

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ РАДИОФИЗИКИ И ЭЛЕКТРОНИКИ  
Кафедра кибернетики

# ИЗУЧЕНИЕ ИНТЕРФЕЙСОВ МИКРОПРОЦЕССОРНЫХ СИСТЕМ

МЕТОДИЧЕСКИЕ УКАЗАНИЯ  
К ЛАБОРАТОРНОМУ ПРАКТИКУМУ

Для студентов специальностей  
G 31 04 02 "РАДИОФИЗИКА",  
G 31 04 03 "ФИЗИЧЕСКАЯ ЭЛЕКТРОНИКА"



МИНСК  
2003

№ 5112  
№ 4.23

08.01.2003  
Рез

**БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ  
ФАКУЛЬТЕТ РАДИОФИЗИКИ И ЭЛЕКТРОНИКИ  
Кафедра кибернетики**

# **ИЗУЧЕНИЕ ИНТЕРФЕЙСОВ МИКРОПРОЦЕССОРНЫХ СИСТЕМ**

**Методические указания  
к лабораторному практикуму**

**Для студентов специальностей:  
G 31 04 02 “Радиофизика”,  
G 31 04 03 “Физическая электроника”**

**МИНСК  
2002**

УДК 536.3(075.83)  
ББК 22.34р30-252.43  
И85

**Автор – составитель:**  
**И. А. Шалатонин**

**Рецензент**  
доктор физико-математических наук,  
профессор *В. В. Ананасович*

Утверждено Ученым советом  
факультета радиофизики и электроники  
28 мая 2002 г., протокол № 12

**Изучение** интерфейсов микропроцессорных систем: Метод. указания  
к лабораторному практикуму / Авт.-сост. И. А. Шалатонин. – Мн.: БГУ,  
2002. – с.  
И85

Методические указания предназначены для студентов 3-5-го  
курсов факультета радиофизики и электроники.

**УДК 536.3(075.83)**  
**ББК 22.34р30-252.43**

© БГУ, 2002

## Содержание.

1. ОБЩИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ .....	5
1.1. ОРГАНИЗАЦИЯ ВВОДА-ВЫВОДА В МИКРОПРОЦЕССОРНОЙ СИСТЕМЕ.....	5
1.1.1. Программный ввод-вывод.....	11
1.1.2. Ввод-вывод по прерываниям .....	13
1.1.3. Ввод-вывод в режиме прямого доступа к памяти. ....	13
1.2. СТРУКТУРА И ТИПЫ ИНТЕРФЕЙСОВ КОНТРОЛЬНО- ИЗМЕРИТЕЛЬНОЙ СИСТЕМЫ НА БАЗЕ МИКРОКОНТРОЛЛЕРА И ПЭВМ.....	17
1.2.1. Параллельный и последовательный порты ПЭВМ (интерфейсы Centronics и RS-232).....	18
1.2.1.1. Параллельный интерфейс Centronics .....	18
1.2.2. Интерфейс RS-232C .....	25
1.2.2.1. Программирование последовательного порта компьютера.....	29
1.2.2.2. Последовательный интерфейс и его стандарты.....	34
1.3. ОРГАНИЗАЦИЯ ПОСЛЕДОВАТЕЛЬНОГО ВВОДА-ВЫВОДА В МИКРОКОНТРОЛЛЕРАХ.....	39
1.3.1. Последовательный асинхронный интерфейс. Устройство последовательного обмена SCI(Serial Communication Interface) ....	41
1.3.1.1 Программирование SCI микроконтроллера в МК MC68HC711 .....	42
1.3.2. Последовательный синхронный периферийный интерфейс SPI.....	43
1.3.3. Синхронный последовательный интерфейс I <sup>2</sup> C .....	44
1.3.4. Интерфейс CAN.....	45
1.3.5. Интерфейс MicroLAN .....	47
1.3.6. Универсальная последовательная шина – USB .....	48
2. ЛАБОРАТОРНЫЕ РАБОТЫ .....	50
<i>Лабораторная работа № 1</i> Программирование параллельного интерфейса K580BB55(i8255) и универсального синхронно- асинхронного приемопередатчика K580BB51 (i8251).....	50
<i>Лабораторная работа № 2</i> Изучение интерфейса Centronics .....	52
<i>Лабораторная работа № 3</i> Изучение последовательного интерфейса RS-232.....	53
<i>Лабораторная работа № 4</i> Программирование SCI порта микроконтроллера MC68HC11E9 .....	55
<i>Лабораторная работа № 5</i> Программирование последовательного периферийного интерфейса (SPI) микроконтроллера	

MC68HC711E9 Сопряжение микроконтроллера с часами реального времени MC68HC68TIP .....	58
<i>Лабораторная работа № 6</i> Программирование интерфейса I <sup>2</sup> S микроконтроллера PCF80C552. Сопряжение микроконтроллера с микросхемой EEPROM AT24C64 .....	59
<i>Лабораторная работа № 7</i> Изучение интерфейса MicroLAN. Сопряжение микроконтроллера MC68HC711E9 с датчиком температуры DS1820 по интерфейсу MicroLan .....	60
ПРИЛОЖЕНИЕ 1 .....	63
ПРИЛОЖЕНИЕ 2 .....	64

# 1. ОБЩИЕ СВЕДЕНИЯ ИЗ ТЕОРИИ

## 1.1. Организация ввода-вывода в микропроцессорной системе

Микропроцессор всегда работает совместно с системой памяти (микросхемы ОЗУ, ПЗУ, ППЗУ), устройствами ввода-вывода (УВВ) информации (рис.1). В памяти хранится программа решаемой задачи, исходные, промежуточные и окончательные результаты (данные); УВВ осуществляет ввод и вывод данных.

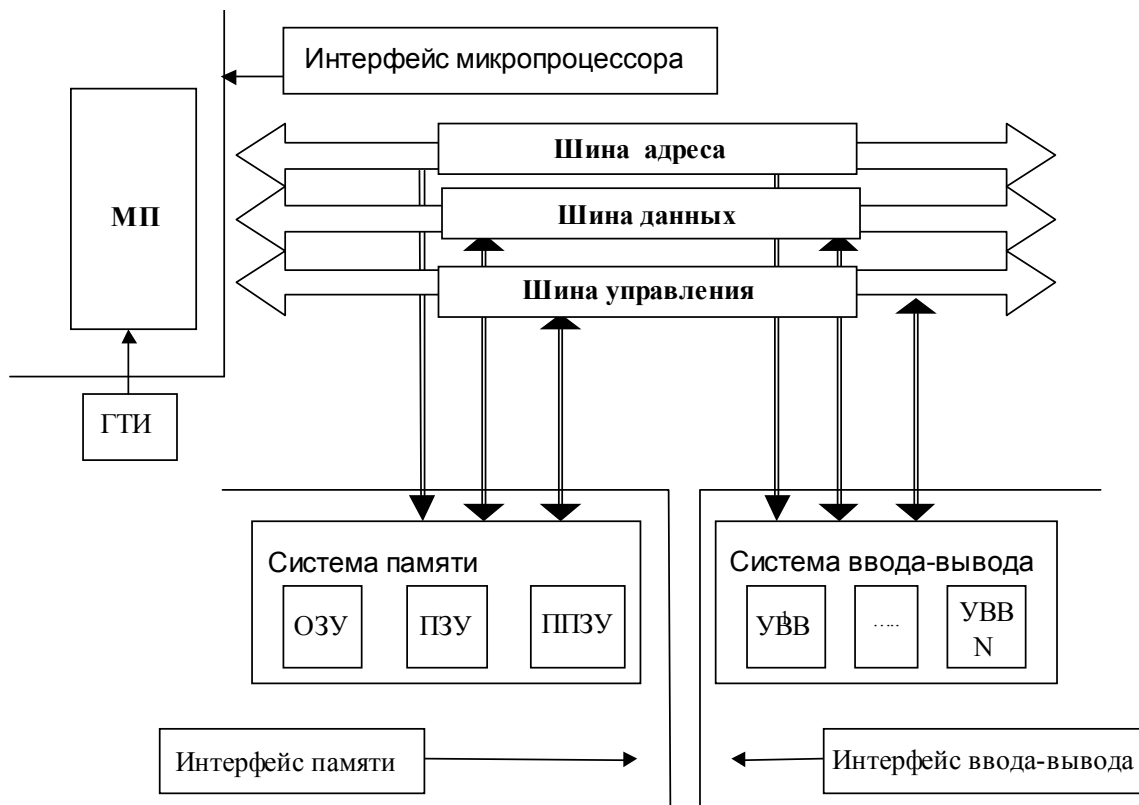


Рис. 1. Обобщенная структура МПС

*МП – микропроцессор,  
ГТИ – генератор тактовых импульсов,  
ПЗУ – постоянное запоминающее устройство,  
ППЗУ – перепрограммируемое запоминающее устройство,  
ОЗУ – оперативное запоминающее устройство  
УВВ – устройство ввода-вывода.*

МП выполняет программу и управляет всеми перечисленными устройствами. Вычислительная система, представленная на рис. 1, называется микропроцессорной системой и реализуется на микропроцессорных БИС. Система построена по модульному принципу и имеет магистральную (шинную) организацию межмодульных связей. В качестве МП используется как микропроцессор с фиксированной системой команд, так и микропрограммируемый МП.

Взаимодействие МП с памятью и УВВ требует выбора способа обращения к устройствам памяти и ввода-вывода, разработки системы адресации и внутреннего интерфейса МП-системы.

По способу организации устройств памяти и УВВ возможны два варианта организации системы:

- разделение адресов;
- выделение подсистемы.

При разделении адресов все элементы системы адресуются как ячейки памяти в том смысле, что для идентификации УВВ и устройств памяти не генерируется никаких специальных сигналов. Адрес, подаваемый на шину, дешифрируется памятью и системой ввода-вывода с целью определения его назначения, и в таких системах можно не использовать специальных команд ввода-вывода.

Все поле адресов распределяется в этом случае между памятью и УВВ. Разделение адресов между памятью и УВВ можно достигнуть выделением одной (или нескольких) адресной линии для указания используемого устройства (память или ввод-вывод). Однако этот способ уменьшает в два (или более) раза емкость адресуемой памяти, поэтому может использоваться лишь в системах с небольшой емкостью памяти.

Использование одной (или более) адресной линии для идентификации устройств можно представить как перевод одной (или более) адресной линии в шину управления.

Достоинство данного способа адресации устройств памяти и УВВ состоит в том, что он позволяет исключить регистр состояния системы. В небольших МП-системах в дополнение к простоте адресации имеются преимущества в использовании команд обращения к памяти вместо команд ввода-вывода. По командам ввода в МП-системах осуществляется просто загрузка требуемых данных в аккумулятор микропроцессора. Команды с обращением к памяти имеют более гибкие возможности по обработке данных. Например, если в команде ADD M содержится адрес входного устройства, то

будет произведено сложение данных входного устройства с содержимым аккумулятора. А команда MOVE В,М произведет загрузку данных из входного устройства в регистр В, если исполнительный адрес команды является адресом входного устройства.

При выделении подсистем вся МП-система разделяется на ряд подсистем (память, ввод, вывод, стек). В рамках каждой подсистемы используется своя совокупность адресов.

В архитектуре магистрального типа важное значение приобретает интерфейс. На рис. 1 показаны интерфейсы МП, системы памяти и системы ввода-вывода (ВВ). В узком смысле интерфейсом (от англ. interfase – сопрягать, согласовать) называют устройство сопряжения; в широком смысле под интерфейсом понимают совокупность аппаратных, программных и конструктивных средств, обеспечивающих взаимодействие функциональных модулей системы. Таким образом, для представленной на рис. 1 микропроцессорной системы необходимым условием высокой эффективности использования является совместимость интерфейсов МП, системы памяти и системы ВВ. Работа рассматриваемой системы синхронизируется генератором тактовых импульсов (ГТИ).

Система работает с командами пересылки информации МП - память, МП-УВВ, память-УВВ.

Для МП-системы разработано три основных режима ввода-вывода:

- программно управляемый,
- по прерываниям,
- режим прямого доступа к памяти (ПДП).

Программно-управляемый ввод-вывод означает, что все действия по обмену информацией инициируются МП, работающим под управлением прикладной программы, а всем другим элементам системы отводится пассивная роль сигнализации о своей готовности (асинхронный способ обмена). Непроизводительные потери времени МП на ожидание сигнала готовности от УВВ могут оказаться неприемлемыми для систем, работающих в реальном времени.

Ввод-вывод по прерываниям подразумевает, что действия по обмену информацией инициируют сами периферийные устройства, генерируя сигнал прерывания. При восприятии сигнала прерывания микропроцессор приостанавливает выполнение текущей программы, временно запоминает ее состояние (как минимум, запоминается содержание PSW, PC), идентифицирует прерывающее устройство и осуществляет обмен информацией. После завершения обмена



восстанавливается состояние прерванной программы и возобновляется ее выполнение (рис. 2). В данном режиме на ввод-вывод расходуется гораздо меньше времени микропроцессора, чем в программно-управляемом ВВ, поэтому система может выполнить больше полезной работы или обслужить больше периферийных устройств.

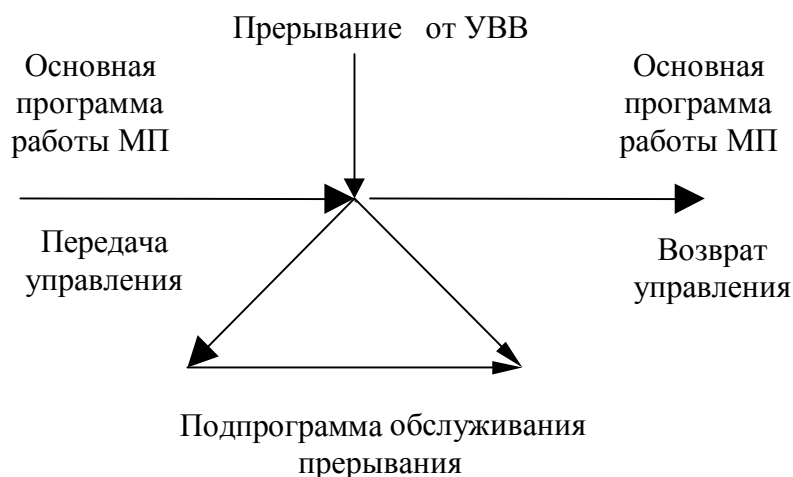


Рис. 2. Обслуживание прерывания в МП-системе

Некоторые периферийные устройства, чаще всего устройства внешней памяти, имеют такую высокую степень передачи, что пропускной способности МП даже в режиме ввода-вывода по прерываниям недостаточно. Связь с такими устройствами удобно реализовать в режиме прямого доступа к памяти (ПДП или DMA). В этом режиме на время передачи действия МП приостанавливаются, и он отключается от шины адреса и данных. Инициирование и управление обменом информацией между периферийными устройствами и системой памяти осуществляется контроллером прямого доступа к памяти (КПДП). Контроллер ПДП позволяет быстродействующим периферийным устройствам обращаться к ОЗУ непосредственно, не прерывая вычислительного процесса МП. Межмодульный обмен осуществляется с помощью магистрали в режиме ПДП. Контроллер на каждом цикле обмена занимает магистраль. За время цикла передается одно слово. МП в режиме ПДП не изменяет своего состояния. Во время цикла обмена МП приостанавливает работу (находится в режиме ожидания), это несколько снижает его производительность.

Условием возможности реализации режима ПДП является способность МП отключаться от своих внешних шин, т. е. переводить буферные регистры данных и адреса по выходу в третье высокое импедансное состояние.

Типовой микропроцессор имеет возможность работать в каждом из трех вышеописанных режимов обмена. Структура интерфейса МП-системы для каждого из режимов приведена на рис. 3.

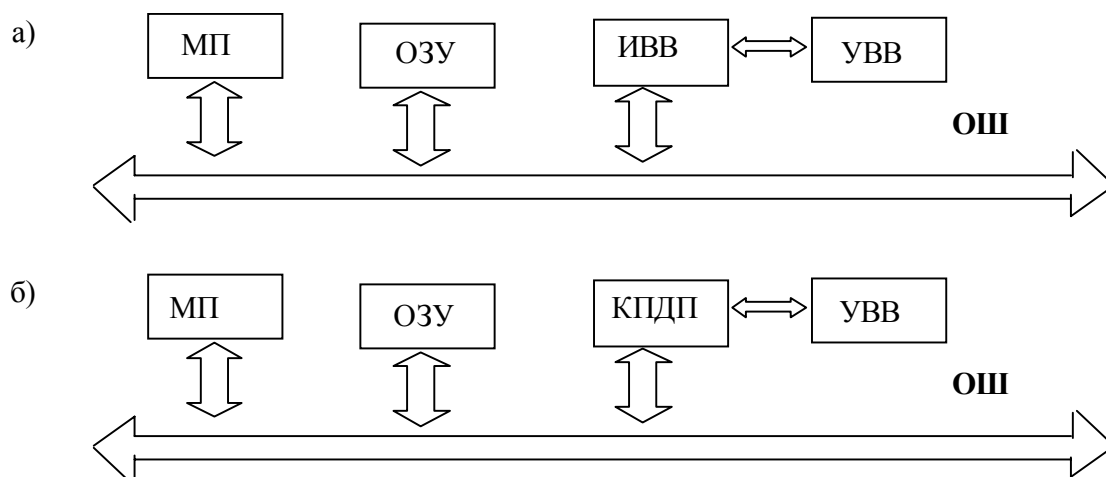


Рис. 3. Структура интерфейса МП-системы:

- а – программно управляемый ВВ, ввод-вывод по прерываниям (ИВВ – интерфейс ввода-вывода, ОШ – общая шина);  
 б – режим ПДП (КПДП – контроллер прямого доступа к памяти)

Функции контроллера интерфейса часто выполняет сам МП.

Наиболее важными факторами, характеризующими интерфейс, являются:

- простота связи,
- скорость,
- адаптивность обмена.

Опыт свидетельствует, что, создавая интерфейс, трудно одновременно удовлетворить всем этим требованиям. Так, программно управляемый интерфейс наиболее прост в аппаратном смысле и обеспечивает высокую адаптивность обмена, однако при этом неизбежны небольшая скорость и потери машинного времени. Интерфейсы по каналам прямого доступа и с прерываниями значительно повышают скорость обмена, однако возрастает сложность интерфейса как в аппаратном, так и в программном отношении, и, как следствие, снижается возможность оперативной модификации обмена.

Передачу информации от внешнего устройства (ВУ) по направлению к процессору называют вводом, а от процессора к внешнему устройству – выводом.

В настоящее время наибольшее распространение получили микроЭВМ с магистральным интерфейсом (рис. 1). Линии связи между процессором, памятью и внешними устройствами разделяют на группы (шины): адреса, данных и управления. Электронная схема, входящая в состав ВУ и обеспечивающая подключение к интерфейсу, называется контроллером. Нередко конструктивно интерфейс оформлен в виде отдельной платы, которая вставляется в электронный блок микроЭВМ и кабелем соединяется с внешним устройством.

