

## 6. Организация ввода/вывода в микропроцессорной системе

В ЭВМ применяются три режима ввода/вывода:

- программно-управляемый ввод/вывод;
- ввод/вывод в режиме прерываний;
- ввод/вывод в режиме прямого доступа к памяти.

Первый из них характеризуется тем, что инициирование и управление осуществляется программой, выполняемой процессором, а внешние устройства играют сравнительно пассивную роль и только сигнализируют о своем состоянии, в частности, о готовности к операциям ввода/вывода. Во втором режиме ввод/вывод иницируется не процессором, а внешним устройством, генерирующим специальный сигнал прерывания. Реагируя на этот сигнал готовности устройства к передаче данных, процессор передает управление подпрограмме обслуживания устройства, вызвавшего прерывание. Действия, выполняемые этой подпрограммой, определяются пользователем, а непосредственными операциями ввода/вывода управляет процессор. Наконец, в режиме прямого доступа к памяти, который используется, когда пропускной способности процессора недостаточно, процессор приостанавливается, он отключается от системной шины и не участвует в передаче данных между основной памятью и быстродействующим ВУ.

### 6.1. Программно-управляемый ввод/вывод

Данный режим характеризуется тем, что все действия по вводу/выводу реализуются командами прикладной программы через порт (рис. 29). Наиболее простыми эти действия оказываются для «всегда готовых» внешних устройств, например индикатора на светодиодах. При необходимости ввода/вывода в соответствующем месте программы используются команды IN или OUT. Такая передача данных называется синхронным вводом/выводом (рис. 30, а).

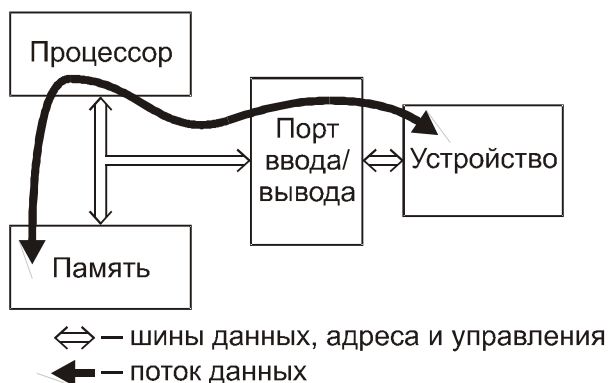


Рис. 29. Аппаратная реализация программно-управляемого ввода/вывода

Однако для большинства ВУ до выполнения операций ввода/вывода необходимо убедиться в их готовности к обмену (рис. 30, б). Такой режим ввода/вывода называют асинхронным. Общее состояние устройства

характеризуется флагом статуса READY, называемым также флагом готовности/занятости (READY/BUSY). Процессор проверяет готовность устройства с помощью команды IN port READY, выставляет флаги командой ANA и осуществляет проверку флагов командой JP. Если флаг READY установлен, то инициируются собственно ввод или вывод данных из порта DATA. Когда же флаг сброшен, процессор выполняет цикл из 2–3 команд с повторной проверкой флага READY до тех пор, пока устройство не будет готово к операциям ВВ. Данный цикл называется циклом ожидания готовности ВУ.

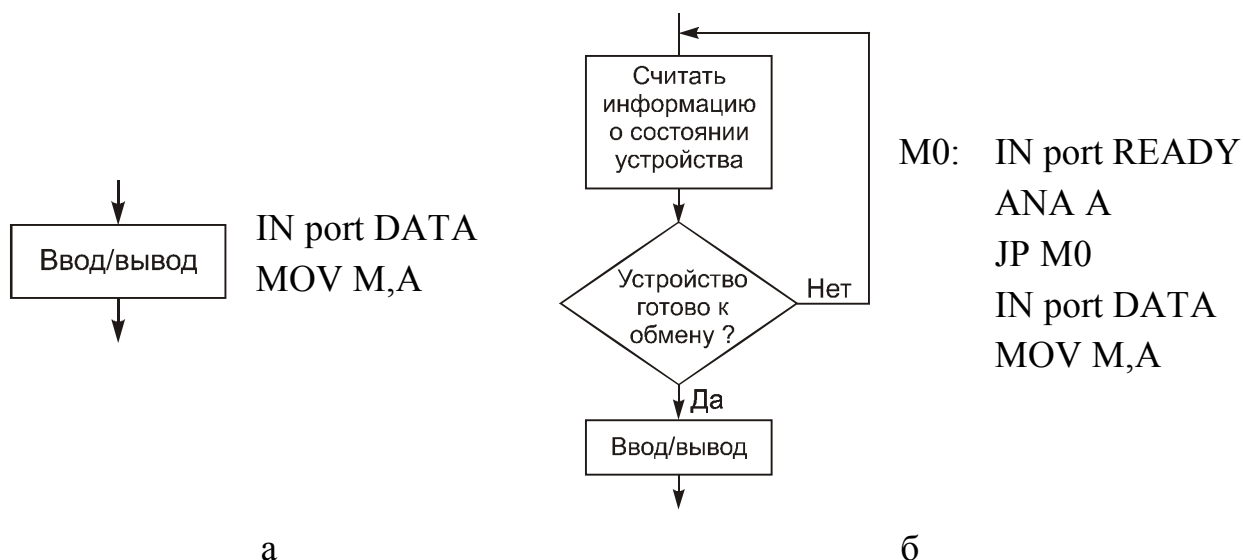


Рис. 30. Выполнение операций ввода/вывода: а – синхронный; б – асинхронный

Основной недостаток программного ввода/вывода связан с непроизводительными потерями времени процессора в циклах ожидания. К достоинствам следует отнести простоту его реализации, не требующей дополнительных аппаратных средств.

