

7.1. Микропроцессор — еще не ЭВМ
 7.2. Схема микропроцессора — еще не ЭВМ
 7.3. Схема микропроцессора — еще не ЭВМ
 7.4. Схема микропроцессора — еще не ЭВМ
 7.5. Схема микропроцессора — еще не ЭВМ
 7.6. Схема микропроцессора — еще не ЭВМ
 7.7. Схема микропроцессора — еще не ЭВМ
 7.8. Схема микропроцессора — еще не ЭВМ
 7.9. Схема микропроцессора — еще не ЭВМ
 7.10. Схема микропроцессора — еще не ЭВМ
 7.11. Схема микропроцессора — еще не ЭВМ
 7.12. Схема микропроцессора — еще не ЭВМ
 7.13. Схема микропроцессора — еще не ЭВМ
 7.14. Схема микропроцессора — еще не ЭВМ
 7.15. Схема микропроцессора — еще не ЭВМ
 7.16. Схема микропроцессора — еще не ЭВМ
 7.17. Схема микропроцессора — еще не ЭВМ
 7.18. Схема микропроцессора — еще не ЭВМ
 7.19. Схема микропроцессора — еще не ЭВМ
 7.20. Схема микропроцессора — еще не ЭВМ
 7.21. Схема микропроцессора — еще не ЭВМ
 7.22. Схема микропроцессора — еще не ЭВМ
 7.23. Схема микропроцессора — еще не ЭВМ
 7.24. Схема микропроцессора — еще не ЭВМ
 7.25. Схема микропроцессора — еще не ЭВМ
 7.26. Схема микропроцессора — еще не ЭВМ
 7.27. Схема микропроцессора — еще не ЭВМ
 7.28. Схема микропроцессора — еще не ЭВМ
 7.29. Схема микропроцессора — еще не ЭВМ
 7.30. Схема микропроцессора — еще не ЭВМ
 7.31. Схема микропроцессора — еще не ЭВМ
 7.32. Схема микропроцессора — еще не ЭВМ
 7.33. Схема микропроцессора — еще не ЭВМ
 7.34. Схема микропроцессора — еще не ЭВМ
 7.35. Схема микропроцессора — еще не ЭВМ
 7.36. Схема микропроцессора — еще не ЭВМ
 7.37. Схема микропроцессора — еще не ЭВМ
 7.38. Схема микропроцессора — еще не ЭВМ
 7.39. Схема микропроцессора — еще не ЭВМ
 7.40. Схема микропроцессора — еще не ЭВМ
 7.41. Схема микропроцессора — еще не ЭВМ
 7.42. Схема микропроцессора — еще не ЭВМ
 7.43. Схема микропроцессора — еще не ЭВМ
 7.44. Схема микропроцессора — еще не ЭВМ
 7.45. Схема микропроцессора — еще не ЭВМ
 7.46. Схема микропроцессора — еще не ЭВМ
 7.47. Схема микропроцессора — еще не ЭВМ
 7.48. Схема микропроцессора — еще не ЭВМ
 7.49. Схема микропроцессора — еще не ЭВМ
 7.50. Схема микропроцессора — еще не ЭВМ
 7.51. Схема микропроцессора — еще не ЭВМ
 7.52. Схема микропроцессора — еще не ЭВМ
 7.53. Схема микропроцессора — еще не ЭВМ
 7.54. Схема микропроцессора — еще не ЭВМ
 7.55. Схема микропроцессора — еще не ЭВМ
 7.56. Схема микропроцессора — еще не ЭВМ
 7.57. Схема микропроцессора — еще не ЭВМ
 7.58. Схема микропроцессора — еще не ЭВМ
 7.59. Схема микропроцессора — еще не ЭВМ
 7.60. Схема микропроцессора — еще не ЭВМ
 7.61. Схема микропроцессора — еще не ЭВМ
 7.62. Схема микропроцессора — еще не ЭВМ
 7.63. Схема микропроцессора — еще не ЭВМ
 7.64. Схема микропроцессора — еще не ЭВМ
 7.65. Схема микропроцессора — еще не ЭВМ
 7.66. Схема микропроцессора — еще не ЭВМ
 7.67. Схема микропроцессора — еще не ЭВМ
 7.68. Схема микропроцессора — еще не ЭВМ
 7.69. Схема микропроцессора — еще не ЭВМ
 7.70. Схема микропроцессора — еще не ЭВМ
 7.71. Схема микропроцессора — еще не ЭВМ
 7.72. Схема микропроцессора — еще не ЭВМ
 7.73. Схема микропроцессора — еще не ЭВМ
 7.74. Схема микропроцессора — еще не ЭВМ
 7.75. Схема микропроцессора — еще не ЭВМ
 7.76. Схема микропроцессора — еще не ЭВМ
 7.77. Схема микропроцессора — еще не ЭВМ
 7.78. Схема микропроцессора — еще не ЭВМ
 7.79. Схема микропроцессора — еще не ЭВМ
 7.80. Схема микропроцессора — еще не ЭВМ
 7.81. Схема микропроцессора — еще не ЭВМ
 7.82. Схема микропроцессора — еще не ЭВМ
 7.83. Схема микропроцессора — еще не ЭВМ
 7.84. Схема микропроцессора — еще не ЭВМ
 7.85. Схема микропроцессора — еще не ЭВМ
 7.86. Схема микропроцессора — еще не ЭВМ
 7.87. Схема микропроцессора — еще не ЭВМ
 7.88. Схема микропроцессора — еще не ЭВМ
 7.89. Схема микропроцессора — еще не ЭВМ
 7.90. Схема микропроцессора — еще не ЭВМ
 7.91. Схема микропроцессора — еще не ЭВМ
 7.92. Схема микропроцессора — еще не ЭВМ
 7.93. Схема микропроцессора — еще не ЭВМ
 7.94. Схема микропроцессора — еще не ЭВМ
 7.95. Схема микропроцессора — еще не ЭВМ
 7.96. Схема микропроцессора — еще не ЭВМ
 7.97. Схема микропроцессора — еще не ЭВМ
 7.98. Схема микропроцессора — еще не ЭВМ
 7.99. Схема микропроцессора — еще не ЭВМ
 7.100. Схема микропроцессора — еще не ЭВМ

МИКРОПРОЦЕССОР КАК ИНТЕГРАЛЬНАЯ СХЕМА

Созданный в 1971 году первый микропроцессор (МП) был предназначен для работы в качестве процессора четырехразрядной ЭВМ и размещался в стандартном корпусе с шестнадцатью выводами. Собственно микросхема была выполнена на кремниевой пластинке размерами $2,8 \times 3,8$ мм и содержала 2250 транзисторов.

В вычислительной технике МП — «кирпичик» для построения компактных высокопроизводительных ЭВМ, в электронике — основной элемент **контроллеров** — устройств управления, непосредственно встраиваемых в объекты.

И в том, и в другом случае основная функция МП — выполнение написанных для него программ, что возможно только при постоянном взаимодействии МП с окружающим миром. Микропроцессор должен откуда-то получать данные и куда-то направлять результаты их обработки. Кроме того, само выполнение программы требует поступления в МП кодов команд из ОЗУ или ПЗУ, которые обычно не входят в его состав.

Использованные в данной и последующих главах названия сигналов, информационных линий, аббревиатуры команд имеют англоязычное происхождение.