Через контроллер процессор обменивается информацией с ВУ, а также управляет его работой (перемоткой ленты, пуском, остановкой) и опрашивает его состояние (готовность, наличие ошибок). Обычно в контроллере для этих целей имеются два отдельных регистра: регистр данных (РД) и регистр состояния (РС).

Каждый разряд регистра состояния имеет свое назначение, определяемое форматом регистра состояния. Например, отдельный бит может использоваться для проверки готовности ВУ: если в нем при чтении РС процессор обнаружил “1”, значит, ВУ готово к вводу-выводу, если же в этом разряде “0” – значит, ВУ не готово (например, выполняет предыдущую операцию ввода-вывода).

Так как в системе может быть несколько различных контроллеров, то каждому из них присваивается свой адрес. Если в контроллере имеются РД и РС, то каждому из этих регистров присваиваются отдельные адреса. В начале каждой операции ввода-вывода процессор формирует адрес требуемого ВУ, контроллеры анализируют этот адрес, сравнивая его с собственным адресом. Контроллер, принявший свой адрес, участвует в операции ввода-вывода, остальные контроллеры не мешают ее выполнению.

При организации ввода-вывода в микроЭВМ необходимо решить ряд важных задач, без чего невозможен правильный обмен данными:

- 1) обеспечить связь только с тем внешним устройством, с которым намечен обмен данными; все другие ВУ не должны мешать чужому вводу-выводу;

- 2) согласовать скорость обмена, так как скорости различных ВУ сильно различаются между собой и отличны от скорости работы процессора и памяти. Выделяют группу медленных ВУ, быстродействие которых связано с работой механических частей ВУ (печатающее устройство, перфоратор и др.), скоростью работы

человека (например, ввод с клавиатуры) и группу быстрых ВУ, скорость обмена данными с которыми может приближаться или достичь быстродействия процессора и памяти (например, магнитные диски, аналого-цифровые и цифро-аналоговые преобразователи);

3) решить проблему синхронизации обмена, т. е. правильного определения моментов начала и окончания передачи данных между устройствами, участвующими в обмене, – передатчиком и приемником данных (без этого нет гарантии правильной передачи информации);

4) согласовать между собой электрические параметры сигналов, которыми обмениваются устройства;

5) обеспечить максимальную скорость обмена минимальными аппаратными и программными средствами.

Противоречивость пятой задачи привела к созданию трех вышеописанных способов обмена данными:

- программный ввод-вывод;
- ввод-вывод по прерываниям;
- ввод-вывод в режиме прямого доступа к памяти.

#### 1.1.1. Программный ввод-вывод

Программный ввод-вывод – это наиболее простой способ обмена данными между процессором и внешним устройством. Всеми действиями по организации обмена управляет процессор.

Известны два типа программно-управляемой передачи данных: синхронная и асинхронная.

Синхронная передача данных характерна для периферийных устройств, для которых известны временные соотношения. При этом типе передачи устройство ввода-вывода должно быть готово к приему или передаче данных за время, равное времени выполнения определенной команды процессора. Синхронная передача реализуется при минимальных затратах технических и программных средств.

Асинхронная передача данных, иногда называемая обменом посредством “рукопожатия”, широко используется в микроЭВМ. При такой передаче данных ЭВМ перед выполнением операции ввода-вывода проверяет состояние периферийного устройства. Обычно при обмене необходимо:

- 1) проверить состояние устройства;
- 2) активизировать устройство, если последнее готово к обмену;
- 3) передать данные (вести или вывести);
- 4) дезактивировать устройство.

На первом шаге выполняется команда ввода информации о состоянии выбранного устройства. Далее выполняется условный переход по содержимому разряда состояния устройства. Если устройство занято, в программе реализуется переход для повторной проверки его состояния. В случае готовности устройства выдается команда на ввод или вывод данных. При завершении передачи устройство деактивируется.

Граф-схема алгоритма асинхронного программного обмена (фрагмента некоторой программы) приведена на рис. 4.

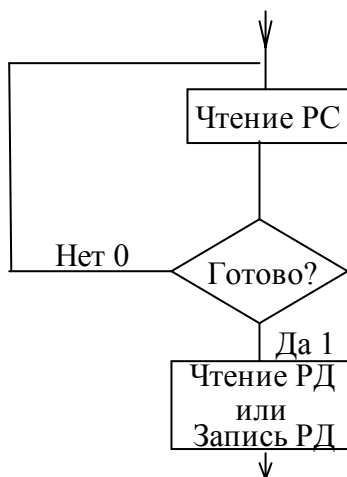


Рис. 4. Граф-схема алгоритма асинхронного программного обмена

При обмене информацией с медленными ВУ (например, печать со скоростью 10 символов в сек.) процессор вынужден большую часть времени ждать готовности ВУ (в данном случае по 0,1 сек. на каждый выводимый символ). Например, чтобы напечатать слово “ВНИМАНИЕ!”, процессор должен заниматься этим в течение 1 сек., хотя собственно подготовка и передача кода одного символа в РД печати занимает обычно несколько мкс, например 10 мкс. Тогда остальные 99 990 мкс, или 99,99 % времени, процессор опрашивает готовность устройства печати. Этот пример показывает, что программный ввод-вывод приводит к простоям процессора, особенно при обмене с медленными ВУ.

Асинхронная передача является идеальной в смысле согласования временных различий между периферийными устройствами и процессором. Недостаток ее в том, что процессор вынужден ожидать, пока периферийное устройство будет готово к обмену. Это приводит не только к непроизводительным затратам времени микроЭВМ (при наличии длительных задержек), но и во многих случаях является

просто недопустимым. Например, в процессах управления в этом случае возникает необходимость сохранения уровня сигналов управления на время ожидания передачи. Методом, позволяющим устранить подобные трудности, является передача данных с прерыванием программы.

#### 1.1.2. Ввод-вывод по прерываниям

Для повышения производительности системы необходимо освободить процессор от опроса готовности ВУ к обмену. Эта функция возлагается на контроллер ВУ. Получив команду ввода-вывода, контроллер передает ее ВУ и следит за временем ее выполнения. По окончании действий в ВУ (печати, перфорации и т. п.) контроллер посылает в процессор сигнал требования прерывания, получает очередную команду ввода-вывода, и действия повторяются. Процессор в этом случае, передав в контроллер очередную команду ввода-вывода, может выполнять другие операции основной программы до получения сигнала требования прерывания. Получив его, он обслуживает это прерывание, т. е. формирует и выдает в контроллер очередную команду ввода-вывода, а затем возвращается к выполнению прерванной основной программы.

В программе вывода текста в этом случае отсутствует ожидание готовности ВУ. Как только устройство печати готово принять очередной символ, контроллер вырабатывает требование прерывания, и на 10 мкс прерывается основная программа, а затем продолжается ее выполнение. Простои процессора сокращаются, так как, пока ВУ не готово, процессор выполняет другие команды основной программы, т. е. процессор и ВУ в это время работают параллельно.

Ввод-вывод по прерываниям, однако, требует усложнения аппаратных средств – создания системы прерываний.

#### 1.1.3. Ввод-вывод в режиме прямого доступа к памяти.

Два рассмотренных выше способа обмена – программный и по прерываниям – имеют малую скорость обмена данными. Для передачи одного слова данных процессор должен выполнить несколько команд, среди них вспомогательные – изменение адреса памяти, изменение и анализ содержимого счетчика слов. В некоторых микропроцессорах невозможно в одной команде передать данные из ВУ в память. Сначала необходимо их принять в процессор, а затем из процессора передать в память. Однако во многих случаях требуется передавать большие массивы информации между памятью и внешним

устройством (например, накопители на гибких магнитных дисках). В этом случае процессор выступает в роли “лишнего звена”, транзитом пропуская через себя информацию. Скорость передачи информации при этом ограничена и не превышает нескольких десятков Кбайт в секунду.

В то же время память микроЭВМ обычно позволяет выполнять чтение-запись данных со скоростью несколько сот Кбайт или даже несколько Мбайт в секунду. Нередко и ВУ позволяют вводить или выводить данные с такими скоростями. Очевидно, что процессор, участвуя в таком обмене, становится “узким местом”, снижает возможные скорости обмена, поэтому в режиме прямого доступа к памяти (ПДП) процессор отключается и не участвует в операциях ввода-вывода. Весь обмен информацией выполняется под управлением контроллера ПДП, который “замещает” процессор на это время, т. е. сам формирует адрес памяти, изменяет его и счетчик слов после каждого чтения-записи, принимает решение об окончании обмена и осуществляет синхронизацию.

Контроллер ПДП – сложное устройство. В случае необходимости обмена контроллер сообщает об этом процессору сигналом “Запрос на захват шины” или “Требование прямого доступа”. Освободив шины, процессор отвечает сигналом “Подтверждение захвата” (“Подтверждение прямого доступа”), и с этого момента контроллер ПДП получает шины интерфейса в свое распоряжение. Режим ПДП позволяет достичь максимально возможной скорости обмена, определяемой физическим быстродействием памяти, ВУ, схем интерфейса. Заметим, что программы обмена в этом случае нет – обмен выполняется аппаратно. Данные передаются из памяти в ВУ и из ВУ в память напрямую, как показано на рис. 5.

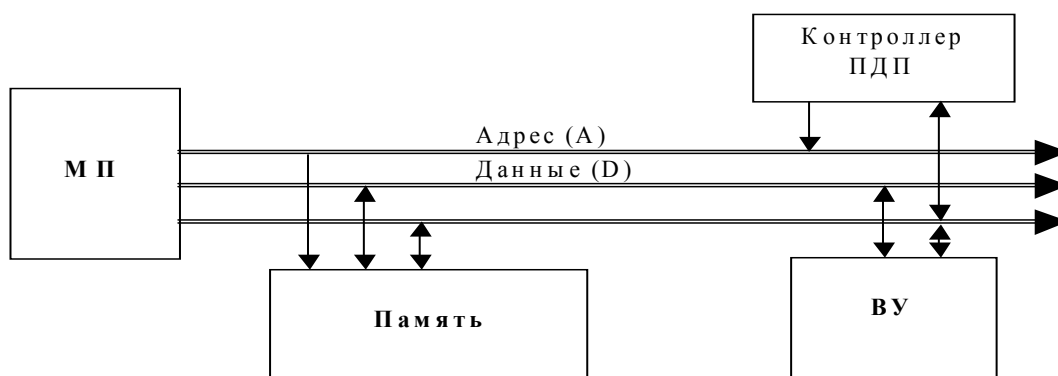


Рис. 5. Передача данных через ПДП

С использованием разнообразных БИС (кристаллов ввода-вывода) упрощается сопряжение, уменьшается количество компонентов, уменьшается стоимость и увеличивается эффективность системы. Самым простым из таких кристаллов является программируемый параллельный интерфейс (ППИ БИС КР580ВВ55, Intel8255). Он представляет собой 40-выводную ИС, содержащую 3 шины ввода-вывода (рис. 6).

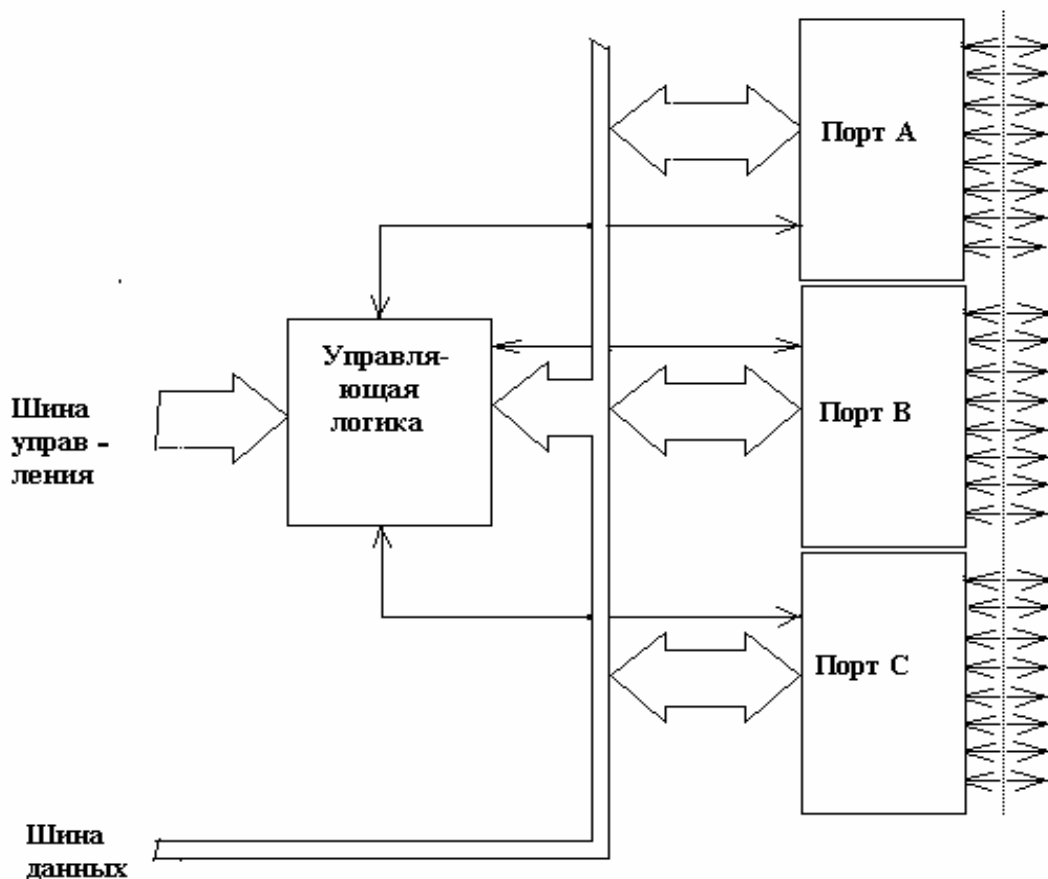


Рис. 6. Программируемый параллельный интерфейс (ППИ БИС КР580ВВ55, Intel8255)

Каждая из шин используется как входная или выходная шина. Направление каждой шины управляется регистром на кристалле. Программа инициализации, находящаяся в ПЗУ системы, устанавливает управляющий регистр в нужную комбинацию входных и выходных шин. Такие БИС являются очень гибкими, так как каждая шина может изменяться программой. Кроме того, они обычно включают управляющую логику для синхронизации работы.

Другой общий тип интерфейсных кристаллов, обеспечивающих последовательные входы и выходы (рис. 7), – универсальные асинхронные передатчики и приемники. Они принимают байт данных



с МП, а затем выдают его по одному биту, т. е. действуют подобно регистру параллельно-последовательного ввода-вывода. Кроме того, могут автоматически вводиться биты старта, остановка и другие синхронизирующие и управляющие сигналы.

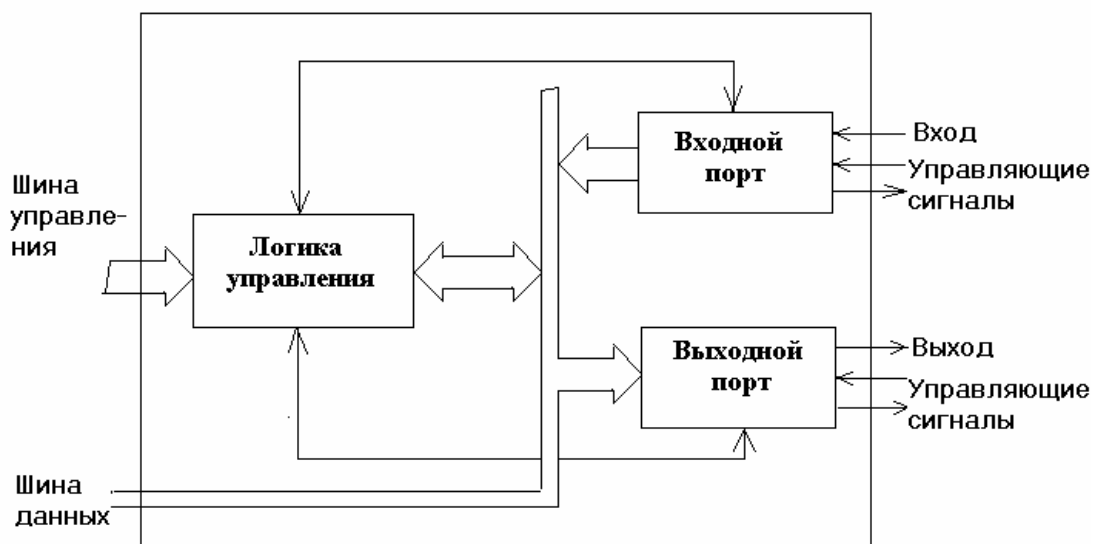


Рис. 7. Интерфейс, обеспечивающий последовательную связь между двумя системами

Формат управляется управляющим регистром, подобным рассмотренному для ППИ. Рассматриваемые интерфейсы также могут обрабатывать данные в другом направлении, преобразовывая поток последовательных битов в параллельную форму, подходящую для непосредственного использования МП. Обычно для связи между микропроцессорной системой и периферийным устройством, таким как ЭЛТ или телетайп, используется последовательный ввод-вывод. Благодаря тому, что информация имеет последовательный формат, для соединения устройств необходимо только 2 провода.

Имеются также разнообразные специализированные интерфейсные кристаллы, включающие контроллеры гибких дисков, контроллеры ЭЛТ, контроллеры ЗУ с прямым доступом, контроллеры клавиатур и дисплея.

## 1.2. Структура и типы интерфейсов контрольно-измерительной системы на базе микроконтроллера и ПЭВМ

Структура контрольно-измерительной системы в общем случае приведена на рис. 8. Она состоит из измерительного датчика, микроконтроллера, осуществляющего управление всей системой, персональной ЭВМ, осуществляющей обработку полученной информации.

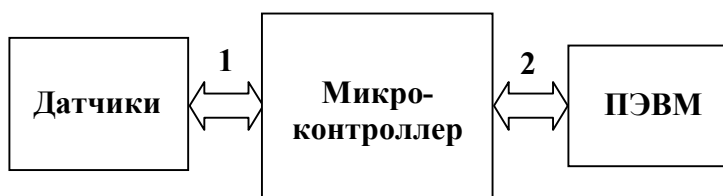


Рис. 8. Структура контрольно-измерительной системы

В частном случае, когда микроконтроллер в состоянии сам осуществить обработку получаемой от датчика информации и отобразить результаты, структурная схема несколько видоизменяется (рис. 9).



Рис. 9. Структура автономной контрольно-измерительной системы

Система индикации может представлять собой жидкокристаллический индикатор либо любой другой, предназначенный для использования в микропроцессорной системе.

Цифрой 1 обозначен интерфейс, посредством которого происходит обмен информацией между датчиками и микроконтроллером. Требования к этому интерфейсу следующие: скорость передачи информации должна соответствовать скорости работы датчиков.

Цифрой 2 обозначен интерфейс, позволяющий связать микроконтроллер с персональным компьютером. Требования к этому

интерфейсу: скорость передачи информации, достаточная для связи компьютера с выбранным микроконтроллером, а также простота реализации процесса передачи информации.