## 6.2. Ввод/вывод в режиме прерываний

При организации в ЭВМ системы прерываний непроизводительные потери времени процессора в циклах ожидания готовности резко сокращаются. Режим с прерыванием программы позволяет организовать обмен данными с внешними устройствами в произвольные моменты времени, не зависящие от программы (определяемые внешними устройствами).

Главное отличие ввода/вывода в режиме с прерыванием от программного ввода/вывода состоит в том, что:

- 1) инициатором является внешний сигнал;
- 2) заранее не известно, в какой момент будет прервана фоновая программа.

При этом каждое периферийное устройство может посылать в процессор сигнал INT (INTerrupt) запроса прерывания, когда оно готово к операциям ввода/вывода. По существу, этот сигнал представляет собой выходной сигнал

триггера, фиксирующего флаг готовности READY. Сигнал INT появляется в произвольные моменты времени, асинхронно по отношению к действиям процессора, и управлять его появлением программа не может. Следовательно, заранее не известно, в какой точке программы и какие периферийные устройства инициируют прерывания, поэтому непосредственно в программе команды ввода/вывода использовать нельзя. Остается одно: реагируя на сигнал INT, процессор должен прервать, т.е. временно приостановить текущую программу, идентифицировать прерывающее устройство, перейти к подпрограмме обслуживания прерываний работы этого устройства, а после ее завершения возобновить выполнение прерванной программы. Подпрограмме обслуживания потребуются внутренние регистры процессора: аккумулятор, программный счетчик, некоторые РОН, и их текущее содержимое будет модифицировано. Но прерванная программа должна возобновиться так, как будто прерывания вообще не было (рис. 31).

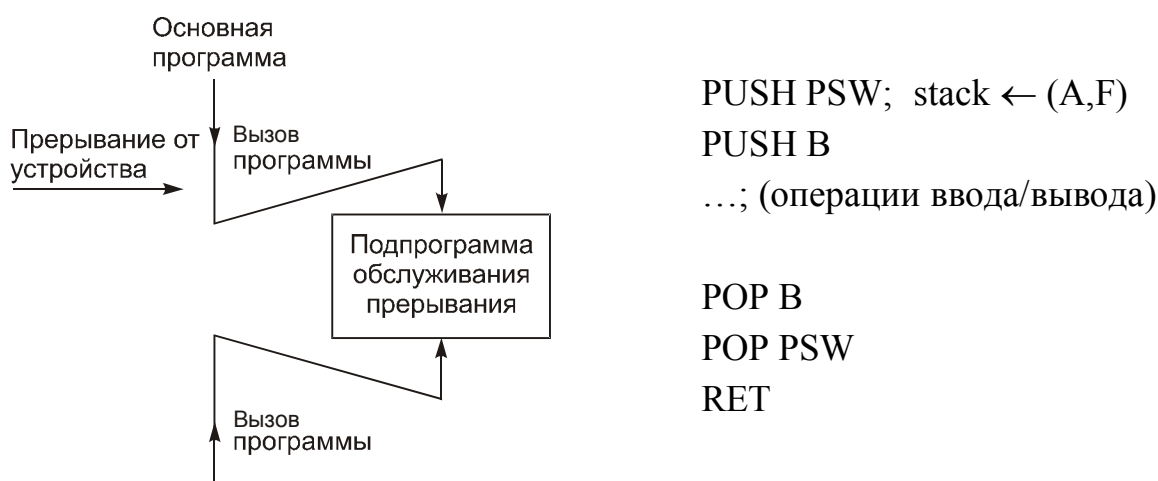


Рис. 31. Реакция на прерывание

Факт обслуживания прерывания влияет на прерванную программу только увеличением времени ее выполнения. Следовательно, содержимое всех регистров, необходимых подпрограмме обслуживания прерывания, следует временно запоминать. В качестве такого временного «хранилища» удобно использовать стек (однако чем больше информации запоминается, тем больше время реакции на прерывание). Предпочтительными с точки зрения повышения производительности микроЭВМ являются уменьшение числа команд, обеспечивающих сохранение информации о прерванной программе, и реализация этих функций аппаратными средствами.

В микропроцессорном комплекте КР580 механизм управления прерываниями реализован в специальной БИС КР580ВН59 (рис. 32), однако общая последовательность реакции на сигнал прерывания примерно одинакова для всех микропроцессоров и содержит следующие шаги:

1. Периферийное устройство генерирует сигнал прерывания, который подается на вход INT процессора; на этой линии по схеме ИЛИ объединяются запросы всех устройств, работающих в режиме прерываний.

2. Процессор завершает текущую команду и, если прерывания разрешены (не замаскированы), формирует сигнал подтверждения прерывания INTA (INT Acknowledgement).

