или УВВ.</u> Сигнал является логическим эквивалентом сигнала ST2/ в максимальном режиме. Используется для того, чтобы отличить доступ к ЗУ от доступа к УВВ. Сигнал M/$\bar{I}O$ устанавливается в такте T4 предшествующего цикла канала и поддерживается до завершающего такта T4 текущего цикла. При подтверждении захвата сигнал находится в третьем состоянии.
27	OP/$\bar{I}P$	—	<u>Выдача (прием) данных.</u> Используется для управления направлением передачи данных через шинные формирователи. Временная диаграмма сигнала OP/$\bar{I}P$ совпадает с диаграммой сигнала M/$\bar{I}O$. При подтверждении захвата локального канала сигнал OP/$\bar{I}P$ находится в третьем состоянии.
26	DE/	—	<u>Разрешение передачи данных.</u> Используется в системах минимальной конфигурации с применением шинных формирователей. Сигнал DE/ уровня лог. 0 выдается в каждом цикле обращения к ЗУ и УВВ и в циклах подтверждения прерывания. В циклах чтения и подтверждения прерывания выдается от середины такта T2 до середины такта T4, а в цикле записи — от начала такта T2 до середины такта T4. При подтверждении захвата сигнал DE/ находится в третьем состоянии.
25	ALE	Выход	<u>Строб адреса.</u> Используется при записи адреса в регистр адреса. Выдается в такте T1 любого цикла канала.

Номер вывода	Обозначение вывода	Тип вывода	Наименование и назначение вывода в максимальном режиме
24	INTA/	Выход	<u>Подтверждение прерывания.</u> Используется как строб чтения в циклах подтверждения прерывания. Сигнал уровня лог. 0 выдается в тактах T2, T3 и T _w каждого цикла подтверждения прерывания.
31, 30	HLD, HLDA	Вход-выход	<u>Захват, подтверждение захвата.</u> Сигнал HLD свидетельствует о том, что другой МП запрашивает захват локального канала. МП, получивший запрос захвата, выдает сигнал подтверждения HLDA в середине такта T4 или холостого такта T5. Одновременно с выдачей сигнала HLDA МП переводит в третье состояние локальный канал и канал управления. Обнаружив переход сигнала HLD в состояние уровня лог. 0, МП переводит в уровень лог. 0 сигнал HLDA, а когда МП потребует выполнить другой цикл канала, он вернет себе управление локальным каналом и каналом управления.

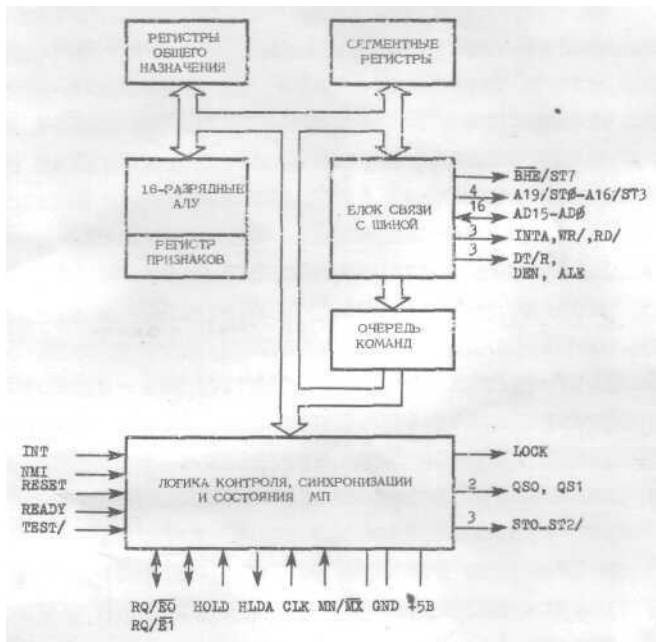


Рис. 16.1. Структурная схема МП KM1810BM86

Регистры общего назначения.
 На рис. 16.2 приведены РОН МП KM1810BM86. ТОН подразделяются на две группы: регистры



Рис. 16.2. Регистры общего назначения МП KM1810BM86

бывает необходимые сигналы управления. В состав УО также входит схема обработки запросов прерывания и схема управления доступом к магистрали.