#### 1.2.1. Параллельный и последовательный порты ПЭВМ (интерфейсы Centronics и RS-232)

Параллельный, последовательный – это наиболее распространенные порты ввода/вывода.

Изначально каждый из этих портов разрабатывался для определенного применения. Параллельные предназначались для соединения компьютеров с принтерами, последовательные – для подключения принтеров, модемов и мыши. Однако они могут использоваться и для других приложений, связанных с сопряжением компьютера с внешними устройствами. Периферийные устройства, созданные для этих портов, легко подключаются к IBM PC-совместимому компьютеру. Принципиальные схемы отличаются мобильностью и могут применяться для решения проблем сопряжения с любым оборудованием, которое оснащено указанными портами. Таким образом, полезно узнать, как они работают и как обеспечивается наиболее эффективное их использование.

##### 1.2.1.1. Параллельный интерфейс Centronics

Порт “Centronic”, или параллельный, – это промышленный стандарт для подсоединения принтеров к компьютеру. Компьютер имеет по крайней мере один такой порт, встроенный в материнскую плату или представляющий собой отдельную интерфейсную карту ввода/вывода. Увеличить количество параллельных портов просто и недорого, можно установить параллельных порта с логическими именами от LPT1 до LPT3. Разъемы порта для компьютера и принтера отличаются друг от друга. Первый – это 25-контактная розетка D-типа, а второй – 36-контактная розетка параллельного типа. Для соединения компьютера с принтером используется принтерный кабель длиной не более 5 м.

Персональный компьютер работает максимум с тремя параллельными портами, которые в MS-DOS имеют логические имена: LPT1, LPT2, LPT3. В адресном пространстве компьютера резервируются базовые адреса этих портов: 3BCh, 378h, 278h. Первый адрес обычно используется, если принтерный порт находится на плате графического адаптера Hercules или EGA. На плате ( ) и материнской плате (motherboard) адрес LPT1 – 378h, а LPT2 – 278h. Для

принтерного порта LPT1 предусмотрено аппаратное прерывание IRQ7, а для LPT2 – IRQ5, хотя на практике они используются очень редко.

Примечание. Установка базовых адресов портов и возможность использования прерываний настраивается перестановкой перемычек (jumpers) на плате Multi I/O Card, описание которых обычно приведено в технической документации на плату.

Начиная с базового адреса, каждый адаптер принтера имеет в адресном пространстве три адреса. При этом первый адрес соответствует регистру данных, посылаемых от компьютера к принтеру. В случае использования TTL-микросхем этот регистр бывает реализован на микросхеме 74LS374. Чтение установленных битов данных можно осуществить по тому же адресу. Физически чтение данных происходит через буфер данных, выполненный, например, на микросхеме 74LS244.

Следующий адрес (базовый адрес плюс единица) позволяет читать регистр статуса адаптера (расположенный, конечно, в принтере) через буферную микросхему (часто 74LS240 или 367–368).

В регистре статуса биты с 3 по 7 позволяют определить состояние некоторых сигналов Centronics:

bit 3 = 0:Error,  
bit 4 = 1:Select,  
bit 5 = 1:Paper Out,  
bit 6 = 0:Acknowledge,  
bit 7 = 0:Busy.

Чтение регистра статуса имеет смысл при передаче данных на принтер для определения состояния принтера и процесса передачи данных.

Адрес третьего порта (базовый адрес +2) соответствует регистру управления интерфейса. Этот регистр (read only – только для чтения), для которого может использоваться микросхема 74LS174, позволяет определить следующие состояния принтера:

bit 0 = 0: сигнал Data Strobe активен,  
bit 1 = 0: сигнал Auto Line Feed включен,  
bit 2 = 0: INIT инициализация принтера,  
bit 3 = 1: Select Input принтер выбран,  
bit 4 = 1: IRQEN прерывание разрешено.

Таким образом, порт состоит из двух регистров на запись и трех на чтение (регистр данных, статуса и управления). Некоторые из

линий регистров аппаратно инвертируются, так что логический сигнал на разъеме противоположен программно-установленному. На рис.1 места инверсии обозначены кружками. Таким образом, на разъем порта выведены 8 выходных линий данных, 5 входных регистра статуса и 4 выходных линии регистра управления (рис 2, 3). Сигнал IRQEN, не выведенный на разъем, разрешает выработку прерываний при поступлении сигнала /ACK.

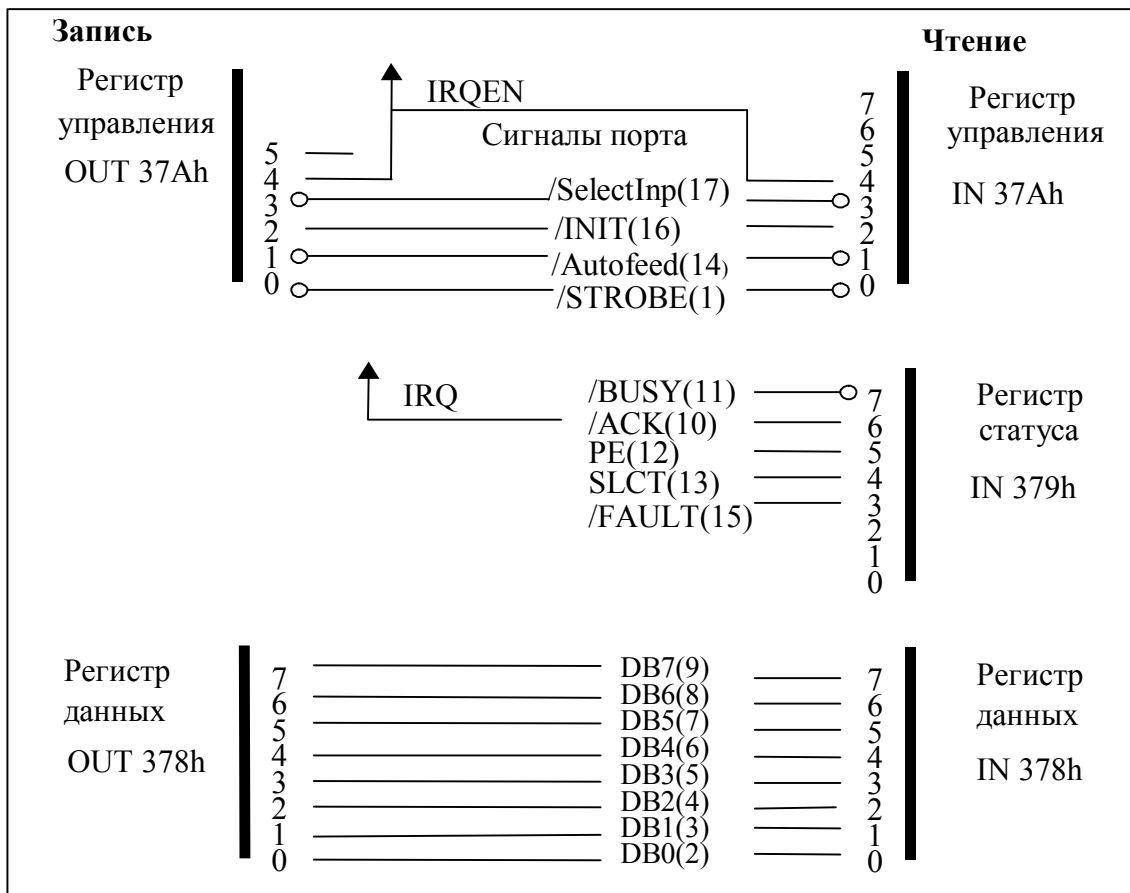


Рис.10. Сигнал интерфейса Centronics

Общая схема параллельного порта внутри ПК представлена на рис. 11.

Восьмибитовые данные заносятся в DD1 во время записи в регистр с адресом “базовый адрес” + 0. Операция осуществляется командой WRITE\_DATA. Эти данные образуют группу. Они считываются компьютером из того же регистра, через DD2 с помощью команды READ\_DATA. Во время чтения выход DD1 должен иметь высокий уровень сопротивления, что достигается подачей на контакт 1 (выход разрешен) DD1 высокого уровня напряжения. Шестибитовое

управляющее слово записывается в DD3 через регистр с адресом “базовый адрес” + 2 при помощи команды WRITE\_CONTROL. Биты с 0 по 3 подаются на выход разъема и образуют группу управления. Некоторые биты инвертируются микросхемами с открытыми коллекторами на выходе (DD6 и DD7). Все выходные линии подключены к питанию +5В через резисторы 4,7 кОм. Состояние этих линий считывается через регистр с адресом, “базовый адрес” + 2 через DD4

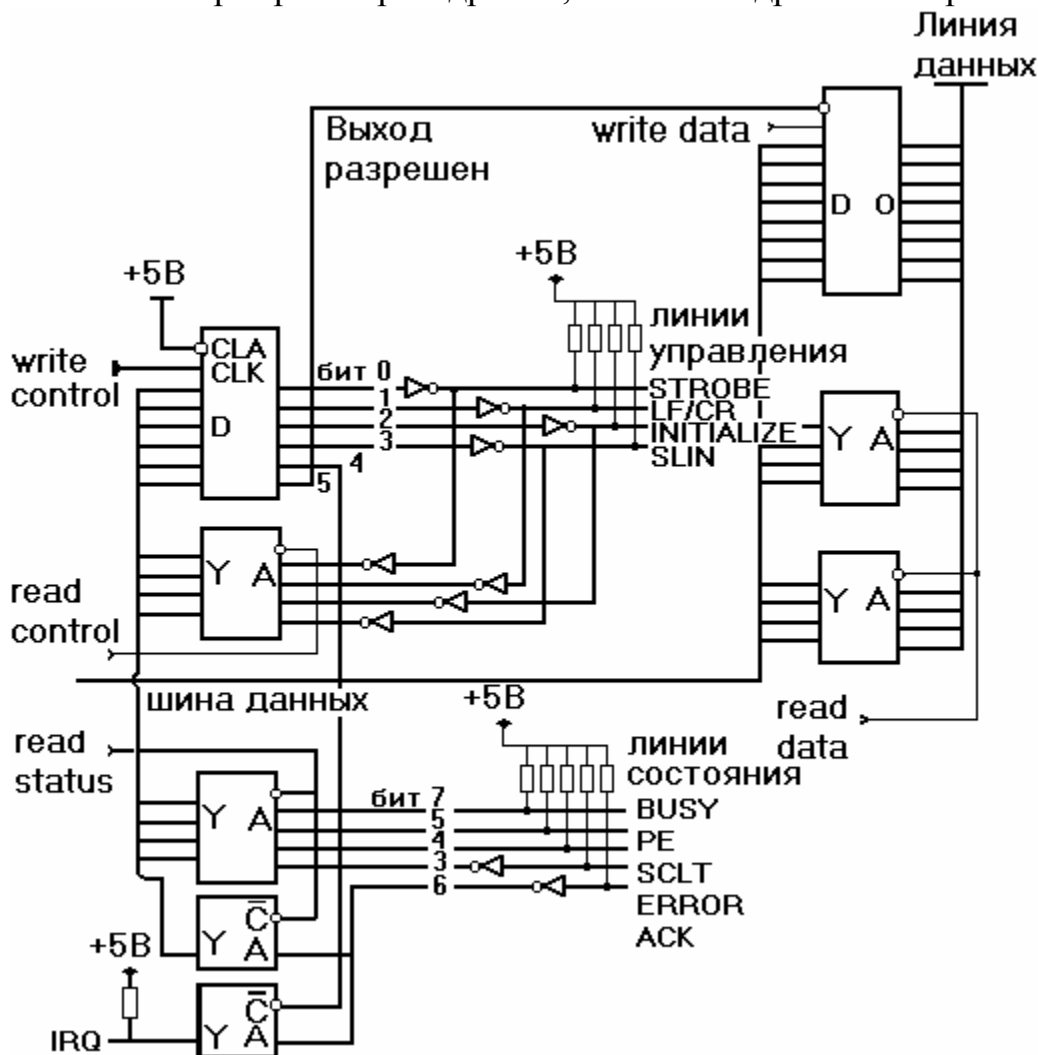


Рис. 11. Общая схема параллельного порта

посредством команды READ\_CONTROL. Четвертый бит управляющего байта разрешает прерывание, а пятый бит открывает или закрывает выход DD1. Состояние пяти контактов разъема порта (группа состояния) компьютер считывает через DD4 с помощью команды READ\_STATUS через регистр с адресом “базовый адрес” + 1.

Входы линии подключены к питанию +5В через резисторы 4,7 кОм, два входа инвертируются.

В первых конструкциях IBM PC контакт “выход” разрешен. DD1 соединялся с “землей” для постоянного открывания выходов. Это была однонаправленная версия параллельного порта. Начиная с IBM PS/2, указанный контакт соединили с пятым битом регистра управления DD3, и порт стал двунаправленным. Следует отметить, что параллельные порты, поставляемые со встроенными картами ввода/вывода, двунаправленные. Для любого контакта следует избегать короткого замыкания и/или соединения с шиной питания. Скорость передачи данных через параллельный порт превышает 1 Мб/с.

На рис. 12 представлена логическая структура параллельного порта.

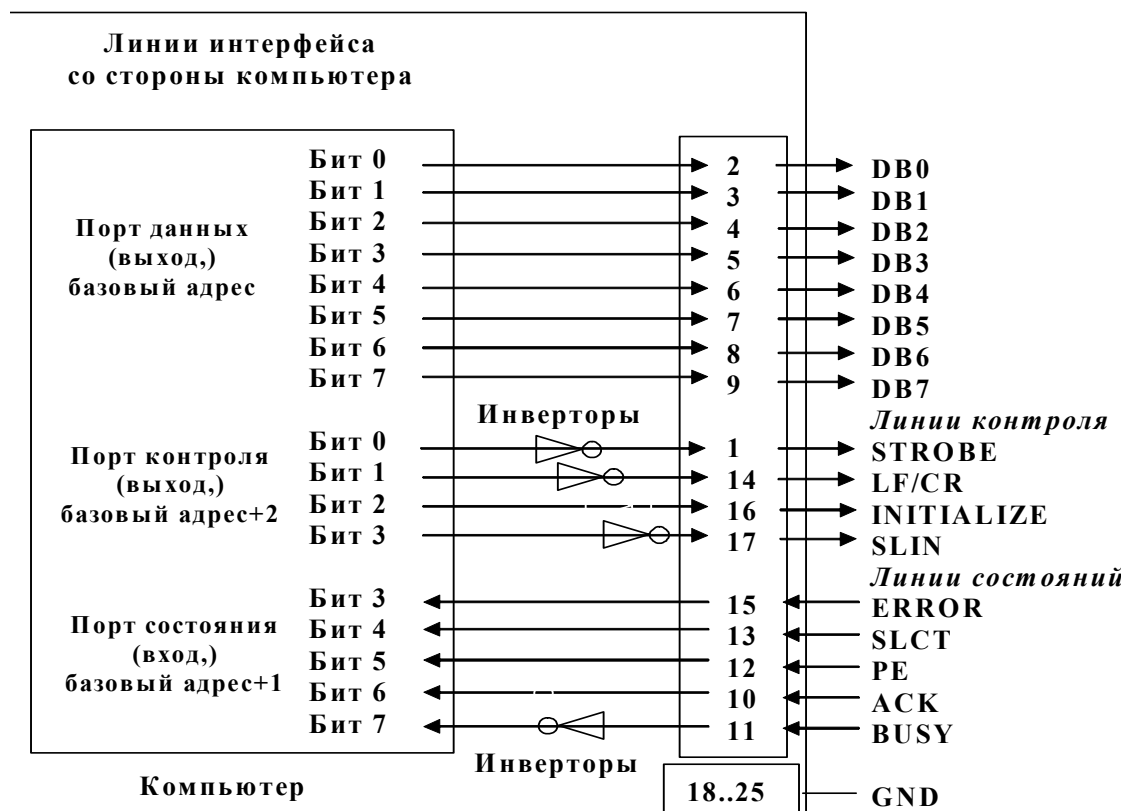


Рис. 12. Логическая структура параллельного порта

Мы в дальнейшем в процессе выполнения лабораторных работ будем детально рассматривать однонаправленный параллельный порт. В нем на вывод могут быть запрограммированы двенадцать линий (четыре регистра управления и восемь регистра данных). Если порт

не двунаправленный, вводить данные можно через линии регистра статуса. Рассмотрим кратко расширенные режимы работы параллельного порта (табл. 1).

Таблица 1.

Расширенные режимы работы параллельного порта

Совместимый протокол		Ниббловый протокол		Байтовый протокол		Протокол EPP		Протокол ECP	
Сигнал	Направление	Сигнал	Направление	Сигнал	Направление	Сигнал	Направление	Сигнал	Направление
/STROBE	0	-	-	HostClk	0	/Write	0	HostClk	0
BUSY	1	PtrBusy	1	PtrBusy	1	/Wait	1	Periph-Ack	1
/ACK	1	PtrClk	1	PtrClk	1	Intr	1	Periph-Clk	1
SELECT	1	Xflag	1	Xflag	1	Xflag	1	Xflag	1
PERROR	1	AckDataReq	1	AckDataReq	1	AckDataReq	1	/Ack-Reverse	1
/FAULT	1	/Data-Avail	1	/Data-Avail	1	/Data-Avail	1	/PeriphRequest	1
/INIT	0	-	-	-	-	/Init	0	/ReverseRequest	0
/AUTOFD	0	Host-Busy	0	Host-Busy	0	/DStrb	0	Host-Ack	0
	0	-	-	Data 1...8	1/0	Data 1...8	1/0	Data 1...8	1/0
/SELECT-IN	0	-	-	-	-	/AStrb	0	ECP-Mode	0

**Совместимый протокол** моделирует простейший принтерный порт. Компьютер защелкивает данные на выходах PD0...PD7 сигналом /STROBE. Принтер выдает сигнал /ASK в подтверждение приема данных и готовности к приему новых данных. Принтер выдает сигнал BUSY, если он не готов к приему данных, а также сигналы PERROR (Paper Error) и /FAULT при возникновении ошибок. Компьютер выдает сигнал /INIT для аппаратного сброса периферийного устройства. Для выбора внешнего устройства компьютер выдает сигнал /SELECTIN. Принтер выдает сигнал SELECT для указания, что он включен. Сигнал AUTOFD (Auto Feed Extension) выдается компьютером для автоматического перевода строки в принтере на каждый символ возврата каретки.

**Ниббловый протокол** служит для ввода в компьютер битов 0...3 и 4...7 по очереди. Внешнее устройство сопровождает каждую



порцию данных сигналом PtrClk. Этот сигнал вызывает прерывание в компьютере, если это разрешено. В ответ компьютер сбрасывает сигнал HostBusy, который затем устанавливается в подтверждение приема данных. Биты 0 и 4 передаются по линии DataAvail, 1 и 5 – по Xflag, 2 и 6 – по AckDataReq, 3 и 7 – по PtrBusy.