3. Запоминается содержимое счетчика команд PC и некоторых других внутренних регистров в стеке (в КР580 автоматически сохраняется только адрес возврата).

4. Процессор идентифицирует прерывающее устройство для перехода к соответствующей подпрограмме обслуживания.

5. Выполняется короткая (30–50 байт) подпрограмма обслуживания прерывания, в которой запрограммированы действия по передаче данных, модификации указателей, проверке окончания операций ввода/вывода.

6. Восстанавливается состояние прерванной программы, для чего сохраненное содержимое регистров извлекается из стека.

7. Возобновляется выполнение прерванной программы; это действие инициируется командой возврата из прерывания RET, являющейся последней командой подпрограммы обслуживания прерывания.

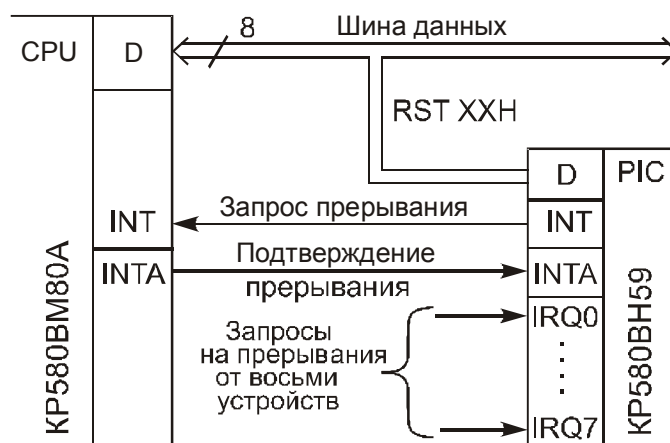


Рис. 32. Организация ввода/вывода в режиме прерываний

Следует отметить, что при выполнении этой последовательности в некоторых процессорах содержимое флагов и аккумулятора сохраняется в стеке автоматически, при этом для возврата используется специальная команда RTI (возврат из прерывания). Также существуют процессоры с несколькими наборами (банками) регистров. При обработке прерываний банки переключаются с основного на дополнительный, что позволяет избежать сохранения содержимого регистров в стеке.

В микроЭВМ обычно используется одноуровневая система прерываний, т.е. сигналы «Запрос прерывания» от всех ВУ поступают на один вход процессора. Поэтому возникает проблема идентификации ВУ, запросившего обслуживание, и реализации заданной очередности (приоритета) обслуживания ВУ при одновременном поступлении нескольких сигналов прерывания.

В контроллере КР580ВН59 реализован следующий механизм идентификации устройств. На этапе выполнения шага 4 приведенной выше последовательности, в

момент подтверждения процессором прерывания сигналом INTA, контроллер выставляет на шину данных команду RST, содержащую в себе номер источника запроса прерывания. Код этой команды имеет следующий вид:

$$RST = 11NNN111_2,$$

где NNN – это двоичный код устройства, запросившего прерывание. В результате дешифрации процессором команды RST формируется стартовый адрес (вектор) подпрограммы-обработчика прерывания:

$$Address = 0000\ 0000\ 00NN\ N000_2.$$

Например, прерывание от устройства №4 сформирует команду RST 4 (код  $11100111_2$ ), которой соответствует стартовый адрес  $0020h$  ( $0000\ 0000\ 00NN\ N000_2$ ). Число NNN также определяет максимальное количество внешних устройств, которое для KP580BH59 равно восьми.

### 6.3. Ввод/вывод в режиме прямого доступа к памяти

Этот метод используется для скоростных внешних запоминающих устройств (ВЗУ), таких, например, как накопители на жестких магнитных дисках. В них обмен производится блоками фиксированного размера от 128 байт и более, причем этот обмен должен осуществляться в строгой последовательности без пропусков, так как пропуск хотя бы одного байта вызовет необходимость в повторном обмене. При этом время, отводимое на обмен одним байтом, строго ограничено скоростью перемещения магнитного носителя относительно магнитных головок и не превышает, как правило, нескольких микросекунд. Обеспечить обмен большими блоками данных с ВЗУ как при помощи программно-управляемого обмена, так и в режиме прерываний процессора невозможно, так как на обмен каждым байтом затрачивается несколько команд, суммарное время выполнения которых превышает допустимое время на обмен одним байтом с ВЗУ.

Поэтому высокоскоростной обмен производится в режиме прямого доступа к памяти (ПДП) без участия процессора. При этом ряд функций реализуется аппаратно, а время обмена одним байтом данных равно одному циклу обращения к памяти. Причем обменом управляет не программа, а специальный контроллер прямого доступа к памяти (микросхема KP580BT57 в комплекте KP580).

Существует две разновидности ПДП. В режиме «прозрачного ПДП» «передача данных выполняется без информирования процессора, для чего используются те интервалы машинных циклов, когда процессор не обращается к памяти, а выполняет внутренние преобразования данных. Такими интервалами в процессоре KP580 являются такты T4 и T5 (части цикла, используемые для внутренних операций). Процессор идентифицирует эти интервалы специальным сигналом, означающим доступность системной шины. Производительность процессора в таком режиме не уменьшается, но сами передачи носят нерегулярный характер, что ведет к уменьшению скорости передачи данных.

