

Режим 0. Синхронный последовательный порт.

Передача

В нулевом режиме последовательный порт работает как обыкновенный сдвиговый регистр. Это позволяет использовать последовательный порт для увеличения количества внешних ножек микросхемы. Использование сдвиговых регистров для этой цели показано на **рисунке 1** и **2**. Передача по последовательному порту начинается после записи байта в регистр данных SBUF. Временная диаграмма сигнала, вырабатываемого последовательным портом микроконтроллера при передаче восьми бит данных приведена на **рисунке 2**. Приём байта по последовательному порту начинается после обнуления флага готовности приёмника RI.

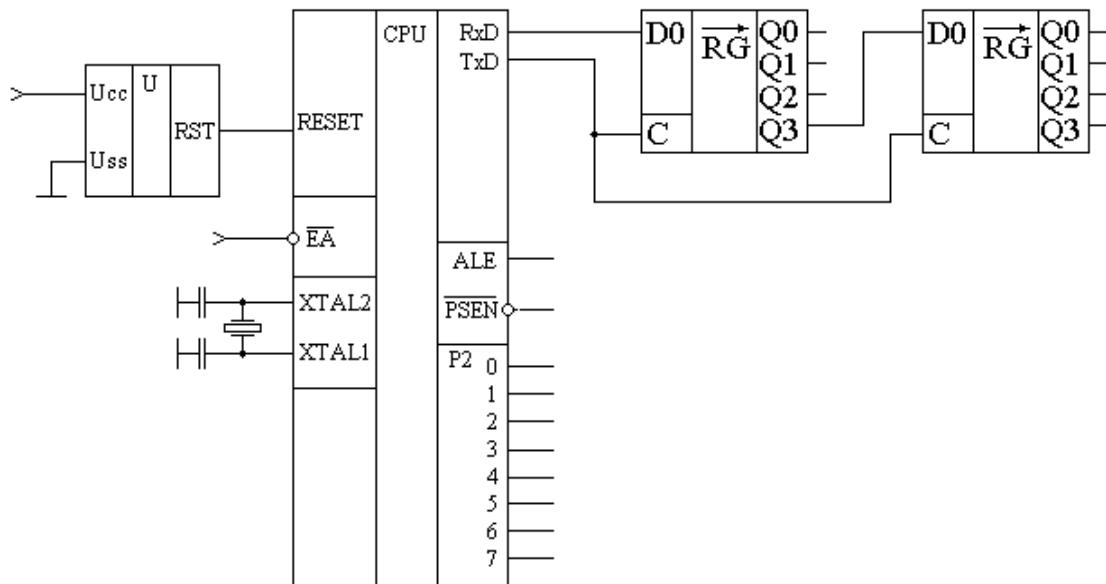


Рисунок 1. Использование нулевого режима работы последовательного порта в качестве расширителя портов.

Для осуществления передачи байта данных достаточно занести его в буфер данных SBUF, как это показано в примере:

```

MOV SCON, #0      ;Настроить последовательный порт на передачу в синхронном режиме
MOV SBUF, A        ;Передать содержимое аккумулятора по последовательному порту
JNB TI, $          ;Подождать окончания передачи
MOV SBUF, #56H    ;Передать по последовательному порту число 56h
JNB TI, $          ;Подождать окончания передачи