**Байтовый протокол** описывает двунаправленную передачу по линиям данных. Внешнее устройство выдает сигнал DataAvail, указывающий на наличие данных к передаче, и сигнал PtrClk вместе с каждым байтом данных. Сигнал HostClk сбрасывается компьютером в подтверждение приема данных. Заметьте, что периферийное устройство не должно интерпретировать этот сигнал как строб данных, передаваемых от компьютера. Xflag выдается внешним устройством и указывает, что оно включено (on-line). Сигналы HostBusy и PtrBusy указывают на занятость компьютера и принтера соответственно при приеме данных. Сигнал AckDataReq сбрасывается внешним устройством в подтверждение HostBusy.

**Протокол EPP** (Enhanced Parallel Port) описывает двунаправленную передачу через порт адресов и данных. Сигнал /Write выдается компьютером и устанавливает направление передачи. Сигнал /Wait выдается внешним устройством, когда оно не готово. Когда этот сигнал сброшен в “0”, порт снимает сигнал IOCHRDY на шине ISA, чтобы удлинить цикл чтения-записи. По окончании передачи адреса или данных сигнал /Wait устанавливается в “1”. Передача данных стробируется сигналом /DStrb, а адреса – /Astrb. Сигнал Intr выдается внешним устройством для выработки прерывания в компьютере. Назначение остальных сигналов аналогично совместимому протоколу.

**Протокол ECP** (Extended Capabilities Port) обеспечивает двунаправленную передачу адресов, данных или RLE (Run Length Encoded) информации. Направление передачи задается компьютером с помощью сигнала /ReverseRequest (если неактивен – от компьютера к внешнему устройству, т. е. в прямом направлении). Внешнее устройство выдает /PeriphRequest для запроса на передачу в обратном направлении. В ответ на /ReverseRequest оно выдает сигнал /AckReverse. При передаче в прямом направлении сигнал HostClk стробирует данные. Этот сигнал снимается при обнаружении сигнала PeriphAck от внешнего устройства и не может быть выдан вновь, пока PeriphAck не будет снят. При передаче в обратном направлении сигнал HostClk не используется, а сигнал PeriphAck обычно сброшен в “0” (“1” указывает на передачу RLE данных). Передача в обратном

направлении стробируется сигналом PeriphClk и подтверждается сигналом HostAck аналогичным образом. При передаче в прямом направлении сигнал HostAck указывает на передачу адреса или RLE (“1”) либо данных (“0”). При работе в режиме ECP сигнал ECPMode установлен в “1”.

Отметим, что конкретная плата портов в вашем компьютере может поддерживать далеко не все из указанных режимов. Тем не менее зачастую их удастся реализовать программным способом хотя бы частично.

### 1.2.2.Интерфейс RS-232C

Широко используемый последовательный интерфейс синхронной и асинхронной передачи данных определяется стандартом EIA RS-232-C и рекомендациями V.24 CCITT. Он изначально создавался для связи компьютера с терминалом и в настоящее время используется в самых различных применениях.

Параметры интерфейса приведены в табл. 2.

*Таблица 2*

Стандарт	EIA RS-232-C, CCITT V.24
Скорость передачи	115 Кбит/с (максимум)
Расстояние передачи	15 м (максимум)
Характер сигнала	Несимметричный по напряжению
Количество драйверов	1
Количество приемников	1
Схема соединения	полный дуплекс, от точки к точке

Интерфейс RS-232-C соединяет два устройства. Линия передачи первого устройства соединяется с линией приема второго, и наоборот (полный дуплекс). Для управления соединенными устройствами используется программное подтверждение (введение в поток передаваемых данных соответствующих управляющих символов). Возможна организация аппаратного подтверждения путем организации дополнительных RS-232 линий для обеспечения функций определения статуса и управления.

Описание выводов интерфейса Centronics приведено в табл. 3.

Таблица 3

Описание выводов интерфейса Centronics

Наименование	Направление	Описание	Контакт (25-контактный разъем)	Контакт (9-контактный разъем)
DCD	IN	Carrie Detect (Определение несущей)	8	1
RxD	IN	Receive Data (Принимаемые данные)	3	2
TxD	OUT	Transmit Data (Передаваемые данные)	2	3
DTR	OUT	Data Terminal Ready (Готовность терминала)	20	4
GND	-	System Ground (Корпус системы)	7	5
DSR	IN	Data Set Ready (Готовность данных)	6	6
RTS	OUT	Request to Send (Запрос на отправку)	4	7
CTS	IN	Clear to Send (Готовность приема)	5	8
RI	IN	Ring Indicator (Индикатор)	22	9

Сигналы интерфейса RS-232C подразделяются на следующие классы:

*Последовательные данные:* – например, TXD, RXD. Интерфейс RS-232C обеспечивает два независимых последовательных канала данных: первичный (главный) и вторичный (вспомогательный). Оба канала могут работать в дуплексном режиме.

*Управляющие сигналы квитирования:* – например, RTS, CTS. Сигналы квитирования – это средство, с помощью которого обмен сигналами позволяет DTE начать диалог с DCE до фактической передачи или приема данных по последовательной линии связи.

*Сигналы синхронизации:* – например, TC, RC. В синхронном режиме (в отличие от более распространенного асинхронного) между устройствами необходимо передавать сигналы синхронизации, которые упрощают контроль целостности сигнала в целях его декодирования.

Интерфейс RS-232C предназначен для подключения к компьютеру внешних устройств (принтера, сканера, модема, мыши и др.), а также для связи компьютеров между собой. Основными преимуществами использования RS-232C по сравнению с Centronics являются возможность передачи на значительно большие расстояния и гораздо более простой соединительный кабель. В то же время работать с ним несколько сложнее. Данные в RS-232C передаются в

последовательном коде побайтно. Каждый байт обрамляется стартовым и стоповыми битами. Данные могут передаваться как в одну, так и в другую сторону (дуплексный режим).

Компьютер имеет 25-контактный (DB25P) или 9-контактный (DB9P) разъем для подключения RS-232C. Назначение контактов приведено в табл. 3.

Наиболее часто используется трех- или четырехпроводная связь (для двунаправленной передачи). Схема соединения для четырехпроводной линии связи показана на рис. 13.

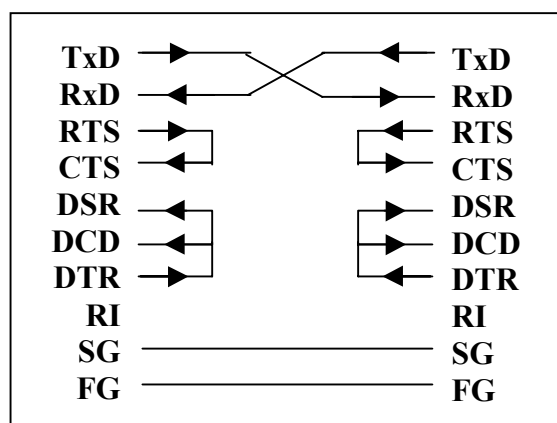


Рис. 13. Схема 4-проводной линии связи для RS-232C

Для двухпроводной линии связи в случае только передачи из компьютера во внешнее устройство используются сигналы SG и TxD. Все 10 сигналов интерфейса задействуются только при соединении компьютера с модемом.

Формат передаваемых данных показан на рис. 14.

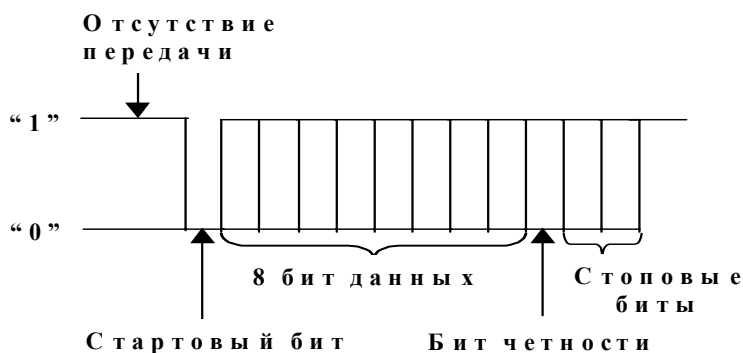


Рис. 14. Формат данных RS-232C

Собственно данные (5, 6, 7 или 8 бит) сопровождаются стартовым битом, битом четности и одним или двумя стоповыми битами.

Получив стартовый бит, приемник выбирает из линии биты данных через определенные интервалы времени. Очень важно, чтобы тактовые частоты приемника и передатчика были одинаковыми (допустимое расхождение – не более 10 %). Скорость передачи по RS-232C может выбираться из ряда: 110, 150, 300, 600, 1200, 2400, 4800, 9600, 19200, 38400, 57600, 115200 бит/с. Все сигналы RS-232C передаются специально выбранными уровнями, обеспечивающими высокую помехоустойчивость связи (рис. 15.). Отметим, что данные передаются в инверсном коде (логической единице соответствует низкий уровень, логическому нулю – высокий уровень).

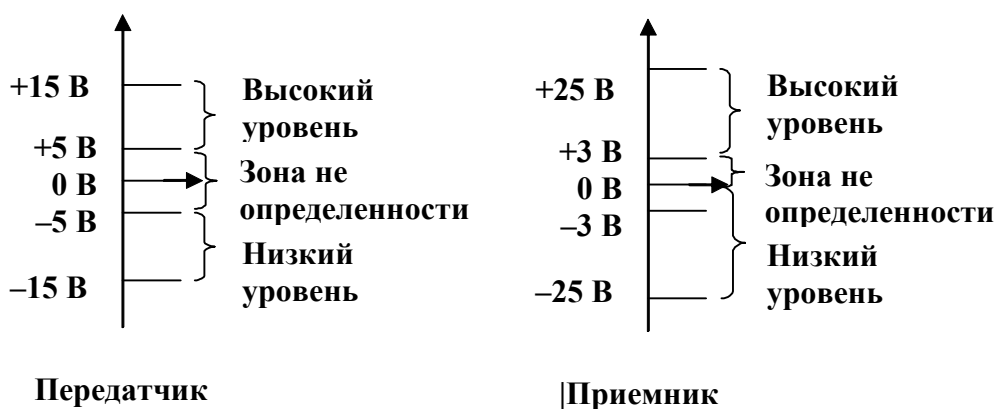


Рис. 15. Уровни сигналов RS-232C на передающем и принимающем концах линии связи

Для подключения произвольного УС к компьютеру через RS-232C обычно используют трех- или четырехпроводную линию связи (см. рис. 1.1), но можно задействовать и другие сигналы интерфейса. Обмен по RS-232C осуществляется с помощью обращений по специально выделенным для этого портам COM1 (адреса 3F8h...3FFh, прерывание IRQ4), COM2 (адреса 2F8h...2FFh, прерывание IRQ3), COM3 (адреса 3E8h...3EFh, прерывание IRQ10), COM4 (адреса 2E8h...2EFh, прерывание IRQ11). Форматы обращений по этим адресам можно найти в многочисленных описаниях микросхем контроллеров последовательного обмена UART (Universal Asynchronous Receiver/Transmitter), например i8250, KP580BB51 и др.

Интерфейс RS-232C/CCITT V24, определенный стандартом Ассоциации электронной промышленности (EIA), подразумевает наличие оборудования двух видов: терминального DTE и связного DCE. Терминальное оборудование, например компьютеры, может посылать и (или) принимать данные по последовательному интерфейсу. Оно как бы оканчивает (terminate) последовательную линию. Связное же оборудование понимается как устройства, которые могут упростить

последовательную передачу данных совместно с терминальным оборудованием (например, модем).

Различие между терминальным и связным оборудованием довольно расплывчато. Произведя незначительные изменения в линиях интерфейса RS-232, можно заставить связное оборудование функционировать как терминальное.

**DTE** – Data Terminal Equipment (COM-порт, принтер, плоттер) – разъем “папа” (male).

**DCE** - Data Communication Equipment (модемы) – разъем “мама” (female).

Общее правило соединения (бывают исключения):

Кабель DTE-DCE: “мама” – “папа”, соединяются одноименные цепи. Переходники 9-25 и 25-9 – аналогично.

Кабель DTE\_DTE (Zero-Modem Cable) – “нуль-модем”: соединяются земли, RD и TD перекрестно, остальные сигналы соединяются по одной из вышеприведенных схем (рис. 2.5).

Наборы передаваемых сигналов могут сокращаться в зависимости от протокола квитирования (Flow Control).

Входы портов буферизованы триггерами Шмитта 1489 (2 шт.) с порогами переключения – 3 В и +3 В, выходы – передатчиками 1488 с выходными уровнями –12...-5 В и +5...+12 В, током короткого замыкания до 20 мА.

На современных системных платах порт COM2 может конфигурироваться на использование **IR** (Infra Red) – беспроводной инфракрасной связи на скорости 115 Кбит/с в стандарте Hewlett Packard IR или Amplitude Shifted Keyed IR. IR-интерфейс имеют блокнотные ПК, некоторые принтеры. Внешний приемопередатчик – “красный глаз” – устанавливается на лицевой панели компьютера и подключается к разъему **IR-Connector** системной платы.

#### 1.2.2.1. Программирование последовательного порта компьютера

Контроллер последовательного интерфейса обеспечивает асинхронный обмен по стандарту **RS-232C**. Порты COM1 - COM4 поддерживаются BIOS INT 14h.

Функции прерывания **INT 14h**:

00h-инициализация (110-9600 бит/с, 5, 7 или 8 бит данных, 1-2 стоп-бита, паритет);

01h-вывод символа (без прерываний);

02h-ввод символа (без прерываний);

03h-опрос состояния модема и линии.

Используемые ячейки BIOS DATA AREA:

0:0400, 0402, 0404, 0406 – адреса портов COM1-COM4;

0:047C, 047D, 047E, 047F – тайм-аут COM1-COM4.

Стандартные **базовые адреса и прерывания:**

COM1: 3F8h-IRQ4;

COM2: 2F8h-IRQ3;

COM3: AT-3E8h, 3E0h, 338h-IRQ4; PS/2-3220h-IRQ3;

COM4: AT-2E8h, 2E0h, 238h-IRQ3; PS/2-3228h-IRQ3;

COM5-COM8: PS/2-4220h, 4228h, 5220h, 5228h -IRQ3;

### **8250/16450/16550 UART**

COM-порты XT/AT базируются на микросхеме UART (Universal Asynchronous Receiver-Transmitter – универсальный асинхронный приемопередатчик), совместимой с i8250-8250A/16450/16550.

Основные отличительные особенности членов семейства 8250:

- 8250 имеет ошибки, учтенные в XT BIOS;
- 8250A – ошибки исправлены, потеряна совместимость с BIOS. Работает в некоторых моделях AT, но не на скорости 9600 бит/с.
- 8250B – исправлены ошибки 8250 и 8250A; восстановлена ошибка в прерываниях – совместима с XT BIOS. В AT работает под DOS, кроме скорости 9600 бит/с.
- 16450 – высокоскоростная версия 8250 для AT. Ошибок 8250 и полной совместимости с XT BIOS не имеет. Минимум, требуемый для OS/2.
- 16550A – имеет работающие 16-байтные FIFO-буферы приема и передачи и возможность использования DMA. Должен применяться в AT при интенсивных обменах на скоростях 9600 бит/с и выше без потери данных.

### **Назначение регистров 8250**

**0R/W - DATA** – регистр данных (DLAB=0).

**0W – DLL** – младший байт делителя (DLAB=0).

**1W – DLH** – старший байт делителя (DLAB=1), делитель  $115200/V$ , где V – скорость бит/с.

**1W – IER** – регистр разрешения прерываний (1 = разрешить прерывание):

Биты 7..4 = 0.

Бит 3 – Mod IE – по изменению состояния модема (любой из линий CTS, DSR, RI, DCD).

Бит 2 – RxL IE – по обрыву/ошибке линии.

Бит 1 – TxD IE – по завершению передачи.

Бит 0 – RxD IE – по приему символа.

### **2R – IR – регистр идентификации прерывания:**

Биты 2..1 – причина прерывания:

11 = ошибка/обрыв линии; сброс – чтением регистра состояния линии.

10 = принят символ; сброс чтением данных.

01 = передан символ; сброс записью данных.

00 = изменение состояния модема; сброс – чтением регистра состояния модема.

Бит 0 – Interrupt Pending – 1 = нет запроса прерывания.

### **3 R/W – LCR – регистр управления линией:**

Бит 7 – DLAB – доступ к делителю (регистрам #0,1).

Бит 6: 1=обрыв линии (посылка нулей).

Бит 5=0 – отмена постоянной четности.

Бит 4: 0=нечетность, 1=четность.

Бит 3: 1=контроль паритета разрешен.

Бит 2: 0=1, 1=2 (1.5 для 5-битного кода) стоп-бит.

Биты 1-0: 00=5, 01=6, 10=7, 11=8 бит/симв.

### **4W – MCR – регистр управления модемом:**

Биты 7..5 = 8.

Бит 4 – Loopback – диагностика.

Бит 3 – OUT2 – выходной сигнал (используется для разрешения IRQ).

Бит 2 – OUT1 – выходной сигнал (свободен).

Бит 1 – RTS: 1=активен (-).

Бит 0 – DTR: 1=активен (-).

### **5R – LSR – регистр состояния линии:**

Бит 7 = 0.

Бит 6 – TEMPT – регистр передатчика пуст, нет передаваемых данных.

Бит 5 – THRE – регистр передатчика пуст, готов принять байт для передачи.

Бит 4 – B1 – индикатор обрыва линии.

Бит 3 – FE – ошибка кадра (стоп-бит).

Бит 2 – PE – ошибка четности.

Бит 1 – OE – переполнение (потеря символа).



Бит 0 – RxRdy – данные приняты; сброс – чтением приемника.

**6R – MSR – регистр состояния модема:**

Бит 7 – состояние линии DCD.

Бит 6 – состояние линии R1.

Бит 5 – состояние линии DSR.

Бит 4 – состояние линии CTS.

Бит 3 – DDCD – изменение состояния линии DCD.

Бит 2 – TER1 – изменение огибающей R1.

Бит 1 – DDSR – изменение состояния линии DSR.

Бит 0 – DCTS – изменение состояния линии CTS.

**7 W – Scratch-Pad Register** – в 8250 отсутствует.

Для программирования приемопередатчика компьютера необходимо знать его базовый адрес в адресном пространстве портов ввода-вывода компьютера. Для его определения следует обратиться к таблице базовых адресов COM-портов (COM – логическое имя последовательного интерфейса). Эта таблица находится по адресу [40:0] в памяти компьютера. Таблица состоит из слов – адресов портов ввода - вывода. В компьютере может быть до четырех COM-портов, соответственно таблица состоит из восьми байт. Если порт отсутствует, в таблице стоит вместо адреса нуль.

Для адресации регистров управления COM-портов используются три адресные линии, но нужно адресовать десять регистров, поэтому бит D7 порта 3 (3D7) используется для переключения функций портов 0 и 1. В табл. 4 приведен список регистров управления.