Другим способом является «ПДП с приостановкой процессора», при котором выполнение команд задерживается на несколько тактов. Как показано на рис. 33, контроллер непосредственно связан с памятью микроЭВМ через шины данных и адреса. При этом возникает проблема совместного использования шин процессором и контроллером ПДП. Для исключения конкуренции двух устройств за право владения шиной контроллер ПДП снабжен двумя управляющими линиями: «Захват шин» (HOLD) и «Подтверждение захвата» (HLDA).

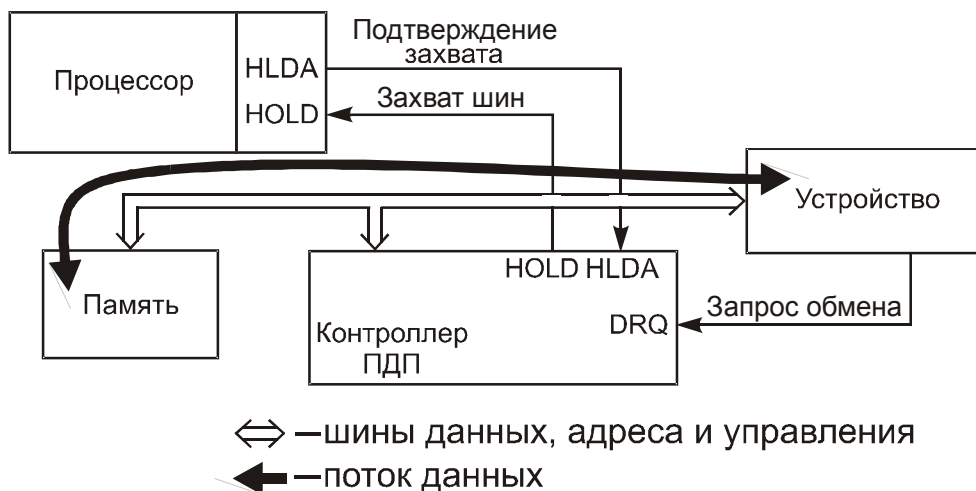


Рис. 33. Организация ввода/вывода в режиме ПДП

Если требуется обмен данными, то периферийное устройство адресует запрос обмена (сигнал DRQ) контроллеру ПДП. После получения от устройства такого запроса контроллер ПДП посылает на вход HOLD микропроцессора сигнал «Захват шин». Процессор, получив этот сигнал, приостанавливает выполнение очередной команды (закончив очередной машинный цикл), отключается от шин системного интерфейса (путем перехода шин в состояние высокого сопротивления) и выдает на системный интерфейс управляющий сигнал «Подтверждение захвата». С этого момента все шины управляются контроллером ПДП, который осуществляет обмен побайтно или блоками данных с памятью микроЭВМ и затем, сняв сигнал захвата, контроллер возвращает управление системным интерфейсом процессору. Как только контроллер ПДП будет готов к обмену следующим блоком данных, он вновь «захватывает» цикл процессора и т.д. В промежутках между сигналами подтверждения процессор может продолжать выполнение команд программы. Тем самым выполнение программы замедляется, но в меньшей степени, чем при обмене в режиме программного ввода/вывода.

Применение в микроЭВМ режима ПДП всегда требует предварительной подготовки, а именно: для каждого ВУ необходимо выделить область памяти, используемую при обмене, и указать ее размер, т.е. число данных, подлежащих пересылке. Следовательно, контроллер ПДП должен обязательно иметь в своем составе регистр адреса и счетчик байт. Перед началом обмена в режиме ПДП необходимо выполнить подготовительные операции по записи в указанные регистры контроллера ПДП начального адреса выделенной внешнему устройству

памяти и ее объема в байтах или словах в зависимости от того, какими порциями информации ведется обмен.

#### 6.4. Параллельная передача данных

Параллельная передача данных между интерфейсом и ВУ является наиболее простым способом обмена. При этом все биты данных передаются одновременно, за один такт. Очевидно, что такой способ обеспечивает высокую скорость обмена, но требует большого числа линий связи (нерационален, например, для подключения удаленных устройств). Однако в связи с сильными взаимными наводками между параллельными линиями, не удастся передавать данные с большой частотой. Поэтому в современных вычислительных системах даже для соединения внутренних узлов все чаще используют последовательный интерфейс (например шина PCI Express).

В настоящее время параллельные интерфейсы стали обязательными компонентами практически всех микропроцессорных систем (рис. 34). Например, в персональном компьютере параллельным является порт LPT. Для организации параллельной передачи данных помимо шины данных, количество линий в которой равно числу одновременно передаваемых бит данных, используется минимальное количество управляющих сигналов.

