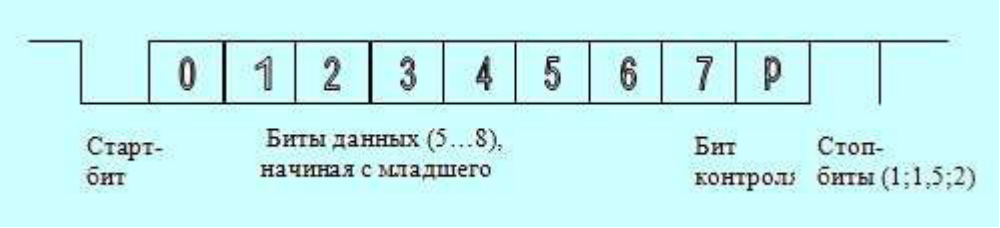
Последовательные интерфейсы

Последовательный интерфейс для передачи данных в одну сторону использует одну сигнальную линию, по которой информационные биты передаются друг за другом последовательно. Такой способ передачи определяет название интерфейса и порта, его реализующего. Линией связи может быть двухпроводная линия, радио- и телефонный канал и др. Информация кодируется уровнем тока, напряжения, частотой сигнала и т.д.

Примеры: последовательный коммуникационный порт (COM-порт), последовательные шины USB и FireWire, PCI Express, интерфейс SATA, интерфейсы локальных и глобальных сетей.

Форматы последовательной передачи данных

Различают синхронную и асинхронную последовательную передачу данных. При асинхронной передаче данные обрамляются стартовым и стоповыми битами, в промежутках между данными по линии связи передаются стоп-биты. При синхронной передаче в отсутствие данных по линии связи циркулируют заранее определенные синхрослова.



Формат асинхронной передачи

Скорость передачи определяется в бит/с. Последовательный интерфейс на физическом уровне может иметь различные реализации, различающиеся способами передачи электрических сигналов, максимальными допустимыми расстояниями и скоростями передачи.

Последовательный порт в персональных компьютерах построен по стандарт RS-232C. В промышленной автоматике широко применяется RS-485, а также RS-422A, встречающийся и в некоторых принтерах.

Стандарт RS-232C определяет 25 линий.