Таблица 4

Обозначение регистров последовательного порта

Адрес	3D7	Чтение/ Запись	Обозначение	Наименование регистра
0	0	Чтение	THR Transmitter Holding Register	Буфер приема
0	0	Запись	RBR Receiver Buffer Register	Буфер передачи
0	1	Чт/зп	DLL Divisor Latch LSB	Делитель частоты, младший байт
1	1	Чт/зп	DLM Divisor Latch MSB	Делитель частоты, старший байт
1	0	Чт/зп	IER Interrupt Enable Register	Регистр разрешения прерываний
2	0	Чт/зп	IIR Interrupt Identification Register	Идентификатор прерывания
3	0	Чт/зп	LCR Line Control	Управление линией

			Register	
4	0	Чт/зп	MCR Modem Control Register	Управление модемом
5	0	Чт/зп	LSR Line Status Register	Состояние линии
6	0	Чт/зп	MSR Modem Status Register	Состояние модема

Для передачи байт-данных записывается в буфер передачи, прием обеспечивается чтением буфера приема. Также возможен вызов прерывания по приему или передаче данных, ошибке при обмене данными или изменению состояния модема (изменению состояния линий DCD,DSR,CTS,RI). Регистр 1 (при 3D7=0) служит для разрешения/запрещения прерываний. Причину прерывания можно узнать, прочитав регистр 2 – идентификатор прерывания. Регистр 3 служит для управления форматом передачи (кроме бита 7). Регистры 4 и 6 – регистры линий управления модемом (DCD,DTR,DSR,RTS,CTS). Регистры 0 и 1 (при 3D7=1) управляют скоростью передачи данных.

Теперь рассмотрим последовательность этапов инициализации COM-порта на примере программы DEMO68HC.PAS (см. приложение 1):

1. Запись в регистр 3 управляющего байта с единицей в 7 разряде:  
 {p-базовый адрес порта};  
 Port [P+3]:=\$83.
2. Программирование делителя частоты:  
 {BR–Baud Rate – скорость передачи данных в бод, обычное значение – 9600 };  
 BT:=Round(115200.0/BR); {BT – коэффициент деления};  
 Port[P+0]:=Lo(BT);  
 Port[P+1]:=Hi(BT).
3. Запись в регистр 3 управляющего байта с нулем в 7 разряде:  
 Port[P+3]:=\$03;{\$03-8 бит данных,1 стоповый бит, без бита четности}.
4. Запрещение прерываний:  
 Port[P+1]:=\$00.

Для того чтобы данные передавались (принимались) правильно, при чтении или записи буфера данных (регистра 0) необходимо ждать приема данных или полной передачи предыдущего байта. Для этого используются биты 5 и 0 регистра 5 – регистра состояния линии:

### 1. Передача байта.

repeat until Port[P+5] and \$20=\$20; { бит 5 регистра 5 равен 1 –  
буфер передачи пуст }

Port[P+0]:=a; {передача байта данных в a}

### 2. Прием байта.

{Ticks:Longint absolute \$40:\$6C – системный таймер с приращением на единицу каждую 1/18,9 с. Используется для выхода из цикла ожидания через 5 с (при отсутствии получаемых данных).

ST := Ticks;

Repeat Until (Port[P+5] and 1=1) or (Ticks-ST>94);

If Port[P+5] and 1=1 then a:=port[P];

{бит 1 регистра 5 равен 1 - получен байт данных a}.

## 1.2.2.2. Последовательный интерфейс

### и его стандарты

RS (в сокращениях типа RS-232, RS-485, RS-422) – это всего на всего Recommended Standard (рекомендованный стандарт). Слово “рекомендованный” означает, что эти стандарты никогда никем не были приняты (в противоположность таким стандартам, как IEEE-1284 или IEEE-1394), они были просто “рекомендованы”. Естественно, это позволяет производителям допускать часто определенные вольности (например, питание по 9-му пину в RS-232 вовсе не оговорено стандартом, однако широко используется).

Далее все RS-протоколы можно приблизительно разделить на полудуплексные (half-duplex) и дуплексные (full-duplex). Правда, деление такое не совсем точно, так как тот же RS-485 может быть и полудуплексным (два провода), и дуплексным (четыре провода), они так и называются: 2-wire (2-проводный) RS-485 и 4-wire (4-проводный) RS-485. Есть еще такой вид протоколов, как симплексный (simplex), но ввиду ряда причин в компьютерной технике не применяется. Чем эти виды протоколов различаются?

Симплексные протоколы позволяют передавать данные только в одну сторону, т. е. только с передатчика на приемник, но не обратно. Хороший пример симплексного протокола – FM радио или телевидение. Применяется он в тех случаях, когда надо просто передать информацию какому-либо устройству без необходимости подтверждения и обратной связи.

Полудуплексные протоколы снимают главное ограничение симплексных протоколов – одностороннюю связь. Они позволяют

двум устройствам обмениваться информацией, причем оба устройства могут быть и приемниками, и передатчиками, но не одновременно, т. е. каждое устройство может либо передавать, либо принимать (классический / рекомендованный / RS-485 именно полудуплексный).

Дуплексные протоколы наиболее совершенные. Применение дуплексного протокола позволяет вести и прием, и передачу информации одновременно, т. е. оба устройства могут быть и приемником, и передатчиком одновременно. Например, RS-232 – дуплексный протокол.

Наиболее используемыми в компьютерной индустрии являются два протокола – RS-232 и RS-485 (RS-422 похож на RS-485). Важное отличие: протокол RS-232 использует небалансный (unbalanced) сигнал, в то время как RS-422/RS-485 использует балансный (balanced) сигнал.

Небалансный сигнал передается по несбалансированной линии, представляющей собой сигнальную землю и одиночный сигнальный провод, уровень напряжения на котором используется, чтобы передать или получить двоичные 1 или 0. Напротив, балансный сигнал передается по сбалансированной линии, которая представлена сигнальной землей и парой проводов, разница напряжений между которыми используется для передачи/приема бинарной информации (все вместе составляет экранированную витую пару).

Не углубляясь в подробности, можно сказать, что сбалансированный сигнал передается быстрее и дальше, чем несбалансированный.

Вот сравнительная таблица для рекомендованных протоколов (надо сказать, что нынешние интерпретации протоколов, особенно RS-232, весьма далеки от рекомендованных):

Сравнительная характеристика последовательных интерфейсов

	<b>RS-232</b>	<b>RS-422</b>	<b>RS-485</b>
<b>Соединения</b>	Одиночный провод	Одиночный провод/много соединений допустимо	Много соединений допустимо
<b>Количество устройств</b>	1 передатчик 1 приемник	5 передатчиков 10 приемников на 1 передатчик	Много передатчиков и приемника
<b>Вид протокола</b>	Дуплексный	дуплексный	полудуплексный

<b>Макс. длина провода</b>	~15.25 м при 19.2Kbps	~1220 м при 100Kbps	~1220 м при 100Kbps
<b>Макс. скорость передачи</b>	19.2Kbps для 15 м	10Mbps для 15 м	10Mbps для 15 м
<b>Сигнал</b>	Небалансный	балансный	балансный
<b>Двоичная 1</b>	-5В минимум -15В максимум	2В мин (B>A) 6В макс. (B>A)	1.5В мин (B>A) 5В макс. (B>A)
<b>Двоичный 0</b>	5В мин. 15В макс.	2В мин (A>B) 6В макс. (A>B)	1.5В мин (A>B) 5В макс. (A>B)
<b>Мин. входное напряжение</b>	(+/-) 3В	0.2В диф.	0.2В диф.
<b>Выходной ток</b>	500мА	150мА	250мА

Следует отметить, что контроллеры RS-232, применяемые в персональных компьютерах, имеют выходной ток, равный 10мА.

Таким образом, последовательный интерфейс на физическом уровне может иметь различные реализации, отличающиеся способами передачи электрических сигналов. На рис.16 приведены схемы соединения приемников и передатчиков и показаны их ограничения на длину линии (L) и максимальную скорость передачи данных (V).

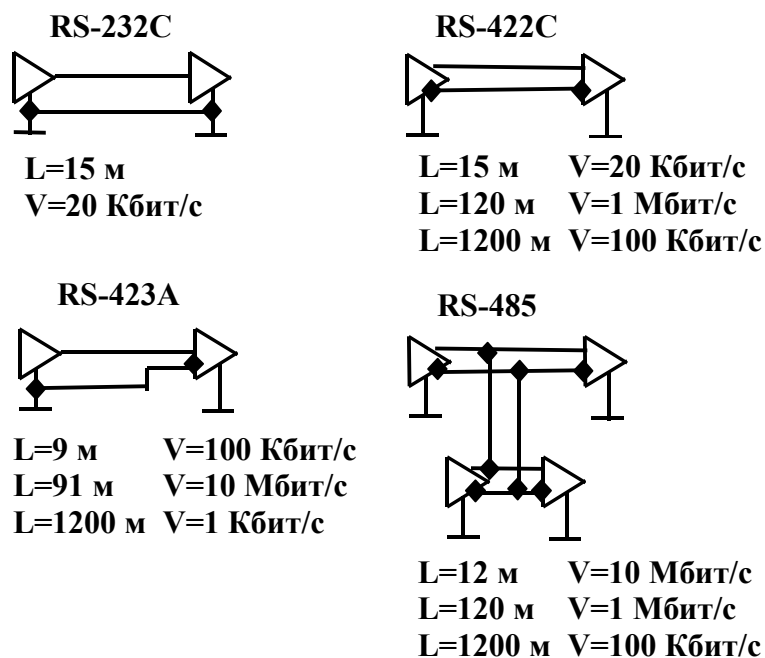


Рис.16. Стандарты последовательного интерфейса

Наибольшее распространение в РС получил простейший из этих – стандарт RS-232C. В промышленной автоматике широко

применяется RS-485, а также RS-422А, встречающийся в некоторых принтерах. Существуют относительно несложные преобразователи сигналов для согласования всех этих родственных интерфейсов.

Интерфейс RS-232С предназначен для подключения аппаратуры, передающей или принимающей данные (ООД – окончное оборудование данных, или АПД – аппаратура передачи данных), к оконечной аппаратуре каналов данных (АКД). В роли АПД может выступать компьютер, принтер, плоттер и другое периферийное оборудование. Этой аппаратуре соответствует аббревиатура DTE – Data Terminal Equipment. В роли АКД обычно выступает модем, этой аппаратуре соответствует аббревиатура DCE - Data Communication Equipment. Конечной целью подключения является соединение двух устройств DTE. Полная схема соединения приведена на рис. 17а. Интерфейс позволяет исключать канал удаленной связи вместе с парой устройств DCE (модемов), соединив устройства с помощью нуль-модемного кабеля (рис. 17б).

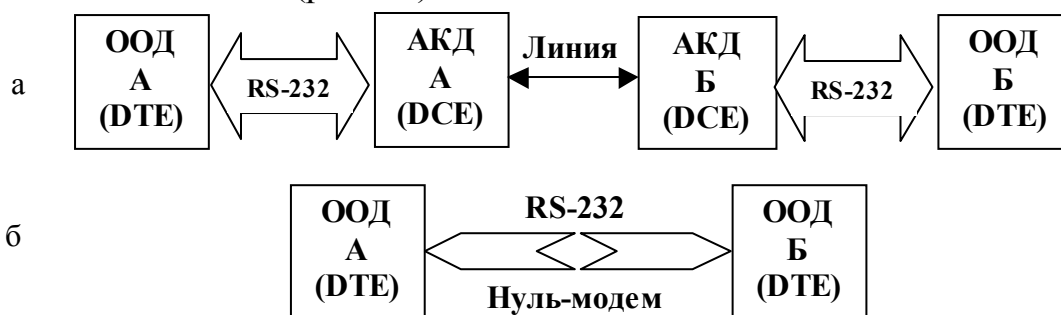


Рис.17. Полная схема соединения:

а – RS-232С;

б – нуль-модемное соединение

Чтобы не составить себе неправильного представления об интерфейсе RS-232С, необходимо отчетливо понимать различия между этими видами оборудования. Терминальное оборудование, например компьютер, может посылать и (или) принимать данные по последовательному интерфейсу. Оно как бы оканчивает (terminate) последовательную линию связи. Связное же оборудование понимается как устройства, способные упростить последовательную передачу данных совместно с терминальным оборудованием. Наглядным примером связного оборудования, как мы уже отмечали, служит модем (модулятор-демодулятор). Он служит соединительным звеном в последовательной цепочке между компьютером и телефонной линией. К сожалению, различие между терминальным и связным оборудованием довольно расплывчато, поэтому возникают некоторые

сложности в понимании того, к какому типу оборудования относится то или иное устройство, например принтер.

Стандарт описывает управляющие сигналы интерфейса, пересылку данных, электрический интерфейс и типы разъемов. Стандарт описывает асинхронный и синхронный режимы обмена, но COM-порты поддерживают только асинхронный режим.

RS-485/RS-422 используют экранированную витую пару, экран в качестве сигнальной земли. Хотя сигнальная земля обязательна, она не используется для определения логического состояния линии. Устройство, управляющее сбалансированной линией (balanced line driver), может (для RS-485 – обязательно, для RS-422 – нет) также иметь входной сигнал “Enable” (Разрешен), который используется для управления выходными терминалами устройства. Если сигнал “Enable” выключен, то это значит, что устройство отключено от линии, причем отключенное состояние устройства обычно называется “tristate” (т. е. третье состояние, вдобавок к двоичным 1 и 0).

Стандарт на RS-485 предусматривает только 32 пары передатчик/приемник, но производители расширили возможности RS-485 протокола, так что теперь он поддерживает от 128 до 255 устройств на одной линии, а используя репитеры, можно продлевать RS-485/RS-422 практически до бесконечности. При использовании RS-485 можно, и в случае длинного провода и/или большого количества устройств необходимо, использовать терминаторы, которые, впрочем, обычно встроены в устройства с RS-485-протоколом, хотя при коротком проводе может наблюдаться даже ухудшение связи при использовании терминаторов. Также стандарт на RS-485 предусматривает использование двухжильной экранированной витой пары, так называемой 2-wire RS-485, но возможно использование и четырехпроводной витой пары (4-wire RS-485); тогда получается полный дуплекс. В таком случае необходимо, чтобы одно из устройств было сконфигурировано как ведущее (Master), а остальные как ведомые (Slave). Тогда все ведомые устройства общаются только с ведущим устройством и никогда не передают ничего напрямую друг другу. В таких случаях обычно RS-422 драйвер используется как ведущее устройство, т. к. RS-422 допускает подключения только как master/slave, а RS-485 устройства как ведомые для общего удешевления системы. Стандарт на RS-422 изначально предусматривает использование четырехжильной экранированной витой пары, но допускает соединения только от одного устройства к другим (до пяти драйверов и до десяти ресиверов на каждый драйвер). RS-422 был

придуман для замены RS-232 в тех случаях, когда RS-232 не удовлетворяет по скорости и дальности передачи.

RS-422 использует строго разделенные две (или больше) пары проводов: одну пару для приема, одну для передачи (и еще по одной на каждый сигнал контроля/подтверждения (control/handshake)). RS-485, благодаря наличию третьего состояния (“tristate”), позволяет обойтись одной парой проводов, что снижает общую стоимость системы при обеспечении связи на большие расстояния. Надо сказать, что сейчас доступно большое количество разнообразных устройств для сопряжения RS-422/RS-485 с RS-232, причем RS-232 часто используется для соединения с компьютером (впрочем, есть и интерфейсные карты RS-422/RS-485 для компьютера), который используется для управления системой. Так же доступны разнообразные устройства (хабы, репитеры, переключатели и пр.) для создания сложных конфигураций RS-422/RS-485 сетей, так что эти интерфейсы таят в себе большие возможности.

### **1.3. Организация последовательного ввода-вывода в микроконтроллерах**

Наличие в составе 8-разрядного МК модуля контроллера последовательного ввода/вывода стало настолько обычным явлением, что лишь самые простые, маловыводные МК в корпусах DIP-116 и DIP-20 не имеют портов последовательного обмена. При анализе структуры более сложных 8-разрядных МК отчетливо прослеживается тенденция наличия в их составе двух контроллеров последовательного обмена, а в некоторых новейших моделях — даже трех (серия AZ семейства HCO8 фирмы Motorola). Задачи, которые решаются средствами модуля контроллера последовательного ввода/вывода, могут быть условно разделены на три группы:

1. Связь встраиваемой МП-системы с системой управления верхнего уровня: промышленным компьютером, программируемым контроллером, офисным компьютером. Наиболее часто для этих целей используются интерфейсы RS-232C и RS-485.
2. Связь с внешними по отношению к МК периферийными ИС встраиваемой МП-системы, а также с датчиками физических величин с последовательным выходом. Для этих целей используются интерфейсы SPI, I2C, а также нестандартные протоколы обмена.
3. Интерфейс связи с локальной сетью в мультимикропроцессорных системах. В системах с числом МК до пяти обычно используют



сети на основе интерфейсов I2C, RS-232C, RS-485 с собственными сетевыми протоколами верхнего уровня. В более сложных системах все более популярным становится протокол CAN.

С точки зрения инженера-схемотехника, упомянутые типы интерфейсов последовательной связи отличаются режимом передачи данных (синхронный или асинхронный), форматом кадра, (число бит в посылке при передаче байта полезной информации) и временными диаграммами сигналов на линиях (уровни сигналов и положение фронтов при переключениях). Напротив, число линий, по которым происходит передача в последовательном коде, обычно равно двум (I2C, RS-232C, RS-485) или трем (SPI, некоторые нестандартные синхронные протоколы). Последнее позволяет спроектировать модули контроллеров последовательного обмена таким образом, чтобы с их помощью на аппаратном уровне можно было бы реализовать несколько типов последовательных интерфейсов. При этом режим передачи (синхронный или асинхронный) и формат кадра поддерживаются на уровне логических сигналов, а реальные физические уровни сигналов, характерные для каждого типа интерфейса, получают с помощью специальных ИС, которые носят название приемопередатчиков, конверторов, трансиверов.

Среди не очень большого множества различных типов встроенных контроллеров последовательного обмена, которые входят в состав тех или иных 8-разрядных МК, сложился стандарт “де-факто” – модуль UART (Universal Asynchronous Receiver and Transmitter). В переводе с английского UART – универсальный асинхронный приемопередатчик. Однако большинство модулей UART, кроме асинхронного режима обмена, способны также реализовать режим синхронной передачи данных. Поэтому более правильным с точки зрения терминологии было бы именовать эти модули USART (Universal Synchronous/Asynchronous Receiver and Transmitter) – универсальный синхронно-асинхронный приемопередатчик. Такой термин тоже существует, но вспоминают о нем редко.

Не все производители МК используют термин UART для обозначения типа модуля контроллера последовательного обмена. Так, в МК фирмы Motorola модуль асинхронной приемопередачи, который поддерживает те же режимы асинхронного обмена, что и UART, принято называть SCI (Serial Communication Interface). Следует отметить, что модуль типа SCI обычно реализует только режим асинхронного обмена, т. е. его функциональные возможности уже по сравнению с модулями типа UART. Однако бывают и исключения: под

тем же именем SCI в МК MC68HC705B16 скрывается модуль синхронно-асинхронной передачи данных.