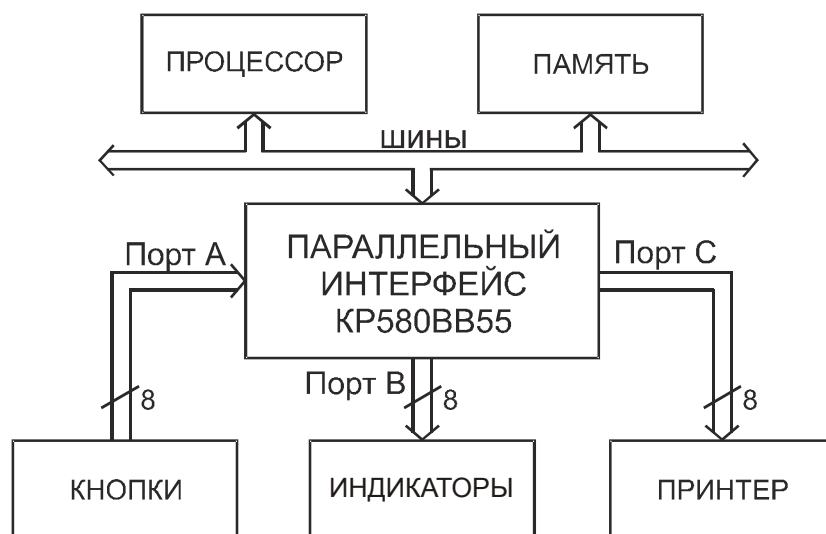


Рис. 34. Параллельный интерфейс в структуре микроЭВМ

Как пример типичного программируемого параллельного интерфейса рассмотрим микросхему КР580ВВ55 (рис. 35). Она предназначена для организации обмена информации между МП и внешними устройствами в параллельном 8-разрядном коде. Подключение периферийного оборудования производится через три двунаправленных параллельных 8-битных порта А, В и С.



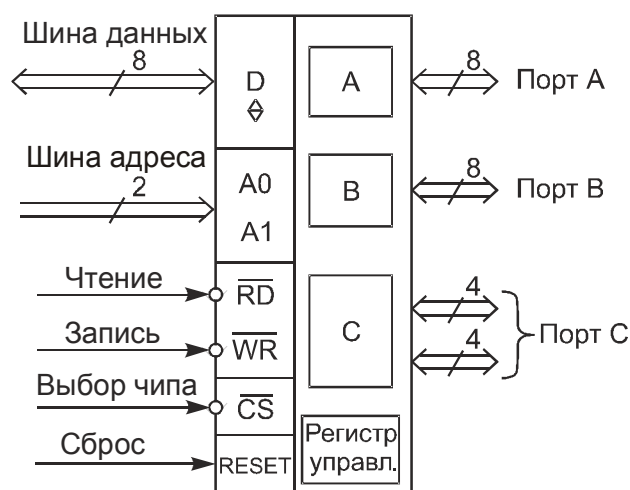


Рис. 35. Программная модель KP580BB55

Интерфейс подключается к системной шине с помощью 14 линий:

$D_{0-7}$  – двунаправленная шина данных с третьим состоянием;

$A_0$ ,  $A_1$  – линии адреса, которые выбирают внутренний регистр интерфейса, подключаемый к шине данных: 00 – порт A, 01 – порт B, 10 – порт C и 11 – регистр управления;

$\overline{RD}$  – чтение информации из адресуемого линиями  $A_{0,1}$  регистра на шину данных (активный уровень – низкий);

$\overline{WR}$  – запись информации с шины данных в адресуемый внутренний регистр интерфейса (активный уровень – низкий);

$\overline{CS}$  – выбор микросхемы (Chip Select) с низким активным уровнем;

$\overline{RESET}$  – сигнал сброса для приведения интерфейса в исходное состояние, при подаче которого регистр управления обнуляется, а все три порта переводятся в режим ввода.

При наличии системного контроллера программирование и обмен данными осуществляется командами ввода IN и вывода OUT. Приемником и источником данных является аккумулятор микропроцессора. При отсутствии системного контроллера порты оказываются в общем адресном пространстве памяти, и ввод/вывод через порты производится командами пересылок между ячейками памяти.

Входы  $A_{0,1}$  интерфейса обычно подключаются к младшим линиям шины адреса, а вход  $\overline{CS}$  используется, если в системе задействовано несколько микросхем портов. Тогда дополнительный дешифратор, подключенный к адресным линиям  $A_{2,3,\dots}$ , определяет активную в текущий момент микросхему интерфейса. Напомним, что KP580BM80A способен адресовать до 256 портов ввода/вывода.

Программирование KP580BB55 заключается в записи в регистр управления управляющего слова, которое определяет один из трех режимов работы интерфейса:



**Режим «0»** базового ввода/вывода осуществляет простой ввод/вывод данных по трем 8-разрядным портам А, В, С и применяется для программно-управляемого обмена данными с медленнодействующими периферийными устройствами. При этом порт С может использоваться как два 4-разрядных канала, а необходимые управляющие сигналы (строб, готовность) могут формироваться программно по отдельным линиям;

**Режим «1»** *стробируемого ввода/вывода* предназначен для однонаправленных передач данных, инициируемых прерываниями. Информация передается через порты А и В, каждый из которых дополнен тремя управляющими сигналами (строб, готовность и прерывание), аппаратно-формируемыми на линиях порта С. Оставшиеся две линии порта С можно использовать как одноканальные порты ввода/вывода;

**Режим «2»** *двунаправленный ввод/вывод*, при котором работает только порт А, а пять линий порта С выполняют функции квитирования и прерываний (две для готовности, две для строба и одна линия прерывания). При этом порт В может работать в режиме 0 или 1.