УО, однако, не имеет непосредственной связи с внешними устройствами. Эту связь осуществляет устройство сопряжения канала (УСК), в состав которого входит блок вычисления адреса, в котором имеется сумматор адреса, сегментные регистры и регистр адреса команды. Сумматор адреса предназначен для вычисления физического адреса команд и данных путем сложения эффективного адреса с содержимым сегментных регистров (базовым адресом текущего сегмента). Выбранные из памяти данные и команды пересылаются в УО через регистр обмена, который можно использовать либо как один 16-разрядный регистр, либо как два 8-разрядных.

данных и регистры индексов и указателей. Все регистры являются 16-разрядными, причем каждый регистр данных может использоваться не только как единый 16-разрядный регистр, но и как 2 8-разрядных регистра, так как старшие и младшие регистры адресуются отдельно. В последнем случае они обозначаются AH (старший байт AX) и AL (младший байт AX) и, аналогично, BH и BL, CH и CL, DH и DL.

Индексные регистры и указатели используются только как 16-разрядные регистры.

РОН можно использовать в большинстве арифметических и логических операций. Кроме того, в некоторых командах предполагается (неявно) использование определенных регистров (табл. 16.4), что дает выигрыш в длине команды.

Регистр	Назначение
AX	Умножение, деление и ввод-вывод слов
AL	Умножение, деление и ввод-вывод байтов, перекодировка, десятичная арифметика
AH	Умножение, деление байтов
BX	Перекодировка
CX	Операции со строками, циклы
CL	Сдвиги и циклические сдвиги на различное число разрядов
DX	Умножение, деление слов
SP	Операции со стеком
SI	Операции со строками
DI	Операции со строками

Регистр адреса команд.

16-разрядный регистр адреса команды (указатель команды) **IP** содержит смещение команды от начала сегмента кода. Базовый адрес сегмента кода, находящийся в сегментном регистре **CS**, и смещение образуют логический адрес команды.

Сегментные регистры.

МП **KM1810BM86** имеет 4 сегментных регистра (рис. 16.3). Область памяти, непосредственно



Рис. 16.3. Сегментные регистры МП **KM1810BM86**

адресуемая МП, достигает 1М байт. Она разделена на логические сегменты, каждый из которых содержит до 64К байт. МП имеет прямой доступ к 4 сегментам одновременно благодаря наличию сегментных регистров, в которых хранятся базовые (начальные) адреса сегментов.

Регистр сегмента кода (**CS**) содержит базовый адрес сегмента кода, из которого выбираются команды. Регистр сегмента данных (**DS**) указывает на текущий сегмент данных. Регистр сегмента стека (**SS**) указывает на текущий сегмент стека. Регистр дополнительного сегмента (**ES**) указывает на начало дополнительного сегмента, который используется для хранения данных.

Регистр признаков.

Регистр признаков представлен на рис. 16.4. Признаки представляют состояние результата выполненной арифметической или логической операции.

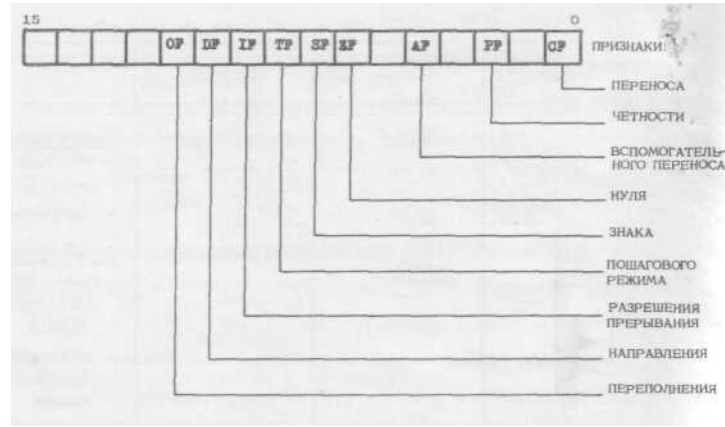


Рис. 16.4. Регистр признаков МП **KM1810BM86**

Признаки состояния:

AF - признак вспомогательного переноса.