```

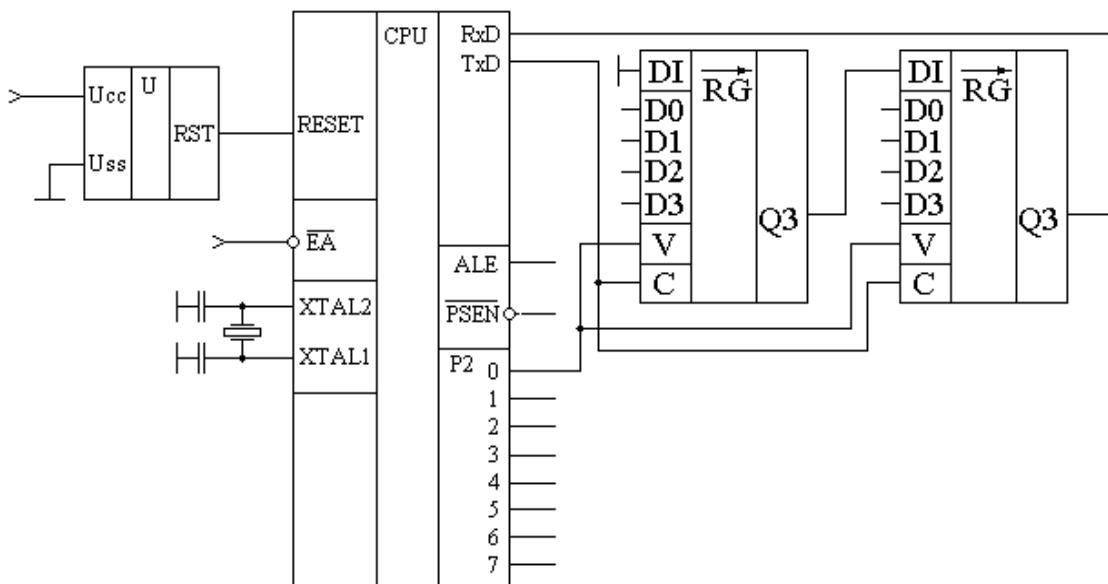


Рисунок 2. Использование нулевого режима работы последовательного порта для ввода информации.

Прием

Для осуществления приёма байта данных достаточно настроить порт на приём в синхронном режиме работы и обнулить флаг приёма RI, как это показано в примере, приведённом на **рисунке 3**.

```
;Настроить режим работы последовательного порта-----
mov SCON,#00010000b      ;настроить последовательный порт на нулевой режим работы
;|||||||+
;|||||||+-----Обнулить флаг приёмника RI
;|||||||+-----Обнулить флаг передатчика TI
;|||||||+-----Обнулить девятый бит приёмника RB8
;|||||||+-----Обнулить девятый бит передатчика TB8
;|||+-----Разрешить работу приёмника
;||+-----В синхронном режиме не имеет значения
;+-----Включить синхронный режим работы последовательного порта
;Так как предыдущая команда обнуляет флаг RI то с этого момента начинается приём байта
JNB RI, $                ;Подождать окончания приёма байта по последовательному порту
MOV A, SBUF               ;и скопировать его в аккумулятор
```

Рисунок 3. Программа считывания одного байта из внешнего регистра по последовательному порту.

Режим 1. Асинхронный восьмиразрядный последовательный порт.

В отличие от нулевого режима работы в первом режиме работы возможен обмен информацией между двумя микроконтроллерами, а не только между микроконтроллером и исполнительными микросхемами. Схема соединения двух микроконтроллеров между собой для обмена информацией приведена на **рисунке 4**. Таким образом может быть построена простейшая многопроцессорная система.