Сигнал прерывания может возникнуть по двум причинам: при заполнении и при освобождении входного буфера. Процессор различает эти два типа прерываний путем анализа содержимого порта С.

Существуют два формата управляющего слова (рис. 36), которые различаются старшим битом:

1XXX XXXX<sub>2</sub> – задание режима;

0XXX XXXX<sub>2</sub> – управление отдельными линиями порта С.

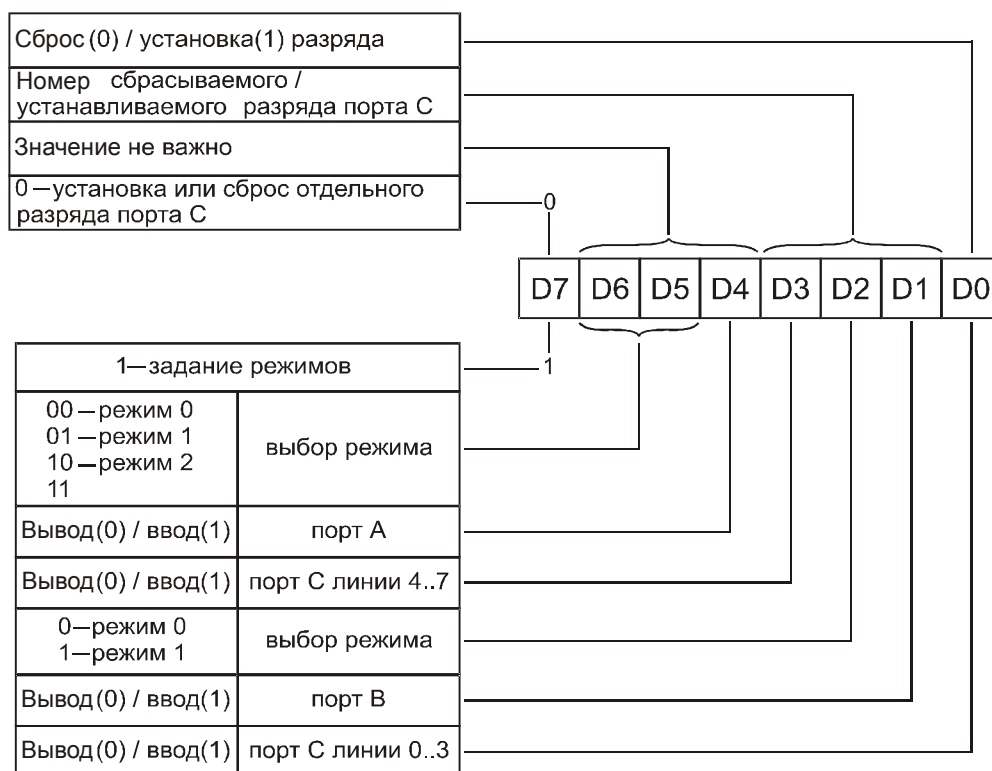


Рис. 36. Формат управляющего слова KP580BB55

#### Примечания:

- обычно программирование интерфейса осуществляется операционной системой или программой монитора, которая запускается по включению питания с адреса 0000H;
- операцию установки и сброса битов порта С можно выполнить и при помощи логических команд, однако этот способ нельзя назвать рациональным;
- управление отдельными линиями порта С требуется для формирования управляющих сигналов (стробирующих импульсов), а также для управления внешними устройствами типа реле (необходимо изменить состояние только одного бита).

### 6.5. Последовательная передача данных

Обмен данными с ВУ по последовательным линиям связи характерен для медленнодействующих или удаленных периферийных устройств. Применение последовательных линий связи обусловлено тем, что линии связи просты по своей организации, а функции автоматического преобразования из последовательного в параллельный формат реализуются аппаратно с помощью программируемого последовательного интерфейса. Для процессора такой интерфейс выглядит устройством параллельного ввода/вывода.

Передача данных по линии может происходить в соответствии со стандартом RS232, который описывает назначение и параметры 25 линий, однако на практике используют только часть из них – обычно 3–5 линий. Величиной скорости передачи является бод, который определяет количество переданных бит в секунду. Для повышения помехозащищенности при передаче данных на значительные расстояния уровни сигналов в линии связи не соответствуют уровням TTL, а имеют амплитуду от  $-12$  до  $+12$  В. Такие сигналы формируют каналные приемопередатчики серии K170.

Типичным представителем контроллера последовательного интерфейса является БИС KP580BB51. Она позволяет осуществить обмен данными как в программно-управляемом режиме, так и в режиме прерывания программы и предназначена для организации асинхронного либо синхронного обмена.

В *асинхронном режиме* используется стартстопный принцип, и передача может начаться в любой момент времени. При этом каждое передаваемое слово передатчик снабжает стартовым и стоповым битами, которыми осуществляется синхронизация запуска работы приемника. Затем в зависимости от применяемого кода передаются от 5 до 8 бит собственно символа. Передача символа завершается необязательным битом четного или нечетного паритета и одним, полутора или двумя стоповыми битами. После этого может быть начат цикл передачи следующего символа. Стандартный формат информационной последовательности при асинхронной связи приведен на рис. 37, а).