Если **AF=1**, значит был перенос из третьего разряда в четвертый, либо заем из четвертого разряда в третий разряд младшего байта 16-разрядного числа. Признак **AF** используется командами десятичной арифметики.

CF - признак переноса.

Если **CF=1**, то был перенос или заем из старшего разряда результата.

OF - признак переполнения.

Если **OF=1**, то это означает, что произошло арифметическое переполнение, т.е. утеряна значащая цифра, когда разрядность результата превышает разрядность приемника результата.

SF - признак знака.

Если **SF=1**, то старший разряд результата равен 1 (результат отрицательный). Если **SF=0**, то результат положительный.

PF - признак четности.

Если **PF=1**, то результат четный. Признак используется для обнаружения сбоев при передаче данных.

ZF - признак нуля.

Если **ZF=1**, то результат операции равен 0.

Кроме признаков состояния имеются еще признаки контроля выполнения программы.

IF - признак разрешения прерывания.

Если **IF=1**, то МП реагирует на внешние маскируемые прерывания. Если **IF=0**, то маскируемые прерывания запрещены.

Признак разрешения прерывания не влияет на внутренние и внешние немаскируемые прерывания

DF - признак направления.

Если **DF=1**, то строки данных обрабатываются от старших адресов к младшим (справа налево). Если **DF=0**, то строки данных обрабатываются слева направо.

TF - признак пошагового режима.

• Если $TF=1$, то МП переходит на пошаговый режим выполнения программы. В таком режиме после выполнения каждой команды МП генерирует прерывание.

Схема предварительной выборки команд.

Распределение функций выборки и выполнения команд между двумя независимыми частями МП приводит к повышению его производительности.

Предварительная очередь команд МП состоит из 3 16-разрядных предварительных регистров команд. Следовательно, МП может хранить до 6 байтов кода команд. Очередь организована по принципу: „первым пришел - первым обслужен“. УСК выбирает слово из памяти всякий раз, когда в очереди освобождается по меньшей мере 2 байта.

УО извлекает коды команд из очереди по мере необходимости. 6-байтная глубина предварительной очереди позволяет удовлетворять запросы УО настолько эффективно, что МП практически не простаивает в ожидании выборки команд из памяти.

Если в очереди освобождается 2 байта и отсутствует запрос на доступ к каналу за операндом памяти, то схема предварительной выборки команд выбирает из памяти следующие байты команды. За одно обращение обычно выбирается 2 байта (если слово размещено по четному адресу).

В предварительной очереди содержится, как правило, не менее 1 байта из потока команд, и устройству обработки не приходится ждать выборки команды. Выполнение команд происходит в той логической последовательности, в которой написана программа, так как в очереди находятся те команды, которые хранились в ячейках памяти, непосредственно следующих за ячейками текущей команды. При передаче управления в другую ячейку памяти последовательность хода программы изменяется. УСК очищает очередь, выбирает команду по адресу перехода, передает ее УО, а после этого начинает новое заполнение очереди. При возврате из подпрограммы или из прерывания также происходит реинициализация очереди команд; адреса команд при этом вычисляются автоматически с помощью схемы адресных констант, имеющейся в сумматоре адреса.

Если МП необходимо выполнить цикл чтения или записи, то предварительная выборка команд приостанавливается на время цикла.