7.1 Микропроцессор — еще не ЭВМ

Двигатель может выполнять различную работу, но требует для этого источника энергии, приспособлений, формирующих движения нужного характера, и объектов воздействия. Работа МП возможна при наличии источников электропитания и, по

крайней мере, двух типов устройств: запоминаящих и *ввода-вывода данных* (УВВ). Связь МП с УВВ и ЗУ осуществляется по *шинам* — системам проводов, несущих сигналы от МП к *внешним устройствам* (ВУ) и обратно. Выработка и прием этих сигналов как в МП, так и в ВУ осуществляется специальными схемами, называемыми интерфейсными. Вообще *интерфейс* (от англ. *interface* — стык, взаимодействие) — это совокупность средств, определяющих логический порядок взаимодействия систем (протокол) и вытекающие из протокола требования к аппаратуре и к программному обеспечению, если обмен данными осуществляется под управлением программы.

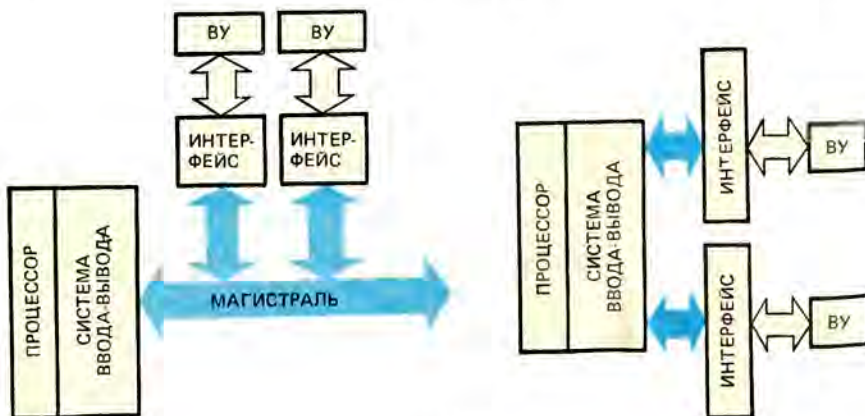
На практике под интерфейсом чаще всего понимают только электрическую схему, содержащуюся в устройстве для сопряжения его с ЭВМ, а интерфейсные средства, входящие в состав УУ процессора, носят название *системы ввода-вывода*. Чтобы упростить интерфейс внешних устройств, самые сложные его функции возлагают на систему ввода-вывода процессора. При наиболее распространенном (магистральном) принципе организации интерфейсы устройств единообразно подключаются к общей системе шин — *магистралам*, исходящей из процессора (рис. 7.1, а). При *радиальной организации интерфейса* к каждому устройству от процессора идет своя система шин (рис. 7.1, б).

Состав ВУ, подключаемых к магистралам МП, зависит от характера решаемых задач. Рассмотрим три основные конфигурации систем на базе МП (рис. 7.2).

Контроллер, или специализированное устройство обработки информации (рис. 7.2, а), помимо МП, содержит ПЗУ для хранения заранее отлаженной программы;

Рис. 7.1.

В вычислительных системах используются магистральный (а) и радиальный (б) принципы организации интерфейса. Физически магистраль выполняется в виде многопроводной шины с разъемами для подключения внешних устройств (ВУ), а в одноплатной микро-ЭВМ — с гнездами (панельками) для микро-схем.



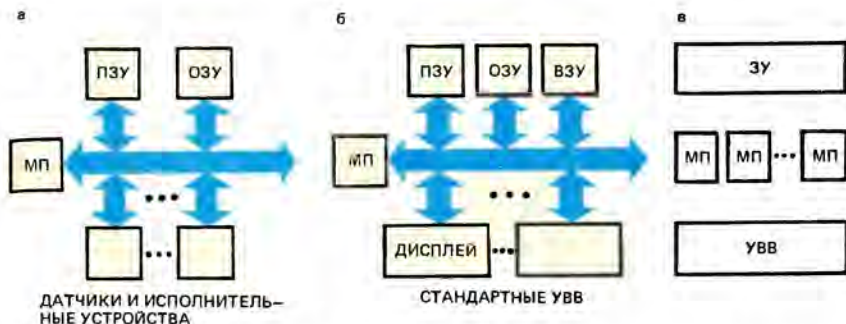


Рис. 7.2.

Основные конфигурации систем на основе микропроцессоров: контроллер, или специализированная микро-ЭВМ (а), микро-ЭВМ общего назначения (б), многопроцессорная ЭВМ, или система распределенной обработки информации (в). Для каждой из систем последнего типа внутренние связи специфичны. Интерфейсные схемы не показаны.

ОЗУ для входных, промежуточных или выходных данных, в качестве которого в простейших случаях служат имеющиеся в МП РОНЫ; интерфейсы используемых датчиков и исполнительных устройств, нередко выполняемые в виде специальных БИС.

Микро-ЭВМ общего назначения (рис. 7.2, б) содержит МП, достаточно емкое ОЗУ, ПЗУ для хранения часто используемых программ, в том числе программ для начальной загрузки. В итоге этой загрузки в ОЗУ помещается программа, обеспечивающая диалог ЭВМ с человеком. Диалог осуществляется обычно посредством терминала — стандартного внешнего устройства, например алфавитно-цифрового дисплея, в котором ввод символов производится с клавиатуры, а вывод — на экран электронно-лучевой трубки. Перечисленные устройства образуют ЭВМ так называемой минимальной конфигурации. Она может быть расширена путем подключения внешних ЗУ (ВЗУ) типа накопителей на магнитных дисках (НМД), на гибких магнитных дисках (НГМД), на магнитных лентах (НМЛ), печатающих устройств, графопостроителей, средств телеобработки, обеспечивающих связь ЭВМ с удаленными объектами, в том числе и с другими ЭВМ (модемы, адаптеры, мультиплексоры передачи данных).

Интерфейсы некоторых из перечисленных устройств выполняют достаточно сложные функции и фактически представляют собой контроллеры. Нередко они строятся на основе собственных микропроцессоров. По этой причине основной МП микро-ЭВМ будем называть в дальнейшем *центральным процессором* (ЦП).

МП как составная часть высокопроизводительной ЭВМ или системы распределенной обработки данных (рис. 7.3, в). В простейших случаях такие важные параметры как длина машинного слова могут быть пропорциональны числу используемых МП. С его увеличением фактически возрастает параллелизм, а значит, — быстродействие обработки

информации. В иных случаях отдельные МП решают различные, независимые друг от друга части сложной задачи. Системы такого типа носят название *мультипроцессорных* или параллельных. Конечно, параллельный характер работы для них следует понимать в более широком смысле, чем параллельную, но одинаковую обработку битов в многоразрядном слове.

7.2. Типы микропроцессоров

По характеру использования различают три основных типа МП.

1. *Однокристалльные микро-ЭВМ* (микроконтроллеры), предназначенные для встраивания их в приборы, станки и другое оборудование. С целью минимизации дополнительно подключаемых ИС для таких МП характерно наличие на одном кристалле хотя бы небольших по объему ОЗУ и ПЗУ, нескольких распространенных интерфейсов ввода-вывода информации, в том числе и в аналоговой форме.