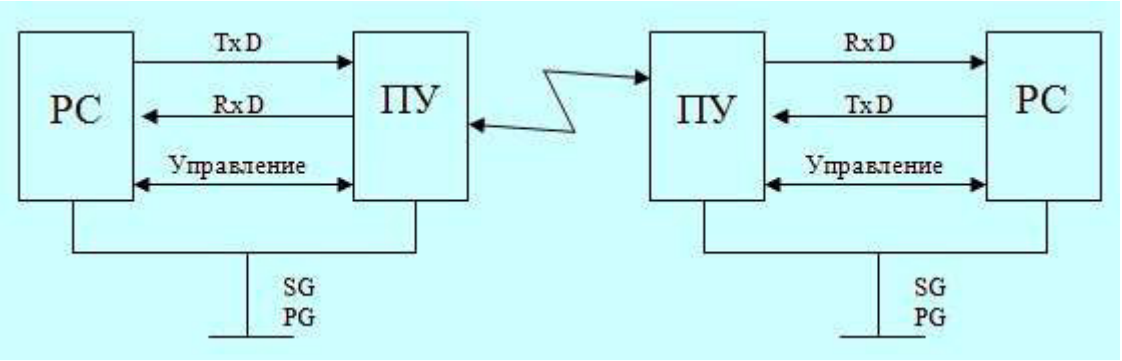


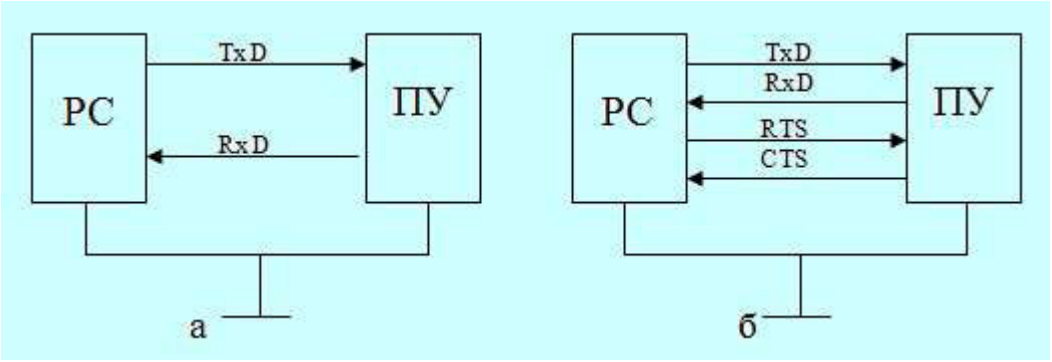
Схема соединения по стандарту RS-232C

На практике используется 10:

- TxD – передаваемые данные,
- RxD – получаемые данные,
- SG, PG – сигнальная и защитная земля.
- 5 линий управления:
- RTS (Request To Send) – выходной сигнал запроса передачи от последовательного порта компьютера (PC) к периферийному устройству (ПУ) (1 – готовность к передаче, 0 – PC в режиме приема);
- CTS (Clear To Send) – входной сигнал готовности ПУ к приему данных (0 – готовность к приему, 1 – готовность к передаче);
- DTR (Data Terminal Ready) – выходной сигнал готовности PC (0 – PC готов передавать данные, 1 – не готов);
- DSR (Data Set Ready) – входной сигнал готовности ПУ передавать данные (0 – ПУ готово, 1 – не готово);
- DCD (Data Carrier Detected) – входной сигнал обнаружения несущей (частоты);
- RI (Ring Indicator) – входной сигнал индикатора вызова (звонка).

Логической единице соответствует отрицательный уровень напряжения, логическому нулю – положительный.

Существуют различные процедуры обмена. В дуплексном режиме, когда одновременно возможны как прием, так и передача данных, протокол обмена обычно реализуется программно. Имеются управляющие коды XON/XOFF, которые разрешают или запрещают передачу данных. В полудуплексном режиме, когда в один и тот же момент возможен либо прием, либо передача данных, используются две управляющие линии RTS и CTS (рисунок б). Когда передача данных происходит в одну сторону, например на принтер, от ПУ требуется только сигнал готовности CTS.



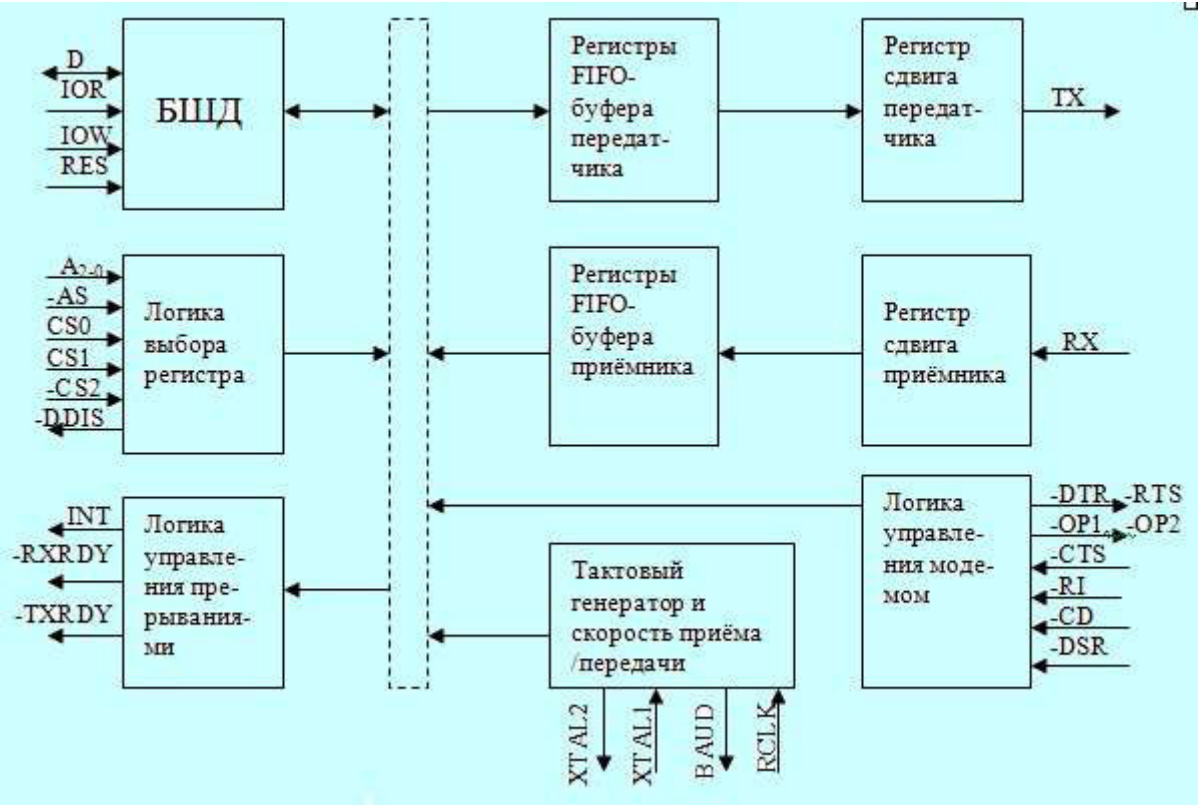
. Варианты соединения компьютера и ПУ при а–дуплексный и б–симплексный обмен