Работая в максимальном режиме, МП выдает сигналы QS1 и QSO состояния очереди команд. В табл. 16.5 приведены значения сигналов QS1 и

Т а б л и ц а 16.5

QS1	QSO	Состояние предварительной очереди команд в предыдущем такте
0	0	Нет операций. (В последнем такте из очереди ничего не выбиралось)
0	1	Первый байт. (Байт, выбранный из очереди, был первым байтом команды)
1	0	Очередь пуста. (Очередь была реинициализована в результате выполнения команды перехода)
1	1	Последующий байт. (Байт, выбранный из очереди, был последующим байтом команды)

QSO и соответствующее состояние предварительной очереди команд в предыдущем такте.

Сигналы состояния МП»

В максимальном режиме МП выдает 8 сигналов состояния, которые могут быть использованы внешними устройствами.

Сигналы ST0-ST2 определяют тип цикла обмена по каналу A/D, который МП начинает выполнять (табл. 16.6). Эти сигналы должны быть декодированы специальной схемой системного контроллера канала.

Т а б л и ц а 16.6

Сигнал состояния			Тип цикла канала
ST2/	ST1/	ST0/	
0	0	0	Подтверждение прерывания
0	0	1	Чтение УВВ
0	1	0	Запись в УВВ
0	1	1	Останов
1	0	0	Выборка команды
1	0	1	Чтение памяти
1	1	0	Запись в память
1	1	1	Пассивный

Сигналы ST3 и ST4 определяют, какой сегментный регистр используется в текущем цикле обмена для вычисления физического адреса ячейки памяти (табл. 16.7).

Сигнал ST5 индицирует состояние триггера

Т а б л и ц а 16.7

Сигнал состояния		Используемый сегментный регистр
ST4	ST3	
0	0	ES
0	1	SS
1	0	CS
		или никакой (при вводе-выводе или прерывании)
1	1	DS

разрешения прерывания регистра признаков. Сигнал ST6 всегда имеет уровень лог. 0.

Состояние сигнала ST7 не определено в процессе работы МП.

Структура прерываний.

МП КМ1810ВМ86 обладает простой и гибкой структурой прерываний. Каждому типу прерывания присвоен отдельный код. МП может обрабатывать до 256 типов различных прерываний. Прерывания могут вызываться устройствами, внешними по отношению к МП, а также командами прерывания и самим МП по некоторым условиям. Прерывания могут быть внутренними и внешними, программируемыми и аппаратными.

Аппаратные прерывания могут быть маскируемыми и немаскируемыми. К внутренним прерываниям относятся прерывания по ошибке деления, по переполнению, по командам INTn и прерывание пошагового режима.

Внешнее немаскируемое прерывание NMI используется для сообщения МП о таких «катастрофических» событиях, как угроза отключения питания, обнаружение ошибки в памяти и ошибки четности канала. Немаскируемые прерывания не могут быть запрещены.

Запрос прерывания по входу INT может маскироваться командой CLI, которая сбрасывает триггер разрешения прерывания IF. Если триггер IP сброшен, то МП игнорирует запрос и переходит к выполнению следующей команды, если же состояние IP - лог. 1, то МП распознает запрос и обслуживает его.

В табл. 16.8 показан приоритет прерываний.

Т а б л и ц а 16.8

Тип прерываний	Приоритет прерываний
Прерывание по ошибке деления, INTn, INTO	Высший
Немаскируемое прерывание (NMI)	
Маскируемое прерывание (INT)	
Пошаговый режим	Низший

Для перехода к подпрограмме обработки прерывания используется таблица векторов прерываний, связывающая код типа прерываний и подпрограмму, предназначенную для обработки данного типа прерываний. Таблица занимает первый К байт памяти. В ней может быть до 256 входов: по одному на каждый тип прерывания, возможный в системе.