2. *МП общего назначения* чаще всего применяются в микро-ЭВМ. Они характеризуются стремлением разработчиков добиться максимального быстродействия при как можно больших длине слова и объеме адресуемой памяти и развитой системой команд, удобной для решения самых разнообразных задач. Такие МП могут быть реализованы в виде одной или нескольких БИС, образующих т. н. базовый микропроцессорный комплект. Каждая из ИС этого комплекта реализует отдельный узел или функциональный блок МП, например АЛУ и УУ.

3. *Секционированные МП*, позволяющие наращивать длину обрабатываемого слова увеличением числа используемых секций, имеют максимальное быстродействие, используются для построения высокопроизводительных ЭВМ и быстродействующих специализированных вычислительных систем.

При характеристике конкретного МП необходимо указать его принадлежность к одному из рассмотренных типов. При этом обычно выделяют следующие параметры:

- 1) длину слова (разрядность);
- 2) объем адресного пространства — максимально возможный диапазон адресов ОЗУ;
- 3) особенности системы ввода-вывода и магистрали;
- 4) количество и функции РОИ;
- 5) способы адресации памяти;
- 6) систему команд и систему микропрограммирования (см. 8.10), если оно возможно;
- 7) быстродействие, обычно оцениваемое количеством операций определенного типа, выполняемых за одну секунду.

Далее будет рассмотрен универсальный шестнадцатиразрядный однокристалльный микропроцессор К1801ВМ1, имеющий быстродействие около 500 тысяч операций в секунду для регистровых команд, который используется в микро-ЭВМ «Электроника

МС1201», входящих в состав диалоговых вычислительных комплексов типа ДВК-1, ДВК-2 и др. Такой выбор не случаен, так как эта ЭВМ — одна из представителей многочисленного семейства машин, чрезвычайно широко распространенных как в нашей стране, так и за рубежом. Все ЭВМ этого семейства совместимы по своему программному обеспечению и отличаются в основном конфигурацией и быстродействием. В их числе: микро-ЭВМ «Электроника-60», «Электроника-81», «Электроника-82», «Электроника-85», машины на основе микропроцессорного комплекта K588, мини-ЭВМ СМ-3, СМ-4, СМ-1420, СМ-1300, СМ-1600, «Электроника-100/25», «Электроника-79» и др. модели. За рубежом машины этого семейства наиболее широко представлены моделями серий PDP-11 (мини-ЭВМ) и LSI-11 (микро-ЭВМ) фирмы DEC. Высокие технические характеристики и надежность перечисленных машин сделали их общепринятым стандартом для решения самых разнообразных задач.

7.3. Способы обмена данными

Представьте себе, что, поставив на плиту чайник, вы удобно уселись в соседней комнате с увлекательной книгой. Маловероятно, что чайник вскипит в тот самый момент, когда, дойдя до конца главы, можно будет без большого сожаления остановиться. Скорее всего, придется периодически отвлекаться, чтобы присматривать за чайником. В более выгодном положении оказывается владелец чайника со свистком, сигнализирующим о закипании воды.

Пример имеет вполне определенное отношение к способам обмена данными между процессором и внешними устройствами.

Как правило, все ВУ работают несинхронно с процессором и, за исключением ОЗУ, значительно медленнее его. Поэтому и для процессора, и для ВУ сигналы магистрали появляются в непредсказуемые моменты времени. Чтобы исключить потери информации в процессе обмена, необходимы специальные меры.

Интерфейс каждого ВУ имеет как минимум два специальных программно-доступных регистра. Один из них — *регистр данных* (РД) — служит в качестве буфера между магистралью и ВУ. Он хранит слово, выработанное в ВУ, до того момента, когда процессор будет в состоянии принять его. В операциях вывода процессор помещает данные в РД, где они могут храниться до момента высвобождения ВУ. В обоих случаях и процессор, и ВУ, передав слово в РД, могут незамедлительно продолжить свою работу, но к факту помещения слова в РД или готовности к его приему должно быть привлечено внимание процессора. Это осуществляется с помощью второго регистра — *регистра состояния* (РС).

В простейшем случае используется только один бит РС, называемый *флагом готовности*. Этот бит устанавливается в единичное состояние, если РД готов к очередному сеансу обмена. И РД, и РС подключены к магистрали, поэтому процессор может определить готовность ВУ при проверке содержимого РС. Если флаг уста-

новлен, производится условный переход к команде считывания или записи, после выполнения которой флаг автоматически сбрасывается.

При неустановленном флаге ВУ процессор циклически продолжает его опрос. Такой способ обмена данными процессора с ВУ называется *асинхронным программным обменом* (рис. 7.4).

Для обмена со сложными ВУ процессору может оказаться недостаточно информации только о готовности ВУ, поэтому состояние ВУ отражается несколькими битами РС.

Основной недостаток асинхронного обмена — бесполезные потери времени процессора на ожидание готовности ВУ. Значительно уменьшить эти потери позволяют *прерывания программы*. В этом случае УУ процессора через специальную линию магистрали реагирует на установку флага ВУ автоматическим переключением с исполняемой программы на заранее подготовленную программу, которая и производит обмен данными.

С реализацией прерываний связано несколько трудностей. Во-первых, по завершению программы обслуживания ВУ необходимо продолжить выполнение прерванной программы, полностью восстановив состояние процессора к моменту прерывания. Во-вторых, при наличии нескольких источников прерываний, подключенных к единственной линии запроса, процессору нужно уметь быстро «распознать» ВУ, нуждающееся в обслуживании. В-третьих, нужно эффективно разрешать конфликтные ситуации, когда, например, прерывания требуют одновременно несколько ВУ, либо новый сигнал прерывания (запрос) приходит в момент, когда еще не кончилась обработка предыдущего запроса (*«вложенные прерывания»*).

Распространенный способ преодоления этих трудностей — *векторная система прерываний*. При этом программы обслуживания ВУ организуются так же, как обычные *подпрограммы**, с той разницей, что их вызов осуществляется не специальной командой, а аппаратно, по запросам с магистрали. Как при обращении к подпрограммам, так и в момент прерывания УУ автоматически заносит содержимое счетчика команд и ССП в стек. Помимо сигнала запроса, ВУ обязано установить на шинах магистрали адрес вектора прерывания в ОЗУ. *Вектор прерывания* — две последовательные ячейки памяти, в которых находятся начальные адрес и состояние подпрограммы обслуживания данного ВУ. Получив адрес вектора прерывания, процессор автоматически загружает СК и ССП содержимым ячеек, образующих вектор, что обеспечивает выполнение подпрограммы обслуживания. В ее конце обязательно должна присутствовать специальная команда возврата из подпрограммы, по которой из стека извлекаются «старые» значения СК и ССП, обеспечивающие продолжение выполнения основной программы с того места, где она была

* Подпрограмма — фрагмент программы, допускающий многократное обращение из различных мест основной программы и обеспечивающий автоматический возврат в то место, откуда произошло обращение (подробнее см. 8.4).



Рис. 7.3.

Программа для асинхронного обмена данными содержит команду тестирования флага, т. е. считывания его состояния в один из битов ССП. Следующая команда — условная передача управления — заставляет процессор вновь повторять тестирование, если принятое состояние флага — нулевое.