Примером последовательного интерфейса является “токовая петля”, или ***ИРПС*** (интерфейс радиальный, последовательный). Физическая линия связи – четырехпроводная дуплексная линия постоянного тока. Логической единице соответствует протекание тока 20 (40) мА, а логическому нулю – его отсутствие. Интерфейс используется для асинхронной передачи данных на расстояние 0…500 м со скоростью 9600 бит/с и ниже.

Последовательный порт UART 16550

Преобразование параллельного кода в последовательный для передачи и обратное преобразование при приеме данных выполняют специализированные микросхемы UART (Universal Asynchronous Receiver-Transmitter) – универсальный асинхронный приемопередатчик. Эти же микросхемы формируют и обрабатывают управляющие сигналы интерфейса. Последовательные порты IBM PC базируются на микросхемах, совместимых на уровне регистров с UART Intel 8250 – 8250/16450/16550A. Это семейство представляет собой усовершенствования начальной модели, направленные на повышение быстродействия, снижение потребляемой мощности и загрузки процессора при интенсивном обмене.

Микросхема 16550A позволяет вести прием/передачу последовательного кода со скоростью от 50 бит/с до 1,5 Мбит/с (при входной тактовой частоте 24 МГц), содержит 16-байтные очереди (FIFO) передатчика и приемника, регистр ошибок при приеме, имеет четыре выбираемых уровня прерывания по заполнению буфера FIFO приемника, стандартный интерфейс с модемом, позволяет использовать прямой доступ в память (DMA) . Совместимость с этой микросхемой обеспечивает большинство универсальных микросхем контроллеров портов ввода/вывода, входящих в чипсеты современных системных плат персонального компьютера.



Структурная схема UART

Назначение выводов приведено в таблице, причем знак минуса перед названием вывода обозначает его активный нулевой уровень.