Каждый вход в таблице описан двойным словом, содержащим адрес процедуры обработки прерывания данного типа. В старшем слове ука-

зателя содержится адрес сегмента, в котором записана процедура, а в младшем слове - смещение процедуры от начального адреса сегмента. Поскольку каждый вход имеет длину 4 байта, то МП может вычислить местонахождение соответствующего входа в таблице для данного типа прерывания простым умножением на 4 (тип x4).

Если в системе используется менее 256 типов прерываний, то старшие ячейки таблицы векторов прерываний, не занятые под адреса перехода, могут быть использованы для любых других целей. Однако младшие ячейки (0H-7FH), занятые указателями предустановленных прерываний (типы 0-4) и зарезервированные, нельзя использовать ни для каких иных целей, так как это нарушит правильное функционирование системы.

ОРГАНИЗАЦИЯ ПАМЯТИ

МП может адресовать до 1 М байта памяти. Память МП представлена как массив 8-разрядных байтов. Функционально она разбита на сегменты. Каждый сегмент выполнен как непрерывная область памяти размером до 64 К байт. Сегменты являются независимыми и произвольно адресуемыми устройствами. Каждому сегменту программно назначается базовый адрес, который является адресом первой ячейки его памяти. Эти адреса кратны 16. Других ограничений на размещение сегментов нет. Сегменты могут соприкасаться, не соприкасаться, перекрываться полностью или частично, не перекрываться. Физическая ячейка памяти может принадлежать одному или нескольким сегментам.

Сегментные регистры содержат базовые адреса 4 текущих сегментов. Программа получает доступ к кодам команд и данным, хранящимся в других сегментах, посредством замены содержимого сегментных регистров на адрес новых сегментов.

Области памяти от 0H до 7FH (128 байт) и от FFFF0H до FFFFFH (16 байт) используются для организации прерываний и системного сброса и не должны использоваться в прикладных программах пользователя для других целей.

Генерация физического адреса.

В системе, основанной на МП КМ1810ВМ86, удобно рассматривать два типа адресов каждой ячейки памяти: физический и логический. Физический адрес - 20-разрядное значение, которое однозначно указывает каждый байт памяти. Двоичный код физического адреса выдается на

внешние шины адреса (данных) по сигналу ALE. Физический адрес может изменяться от он до FFFFFH. Все передачи информации меащу МП и памятью используют физический адрес.

Программисты имеют дело с логическим адресом чаще, чем с физическим, что позволяет писать программы, не зная, где они будут расположены в памяти. Логический адрес состоит из сегментного значения и значения относительного адреса. Для любой ячейки памяти значение относительного адреса указывает расстояние в байтах от текущей ячейки памяти до начала сегмента. Базовый адрес сегмента - это адрес первой ячейки памяти данного сегмента. Базовый адрес сегмента и значение относительного адреса являются 16-разрядными беззнаковыми величинами.

Генерация физического адреса осуществляется сдвигом значения базового адреса сегмента на 4 бита и сложением его со значением относительного адреса (рис. 16.5).

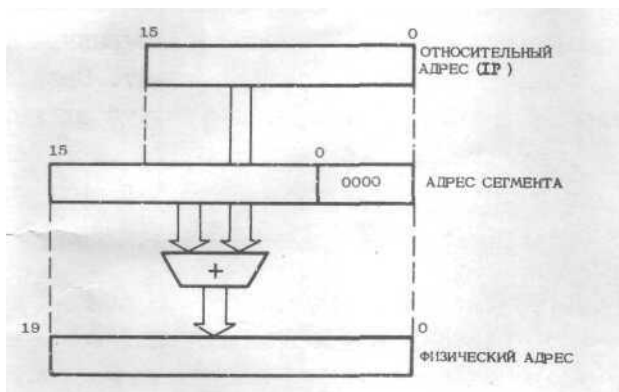


Рис. 16.5. Генерация физического адреса

Адресация внешней памяти.

Адресуемая область памяти (1М байт) поделена физически на две части, так называемые банки, размером до 512К байт.