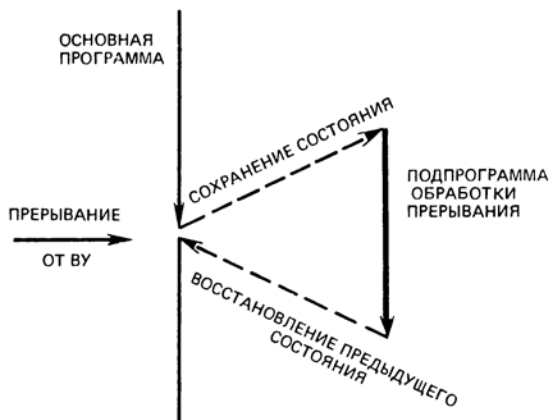
прервана (рис. 7.4). Основным принцип стека «первым вошел — последним вышел» обеспечивает правильную последовательность возврата в основную программу в случае вложенных прерываний.

Единичное значение специального бита разрешения прерывания в ССП блокирует всю систему прерываний, а соответствующие комбинации битов в РС запрещают выдачу на магистраль запросов на прерывание от отдельных устройств или, что то же самое, осуществляют *маскирование прерываний*.

Гибкость системы прерываний может быть повышена назначением *приоритетов* как основной программе, так и подпрограммам обслуживания ВУ. С этой целью используются выделенные группы битов в ССП и векторах прерываний. В «старших» ЭВМ упомянутого семейства допустимы 4 уровня приоритета (от 4-го до 7-го). Получив запрос на прерывание, процессор реагирует на него только тогда, когда приоритет ВУ больше приоритета исполняемой программы. Назначение ВУ различных приоритетов позволяет процессору легко разрешать конфликты, возникающие при одновременном запросе на прерывание от нескольких устройств. Об установке битов приоритета и разрешения прерываний обязан

Рис. 7.4.

Реакция процессора на запрос прерывания от ВУ состоит в «незаметном» для основной (фоновой) программы выполнении программы обработки прерывания.



позаботиться программист. После включения процессора все эти биты автоматически очищаются. Их установка обычно производится специальными командами в самом начале выполнения программы.

Прерывания — не только способ совершенствования механизма обмена данными с ВУ. Очень удобны т. н. *внутренние прерывания*, происходящие при сбоях в работе аппаратуры, неисправностях электропитания, переполнении памяти, недопустимых командах, попытках деления на нуль и т. п. Другие прерывания (программные) инициируются специальными командами *TRAP* («ловушка»). В сложных программных системах «ловушки», вызываемые командами *EMT*, используются для переключения процессора на выполнение программ, реализующих некоторые стандартные для данной системы функции. Внутренние прерывания имеют свои векторы и обрабатываются аналогично внешним, но в первую очередь.

Обмен данными путем прерываний повышает эффективность работы процессора, исключая непроизводительные циклы ожидания готовности ВУ. Тем не менее определенные потери времени остаются. Действительно, часть времени отнимает загрузка и выгрузка стека, выполнение команды возврата. Более того, подпрограмма обслуживания, использующая РОНЫ, обязана переслать в ОЗУ их содержимое в момент входа в прерывание и восстановить его перед возвратом, иначе может нарушиться выполнение прерванной программы. Неизбежны «накладные» расходы и на извлечение из ОЗУ самих команд обмена.

Минимальные потери времени и максимальную скорость обеспечивает обмен данными путем *прямого доступа в память* (ПДП). Получив через особую линию запрос на такого рода обмен, процессор почти полностью отключается от магистрали, позволяя ВУ взаимодействовать непосредственно с ОЗУ. Обмен словом или байтом в этом случае приостанавливает работу процессора на длительность одного цикла ОЗУ. Говорят, что ВУ заимствует или «ворует» этот цикл у процессора, совершенно не изменяя его состояния.

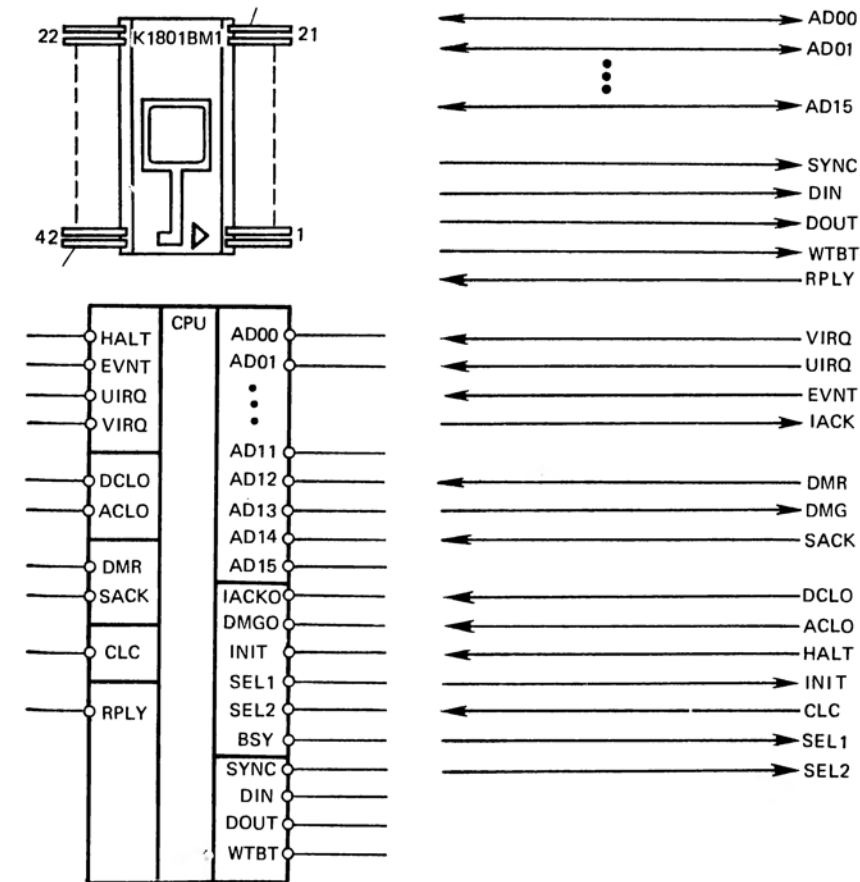
Обмен данными посредством ПДП имеет наивысший приоритет. Если потребность в таком обмене возникает не слишком часто, то процессор, выполняя любую программу, практически не «чувствует» его и оба процесса происходят как бы одновременно. Выигрыш в быстродействии особенно велик при необходимости передачи больших массивов информации, например, при работе с ВЗУ: для самого обмена не нужна программа, а значит, нет потерь времени на выполнение и извлечение команд из ОЗУ. Достоинства ПДП имеют негативную сторону в виде заметного усложнения интерфейса ВУ, полностью берущего на себя управление магистралью.