Назначение выводов UART 16550A

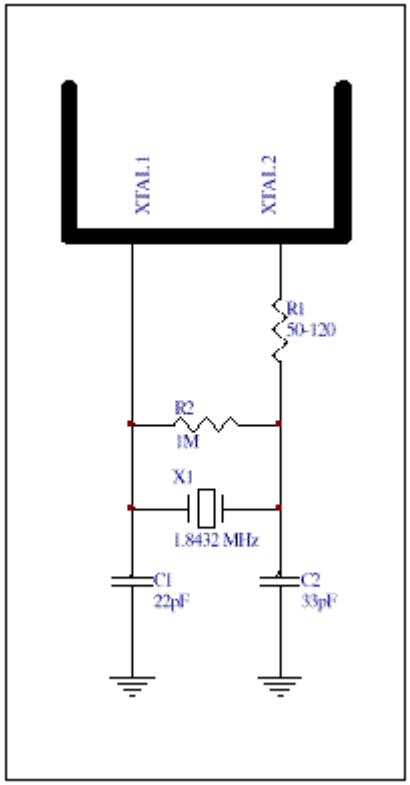
Вывод	Назначение
D7-0	Двунаправленная шина данных
A2-0	Адресные входы для выбора внутренних регистров
-IOR,IOR	Стробы чтения данных
-IOW, IOW	Стробы записи данных
-AS	Строб адреса (на входах A2-0 находится адрес)
CS_0, CS_1, $-CS_2$	Выбор кристалла
INT	Выходной сигнал запроса прерывания
-RxRDY	Выходной сигнал готовности приемника
-TxRDY	Выходной сигнал готовности передатчика
-BAUDOUT	Выход генератора скорости приема/передачи. Должны быть соединены выводы RCLK и BAUDOUT, если прием осуществляется с той же скоростью, что и передача
RCLK	Вход тактирования приемника
-DDIS	Выходной сигнал запрещения обращения

-OP1, -OP2	Выходные биты, определяемые пользователем (биты 2 и 3 MCR)
$XTAL_1$, $XTAL_2$	Выводы для подключения кристалла осциллятора
CD, CTS, DSR, DTR, RTS, RI	Выводы связи с модемом
RX	Принимаемые данные
TX	Передаваемые данные
RESET	Сброс

Микросхема UART 16550A с программной точки зрения представляет собой набор регистров, доступ к которым определяется адресом A2-0 и значением бита 7 регистра LCR. В схеме 8250 отсутствует регистр управления FCR и все возможности DMA и FIFO (и соответствующие биты в других регистрах). Адреса регистров и их содержимое после сброса приведены в таблице.

Адрес A2-0	Бит LCR7	Регистр	Назначение	Чтение/запись R/W	После сброса
000	0	THR	Данные передатчика	W	xxh
000	0	RHR	Данные приемника	R	xxh
000	1	DLL	Младший байт делителя	R/W	xxh
001	1	DLH	Старший байт делителя	R/W	xxh
001	0	IER	Регистр разрешения прерываний	W	00h
010	x	ISR	Регистр состояния прерываний	R	01h
010	x	FCR	Регистр управления FIFO	W	00h
011	x	LCR	Регистр управления линией	W	00h
100	x	MCR	Регистр управления модемом	W	00h
101	x	LSR	Регистр состояния линии	R	60h
110	x	MSR	Регистр состояния модема	R	x0h
111	x	SPR	Рабочий регистр	R/W	ffh

Схема подключения *кварцевого осциллятора* частотой 1,8432 МГц (используется в IBM PC) приведена на рисунке.



Генератор делит входную частоту на делитель от 1 до 2^{16}-1. Схема последовательного порта делит частоту еще на 16. Необходимое значение делителя D можно вычислить по формуле

$$D = f_{вх}/(16*V),$$

где $f_{вх}$ – входная частота в Герцах (частота осциллятора), V – скорость приема/передачи в бит/с.

Форматы регистров управления и состояния

Регистр THR – промежуточный регистр данных передатчика. Данные, записанные в этот регистр, будут пересланы в выходной сдвигающий регистр (когда он будет свободен), из которого поступят на выход при наличии разрешающего входного сигнала CTS. Бит 0 передается и принимается первым. При длине посылки менее 8 бит старшие биты игнорируются.

Регистр RHR – промежуточный регистр принимаемых данных. Данные, принятые входным сдвигающим регистром, помещаются в регистр RHR, откуда могут быть считаны процессором. При длине посылки менее 8 бит старшие биты в регистре имеют нулевое значение.

Регистр LCR регистр управления линией, определяет формат передаваемых (принимаемых) данных.