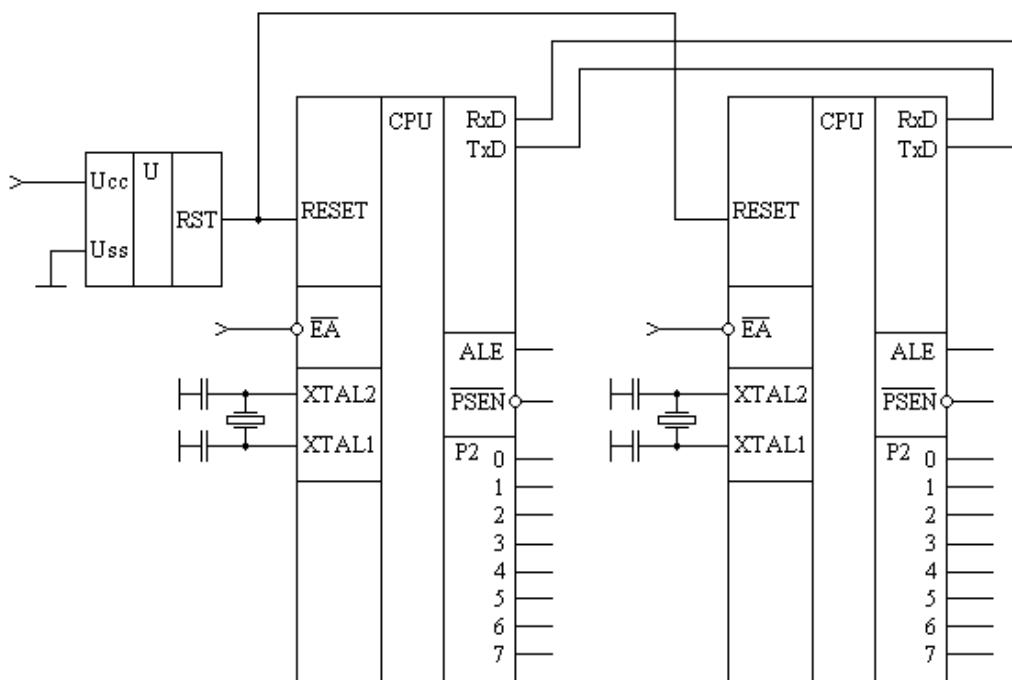


Рисунок 4. Схема обмена информацией между двумя микроконтроллерами по последовательному порту.

В качестве примера работы с последовательным портом приведём подпрограмму инициализации последовательного порта для работы в асинхронном режиме на скорости 9600 бит/с. Настройка режима работы последовательного порта в этой программе производится записью двоичной константы в регистр управления SCON. Выбор конкретных значений бит двоичной константы подробно приводится в комментариях к программе.

Так как для последовательного порта требуется таймер, то, кроме настройки последовательного порта, при помощи регистра выбора режима работы таймеров TMOD настраивается таймер T1. Эта настройка также производится при помощи записи двоичной константы.

Скорость работы последовательного порта настраивается записью константы 0FDh в старший байт таймера T1.

```
;*****  
;НАСТРОЙКА ПОСЛЕДОВАТЕЛЬНОГО ПОРТА  
;*****  
;Настроить режим работы последовательного порта-----  
mov SCON,#01110000b ;настроить последовательный порт на первый режим работы  
;|||||||  
;|||||||+----Обнулить флаг приёмника RI  
;|||||||+----Обнулить флаг передатчика TI  
;|||||||+----Обнулить девятый бит приёмника RB8  
;|||||||+----Обнулить девятый бит передатчика TB8  
;|||+----Разрешить работу приёмника  
;|||+----Проверять ошибку кадра (приём нулевого бита на месте стоп-  
бита)  
;+-----Включить асинхронный режим работы последовательного порта  
  
;Настроить режим работы таймера T1 -----  
anl TMOD,#00001111b ;Подготовить таймер T1 к настройке (таймер T0 не трогать!)  
orl TMOD,#00100000b ;перевести таймер T1 во второй режим работы (таймер T0 не  
трогать!)  
;|||||  
;|||+----Перевести таймер T1 в режим автозагрузки  
;|+----Синхронизироваться от внутреннего генератора  
;+----Запретить управление таймером от ножки INT1  
  
;Настроить таймер на генерацию 3-х микросекундного интервала времени-----  
mov TH0, #fdh ;Загрузить старший байт таймера  
mov TL0, #fdh ;Загрузить младший байт таймера  
setb TR1 ;Включить таймер 1  
  
;*****  
;РАБОТА С ПОСЛЕДОВАТЕЛЬНЫМ ПОРТОМ  
;*****  
JNB RI, $ ;Подождать окончания приёма байта по последовательному порту  
MOV A, SBUF ;и скопировать его в аккумулятор
```

Рисунок 5. Программа приёма одного байта по последовательному порту.

Возможность работы в асинхронном режиме позволяет использовать последовательный порт для связи с универсальным компьютером через его последовательный COM порт. К сожалению уровни последовательного порта микроконтроллера не совпадают с уровнями последовательного порта компьютера, поэтому для подключения микроконтроллера к последовательному порту компьютера используются специализированные микросхемы согласования уровней. Эти же микросхемы обеспечивают защиту микроконтроллера от вывода из строя статическим потенциалом при подключении разъёмов.

Обычно для работы используются только сигнальные цепи COM-порта компьютера. Тем не менее, оставшиеся буферы интерфейсной микросхемы могут быть использованы для контроля питания микроконтроллерной схемы. Типовая схема подключения компьютера к последовательному порту микроконтроллеров семейства MCS-51 с применением микросхемы ADM3202 приведена на **рисунке 6**.

Использование последовательного порта компьютера позволяет не только управлять микроконтроллерным устройством, используя клавиатуру компьютера, но и отображать внутреннюю информацию этого устройства, используя дисплей компьютера. Это значительно расширяет возможности ввода и вывода информации в микроконтроллерных устройствах. В последнее время дополнительно появилась возможность заносить программу во внутреннюю память наиболее современных микроконтроллеров.