Магистраль, память и ВУ

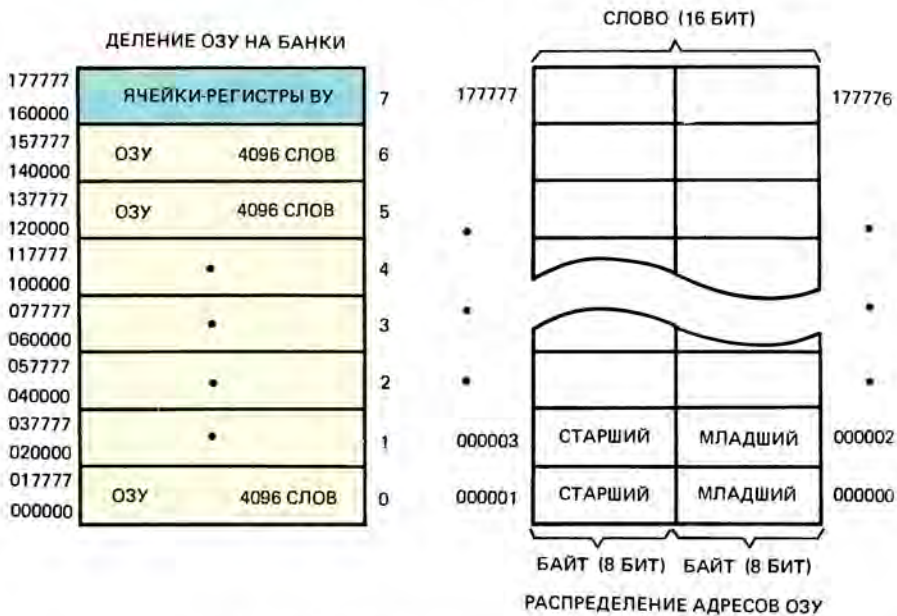
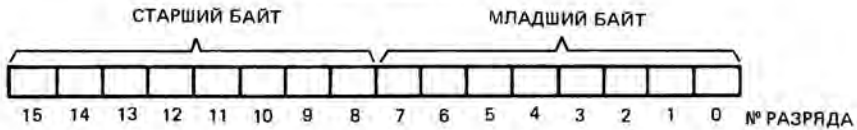
Линии, образующие магистраль передачи информации, можно разделить на три группы: *информационную, адресную и управляющую*. В полном составе все эти шины имеются в магистралях «старших» моделей семейства малых ЭВМ и называются «общая шина». Для магистралей микропроцессоров с целью сокращения числа линий характерно совместное использование одной шины для поочередной передачи вначале адреса, а затем — данных. Именно так построена магистраль микропроцессора К1801ВМ1, носящая название *МПИ* (магистраль передачи информации).

Основу МПИ составляет 16-разрядная *двухнаправленная шина*

БИС МП К1801 ВМ1 оформлена в металлокерамическом корпусе, имеющем 42 вывода, которые предназначены для передачи информационных и управляющих сигналов МПИ, а также подключения к источнику питания.



ОРГАНИЗАЦИЯ МАШИННОГО СЛОВА



16-разрядные машинные слова позволяют указывать $2^{16} = 65536 = 64\text{К}$ различных адресов. Старшие $8\text{К} = 8192$ адреса отведены для регистров ВУ, а остальные — для ОЗУ и ПЗУ. Эта часть «адресного пространства» делится на страницы, или банки по 8К адресов, что упрощает создание ЗУ с нужной пропорцией между ОЗУ и ПЗУ (блоки памяти выпускаются с емкостью, кратной 8К адресов).

адреса/данных. Линии управления можно объединить в четыре группы, как это показано на рисунке 7.5.

Каждое ВУ, точнее, любой из его регистров (РД или РС) имеет свой номер — адрес. Этим адресом процессор снабжает любую порцию выводимой информации (слово или байт). Аналогично, каждому слову, вводимому в процессор, предшествует адрес ВУ «отправителя». ЗУ подключаются к МПИ почти так же, как и остальные ВУ, с той лишь разницей, что ячейки памяти имеют другие адреса (рис. 7.6).

Может возникнуть вопрос, почему до сих пор размеры памяти определялись числом адресов, а не слов. Это объясняется тем, что рассматриваемые процессоры способны обращаться не только

к шестнадцатиразрядным словам, но и к каждому из составляющих их байтов. При этом свои адреса имеют каждый из двух байтов любого слова: четные — для младших байтов и нечетные — для старших. Поэтому число слов всегда в два раза меньше числа адресов. Так, выраженные в словах объемы всего адресного пространства и одного банка составляют соответственно 32К и 4К. Для однозначности принято, что адрес слова совпадает с адресом его младшего байта.

Информационное «рукопожатие»

Передача данных между любыми двумя устройствами, подключенными к магистрали, осуществляется по принципу «управляющий — управляемый». Управляющим в каждый момент времени может быть только одно устройство, и именно оно осуществляет все функции по управлению передачей данных по МПИ. Как правило, хозяином МПИ является сам процессор. Перехватить управление магистралью разрешается лишь устройствам прямого доступа в память, таким, как контроллеры накопителей на магнитных лентах, дисках и т. п., и лишь на период обмена с ОЗУ большими порциями данных. В этом случае процессор «освобождает» магистраль, переводя свои выходы в состояние с высоким выходным сопротивлением, и приостанавливает выполнение программы до окончания работы ВУ по пересылке данных.

Сначала рассмотрим случай, когда управляющим устройством является процессор, исполняющий некоторую программу. При исполнении команд он обращается к ячейкам ОЗУ и регистрам внешних устройств с целью:

- 1) выборки команды из ОЗУ;
- 2) ввода данных из ОЗУ/считывания содержимого регистров ВУ;
- 3) записи данных в ОЗУ/регистры ВУ.

В любом случае обмен осуществляется с помощью одного или нескольких *циклов* магистрали, за каждый из которых по шине адреса/данных передается одно машинное слово или байт данных. Каждый цикл имеет свой *протокол*, т. е. порядок обмена сигналами между ведущим и ведомым устройствами. В МПИ реализуются циклы 4 видов:

- 1) ввод данных;
- 2) вывод данных (вывод слова);
- 3) вывод байта;
- 4) ввод-пауза-вывод.

Общее для всех циклов — *асинхронный характер*. В данном случае это означает, что на каждый управляющий сигнал должен поступить сигнал ответа от управляемого устройства (обмен «рукопожатием»), причем время ожидания сигнала ответа (таймаут) ограничено величиной 10 мкс. Если в течение таймаута управляющее устройство не получило ответа, происходит внутреннее прерывание.

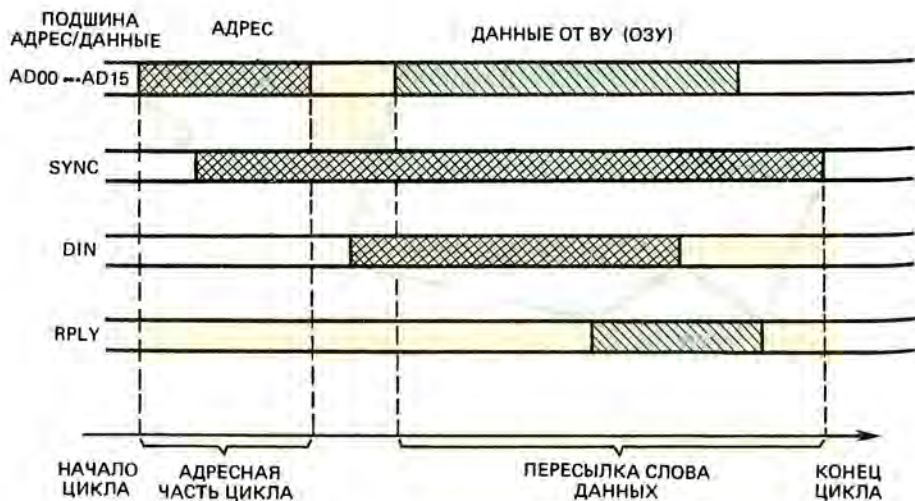
Асинхронное выполнение циклов передачи данных позволяет вести с максимально высоким темпом обмен информацией между устройствами, обладающими самым различным быстродействием.