Назначение битов регистра следующее:

- биты 1-0 – длина символа, 00 - 5 бит, 01 - 6 бит, 10 - 7 бит, 11 - 8 бит;
- бит 2 - число стоп-битов, 0 - 1, 1 - 1.5, если длина символа равна 5, иначе 2;
- бит 5-3 – управление контрольным битом, xx0 – нет бита, 001 – четность, 011 – нечетность, 101 – «1», 111 - «0»;
- бит 6 - 1- установка перерыва (посылка нулей в качестве сигнала отдаленной станции);
- бит 7 – 1- доступ к делителю.

Обычно биты 6-7 сброшены в 0. Остальные описывают значения, определяемые протоколом обмена.

Регистр LSR – регистр состояния линии. Единичное значение бита:

- бит 7 – ошибка принятых данных в режиме FIFO (буфер содержит хотя бы один символ, принятый с ошибкой формата, паритета или обрывом). Сбрасывается чтением данных. В не FIFO режиме всегда 0;
- бит 6 – регистр передатчика пуст (нет данных для передачи ни в сдвиговом регистре, ни в буферных);
- бит 5 – регистр передатчика готов принять байт для передачи (TxRDY);
- бит 4 – индикатор обрыва линии (вход приемника находится в состоянии 0 не менее чем время посылки символа);
- бит 3 – ошибка стоп-бита;
- бит 2 – ошибка контрольного бита;
- бит 1 – ошибка переполнения (прием начат до того, как предыдущий символ (который теряется) выгружен из сдвигающего регистра);
- бит 0 – принятые данные готовы (RxRDY). Сброс – чтение данных.

Индикаторы ошибок – биты 1-4 – сбрасываются после чтения регистра LSR. В режиме FIFO ошибки хранятся вместе с каждым символом. В LSR они устанавливаются и вызывают прерывание в тот момент, когда символ с ошибкой первый в очереди на считывание. В случае обрыва в буфер FIFO заносится только один «обрывной» символ.

Регистр IER – регистр разрешения прерываний. Единичное значение бита разрешает прерывание от соответствующего источника:

- бит 3 – по изменению состояния модема (любой из линий CTS, DSR, RI, DCD);
- бит 2 – по обрыву/ошибке линии;
- бит 1 – по завершению передачи;
- бит 0 – по приему символа.

Биты 7-4 не используются.

Регистр ISR – регистр состояния прерывания и признак режима FIFO. Назначение битов регистра ISR:

- биты 7-6 – признак режима FIFO, 11 – режим 16550A, 10 – режим 16550 (схема имела ошибки), 00 – нет режима FIFO;
- биты 5-4 не используются;
- бит 3 – прерывание по тайм-ауту (не в режиме FIFO);
- биты 2-1 – причина прерывания с наивысшим приоритетом (не в режиме FIFO):
  11 – ошибка/обрыв линии, сброс чтением регистра LSR;
  10 – принят символ, сброс чтением данных;
  01 – передан символ, сброс записью данных;
  00 – изменение состояния модема, сброс чтением регистра MSR;
- бит 0 – признак необслуженного запроса прерывания, 1 – нет запроса, 0 – есть запрос.

Для упрощения программного анализа UART выстраивает внутренние запросы прерывания по 4-уровневой приоритетной системе. Порядок приоритетов (по убыванию) следующий:

изменение состояния линии,

RxRDY (принят символ или тайм-аут),

TxRDY (освобождение регистра передатчика),

изменение состояния модема.

Состояние тайм-аута фиксируется тогда, когда за 4-кратный интервал времени передачи символа не передано (или не принято) ни одного символа (хотя в буфере имеется).

При возникновении условий прерывания UART указывает на источник с высшим приоритетом до тех пор, пока он не будет сброшен соответствующей операцией. Только после этого будет выставлен запрос с указанием следующего источника.

В режиме FIFO индикатором тайм-аута является код ISR3-1 = 110, сбрасывается чтением регистра данных приемника. Остальные причины прерывания кодируются, как рассмотрено выше, но с нулевым битом 3.