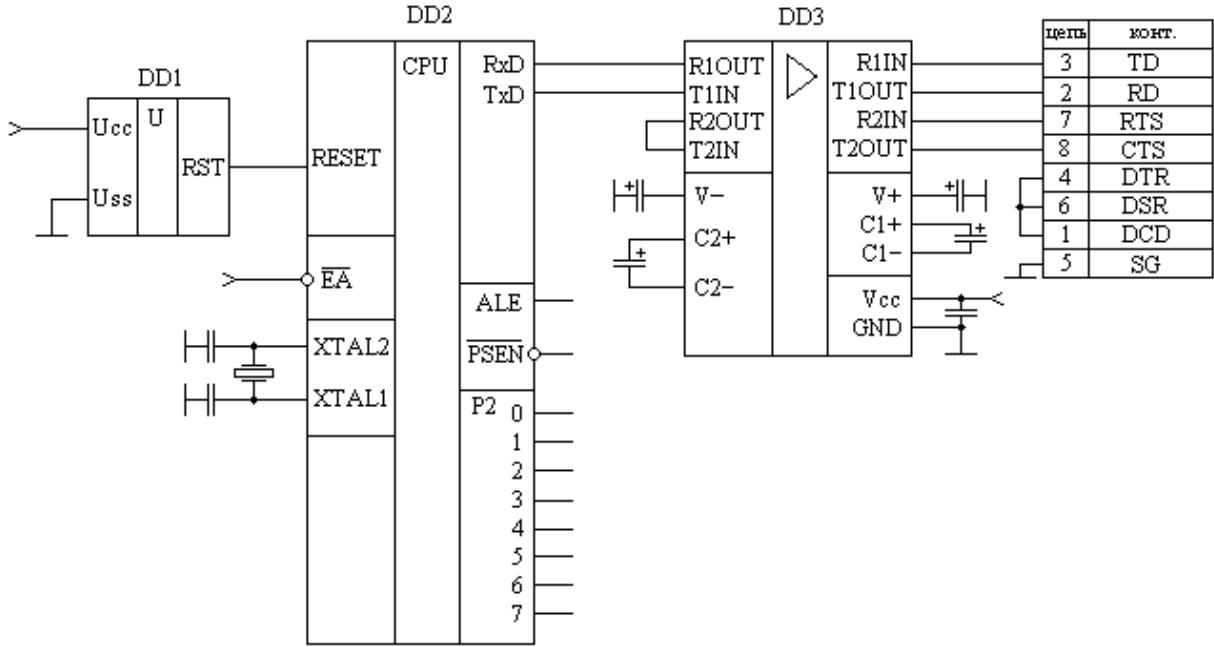


Рисунок 6. Подключение последовательного порта микроконтроллеров семейства MCS-51 к последовательному COM порту компьютера.

Режим 2. Асинхронный девятиразрядный последовательный порт с фиксированной скоростью передачи.

Ещё большие возможности для построения устройств предоставляет девятиразрядный режим работы при реализации многопроцессорных систем. Параллельные порты микроконтроллеров семейства MCS-51 построены по схеме с открытым стоком. Это позволяет объединять несколько выходов передатчиков в одну шину. Такое выполнение выходных каскадов микросхем облегчает построение многопроцессорных систем. В многопроцессорной системе один процессор должен быть главным (*master*), остальные - подчинёнными (*slave*). Естественно команды главного процессора должны восприниматься подчинёнными процессорами, поэтому выход передатчика главного процессора соединяется со входами приёмников подчинённых. Выходы же передатчиков подчинённых процессоров объединяются и подключаются ко входу приёмника главного процессора. Схема примера многопроцессорной системы приведена на **рисунке 7**.