Циклы магистрали

Каждый из 4 циклов МПИ начинается с адресации ячейки ОЗУ или регистра ВУ, к которому будет производиться обращение. В ходе *адресной части цикла* процессор устанавливает нужный адрес на шине адреса-данных (линии AD00-AD15) и вырабатывает **сигнал синхронизации адреса SYNC** (SYNCronization), который действует в течение всего цикла. С появлением сигнала SYNC переданный адрес воспринимается всеми управляемыми устройствами, подключенными к МПИ, и адресная часть цикла заканчивается. За адресной частью цикла следует информационная, в которой выполняется конкретная операция: ввод, вывод, вывод байта, ввод-пауза-вывод. В информационной части цикла принимает участие лишь то управляемое устройство, которое имеет в своем составе ячейку (регистр) с адресом, принятым в адресной части цикла, что обеспечивается так называемым селектором (де-

После адресной части цикла «ввод» ЦП подает сигнал DIN (Data IN — ввод данных). В ответ на это ВУ устанавливает требуемые данные на линиях AD00-AD15 и формирует сигнал ответа RPLY (RePLY — отзыв). Получив сигнал RPLY, процессор принимает данные и снимает сигнал DIN. Вслед за этим устройство снимает сигнал RPLY, а процессор — сигнал SYNC. На этом и следующих рисунках используются такие обозначения:

— сигнал передается управляющим устройством (процессором);
 - сигнал передается управляемым устройством;





После адресной части цикла «вывод» ЦП устанавливает на шине адреса/данных выводимое слово (или байт), а затем формирует сигнал DOUT (Data OUT — вывод данных). Выбранное ВУ, получив сигнал DOUT, принимает данные с шины адреса/данных, после чего формирует сигнал ответа RPLY. Получив ответ, процессор снимает сигналы DOUT, SYNC и данные с линий AD00-AD15. WTBT устанавливается только в цикле «вывод байта».

шифратором) адреса. Назовем такое устройство *выбранным*.

Цикл «ввод». В течение цикла «ввод» (рис. 7.7) производится передача машинного слова из ОЗУ или регистра ВУ в ЦП. При исполнении любой команды ЦП по крайней мере один раз выполняет цикл «ввод» с целью ее выборки.

Цикл «ввод-пауза-вывод» напоминает цикл «ввод», но по окончании приема данных сигнал SYNC не снимается, а сразу следует информационная часть цикла «вывод». Адрес, принятый в адресной части цикла, сохраняется неизменным в течение всего времени действия сигнала SYNC. WTBT устанавливается при операциях с байтами.



Циклы «вывод» и «вывод байта». При выполнении циклов «вывод» и «вывод байта» производится передача из процессора в ячейку ОЗУ или регистр ВУ соответственно одного машинного слова или байта. В цикле «вывод байта» передается лишь 1 байт данных. При этом в информационной части цикла устанавливается сигнал **WTBT** (Write Byte — запись байта), как это показано на рисунке 7.8.

Цикл «ввод-пауза-вывод». При выполнении многих команд (например, увеличения на 1) возникает необходимость в модификации содержимого выбранной ячейки ОЗУ. Эту операцию можно произвести с помощью последовательного исполнения цикла «ввод» и цикла «вывод», но поскольку адрес ячейки не меняется, то можно обойтись без его повторной передачи, а два отдельных цикла объединить в один. Такой цикл называется *циклом «ввод-пауза-вывод»* (рис. 7.9).

Прерывание и прямой доступ в память

Устройство, которому требуется прервать программу, вырабатывает сигнал **VIRQ** (Vector Interrupt ReQuest — запрос векторного прерывания). Процессор, получив запрос **VIRQ**, удовлетворяет требование путем выдачи сигналов **DIN** и **IACK O** (Interrupt Acknowledge Output — выходной сигнал предоставления прерывания). Получив сигналы **DIN** и **IACK O**, прерывающее устройство снимает сигнал **VIRQ** и устанавливает на линиях **AD00-AD15** адрес «своего» вектора прерывания, сопровождая его сигналом ответа **RPLY** (как и в цикле «ввод»). Приняв адрес вектора прерывания, ЦП снимает сигналы **DIN** и **IACK O** и переходит к подпрограмме обслуживания прерывания, а устройство снимает сигнал **RPLY**. На этом процесс предоставления прерывания заканчивается. Временная диаграмма этой процедуры представлена на рисунке 7.10.

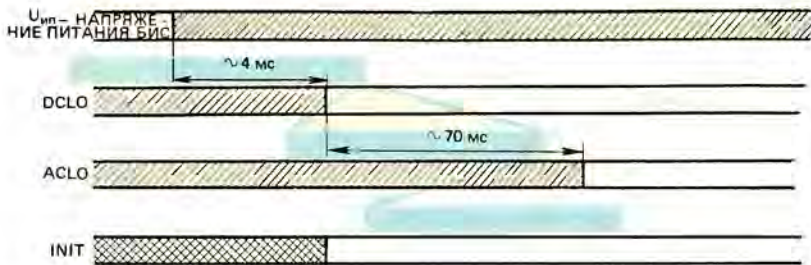




В отличие от остальных сигналов МПИ, сигналы предоставления прерывания и прямого доступа в память проходят «транзитом» через все ВУ. Обозначение входного для ВУ сигнала предоставления принято оканчивать буквой I (Input), а выходного — O (Output). Поэтому одна и та же линия, соединяющая два устройства, именуется по-разному с каждого конца.

В МПИ имеются специальные линии для организации двух так называемых *радиальных прерываний*, не требующих передачи адреса вектора прерывания. Эти линии носят название **UIRQ** (User Interrupt ReQuest — запрос пользовательского прерывания) и **EVNT** (EVeNT — событие, для прерывания от таймера микропроцессорной системы). При подаче сигналов на эти линии процессор автоматически выполняет переход на подпрограммы обработки прерываний с адресами векторов соответственно 270₈ и 100₈. Сигналы IACK O и DIN при этом не вырабатываются, а сигнала ответа RPLY не требуется. Линия EVNT обычно возбуждается с частотой 50 Гц, что позволяет специальной программе вести учет текущего времени («часы»). Линия UIRQ предназначена для использования разработчиком микропроцессорной системы по своему усмотрению.