Модули типа UART в асинхронном режиме работы позволяют реализовать протокол обмена для интерфейсов RS-232C, RS-422A, RS-485, в синхронном режиме – нестандартные синхронные протоколы обмена и в некоторых моделях – SPI. МК фирмы Motorola традиционно имеют в своем составе два модуля последовательного обмена:

- модуль SCI с возможностью реализации только протоколов асинхронной приемопередачи для интерфейсов RS-232C, RS-422A, RS-485;
- модуль контроллера синхронного интерфейса в стандарте SPI.

Протоколы интерфейсов локальных сетей на основе МК (I2C и CAN) отличает более сложная логика работы. Поэтому контроллеры CAN интерфейса всегда выполняются в виде самостоятельного модуля. Интерфейс I2C с возможностью работы как в ведущем, так и ведомом режиме также обычно поддерживается специальным модулем (модуль последовательного I2C порта в МК 8C552 фирмы Philips). Но если реализуется только ведомый режим I2C, то в МК PIC16 Microchip он успешно сочетается с SPI: настройка одного и того же модуля на один из протоколов осуществляется путем инициализации.

#### 1.3.1. Последовательный асинхронный интерфейс. Устройство последовательного обмена SCI(Serial Communication Interface)

В микроконтроллерах семейства MC68HC11 устройство последовательного обмена называется SCI(Serial Communication Interface). На плате стенда расположена микросхема MC145407, преобразующая уровни сигналов на выводах микроконтроллера (0/+5в) в уровни интерфейса RS-232 (+/-12 в). В стенде используется наиболее простой вариант обмена информацией(нуль-модем), при котором используются только три сигнала: TxD (Transmit Data – передача данных), RxD (Recieve Data – Прием данных) и GND (Ground – земля).

Для того чтобы информация передавалась корректно, необходимо запрограммировать SCI в микроконтроллере и последовательный порт (COM) в компьютере на одинаковые скорости передачи и форматы передачи байт.

### 1.3.1.1 Программирование SCI микроконтроллера в МК MC68HC711

Для программирования SCI микроконтроллера используются 5 регистров (см. раздел 1.1.6) из блока, находящегося по адресу \$1000:

\$102B – BAUD – Регистр управления скоростью передачи.

\$102C – SCCR1 – Управление 8/9 битовым режимом передачи данных и активизацией SCI.

\$102D – SCCR2 – Разрешение/запрещение прерываний и приема/передачи.

\$102E – SCSR – Регистр состояния SCI.

\$102F – SCDR – Регистр данных.

Также для работы с SCI необходимо разрешить соответствующие линии порта D, используемые как линии TxD и RxD, но, так как для работы отладчика PCBUG11 используется SCI, эти линии уже настроены на необходимый режим работы, и инициализировать их необязательно. Рассмотрим инициализацию SCI на примере программы TESTCOM: (см. приложение 2).

1. Загрузка базового адреса блока регистров управления в индексный регистр микроконтроллера:

```
LDX #REGBAS * X=$1000 *;
```

2. Включение 8-битового режима обмена данными:

```
CLR SCCR1,X;
```

3. Инициализация делителя частоты и регистра управления SCCR2

```
LDD #$302C;
```

```
STAA BAUD,X BAUD=$30 => скорость обмена – 9600 бод;
```

```
STAB SCCR2,X SCCR2=$2C => прием и передача разрешены;
```

4. Запрещение всех прерываний

```
LDAA #$D0;
```

```
TAP CCR=$D0 => прерывания запрещены.
```

Прием и передача осуществляются посредством чтения и записи регистра данных SCDR. Перед обращением к этому регистру необходимо считать статус SCI из регистра SCSR. Бит 5 этого регистра (RDRF) устанавливается в “1”, если принят полный байт данных. При передаче полного байта данных устанавливается в “1” бит 7 (TDRE), который может рассматриваться как знаковый бит. При ошибках в “1” устанавливаются биты 3 (OR) или 1 (FE).

1. Прием байта:

```
INSCI EQU *
```

```
LDAB SCSR+REGBAS – Получение содержимого регистра SCSR в ACCB;
```

BITB #(FE+OR) – Проверка флагов ошибки;  
BNE START – Перезапуск при ошибке;  
ANDB #RDRF – Проверка бита RDRF;  
BEQ INSCI – Переход, если бит RDRF равен нулю;  
LDAA SCDR+REGBAS – Чтение байта данных;  
RTS – Возврат из процедуры с данными в ACCA;

2. Передача байта:

OUTSCI LDAB SCSR+REGBAS – Получение содержимого регистра SCSR в ACCB;  
BPL OUTSCI – Переход, если бит знака (TDRE) равен 0;  
STAA SCDR+REGBAS – Передача байта;  
RTS – Возврат.

1.3.2. Последовательный синхронный периферийный интерфейс SPI

Последовательный периферийный интерфейс SPI (Serial Peripheral Interface) предназначен для связи МК с периферийными устройствами микропроцессорной системы, основой которой он является. Наиболее часто эти устройства расположены на одной плате с МК, реже – это вынесенные пульта управления, индикаторные панели и т. п. В качестве периферийных устройств могут использоваться как простейшие сдвиговые регистры, так и сложные периферийные ИС со встроенными контроллерами управления, такие, как ЦАП, сигма-дельта АЦП с цифровой фильтрацией, последовательные запоминающие устройства типа FLASH или EEPROM, энергонезависимые ОЗУ и т. д.

SPI представляет собой 3-проводной интерфейс для передачи данных, в котором используются две линии: MISO (MASTER In, SLAVE Out) – линия передачи данных от ведущего к ведомому, MOSI (MASTER Out, SLAVE In) – линия передачи данных от ведомого к ведущему, одна линия – SCK – используется для синхронизации передатчика и приемника. Также может присутствовать линия выбора устройства – SS в МП-системах с несколькими периферийными устройствами.

Устройство с SPI-интерфейсом может работать как ведущее (MASTER), так и ведомое (SLAVE).

Максимальная скорость передачи по SPI-интерфейсу – 10 Мегабод. Максимальная протяженность линий связи – 10 м.

Для реализации интерфейса специальных интегральных схем не требуется.

В кадре сообщения присутствует 8 бит информации.

Временные диаграммы работы SPI-интерфейса приведены на рис. 18.

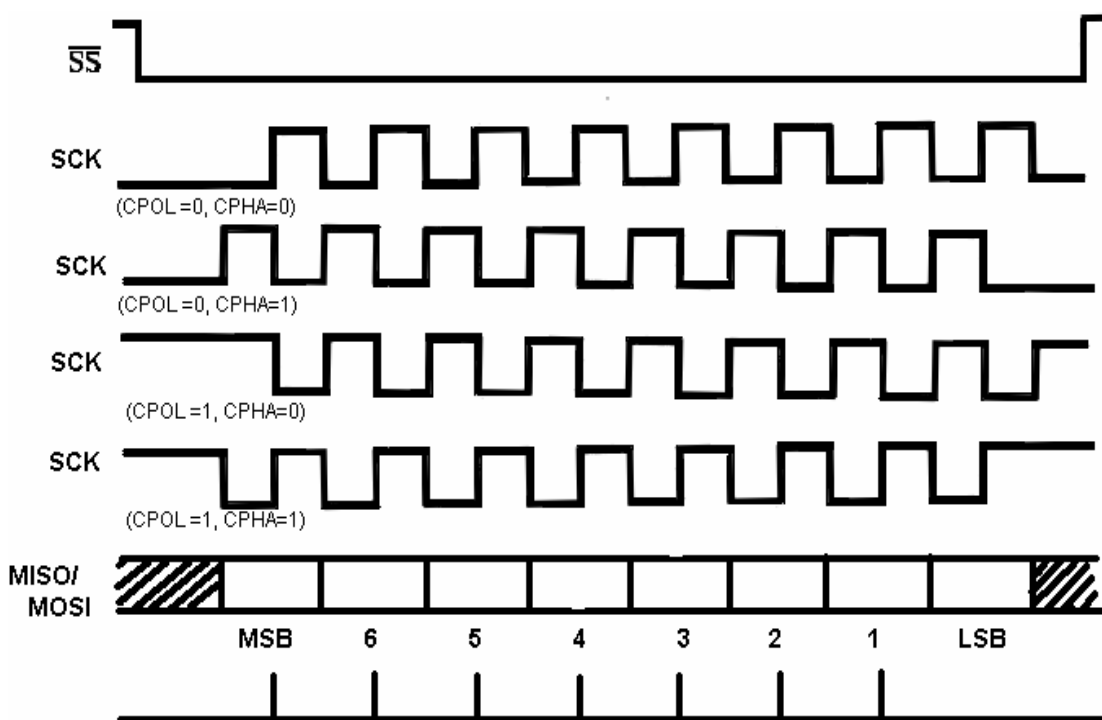


Рис. 18. Временные диаграммы работы SPI интерфейса

На изображении приведены четыре разрешаемых варианта полярности синхросигнала.

Данные по шине передаются старшим битом вперед.

### 1.3.3. Синхронный последовательный интерфейс I<sup>2</sup>C

Интерфейс I<sup>2</sup>C (синхронный последовательный интерфейс I<sup>2</sup>C) является двухпроводным последовательным интерфейсом, разработанным фирмой Philips. Изначально в стандартном режиме шина была предназначена для скорости передачи данных до 100 кбит/с. В усовершенствованном быстром режиме поддерживается передача данных со скоростью до 400 кбит/с. На одной шине одновременно могут работать устройства и стандартного, и быстрого режима. Основными свойствами шины являются следующие:

- двунаправленная передача данных между главными и подчиненными устройствами,
- многоабонентская связь (нет центрального главного узла),
- арбитраж между одновременно передающими устройствами без разрушения целостности передаваемых данных.



сеть контроллеров) является последовательным протоколом высокоскоростной и высоконадежной передачи данных в широковещательном (broadcast) режиме и мультимастерной среде. Удачное сочетание низкой стоимости подключения, простоты и надежности с проверкой временем и широкой доступности элементной базы и инструментальных средств разработки – одно из основных достоинств CAN-технологии.

Положения, закрепленные в используемой на сегодня спецификации 2.0 А/В фирмы Bosch и международном стандарте ISO 11898, соответствуют двум начальным уровням (физическому и канальному) 7-уровневой модели взаимодействия открытых систем ISO/OSI. Ряд оригинальных технических решений, реализованных при разработке протокола, наилучшим образом позволили сориентировать на решение задач контроля и управления.

Шинная топология, являющаяся основой CAN, требует наличия механизма адресации узлов, однако в CAN нет адресов как таковых, сообщение принимается всеми узлами. Любое передаваемое сообщение имеет определяющий его содержание уникальный идентификатор (ID), на основании которого каждый узел фильтрует “свои сообщения” и “решает”, реагировать или нет на сообщение, транслируемое в данный момент. Неоспоримыми преимуществами отсутствия адресации являются теоретически неограниченное количество узлов и простота их добавления и отключения.

Физическая среда передачи данных в CAN может быть самой разной – витая пара, плоский кабель, оптоволокно, а также радиоканалы и даже линии электропередач. Основным ограничением протяженности шины является лишь предельно допустимая суммарная задержка распространения сигнала для заданной скорости передачи (в кабеле, трансиверах, входных цепях контроллеров и т. д.). В соответствии с рекомендациями ISO 11898 при использовании стандартных трансиверов и быстродействующих оптопар (для гальванической развязки) максимальная протяженность сети при скорости передачи 1 Мбит/с ограничена девятью метрами. Предельная рекомендуемая протяженность сети в соответствии с тем же стандартом достигается при снижении скорости передачи до 50 кбит/с. В документах промышленной CAN-группы CiA (CAN in Automata) приведены следующие, полученные практически путем соотношения: скорость-протяженность для проводной сети без гальванической развязки: 1 Мбит/с — 30 м; 500 кбит/с — 100 м; 125 кбит/с — 500 м; 20 кбит/с — 2500 м; 10 кбит/с — 5000 м.

Число линий связи – 2, число информационных бит в кадре сообщения до – 64, число служебных бит в кадре сообщения до – 64.

Протокол требует наличия специальных микросхем для реализации интерфейса (прим. MC33388, MC33389 фирмы Motorola).

#### 1.3.5. Интерфейс MicroLAN

Интерфейс представляет собой однопроводной интерфейс, разработанный фирмой Dallas Semiconductors.

Максимальная скорость передачи по интерфейсу – 16,3 Кбит/с. Любой промышленный микроконтроллер, такой как 8051, работающий на тактовой частоте более 1,8 МГц, а также любой последовательный порт RS-232, управляемый микросхемой 8250, может легко поддерживать протокол обмена MicroLAN. Протокол MicroLAN позволяет адресовать до  $2^{56}$  устройств, подключенных к шине данных.

Приборы MicroLAN используют конфигурацию с открытым стоком при значении напряжения питания от 2,8 V (минимальное напряжение питания) до 6 V (максимальное напряжение питания). Любое напряжение, превышающее 2,2 V, рассматривается как логическая “1”, или ВЫСОКИЙ уровень, а напряжение ниже 0,8 V считается логическим “0”, или НИЗКИМ уровнем.

Не любой прибор, имеющий встроенный интерфейс MicroLAN, способен запитываться непосредственно от шины. Напряжение питания должно составлять минимум 2,8V для того, чтобы зарядить внутренний конденсатор устройства, обеспечивающий питание прибора при НИЗКОМ уровне напряжения шины. Номинал конденсатора питания составляет приблизительно 800 pF. Эта емкость заряжается на протяжении короткого промежутка времени, когда подается питание MicroLAN. После этого требуется только небольшая подзарядка конденсатора для восстановления полного заряда.

Все приборы, предназначенные для работы на шине MicroLAN, содержат встроенный сетевой контроллер, позволяющий многочисленным приборам работать в составе общей сети. Это позволяет построить распределенную систему сбора и хранения информации, использующую только одну общую линию данных к ведущему шины.

Любая сеть всегда требует наличия идентификационных номеров всех узлов в пределах сети. Не все микросхемы MicroLAN содержат область ПЗУ, в которой записан уникальный для каждой микросхемы регистрационный номер, используемый в качестве идентификатора узла. Пользователю не нужно волноваться относительно возможности



конфликта идентификаторов узлов, так как производителем гарантируется уникальность каждого из них.

Для передачи данных по шине MicroLAN используется чередование импульсов низкого уровня различной длительности: любая транзакция начинается с передачи условия сброс–импульс низкого уровня длительностью минимум 480 мкс, затем можно передавать данные на шину – последовательность логических нулей и единиц.

Логический нуль – это импульс низкого уровня длительностью минимум 60 мкс и максимум 120 мкс.

Логическая единица – импульс длительностью менее 60 мкс.

#### 1.3.6. Универсальная последовательная шина – USB

USB (Universal Serial Bus – универсальная последовательная шина) является промышленным стандартом расширения архитектуры PC, ориентированным на интеграцию с телефонией и устройствами бытовой электроники.

Версия 1.0 была опубликована в январе 1996 г. Архитектура USB 1.1 определяется следующими критериями:

- Легко реализуемое расширение периферии PC.
- Дешевое решение, поддерживающее скорость передачи до 12 Мбит/с.
- Полная поддержка в реальном времени передачи аудио и (сжатых) видеоданных.
- Гибкость протокола смешанной передачи изохронных данных и асинхронных сообщений.
- Интеграция с выпускаемыми устройствами.
- Доступность в PC всех конфигураций и размеров.
- Обеспечение стандартного интерфейса, способного быстро завоевать рынок.
- Создание новых классов устройств, расширяющих PC.

С точки зрения конечного пользователя, привлекательны следующие черты USB:

- Простота кабельной системы и подключений.
- Скрытие подробностей электрического подключения от конечного пользователя.
- Самоидентифицирующиеся ПУ, автоматическая связь устройств с драйверами и конфигурирование.

- Возможность динамического подключения и конфигурирования ПУ.

С середины 1996 г. выпускаются PC со встроенным контроллером USB, реализуемым чипсетом. Уже появились модемы, клавиатуры, сканеры, динамики и другие устройства ввода/вывода с поддержкой USB, а также мониторы с USB-адаптерами – они играют роль концентраторов для подключения других устройств.

Спецификация 2.0 обеспечивает подключение периферийных устройств со скоростью 480 Мбит/с, что примерно в сорок раз выше, чем в нынешней версии USB 1.1. Более высокое быстродействие улучшит работу таких приложений, как интерактивные игры и цифровая обработка изображений. Новый интерфейс будет применяться в цифровых видеокамерах, сканерах, принтерах, внешних накопителях и широкополосных устройствах доступа и Internet.

Спецификация 2.0 предусматривает использование таких же кабелей и разъемов, как и действующий стандарт 1.1. Однако напряжение питания, используемое контроллерами, уменьшится с 3,3 В до 400 мВ. Кроме того, на обоих концах шины теперь будут необходимы терминаторы.

## 2. ЛАБОРАТОРНЫЕ РАБОТЫ

### *Лабораторная работа № 1*

#### **ПРОГРАММИРОВАНИЕ ПАРАЛЛЕЛЬНОГО ИНТЕРФЕЙСА К580ВВ55(i8255) И УНИВЕРСАЛЬНОГО СИНХРОННО-АСИНХРОННОГО ПРИЕМОПЕРЕДАТЧИКА К580ВВ51 (i8251)**

**ЦЕЛЬ РАБОТЫ:** изучить режимы работы программируемых интерфейсных компонентов, изучить построение параллельных и последовательных портов ввода-вывода МПС на примере микросхем программируемого параллельного интерфейса (ППИ) КР580ВВ55 (Intel 8255) и универсального синхронно-асинхронного приемопередатчика (УСАПП) КР 580ВВ51 (Intel 8251).

#### **Задания для домашней подготовки.**

1. Изучить методические указания к лабораторной работе “Организация интерфейсов микропроцессорных систем обработки информации на базе программируемых компонентов”.
2. Изучить описание параллельного и последовательного порта ввода-вывода по методическому пособию “Аппаратно программные средства на базе микропроцессора КМ1810ВМ86”. Часть 1 уроки 9,10.
3. Прочсть принципиальные электрические схемы подключения адаптеров К580ВВ55 и К580ВВ51 в учебной микроЭВМ (приложение 1,2).

#### **Порядок выполнения лабораторной работы**

1. Исследовать с помощью задающих воздействий, подаваемых с тумблерных регистров универсального стенда для исследования БИС, работу параллельного интерфейса К580ВВ55(i8255) и универсального синхронно-асинхронного приемопередатчика К580ВВ51.
2. Реализовать заданные преподавателем режимы работы микросхем параллельного интерфейса К580ВВ55(i8255) и универсального синхронно-асинхронного приемопередатчика К580ВВ51.
3. Составить карту адресов внешних устройств для учебной микроЭВМ “Микролаб”.