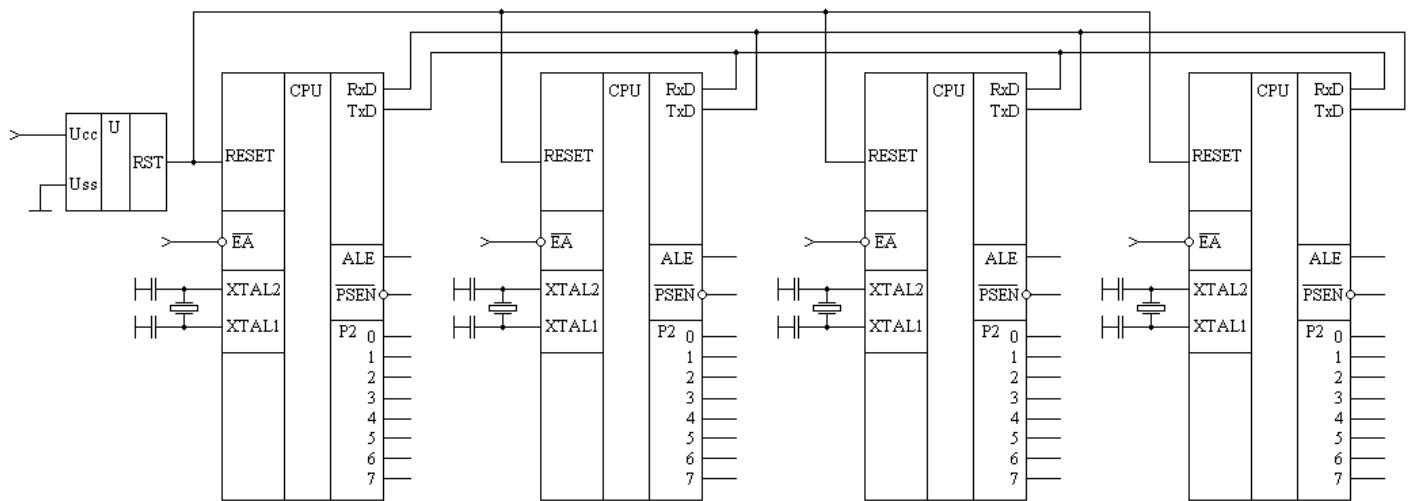


Рисунок 7. Схема соединения нескольких микроконтроллеров между собой по последовательному порту, работающему в асинхронном режиме.

Команды главного процессора могут быть обращены к конкретному подчинённому процессору, поэтому в состав команд включается адрес подчинённого процессора. При работе в шине необходимо уметь отличать адресную информацию от данных. Это можно осуществить при помощи девятого бита. Обычно при передаче адреса в девятый бит

записывают единицу, а в байтах данных и команд - 0. Таким образом микроконтроллер, даже подключившийся к шине позднее остальных, легко может осуществить синхронизацию с многопроцессорной шиной. Временная диаграмма работы многопроцессорной шины приведена на **рисунке 8**.

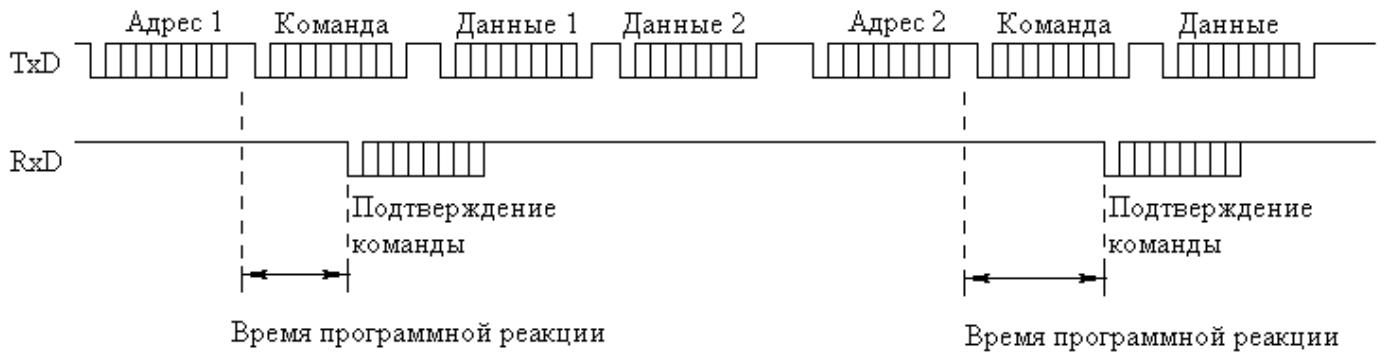


Рисунок 8. Временная диаграмма работы многопроцессорной шины.

При настройке и работе с портом во втором режиме никаких особенностей не возникает, поэтому можно воспользоваться примером программы, приведённом для нулевого режима работы. Все особенности работы сосредоточены на протокольном уровне, а это не входит в задачу рассмотрения последовательного порта.