Отметим одну важную особенность в организации прерываний. Прерывание программы, как и предоставление прямого доступа в память, происходит по инициативе внешнего устройства. В системе с множеством внешних устройств возникает опасность того, что в один и тот же момент прервать программу понадобится не одному, а сразу нескольким ВУ. По этой причине в МПИ предусмотрена защита от ошибочных ситуаций, связанных с одновременной передачей сразу нескольких векторов прерывания или «захватом» магистрали сразу несколькими устройствами прямого доступа в память. Она заключается в том, что линии, несущие сигналы IACK O и DMG O (выходные сигналы предоставления прерывания и прямого доступа в память), проходят последовательно через все устройства (рис. 7.11). Любое устройство, получив в ответ на требование прерывания или прямого доступа в память соответствующий сигнал предоставления, обязано прежде всего запретить его дальнейшее распространение, т. е. прохождение к следующему устройству, а затем реализовать операции прямого доступа или прерывание программы. Если же оно не требовало прерывания или прямого доступа в память, то соответствующие



С целью прямого доступа в память ВУ вырабатывает сигнал требования ПДП DMR (Direct Memory access Request). Завершив текущую команду, процессор приостанавливает выполнение программы и формирует сигнал DMG O (Direct Memory access Grant — предоставление ПДП). Приняв сигнал DMG O, ВУ запрещает его дальнейшее прохождение и формирует сигнал SACK (Selection ACKnowledge — подтверждение выбора). В ответ процессор снимает сигнал DMG O и «замораживает» на период действия сигнала SACK, т. е. до окончания обмена.

сигналы предоставления должны быть переданы на следующее устройство.

Вам приходилось, наверное, видеть, как в час «пик» полупустой трамвай за несколько секунд заполняется пассажирами. Большинство из них сразу же пытается передать «пробить» проездные талоны. В суматохе уже не так просто разобраться, кому какой талон принадлежит. Чаще всего поступают так: кто стоит ближе к компостеру, получает первый попавшийся, а остальные передает дальше по цепочке. Что-то похожее имеет место при приоритетной организации процедур предоставления прерываний и прямого доступа в память. Таким образом устройство, расположенное на магистрали ближе к процессору, обладает и большим приоритетом на обмен.

Процедура предоставления *прямого доступа в память* (ПДП) связана с полным освобождением процессором магистрали на период пересылки данных между ОЗУ и ВУ (рис. 7.12). Управление МПИ при этом берет на себя само внешнее устройство. После окончания пересылок ВУ освобождает МПИ и процессор продолжает выполнение прерванной программы, для которой сеанс прямого доступа в память проходит почти «незаметно».

Начальный пуск и синхронизация микропроцессора

МПИ содержит несколько вспомогательных линий, осуществляющих управление запуском микропроцессора после включения питания. К ним относятся линии **ACLO** (Alternative Current LOw — авария сетевого питания), **DCLO** (Direct Current LOw — авария источника питания), **INIT** (INITialization — начальная установка магистрали), **SEL1** и **SEL2**. Сигналы ACLO и DCLO вырабатываются источником питания микропроцес-

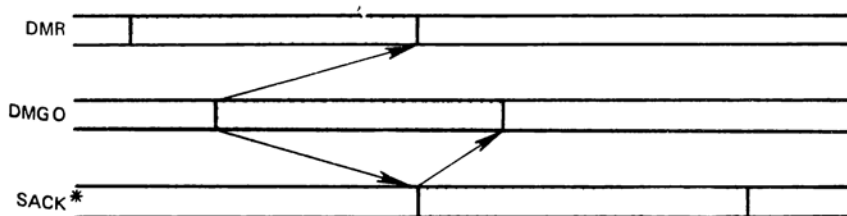
сорной системы и являются входными для ЦП (рис. 7.13).

Сигнал INIT подается процессором при выполнении специальной команды «сброс», запуске программы, в момент включения питания и служит для установки всех устройств на магистрали в исходное состояние.

Начальный пуск микропроцессора представляет собой переход к исполнению некоторой программы с определенного адреса в момент окончания действия сигнала ACLO. Адрес начального пуска микропроцессора задается с помощью специального внешнего регистра начального пуска (РНП), имеющего адрес 177716_8 . По окончании действия сигнала ACLO микропроцессор автоматически осуществляет чтение содержимого РНП, заносит его в счетчик команд и приступает к выполнению программы со сформированного таким образом адреса. Для упрощения устройства регистра начального пуска микропроцессор имеет специальный вывод SEL1, на котором появляется сигнал в момент чтения РНП. По этому сигналу схема, реализующая РНП, должна выставить на линии AD00-AD15 адрес начального пуска, а подача сигнала RPLY при этом не требуется. Отметим, что с помощью РНП можно задавать адрес начального пуска, кратный 400_8 (младший байт СК очищается).

Обычно в качестве адреса начального пуска используют *стартовый адрес* программы начального загрузчика операционной системы с магнитного диска, записанной в постоянном запоминающем устройстве, либо (при использовании МП в качестве контроллера) стартовый адрес управляющей программы, также хранящейся в ПЗУ. Во многих микро-ЭВМ используется специальное ПЗУ, содержащее *программу-эмулятор пульта*, которой может передаваться управление в момент начального пуска. Эта программа позволяет использовать терминал микро-ЭВМ в качестве пульта управления, который обычно имеется у мини- и больших ЭВМ. Программа-эмулятор принимает с терминала и выполняет некоторый набор команд типа «распечатать содержимое ячейки ОЗУ», «начать выполнение программы с определенного адреса» и т. п. (подробнее об этом см. 8.7). Еще одна особенность МП K1801BM1 и большинства других — необходимость в так называемом «так-

Для осуществления нормального запуска МП K1801BM1 сигналы ACLO и DCLO должны подаваться в указанном на временной диаграмме порядке при включении питания и в обратном порядке при выключении.



товом питании», которое представляет собой синхросигнал, вырабатываемый отдельным генератором и подаваемый на специальный вход синхронизации микропроцессора CLK. Он осуществляет синхронизацию работы внутренних узлов МП, т. е. является «жизненно важным» для нормальной работы микропроцессорных БИС.

Радиальные интерфейсы

Очень большая, но конечная, скорость распространения электрических сигналов по проводам заставляет снижать темп обмена данными через МПИ с увеличением ее длины. Как следствие все входящие в систему ВУ должны размещаться в непосредственной близости от ЦП, что не всегда удобно. Поэтому для подключения удаленных устройств часто используют *контроллеры радиальных интерфейсов*. К каждому из таких контроллеров можно подключить лишь одно ВУ, соединив его с контроллером более длинной низкоскоростной линией связи, которая состоит из относительно небольшого числа проводов. Простейший *радиальный интерфейс ИРПР* (интерфейс радиальный параллельный) позволяет передавать данные в параллельном виде на расстояние до 15...20 м с помощью следующих сигналов:

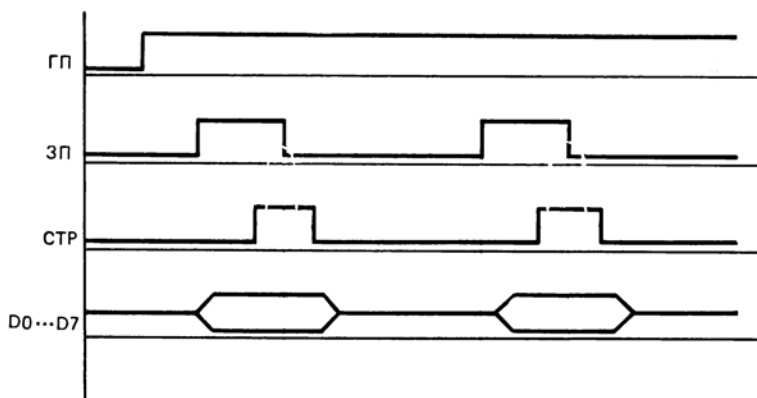
ГП (*готовность приемника*) — формируется приемником информации и означает, что приемник работоспособен и подключен;

ЗП (*запрос приемника*) — означает, что приемник готов к приему очередного байта данных:

СТР (*строб источника*) — поступает от источника и информирует приемник о том, что на линии данных установлен очередной байт;

D0...D7 — данные.