Один банк связан с младшей половиной 16-разрядного канала данных (D7-D0), а другой - со старшей (D15-D8). Адрес (A19-A1) используется для того, чтобы адресовать ячейки памяти как в младшем, так и в старшем банках. Младший разряд адреса A0 не используется при адресации памяти, но используется для выбора банка памяти. Младший банк, содержащий четно-адресуемые байты, выбирается, когда A0=0. Старший банк, содержащий нечетноадресуемые байты (A0=1), выбирается по отдельному сигналу разрешения старшей половины канала (ВНЕ/=0). Механизм выбора банка по сигналам ВНЕ/и A0 расшифровывается в табл. 16.9.

ВНЕ/	A0	Пересылаемый байт
0	0	Оба байта
0	1	Старший байт в(из) нечетный адрес
1	0	Младший байт в(из) четный адрес
1	1	Ничего

Режимы адресации.

Адрес, вычисляемый МП для адресации операнда в памяти, называется исполнительным адресом (EA). Для вычисления исполнительного адреса в кодах машинных команд имеются поля mod (2 бита), r/m (3 бита) во втором байте И ПОЛЯ <disp - low (байт) И disp-high. Поля disp-low и disp-high расположены в 2 последующих байтах команды (перед данными). В зависимости от содержимого поля mod может отсутствовать байт disp-high или оба байта disp-low и disp-high. СПОСОБЫ ВЫЧИСЛЕНИЯ исполнительного адреса операнда памяти приведены в табл. 16.10.

Т а б л и ц а 16.10

r/m	Исполнительный адрес
000	$EA = (BX)^1 + (SI) + DISP$
001	$EA = (BX) + (DI) + DISP$
010	$EA = (BP) + (SI) + DISP$
011	$EA = (BP) + (DI) + DISP$
100	$EA = (SI) + DISP$
101	$EA = (DI) + DISP$
110	$EA = (BP) + DISP^2$
111	$EA = (BX) + DISP$

Примечание: (BX) — содержимое регистра BX и т.д.; ²За исключением случая mod=00 и r/t=110 (тогда EA=disp-low: disp-high).

ЕСЛИ mod=00, ТО DISP=0², байты disp-low и disp-high ОТСУТСТВУЮТ; ЕСЛИ mod=01, ТО DISP представлен байтом disp-low, знаково «растянутым» до 16 бит, байт disp-high отсутствует; если mod=10, ТО DISP=disp-low, disp-high, ЕСЛИ mod=11, то поле r/m используется МП также, как поле reg.

Прямая адресация.

Прямая адресация (рис. 16.6) - простейший

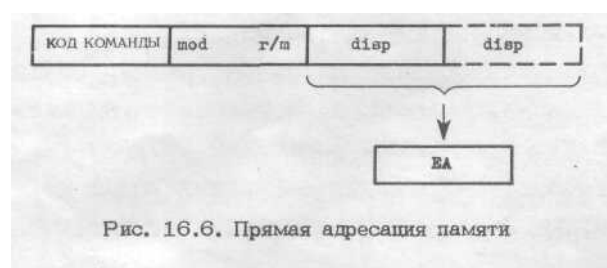


Рис. 16.6. Прямая адресация памяти

способ адресации памяти. При этом регистры не используются. Исполнительный адрес берется непосредственно из поля смещения команда. Прямая адресация обычно используется для доступа к простым переменным.

Регистровая косвенная адресация.

Исполнительный адрес памяти может быть получен непосредственно из базовых или индексных регистров (рис. 16.7). Одна команда может работать со множеством различных областей памяти, если базовому или индексному регистрам присваиваются новые значения.