Регистр FCR – регистр управления FIFO. Назначение бит:

- биты 7-6 – уровень заполнения FIFO-буфера, при котором вырабатывается прерывание «принят символ»:
  00 – 1 байт, 01 – 4 байта, 10 – 8 байт, 11 – 16 байт;
- биты 5-4 = 0;
- бит 3 – разрешение операций DMA, 0 – операции DMA типа 0, 1 – типа 1;
- бит 2 – сброс счетчика FIFO-передатчика. Запись единицы в этот бит приводит к сбросу счетчика FIFO, сдвигающий регистр не сбрасывается;
- бит 1 - сброс счетчика FIFO-приемника. Запись единицы в этот бит приводит к сбросу счетчика и буфера FIFO, сдвигающий регистр не сбрасывается;
- бит 0 – разрешение (единицей) режима FIFO для передатчика и приемника, буферы FIFO при этом очищаются. Если бит 0 равен 0, то другие биты регистра FCR не программируются (должны быть равны 0).

Поясним режимы DMA. При DMA типа 0 (FCR.0 = 0 или FCR.0 = 1, но FCR.3 = 0) активный уровень сигнала TxRDY формируется, если нет символов в буфере FIFO и в регистре передатчика, активный уровень сигнала RxRDY – если есть хотя бы один символ в буфере FIFO приемника.

В случае DMA типа 1 активный уровень сигнала TxRDY формируется, когда хотя бы один и более регистров буфера FIFO передатчика пусты, активный уровень сигнала RxRDY – когда достигнут уровень заполнения буфера.

Регистр MCR имеет адрес порта на 4 больше, чем базовый адрес используемого коммуникационного канала. Вот значение его битов:

- биты 7-5 - всегда 0;
- бит 4 - 1 - выход UART замкнут на вход (диагностический режим);
- бит 3 - выходной сигнал OP2, в диагностическом режиме поступает на вход MSR.7;
- бит 2 - выходной сигнал OP1, в диагностическом режиме поступает на вход MSR.6;
- бит 1- управление выходом RTS, 1 - RTS активен;
- бит 0 - управление выходом DTR, 1 - DTR активен.

Обычно установлены биты 0 и 1 регистра управления модемом, а остальные равны 0. Бит 2=0, за исключением случаев, когда производитель модема предназначил его для специального использования. Бит 3 установлен только в случае, когда используются прерывания. Наконец, бит 4 предоставляет возможность тестирования коммуникационных программ без установления реальной связи.

Формат ***регистра состояния модема MSR***:

биты 7-4 – состояние (1 – активный уровень),) выводов, соответственно DCD, RI, DSR, CTS;
биты 3-0 – изменение состояния (1 – произошло изменение) выводов, соответственно DCD, RI, DSR, CTS

Работа с последовательным портом на уровне регистров

BIOS поддерживает 4 порта коммуникации. Их базовые адреса хранятся в ячейках 0:0400 для COM1, 0:0402 для COM2 и т.д. (Базовый адрес это двухбайтовый адрес порта, который является младшим из группы адресов портов, дающих доступ к UART.) Базовый адрес может иметь значение 3F8H, 2F8H, 3E8H (3E0H, 338H), 2E8H (2E0H, 238H). При инициализации BIOS проверяет наличие портов именно в этом порядке и, соответственно, присваивает обнаруженным портам логические имена COM1, COM2, COM3 и COM4. Операционная система MS DOS поддерживает только два COM-порта.

Для удобства, мы в дальнейшем будем всегда нумеровать регистры 3FxH, но все сказанное в равной степени применимо и к регистрам других портов.

Доступ к 12 регистрам UART осуществляется через восемь адресов портов с номерами 3F8H - 3FFH.

Независимо от того, занимаемся ли мы вводом или выводом, как минимум 4 регистра микросхемы 16550A должны быть инициализированы для операций обмена. Это регистры делителя скорости обмена, регистр управления линией LCR и регистр разрешения прерывания IER.

1. Инициализация скорости обмена

Делитель скорости обмена – это число, на которое надо разделить частоту системных часов (1,18432 МГц), чтобы получить желаемую скорость обмена. Например, для скорости обмена 1200 бит/с делитель скорости обмена должен быть равен 96, поскольку

D= fвх/(16*V) = 1184320/(16*1200) =96.