4. Написать программу организации режима бегущих огней на светодиодной линейке, подключаемой к порту с адресом FFFA в учебной микроЭВМ “Микролаб” (минимальное время горения одной лампочки – 3с).
5. Используя один из имеющихся трансляторов (MASM или TASM), получить листинг программы и, введя код программы в “Микролаб”, запустить программу на выполнение.
6. Составить карту прошивки ПЗУ на базе м/с КР556РТ4, реализующую возможность адресации портов внешних устройств в “Микролабе” в соответствии с картой памяти на основе анализа принципиальных электрических схем в приложениях 1 и 3.

### ***Содержание отчета***

#### **Отчет должен содержать**

1. Программные модели БИС К 580ВВ55 (I 8255), К 580ВВ51 (I8251).
2. Схемы подключения к “Микролаб” внешних устройств.
3. Структуру управляющих слов, обеспечивающих заданные преподавателем режимы работы микросхем.
4. Карту прошивки дешифратора внешних устройств на базе микросхемы КР556РТ4 для учебной микроЭВМ “Микролаб”.

#### **Контрольные вопросы**

1. Дать определение порта ввода-вывода.
2. Сколько 8-разрядных портов включает в себя БИС КР580ВВ55А?
3. Описать формат управляющего слова, задающего режим работы программируемого параллельного интерфейса (ППИ).
4. Описать основные режимы работы ППИ.
5. С помощью каких сигналов происходит обмен информацией между портами ввода-вывода и шиной данных?
6. Описать отличия последовательного порта ввода-вывода и параллельного порта.
7. Назначение буфера шины данных в ППИ.
8. Описать формат слова выбора режима в микросхеме I8251.
9. Описать работу I8251 в асинхронном режиме.
10. Описать работу I8251 в синхронном режиме.

## Лабораторная работа № 2

### ИЗУЧЕНИЕ ИНТЕРФЕЙСА CENTONICS

ЦЕЛЬ РАБОТЫ: изучить интерфейс Centonics, ознакомиться с работой параллельных портов ПЭВМ.

#### Задания для домашней подготовки

1. Ознакомиться с параллельным интерфейсом (LPT-порты) и последовательным интерфейсом ПЭВМ (COM-порты).
2. Построить программную модель LPT-порта [в режиме SPP (Standard Parallel Port)].
3. Составить таблицы описания выводов LPT-порта.
4. Изучить особенности возможных режимов работы LPT-портов ПЭВМ типа IBM PC AT. (SPP, EPP, ECP).
5. Изучить по данным из Internet работу микросхемы DS1620.

#### Порядок выполнения лабораторной работы

1. Подключить к ПЭВМ к порту LPT схему для определения правильного прохождения сигналов по шине данных в стандарте Centronics (рис. 20).

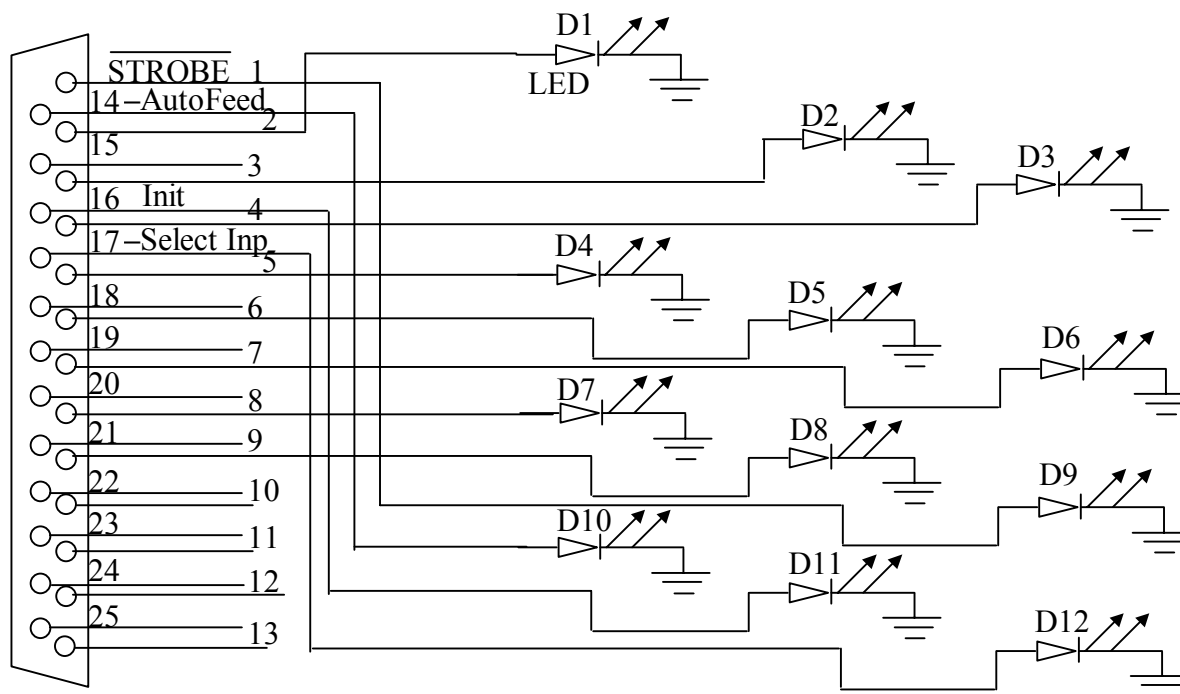


Рис.20. Схема для определения правильного прохождения сигналов по шине данных в стандарте Centronics

2. Написать на языке Ассемблера программу тестирования параллельного порта, которая обеспечивает режим бегущей единицы на выходных линиях LPT-порта (минимальное время горения одной лампочки – 3с).
3. Построить на базе микросхемы DS1620, подключенной к LTP-порту, электронный термометр. Написать ПО к термометру. (см. приложение 3).
4. Построить на базе двух фотоприемников и двух лазерных излучателей схему измерения времени движения тела вдоль наклонной плоскости. Написать ПО к схеме. (см. приложение 4).

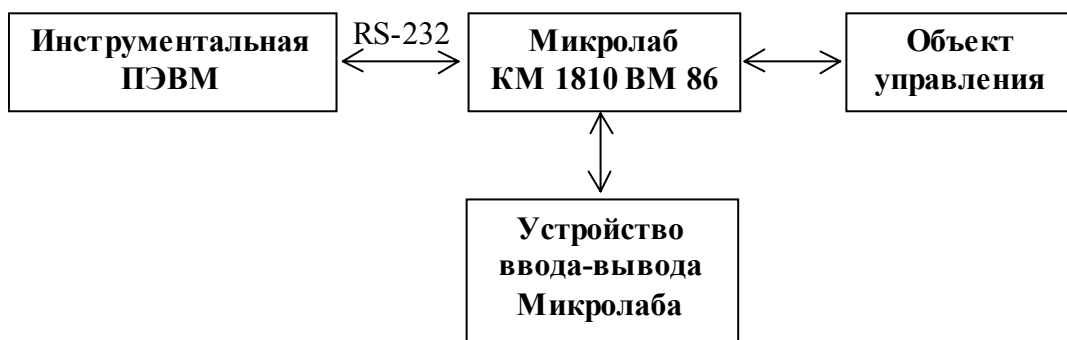
### ***Содержание отчета***

1. Программную и логическую модель LPT-порта компьютера.
2. Схему тестирования LPT-порта компьютера.
3. Тексты программ, разработанных в ходе выполнения лабораторных работ.

### ***Лабораторная работа № 3***

#### **ИЗУЧЕНИЕ ПОСЛЕДОВАТЕЛЬНОГО ИНТЕРФЕЙСА RS-232**

ЦЕЛЬ РАБОТЫ: изучить программирование интерфейса RS-232 в



составе лабораторного комплекса приведенного на рис. 21.

*Рис. 21. Структура лабораторного комплекса*

#### **Задания для домашней подготовки**

1. Ознакомиться с последовательным интерфейсом RS-232 (COM-порты ПЭВМ).
2. Построить программную модель COM-порта.
3. Составить таблицы описания выводов COM-порта.

4. Изучить особенности возможных режимов работы СОМ-порта ПЭВМ.

### **Порядок выполнения лабораторной работы**

1. Собрать лабораторный комплекс в соответствии с рисунком. Соединить учебную микроЭВМ “Микролаб” и ПЭВМ с использованием “нуль-модемного” (Null Modem) кабеля. (Тип монитора – клавиатурный).
2. Написать программы для ПЭВМ и “Микролаб”, обеспечивающие передачу байта через СОМ-порт со стороны ПЭВМ, его прием учебной микроЭВМ, инвертирование и передачу назад в ПЭВМ. ПЭВМ должна принять инвертированный байт назад, сложить его с переданным и результат вывести на экран ПЭВМ. Программу для ПЭВМ написать с использованием языка Паскаль, для “Микролаб” – языка Ассемблера. Формат передачи данных: длина символа 8 бит, стоповый бит, без бита четности, скорость 9600, коэффициент пересчета скорости для КР580ВВ51А равен 64

### ***Содержание отчета***

1. Программную модель СОМ-порта компьютера.
2. Тексты программ, разработанных в ходе выполнения лабораторных работ.
3. Схему соединения компьютеров с использованием нуль-модемного кабеля.

## **КОНТРОЛЬНЫЕ ВОПРОСЫ**

***к лабораторным работам № 2 (Изучение интерфейса Centonics),  
№ 3 (Изучение последовательных интерфейсов RS-232)***

1. *Опишите структуру LPT-портов РС.*
2. *Опишите структуру СОМ-портов РС.*
3. *Опишите режимы работы современных LPT-портов компьютера.*
4. *Опишите скорости передачи информации через СОМ-порты современных компьютеров.*
5. *Постройте программную модель UART i8250.*
6. *Как формируется установка делителя частоты для i8250?*
7. *Сформируйте последовательность управляющих посылок, устанавливающих UART в режим 9600 бит/с, длиной символа 8 бит, без разряда четности с 1-стоповым битом.*

8. *Какая команда в `rs-com.pas` запрещает прерывание от последовательного порта?*
9. *Какие команды в `rs-com.pas` организуют прием информации?*
10. *Какие команды в `rs-com.pas` производят передачу информации?*
11. *По какому адресу в памяти IBM PC находится базовый адрес последовательного порта COM2?*
12. *Постройте программную модель КР580ВВ51 (Intel 8251).*
13. *Сформируйте управляющее слово, настраивающее i8251 на работу в асинхронном режиме со скоростью передачи 1/64, длиной символа 8 бит данных, без разряда четности с 1-стоповым битом.*
14. *Сформируйте слово команды (слово приказа) контроллеру i8251 для разрешения приема/передачи.*
15. *Какие команды в программе `micro-com.asm` организуют цикл записи/чтения буфера данных?*
16. *Для каких целей используется команда сброса для i8251?*
17. *Опишите параметры физических сигналов на выходе интерфейса RS-232.*

#### ***Лабораторная работа № 4***

### **ПРОГРАММИРОВАНИЕ SCI ПОРТА МИКРОКОНТРОЛЛЕРА MC68HC11E9**

**ЦЕЛЬ РАБОТЫ:** изучить архитектуру последовательного интерфейса связи (SCI), микроконтроллера MC68HC711E9, назначение всех регистров, освоить программирование работы последовательного интерфейса во всех режимах в составе измерительного комплекса, приведенного на рис. 22.



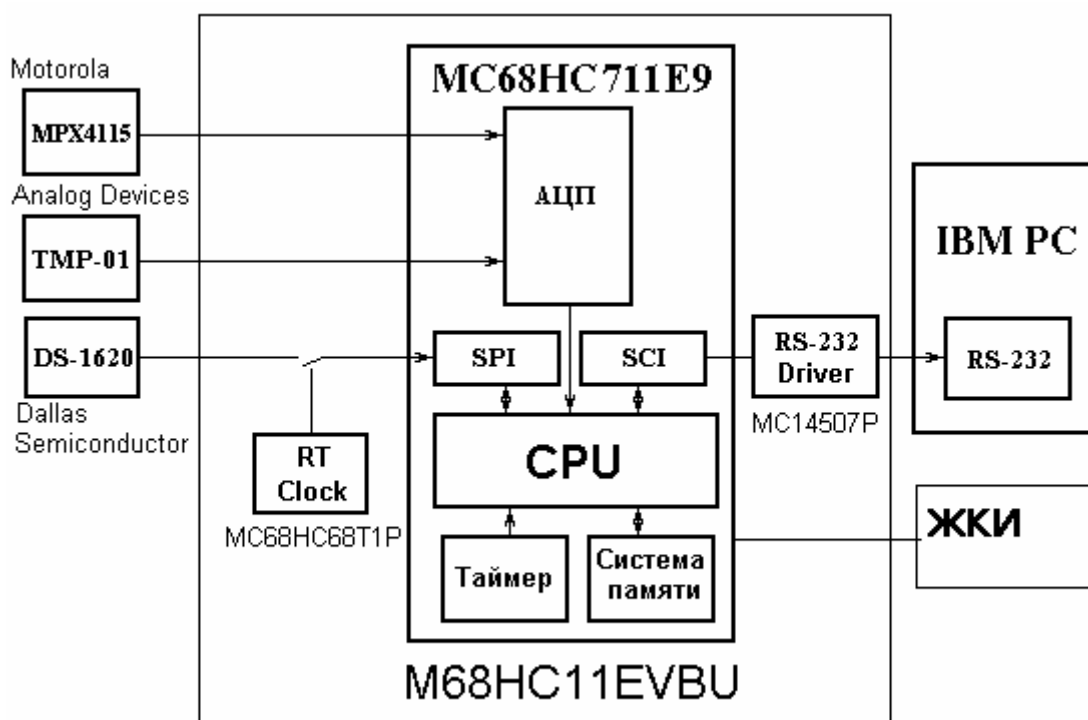


Рис. 22. Структура измерительного комплекса

### **Задание для домашней подготовки**

Изучить последовательный интерфейс связи (SCI).

### **Задание к лабораторной работе**

1. Собрать лабораторный комплекс в соответствии с рис. 22.
2. Изучить работу с транслятором AS11.
3. Изучить сигналы последовательного интерфейса SCI (см. электронный вариант книги по описанию микроконтроллера MC68HC11E9).

### **Выполнение лабораторной работы**

1. Написать программу связи по COM-порту компьютера с микроконтроллером (программа demob8HC) на языке PASCAL.
2. Написать программу для микроконтроллера на языке Ассемблер, обеспечив прием байта от компьютера по последовательному интерфейсу, инвертирование, возврат в ПЭВМ и отображение на экране дисплея (test.com).
3. Обеспечить запись разработанной программы в RAM и EEPROM микроконтроллера с помощью программы upload (см. приложение 3).

### ***Содержание отчета***

### **Название и цель работы**

1. Диаграмма развития микроконтроллеров и микропроцессоров фирмы “Motorola”.
2. Структура микропроцессорной продукции фирмы “Motorola”.
3. Программная модель микроконтроллера.
4. Структура лабораторного стенда.
5. Текст программы DEMO68HC и текст программы TEST.com.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Условные обозначения компонентов фирмы “Motorola”.
2. Нарисовать программную модель микроконтроллера.
3. Рассказать об отладчике PCbug11.
4. Рассказать о командах МК на примере программы.
5. Назовите известные вам режимы адресации.
6. Объяснить режим работы МК M68HC711E9.
7. Нарисовать структуру памяти МК в различных режимах работы.
8. Рассказать об особенностях программирования COM-порта ПЭВМ.
9. Объяснить назначение функциональных узлов SCI .
10. Форматы кадров последовательной передачи информации, поддерживаемые SCI.
11. Функциональные характеристики SCI.
12. Функции приемника SCI.
13. Функции передатчика SCI.
14. Функции регистров управления SCCR1 и SCCR2.
15. Роль регистра состояния SCSR, функциональное назначение битов регистра.
16. Функции битов регистров управления SCCR1 и SCCR2.
17. Возможности установки скорости передачи информации в SCI.
18. Прерывания, вырабатываемые в SCI; возможность управления ими.
19. Работа SCI в режиме передачи.
20. Работа SCI в режиме приема.
21. Составить схему соединения двух однокристальных ЭВМ MC68HC711E9 с использованием интерфейса SCI.
22. Как приемник контроллера SCI распознает стартовый бит?

## *Лабораторная работа № 5*

### **ПРОГРАММИРОВАНИЕ ПОСЛЕДОВАТЕЛЬНОГО ПЕРИФЕРИЙНОГО ИНТЕРФЕЙСА (SPI) МИКРОКОНТРОЛЛЕРА MC68HC711E9 СОПРЯЖЕНИЕ МИКРОКОНТРОЛЛЕРА С ЧАСАМИ РЕАЛЬНОГО ВРЕМЕНИ MC68HC68TIP**

**ЦЕЛЬ РАБОТЫ:** изучить архитектуру последовательного периферийного интерфейса микроконтроллера **MC68HC711E9**, назначение всех регистров; освоить программирование работы последовательного периферийного интерфейса во всех возможных режимах в составе измерительного комплекса, приведенного на рисунке.

#### **Задание для домашней подготовки**

Изучить режимы работы последовательного интерфейса связи SPI (см. электронный вариант книги по описанию микроконтроллера MC68HC11E9).

#### **Задания к лабораторной работе**

1. Сдать коллоквиум по теме “Последовательный периферийный интерфейс микроконтроллера **MC68HC711E9**”.
2. Пользуясь общей схемой стенда, с помощью переключателей задать режим работы контроллера с часами реального времени MC68HC68TIP по SPI.
3. На языке ассемблера контроллера написать программу связи микроконтроллера и часов реального времени MC68HC68TIP. Отображения текущего времени производить на экране ПЭВМ.
4. Оформить отчет по лабораторной работе, включающий: схему принципиальную электрическую стенда, теоретическую часть, тексты программ на языке Ассемблера и Pascal.

### **КОНТРОЛЬНЫЕ ВОПРОСЫ**

1. Объяснить сопряжение МК с часами реального времени MC68HC68TIP.
2. Рассказать о АЦП микроконтроллера M68HC711E9.
3. Что такое таймерная система МК M68HC711E9?
4. Рассказать о средствах отладки для МК фирмы “Motorola”.
5. Объясните назначение функциональных узлов SPI по структурной схеме.

6. Форматы кадров последовательной передачи информации, поддерживаемые SPI.
7. Функциональные характеристики SPI.
8. Функции регистра управления SPCR.
9. Роль регистра состояния SPSR, функциональное назначение битов регистра.
10. Функции битов регистров управления SPCR.
11. Возможности установки скорости передачи информации в SPI.
12. Прерывания, вырабатываемые в SPI; возможность управления ими.
13. Работа SPI в режиме передачи.
14. Работа SPI в режиме приема.
15. Составьте схему соединения двух (трех, четырех, пяти) однокристальных ЭВМ MC68HC711E9 с использованием интерфейса SPI.