В *синхронном режиме* данные передаются целыми массивами слов. Для синхронизации запуска приемника используется не один бит, а одно или два слова синхронизации SYN1 и SYN2, после которых передаются 5–8-битные коды символов с необязательными битами четного или нечетного паритета (рис. 37, б). Приемник перед началом работы находится в режиме активного ожидания, принимая поступающую информацию, и осуществляет ее проверку на совпадение с кодом синхронизации. Совпадение свидетельствует о начале информационного массива данных.

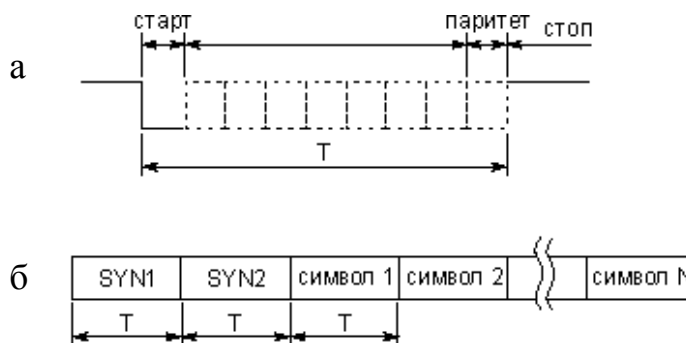


Рис. 37. Формат обмена: а – асинхронного; б – синхронного

На рис. 37, а видно, что асинхронному режиму свойственна значительная избыточность. Если, например, код символа содержит 5 бит, то вместе с ним могут передаваться до 4 служебных бит. Непроизводительное использование линии связи доходит до 44 %, поэтому асинхронный режим более выгоден с точки зрения передачи данных малой размерности, при этом скорость передачи составляет 9,6 Кбод. Для больших массивов целесообразно применение синхронного режима, в котором можно получить скорость 115,2 Кбод.

Рассмотрим основные элементы интерфейса КР580ВВ51 (рис. 38). Он имеет буферы приема/передачи данных и блок управления с регистрами, определяющими скорость обмена, длину символа, число стоповых бит, режим и условия контроля (четный или нечетный паритет). Связь с системной шиной и внешним устройством осуществляется по следующим линиям:

D – двунаправленная 8-разрядная шина данных с третьим состоянием;

C/D – команды/данные – линия идентификации передачи данных или управляющих слов;

RD – чтение информации из адресуемого линиями  $A_{0,1}$  регистра на шину данных (активный уровень – низкий);

WR – запись информации с шины данных в адресуемый внутренний регистр интерфейса (активный уровень – низкий);

CLK – синхронизация – подключается к системному генератору;

CS – выбор микросхемы (Chip Select) с низким активным уровнем;

RESET – сигнал сброса для приведения интерфейса в исходное состояние, при подаче которого регистр управления обнуляется, а все три порта переводятся в режим ввода;

TxD – выход передатчика;  
 RxD – вход приемника;  
 TxC – управление скоростью передачи данных;  
 RxC – управление скоростью приема данных.

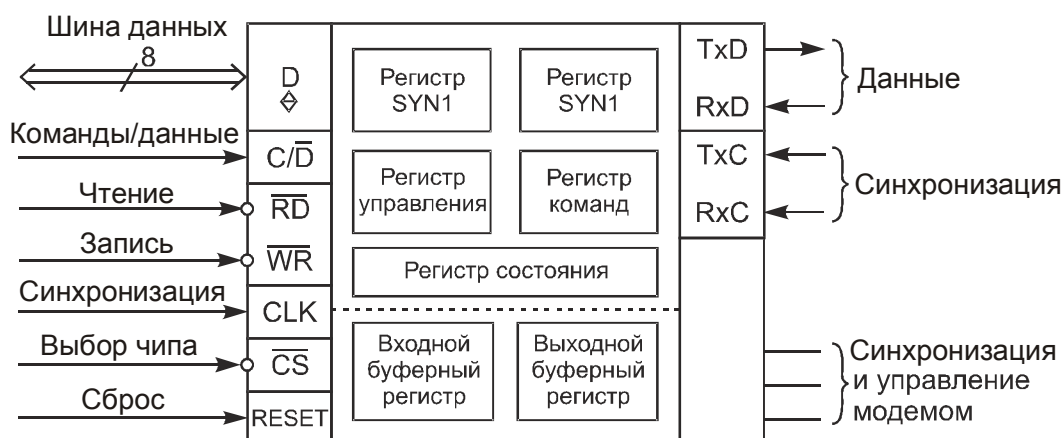


Рис. 38. Программная модель KP580BB51

Перед началом работы интерфейс устанавливается в исходное состояние подачей сигнала RESET. Затем необходимо произвести инициализацию управляющим словом (рис. 39).