Чем больше делитель, тем меньше скорость обмена. Скорости обмена 300 и меньше требуют двухбайтного числа для делителя. Старший байт посылается в 3F9H, а младший в 3F8H. В обоих случаях бит 7 регистра LCR должен быть предварительно установлен в 1. Вот некоторые значения, требуемые для обычных скоростей обмена:

Скорость обмена	3F9H	3F8H	Скорость обмена	3F9H	3F8H
110	04H	17H	4800	00H	18H
300	01H	80H	9600	00H	0CH
600	00H	C0H	19200	00H	06H
1200	00H	60H	38400	00H	03H
1800	00H	40H	57600	00H	02H
2400	00H	30H	115200	00H	01H
3600	00H	20H			

Нужно устанавливать регистры скорости обмена первыми, так как они единственные, которые требуют, чтобы был установлен бит 7 в регистре контроля линии. После этого надо изменить содержимое регистра контроля линии, сбрасывая 7-й бит, чтобы все остальные доступы к регистрам были правильными.

2. Инициализация регистра LCR

Значение битов регистра управления линией, адрес порта которого равен 3FBH, следующее:

биты 1-0 – длина символа, 00 - 5 бит, 01 - 6 бит, 10 - 7 бит, 11 - 8 бит;

бит 2 - число стоп-битов, 0 - 1, 1 - 1.5, если длина символа равна пяти, иначе 2;

бит 5-3 – управление контрольным битом, xx0 – нет бита, 001 – четность, 011 – нечетность, 101 – «1», 111 – «0»;

бит 6 - 1- установка перерыва (посылка нулей в качестве сигнала отдаленной станции);

бит 7 – 1- доступ к делителю.

3. Регистр разрешения прерывания

Значение битов регистра разрешения прерываний, адрес порта которого равен 3F9H, следующее:

бит 3 – по изменению состояния модема (любой из линий CTS, DSR, RI, DCD);

бит 2 – по обрыву/ошибке линии;

бит 1 – по завершению передачи;

бит 0 – по приему символа.

Если прерывания не используются, нужно произвести запись 0 в регистр разрешения прерывания, чтобы быть уверенным, что прерывания запрещены. Регистр идентификации прерывания можно игнорировать.

Инициализация остальных регистров связана с модемами. Модемы нужны только для связи с удаленными устройствами, а не для управления близлежащими устройствами, такими как последовательный принтер.

В данном примере порт COM1 инициализируется для скорости обмена 1200 бод, 7-битных данных, контроля на четность и одного стоп-бита, разрешаем прерывания по приему и передаче символа.

```
; получаем базовый адрес COM1
Addr equ 3F8H
MOV  DX,Addr      ;получаем базовый адрес COM1
; инициализируем регистры делителя скорости обмена на 1200 бод
MOV  AL,10000000B    ;устанавливаем бит 7 регистра управления линией
MOV  DX,Addr+3
OUT  DX,AL           ;посылаем байт
MOV  DX,Addr+1      ;указываем на старший байт делителя скорости обмена
MOV  AL,0            ;старший байт для 1200 бод
OUT  DX,AL           ;посылаем старший байт для 1200 бод
MOV  DX,Addr         ;указываем на младший байт делителя
MOV  AL,60H          ;младший байт делителя для 1200 бод
OUT  DX,AL           ;посылаем младший байт
; инициализируем регистр управления линией
MOV  AL,01010b       ;длина данных 7 битов, 1 стоп-бит, бит четности
MOV  DX,Addr+3       ;указываем на регистр контроля линии
OUT  DX,AL           ;посылаем инициализационное значение
; инициализируем регистр разрешения прерывания
MOV  DX,Addr+1       ;указываем на регистр разрешения прерывания
MOV  AL,11b         ;разрешаем прерывания по приему и передаче символа
OUT  DX,AL           ;посылаем байт
```

Передача данных проще чем прием, поскольку программа имеет полный контроль над составом данных и скоростью, с которой они должны посылаться.