В интерфейсе ИРПР прием данных приемником осуществляется по сигналу СТР. Признаком получения данных служит снятие сигнала ЗП.



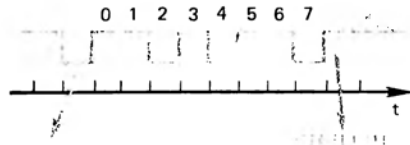


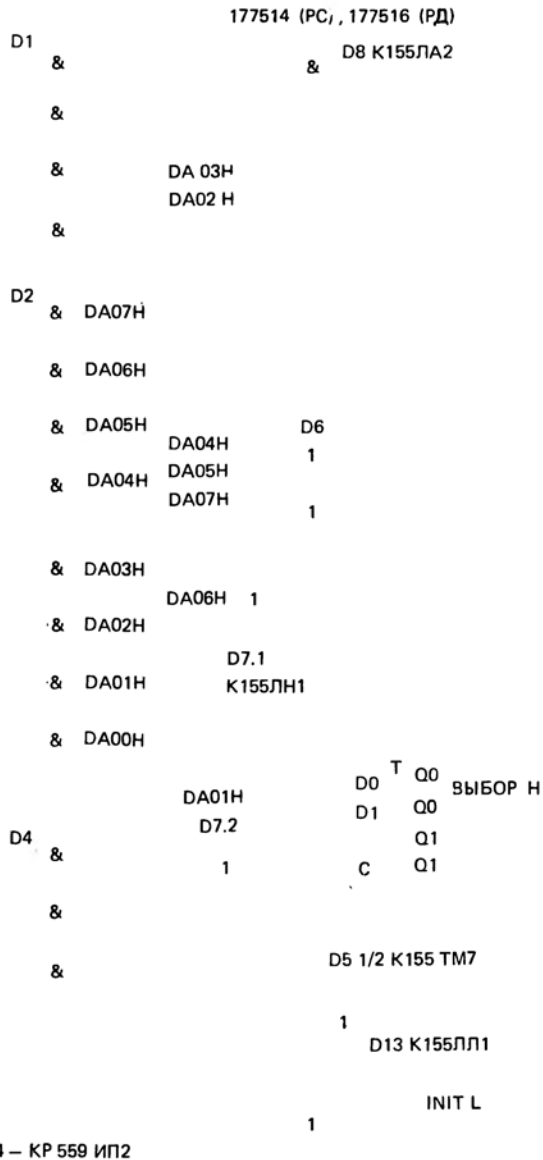
Рис. 7.13

При обмене посредством ИРПС байт данных снабжается стартовым, одним или двумя стоповыми битами и высылается в линию связи последовательно, т. е. каждый бит передается в течение строго фиксированного промежутка времени, называемого битовым интервалом. Единица скорости передачи данных — бод, т. е. бит в секунду. На практике используют следующие стандартные скорости передачи: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19 200 бод.

Протокол обмена посредством ИРПР поясняется рисунком 7.14. Для передачи данных в обоих направлениях количество линий удваивается.

Пример конкретной реализации ИРПР для желающих уяснить устройство интерфейсов представлен на рисунке 7.16. Со всеми элементами этой схемы читатель уже знаком. Исключение — специальные приемники и передатчики для обмена сигналами с МПИ. Приемники обладают низкими входными токами и высокой помехоустойчивостью. Передатчики имеют мощный выходной каскад и, подобно тристабильным элементам, допускают объединение своих выходов для реализации функции «монтажное ИЛИ». Работа интерфейса отвечает описанным выше требованиям протоколов МПИ и ИРПР. Непользуемые разряды РС и все разряды РД в цикле «ввод» считываются в процессор нулями.

Относительно большое число линий является серьезным недостатком интерфейса ИРПР. Более удобен в этом отношении интерфейс ИРПС, или «токовая петля», применяемый для передачи данных между вычислительной системой и удаленными на расстояние до нескольких километров видеотерминалами, печатающими устройствами. В интерфейсе ИРПС передача данных в каждом направлении осуществляется по линии связи, состоящей из двух скрученных проводов (витая пара). Для передачи логической единицы источник данных посылает в линию ток величиной 20 или 60 мА в течение фиксированного промежутка времени, называемого битовым интервалом, а при передаче логического нуля на время битового интервала ток выключается. В зарубежной аппаратуре при передаче данных вместо импульсов тока зачастую используют уровни напряжения: единица представляется положительным напряжением от 3 до 12 В, ноль — отрицательным напряжением



Упрощенная схема байтового ИРПР для устройства печати работает в режиме программного обмена. Интерфейс имеет два программно-доступных регистра — РС с адресом 177514 и РД с адресом 177516. РС доступен процессору только для счи-

		D9	
DA07H	D0	RG	Q3
DA06H	D1		Q1
DA 05H	D2		Q2
DA04H	D3		Q1
		C	
		R	
DA03H	D0	RG	Q0
DA02H	D1		Q1
DA01H	D2		Q2
DA00H	D3		Q3
		C	
		R	D10

D9, D10 — К 155 ТМ8

		D7.3		D	T
&	ВЫВОД РД	1		C	
				R	СТР
&	ВВОД РС	INIT L		S	T
		"ЛОГ. 1"		D	
D12.1	D12.2			C	
&	&			R	ФЛАГ
▷	▷				

D11 К155 ТМ2

D12 КР559 ИП1

ывания, а РД — для записи. Сигналы с суффиксом L передаются активным низким уровнем, т. е. напряжение 0...0,5 В означает наличие сигнала, а 2,4...5 В — отсутствие (логический «0»). Сигналы с префиксом В относятся к МПИ.

той же величины. Перед передачей с помощью сдвиговых регистров производится преобразование данных из параллельной формы в последовательную, а в точке приема — обратное преобразование (рис. 7.15). В процессе преобразования в последовательный код к данным добавляют специальные служебные биты — стартбит (логический «0») и один или более стоп-битов (логическая «1»), которые используются приемником для определения моментов начала и конца передачи байта. В отсутствие передачи данных линия находится в состоянии логической единицы. Все необходимые для последовательной передачи преобразования данных выполняются с помощью асинхронных приемопередатчиков, реализованных обычно в виде БИС.

Более сложные преобразования данных необходимы при передаче цифровой информации по обычной телефонной сети, поскольку в этом случае передача и прием ведутся по одной и той же паре проводов. Кроме того, телефонный канал пропускает сигналы в узком диапазоне частот — от 300 до 3000 Гц. Поэтому для передачи цифровой последовательности применяют специальный адаптер телефонной линии — модем (сокращенно от модулятор — демодулятор), в котором при передаче логические ноль и единица представляются тональными посылками двух различных частот (например, 1080 и 1750 Гц), а принимаемый с линии сигнал пропускается через фильтры и, таким образом, демодулируется. В силу частотных ограничений темп передачи данных по телефонным линиям связи обычно не превышает 600 бит/с.