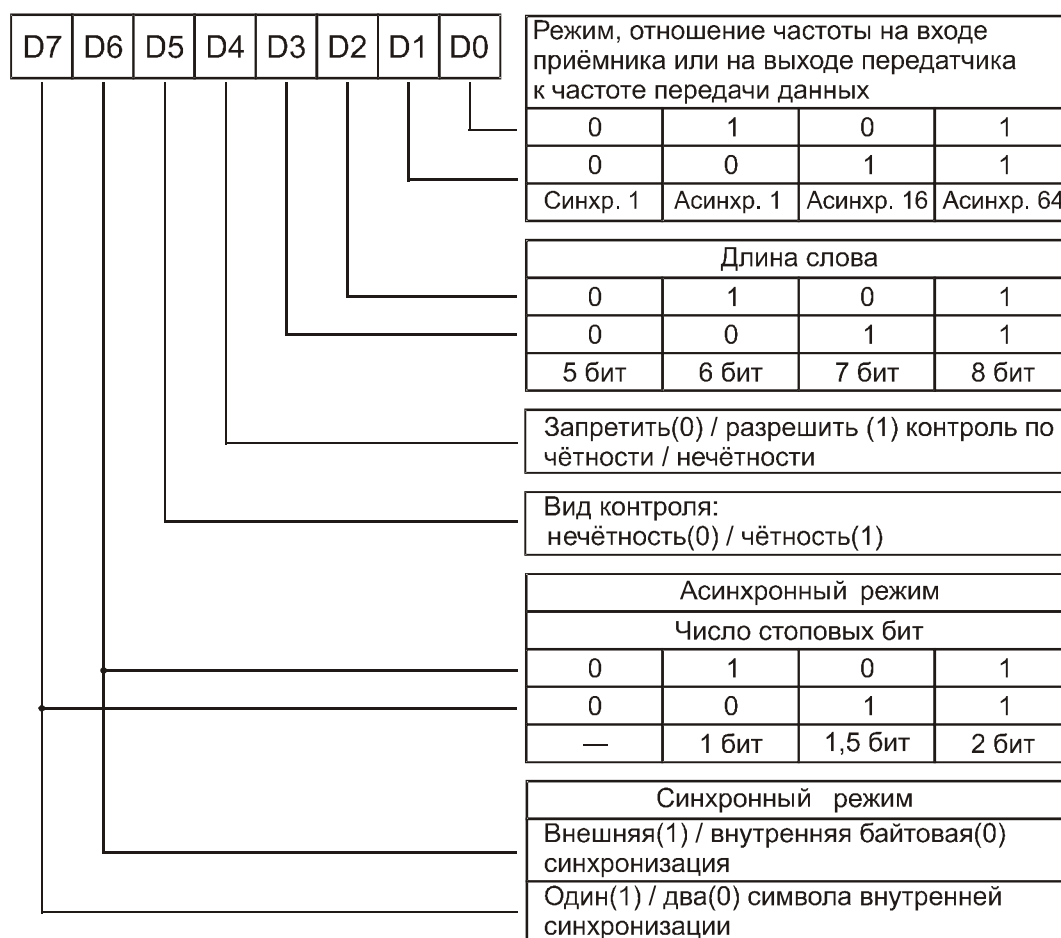


Рис. 39. Формат слова управления KP580BB51

Если задан синхронный обмен, то вслед за управляющим словом подаются коды одного или двух символов синхронизации. В процессе передачи данных на БИС подаются команды для оперативного указания конкретных операций: приема, передачи, перехода в режим активного ожидания (рис. 40).



Рис. 40. Формат слова команды KP580BB51

Для обеспечения программно-управляемого обмена предусмотрена возможность программного считывания слова-состояния (рис. 41), которое содержит информацию о текущем состоянии приемного и передающего буферов, наличии ошибок приема и передачи. Скорость обмена определяется как частотой входа CLK, так и программно и может составлять 1/16 или 1/64 этой частоты.

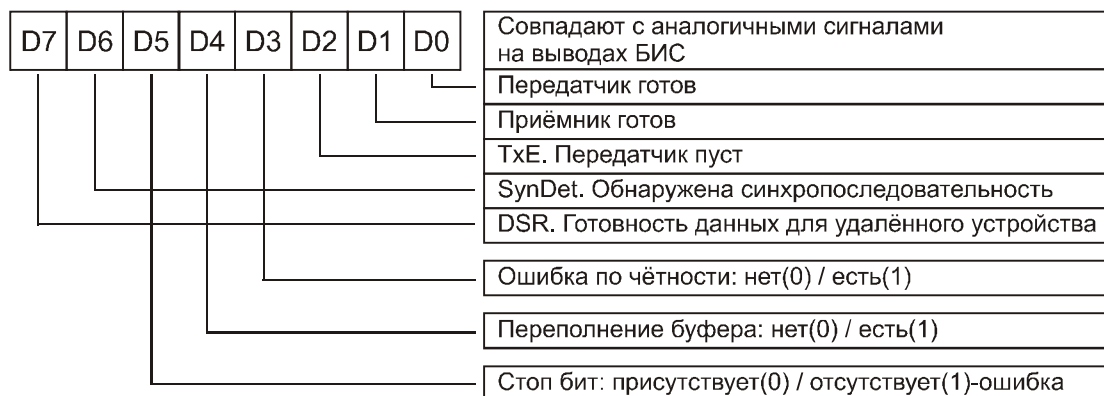


Рис. 41. Формат слова состояния KP580BB51

Следует отметить, что одна из команд выполняет сброс интерфейса KP580BB51, что эквивалентно подаче сигнала RESET.