Когда байт данных помещается в регистр хранения передатчика, то он автоматически выводится в последовательный канал через регистр сдвига передатчика, который отправляет данные в последовательный канал. Нет необходимости в импульсе бита строба, как это делается в случае параллельного адаптера.

Бит 5 регистра статуса линии показывает: свободен ли регистр хранения передатчика для приема данных. Регистр постоянно проверяется до тех пор, пока бит 5 не станет равным 1.

После этого в регистр хранения передатчика посылается очередной байт из того места, откуда они берутся. В процессе передачи бит 5 равен 0 и только тогда, когда он опять станет равным 1, в регистр хранения передатчика может быть послан следующий символ. Этот процесс повторяется до тех пор, пока это нужно.

В следующем примере даны основные понятия об этой процедуре. Конечно, она может быть сделана необычайно сложной (в частности, программирование связи требует особо тщательных процедур обнаружения ошибок и восстановления при сбоях). В примере предполагается, что коммуникационный порт уже инициализирован. Первая часть – это цикл проверки ошибок и приема символов.

```
;---ждем, пока все будет готово для посылки символа
KEEP_TRYING:  MOV  DX,Addr+5   ;базовый адрес
                  ;указываем на регистр статуса линии
   IN   AL,DX             ;получаем байт статуса
   TEST AL,00011110B      ;проверяем на ошибку
   JNZ  ERROR_ROUTINE     ;если есть, то на процедуру обработки
   TEST AL,00000001B      ;проверяем, получены ли данные
   JNZ  RECEIVE           ;если да, то на процедуру приема
   TEST AL,00100000B      ;проверяем готовность к передаче
   JZ   KEEP_TRYING       ;если нет, то возвращаемся назад
;---передаем символ, принимаемый с клавиатуры
   MOV AL, …
   MOV  DX,Addr              ;адрес регистра хранения передатчика
   OUT  DX,AL             ;посылаем символ
   JMP  SHORT KEEP_TRYING ;возвращаемся к началу цикла
```

Получение данных

Коммуникационная программа готова принимать данные, как только инициализируется коммуникационный порт и установится связь с удаленной станцией.

В зависимости от сложности используемого протокола обмена, принимаемые данные могут требовать простой или сложной обработки. Может быть получен один из набора управляющих кодов. Те из них, которые являются ограничителями данных, чаще обнаруживаются при синхронном обмене.

При получении данных без использования коммуникационного прерывания программа должна постоянно проверять регистр статуса линии, адрес порта которого на 5 больше базового адреса используемого коммуникационного адаптера.

Бит 0 этого регистра будет равен нулю до тех пор, пока не будет получен символ в регистр данных приемника. Когда бит 0 становится равным 1, то надо немедленно считать его из регистра, чтобы на него не наложился следующий принимаемый символ. После того как символ считан, бит 0 опять становится равным 0 и остается таковым, пока не прибудет новый символ.

Следующий пример частично дублирует содержимое предыдущего раздела, относящегося к передаче символов. Как и в том случае, код начинается с бесконечного цикла. Можно объединить эти 2 процедуры с процедурами инициализации для создания законченной процедуры ввода/вывода через коммуникационный канал.

```
KEEP_TRYING:   MOV  DX,Addr+5   ;базовый адрес
                    ;указываем на регистр статуса линии
   IN   AL,DX          ;получаем байт статуса
   TEST AL,00011110B   ;проверяем на ошибку
   JNZ  ERROR_ROUTINE  ;если да, то на обработку ошибки
   TEST AL,00000001B   ;проверяем, получены ли данные
   JNZ  RECEIVE        ;на процедуру приема данных
   TEST AL,00100000B   ;проверяем готовность к передаче
   JZ   KEEP_TRYING    ;если нет, то к началу цикла
   (здесь расположена процедура передачи)
;---получаем данные и выводим их на экран
RECEIVE:   MOV  DX,Addr        ;базовый адрес
   IN   AL,DX          ;читаем полученный символ
   CMP  AL,19          ;проверка на XOFF
   JE   XOFF_ROUTINE   ;
   (и т.д.)

   JMP  SHORT KEEP_TRYING   ;возвращаемся на начало цикла
```