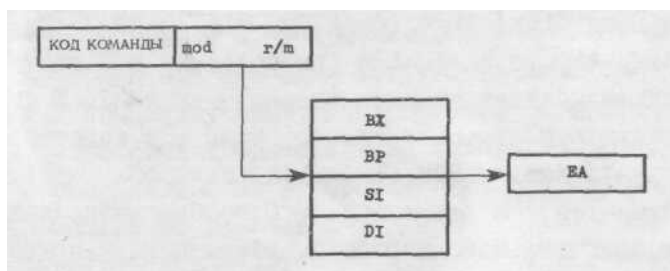


Рис. 16.7. Регистровая косвенная адресация памяти

Базовая адресация.

В базовой адресации (рис. 16.8) исполнительный адрес является суммой значения смещения и содержимого регистров *bx* или *bp*. Базовая

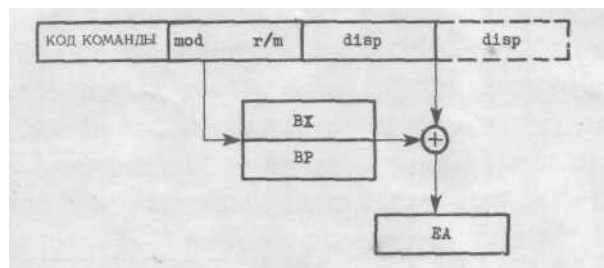


Рис. 16.8. Базовая адресация памяти

адресация обеспечивает возможность адресации структур данных, которые могут быть размещены в различных областях памяти. Базовый регистр в этом случае указывает адрес структуры, а элемент в структуре адресуется с помощью смещения. Обращение к другой структуре возможно путем замены значения базового регистра.

Индексная

В индексной адресации исполнительный адрес вычисляется как сумма смещения и содержимого индексного регистра *si* или *di* (рис. 16.9). Индексная адресация часто используется для обращения к элементам массивов. Смещение указывает начало массива, а значение индексного регистра - элемент в массиве (первый элемент в массиве выбирается, если индексный регистр содержит 0). С помощью простых арифметических

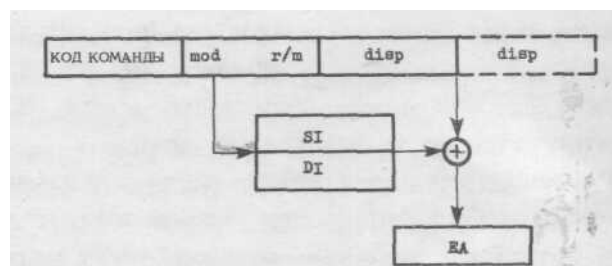


Рис. 16.9. Индексная адресация памяти

действий над содержимым индексного регистра можно выбирать любые элементы массива.

Базово-индексная адресация.

Базово-индексная адресация генерирует исполнительный адрес как сумму значений базового регистра, индексного регистра и смещения (рис. 16.10).

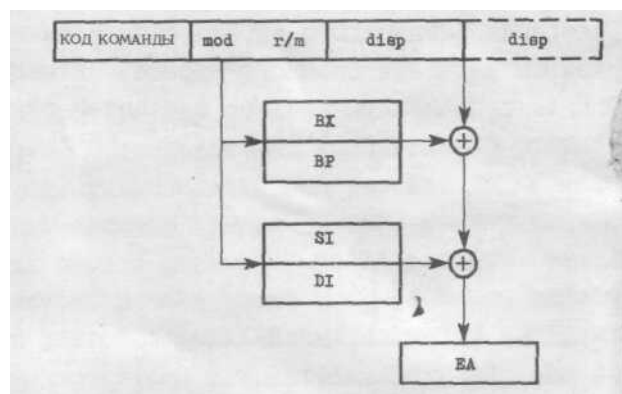


Рис. 16.10. Базово-индексная адресация памяти

Гибкость базово-индексной адресации обеспечивается тем, что два компонента адреса могут изменяться во время выполнения действий. Базово-индексная адресация может использоваться для обеспечения доступа к массивам, расположенным в стеке. Регистр *BP* в этом случае содержит смещение начала структуры массива в стеке относительно вершины стека. Расстояние от начала массива до выбираемой структуры извлекается из смещения, заданного в команде, а индексный регистр используется для доступа к конкретному элементу массива. Матрицы (двумерные массивы) также удобно обрабатывать, применяя базово-индексную адресацию

Адресация строк данных.