### *Лабораторная работа № 6*

#### **ПРОГРАММИРОВАНИЕ ИНТЕРФЕЙСА I<sup>2</sup>C МИКРОКОНТРОЛЛЕРА PCF80C552. СОПРЯЖЕНИЕ МИКРОКОНТРОЛЛЕРА С МИКРОСХЕМОЙ EEPROM AT24C64**

**ЦЕЛЬ РАБОТЫ:** изучить архитектуру последовательного интерфейса связи I<sup>2</sup>C микроконтроллера PCF80C552, освоить программирование работы последовательного интерфейса во всех режимах в составе информационно-измерительного комплекса, приведенного на

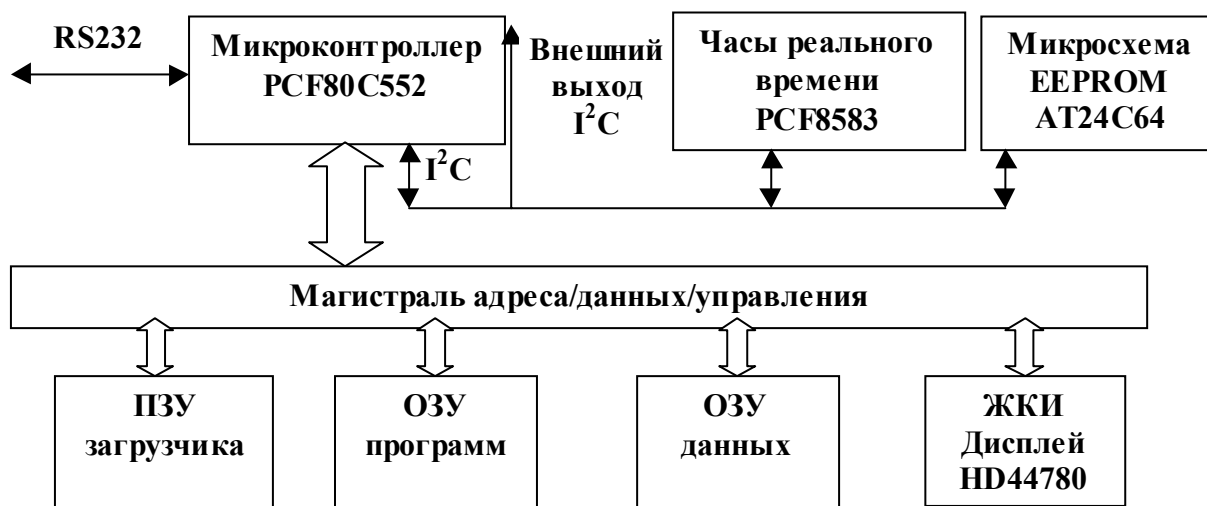


рис. 23.

Рис.23. Структурная схема лабораторного стенда EB-552

### Задание для домашней подготовки

Изучить электронную документацию на микроконтроллер PCF80C552 и микросхему EEPROM AT24C64.

### Задание к лабораторной работе

1. Написать программу, обеспечивающую формирование записной книжки (список группы) по интерфейсу I2C на базе микросхемы EEPROM AT24C64. По нажатию одной из кнопок INT0 или T0 обеспечить вывод записей на экран ЖКИ HD44780.

## *Лабораторная работа № 7*

### **ИЗУЧЕНИЕ ИНТЕРФЕЙСА MICROLAN. СОПРЯЖЕНИЕ МИКРОКОНТРОЛЛЕРА MC68HC711E9 С ДАТЧИКОМ ТЕМПЕРАТУРЫ DS1820 ПО ИНТЕРФЕЙСУ MICROLAN**

ЦЕЛЬ РАБОТЫ: изучить характеристику однопроводного интерфейса MicroLAN, освоить программирование работы последовательного интерфейса во всех режимах в составе информационно-измерительного комплекса, приведенного на рис. 24.

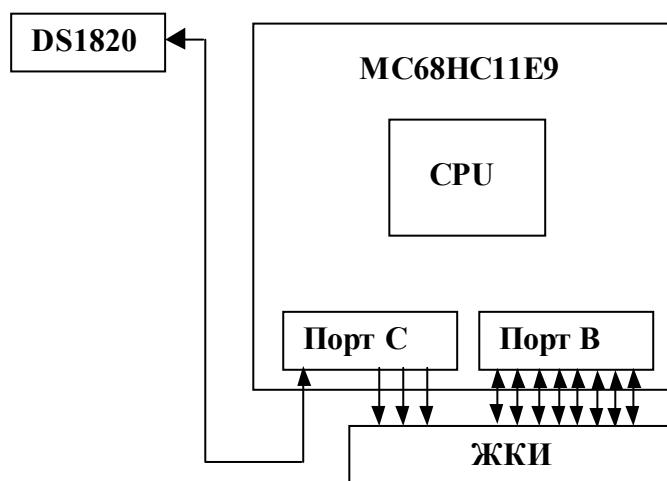


Рис. 24. Структурная схема разработанной системы

Управляющий элемент – микроконтроллер MC68HC711E9.

DS1820 – температурный датчик со встроенным однопроводным интерфейсом MicroLAN, подключаемый к контакту 7 порта С.

ЖКИ – жидкокристаллический индикатор WH1602.

Микроконтроллер MC68HC11E9 фирмы “Motorola” программно эмулирует MicroLAN интерфейс с использованием одного из портов ввода-вывода.

### **Задание для домашней подготовки**

Изучить электронную документацию на микроконтроллер MC68HC11E9 и датчик DS1820.

### **Порядок выполнения работы**

1. Написать программу измерения температуры с отображением результата на экране ЖКИ.
2. Разработать схему и структуру локальной сети для многоточечного измерения температуры.

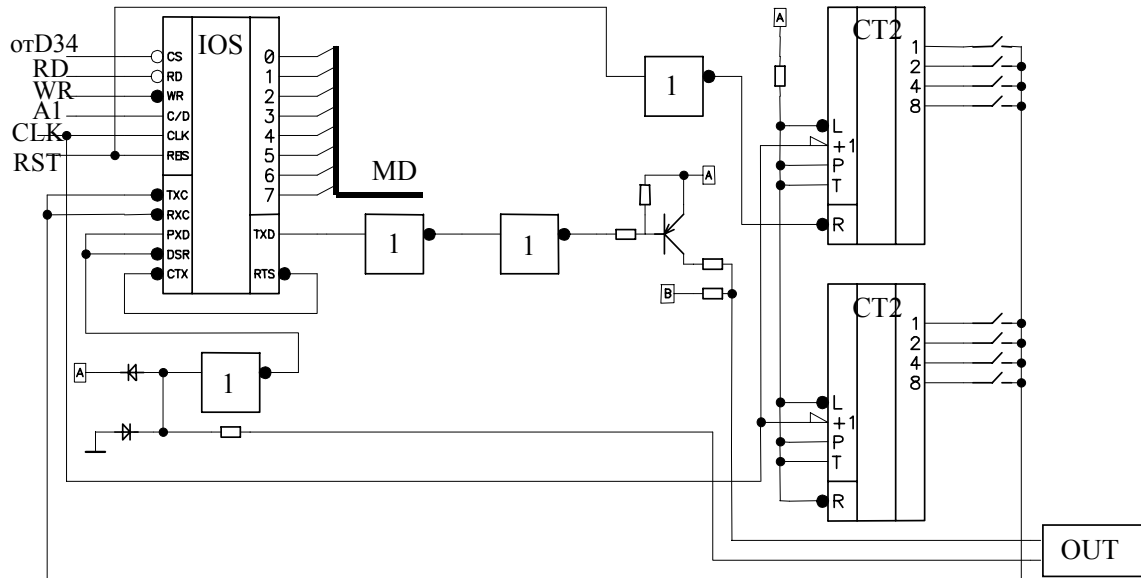
### **Литература**

1. Семейство 80C51, 80X552. 1997 КТЦ-МК. 1997.
2. 80C552. Single chip microcontroller, Philips Electronics North America Corporation. 1995.
3. The I<sup>2</sup>C bus and how to use it. Philips. 1997.
4. Крамарухин Ю. Е. Приборы для измерения температуры.
5. Спектор С. А. Электрические измерения физических величин/ Энергия. 1987 .
6. Нуберт Г. П. Измерительные преобразования неэлектрических величин/ Энергия. 1970.
7. Новиков Ю. В., Калашиников О. А., Гуляев С. В. Разработка устройств сопряжения / М.: Эком. 1997.
8. Томпкинс У., Уэбстер Дж. Сопряжение датчиков и устройств ввода данных с компьютерами IBM PC / М.: Мир., 1992.
9. Романов С. Е. 8-разрядные МК фирмы “Motorola”. Электроника и радиотехника.
10. Романов С. Е. Интегральные измерители давления. Электроника и радиотехника / № 9. 1998.
11. MPX series pressure detectors. Motorola.
12. Data Book. Analog Devices.
13. Data Sheets / Dallas Semiconductors.
14. MC68HC11E9. HCMOS single chip microcontroller. Advanced information / Motorola.
15. MC68HC11EVBU. Evaluation board. User's manual / Motorola.

16. *Ан Лей*. Сопряжение ПК с внешними устройствами / Пер. с англ. М.: ДМК Пресс, 2001.
17. Data Sheets /Analog Devices.
18. The SPI bus and how to use it / Motorola.
19. MicroLAN interface data sheet / Dallas Semiconductors.
20. CAN interface data sheet / Bosh.

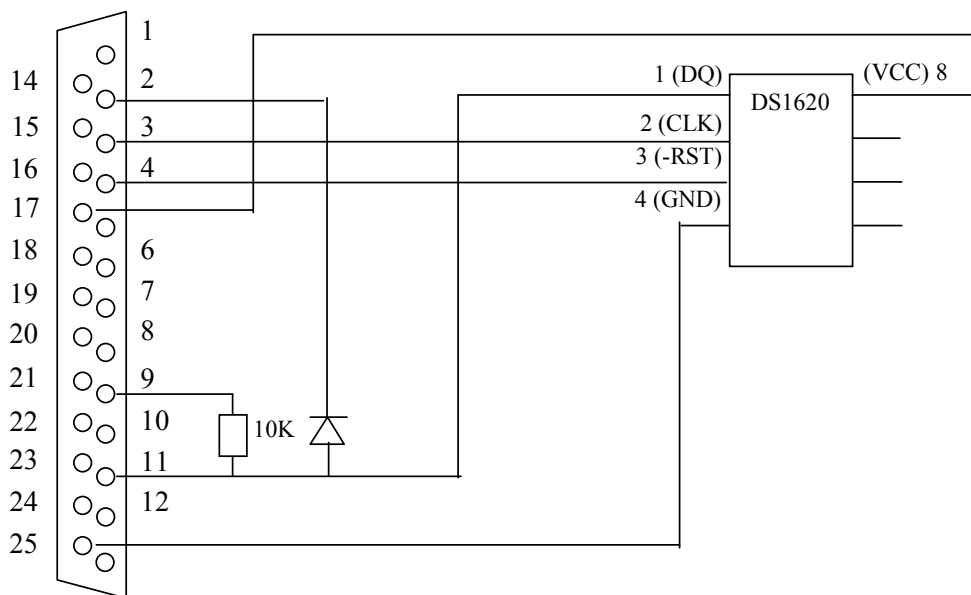
## ПРИЛОЖЕНИЕ 1

Схема подключения адаптера последовательной связи КР580ВВ51 в комплекте “Микролаб”.



## ПРИЛОЖЕНИЕ 2

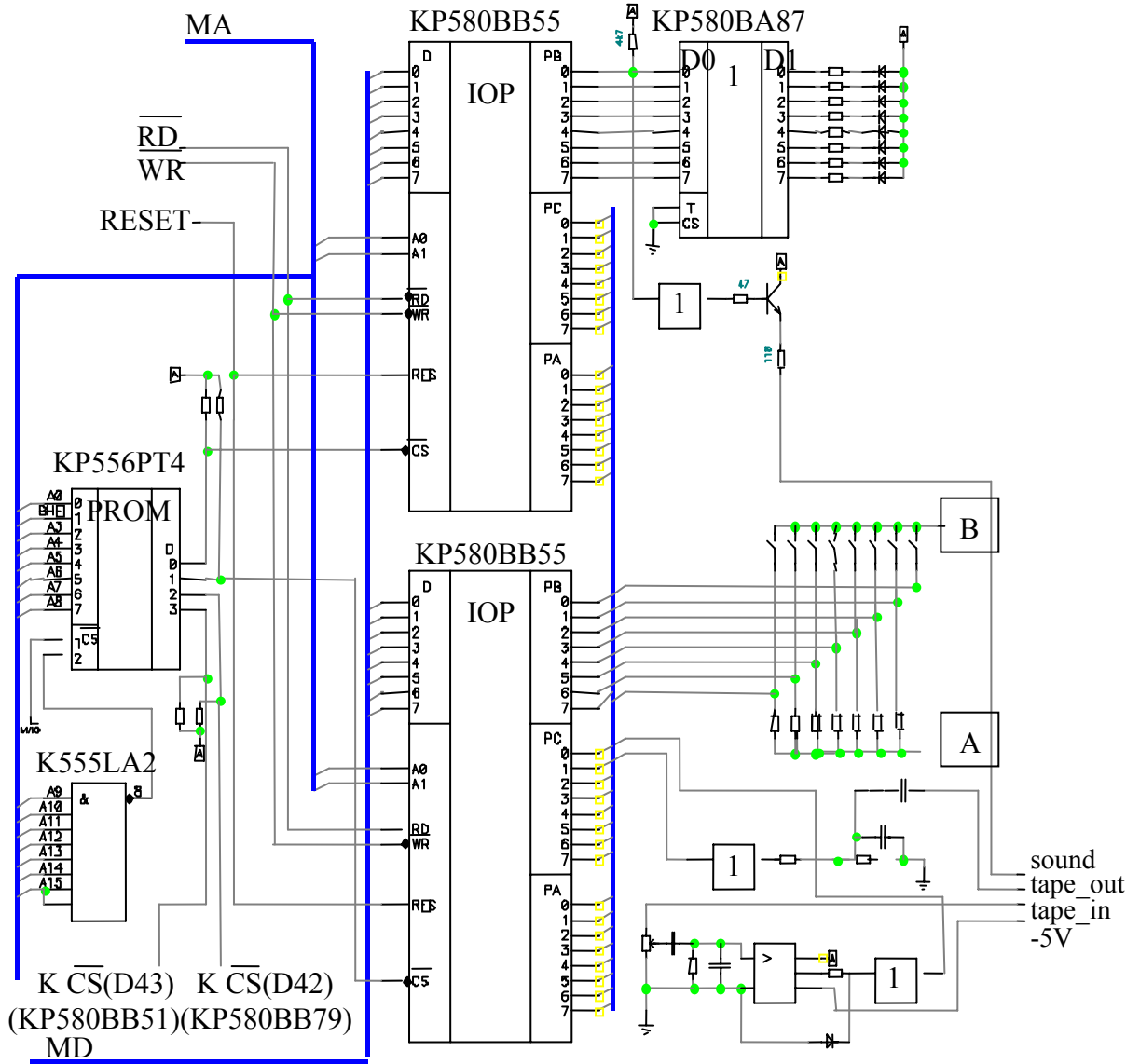
Схема подключения электронного датчика температуры DS1620 к LPT порту компьютера.





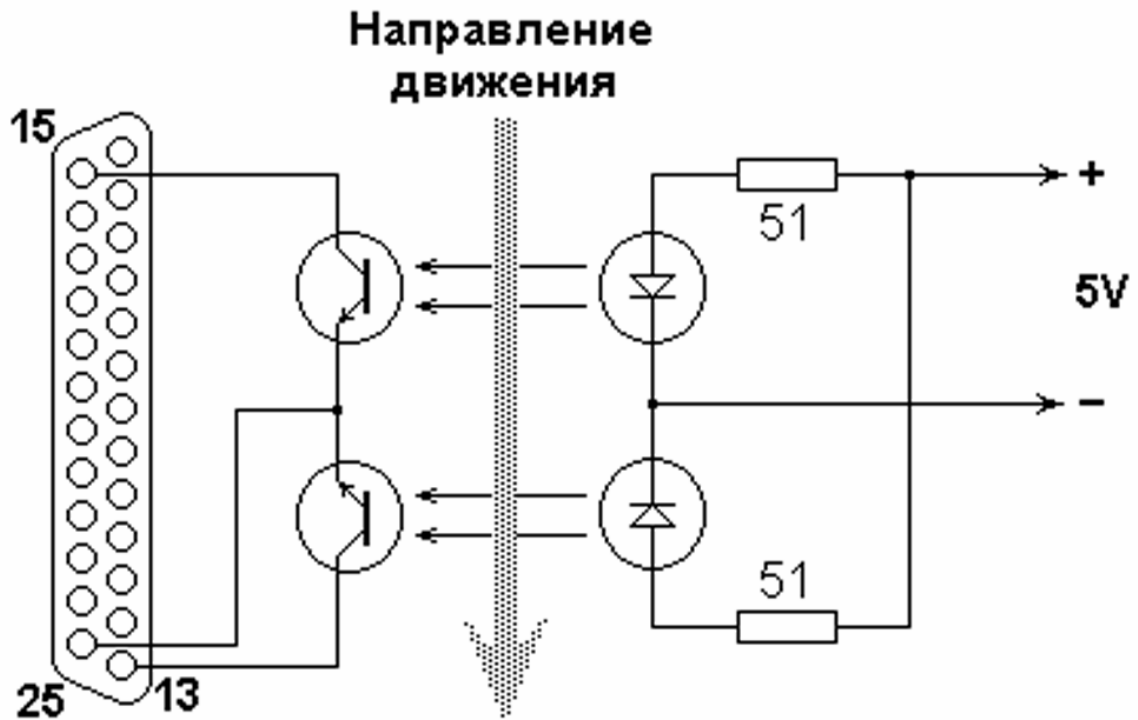
### ПРИЛОЖЕНИЕ 3

Схема подключения параллельного программируемого интерфейса (ППИ) в комплекте “Микролаб”.



## ПРИЛОЖЕНИЕ 4

Схема измерения времени движения тела вдоль наклонной плоскости.



Учебное издание

## **ИЗУЧЕНИЕ ИНТЕРФЕЙСОВ МИКРОПРОЦЕССОРНЫХ СИСТЕМ**

Методические указания  
к лабораторному практикуму

Для студентов специальностей  
G 31 04 02 “Радиофизика”,  
G 31 04 03 “Физическая электроника”

**А в т о р - с о с т а в и т е л ь**  
**Шалатонин Иван Алексеевич**

В авторской редакции  
*Технический редактор Г. М. Романчук*  
*Корректор Р. П. Кадырко*

Ответственный за выпуск *И.А.Шалатонин*

Подписано в печать 2002. Формат 60×84/16. Бумага офсетная. Гарнитура Таймс.  
Усл. печ. л. Уч.- изд. л. Тираж экз. Зак.

Белорусский государственный университет.  
Лицензия ЛВ №315 от 14.07.98.  
220050, Минск, проспект Франциска Скорины, 4.

Отпечатано на копировально-множительной технике  
факультета радиофизики и электроники БГУ  
220064, Минск, ул. Курчатова, 5.