Команды обработки строк данных для доступа к своим операндам не используют обычную систему адресации памяти. Когда выполняется команда обработки строк, регистр *si* определяет адрес байта или слова элемента строки источника, а регистр *di* - адрес байта или слов элемента строки приемника. В повторяющихся операциях обработки элементов строк МП авто

матически корректирует содержимое регистров SI и DI для обращения к последовательно расположенным байтам или словам.

Адресация внутренних регистров.

Команды, обрабатывающие только регистровые операнды, более компактны и наиболее **быстро** выполняются во всех форматах команд. Регистры могут быть источниками операндов, приемниками операндов или тем и другим. Для адресации внутренних регистров в коде команд используется поле red или поле r/m, которое при значении mod=11 трактуется как поле red.

Таблица 16.11

reg	Регистр общего назначения		reg	Сегментный регистр
	W ¹ =1	W=0		
000	AX	AL	00	ES
001	CX	CL	01	CS
010	DX	DL	10	SS
011	BX	BL	11	DS
100	SP	AH		
101	BP	CH		
110	SI	DH		
111	DI	BH		

Поле red адресует регистры согласно таблице 16.10.

ВРЕМЕННЫЕ ДИАГРАММЫ РАБОТЫ МП

Временные диаграммы основных сигналов МП показаны на рис. 16.11.

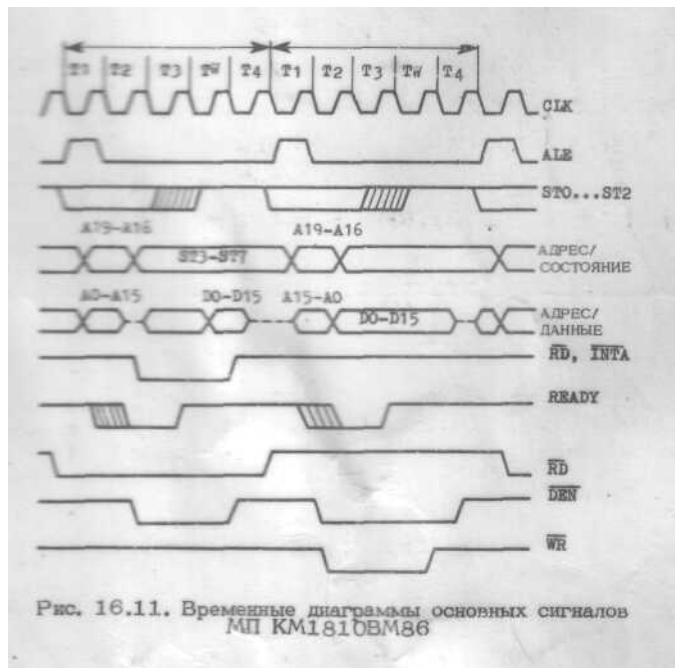


Рис. 16.11. Временные диаграммы основных сигналов МП KM1810BM86



Авторы: В.Я. Смолин, В.Е. Лякаев, Т.П. Власова, О.Н. Захарова, Е.Н. Чардымова, И.Ю. Хорева, В. Залавин, Н.Я. Кобица, О.С. Леонтьева, И.Б. Яковлев, И.В.

устройств на базе микропроцессора
KM1810BM86

Методическое пособие

Редактор В.П. Пермичева, художник С.Ю. Чугунов, М.В. Белов, технический редактор В.В. Воробьева

Подписано в печать 24.05.89. № 01166. Формат бумаги 60x90/8. Объем 10 п.л., 11 уч.-изд. л. Зак. 11105. Тираж 300 экз.