CpE 213
Digital Systems Design
Serial Communication

Lecture 22

Monday 11/21/2005

UMR UNIVERSITY OF MISSOURI-ROLLA
The Name. The Degree. The Difference.

Overview

- Assignment 10 due 11/28
- Serial communication

Serial Communication with the 8051

Serial Communication

- Serial communication means that information is transmitted from source to destination over a single pathway.
- Serial data transmission system is divided into 2 categories:
  - Synchronous transmission
    - transmits a block of data and a clock signal or known bit pattern that can be used for synchronization.
    - A block number and check sum are also transmitted.
    - Since digital data is transmitted as a stream of bits, synchronization is required between the source and destination of the data.

Serial Communication (Cont.)

- Asynchronous transmission
  - transmits one character at a time by adding start and stop bits to the character code.
  - Asynchronous here means “asynchronous at the byte level” i.e. there may be a variable-length gap between each byte. But within each byte, the bits are still synchronized; their durations are the same.
- The 8051 and the IBM-compatible PCs support only asynchronous serial communication.

Serial Transmission Modes

- The transmission mode is used to define the direction of signal flow between two linked devices.
- There are three transmission modes:
  - Simplex transmission means that data flows in one direction only.
  - Half-duplex transmission allows data to flow in both directions, but not at the same time.
  - Full-duplex transmission allows data to flow in both directions at the same time.

Serialization and De-serialization

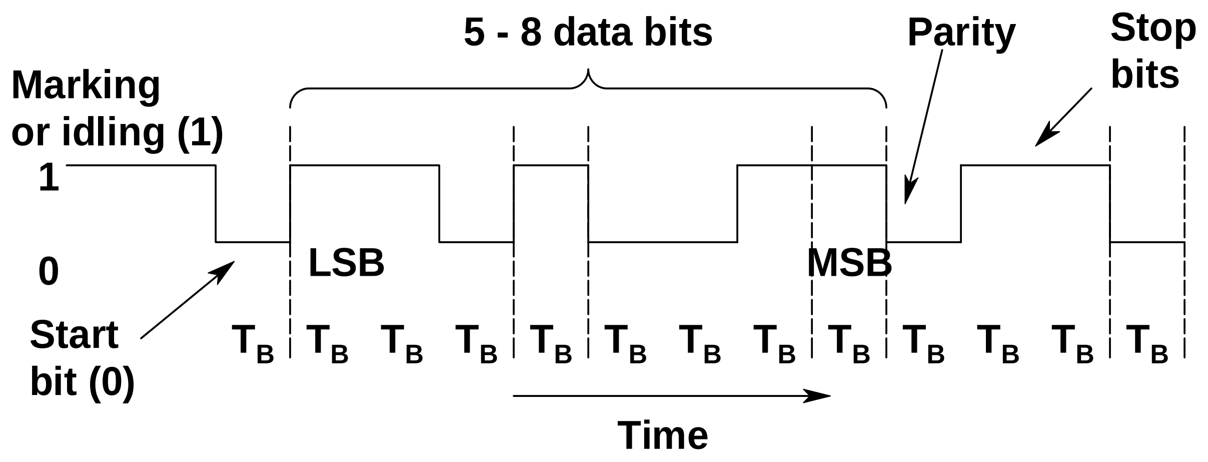
- Serialization: converting bit-parallel bytes off a data bus to a serial bit stream for transmission over a single channel.
  - Uses shift registers
  - Uses the TRANSMITTER part of a special integrated circuit developed for asynchronous systems.
- De-serialization: converting a serial stream of bits to bit-parallel bytes for use within the computer data bus.
  - Uses shift registers
  - Uses the RECEIVER part of a special integrated circuit developed for asynchronous systems.

Operations of Asynchronous Transmission

- The line is normally high.
- A START bit (a low bit) is transmitted to synchronize the receiver to the byte.
- The data bits are sent, LSB first.
- An optional PARITY bit is appended. It is used to check if any bit in the received byte was corrupted or not.
- Either 1 or 2 STOP bit(s) inform(s) the receiver of the end of the byte.
- This procedure happens for every byte.

Modern standard

Serial Data Signal



- Baud rate is the number of signal changes per second

$$\text{Baud rate} = \frac{1}{T_B}\ \text{s}^{-1}$$

- Standard baud rates are: 110 (the old teletype or TTY rate), 300 (early PC modems), 1200, 2400, 9600, 14.4K, 19.2K, 28.8K, 38.4K, and 57.6K

Baud Rate vs. Bits/Sec(bps)

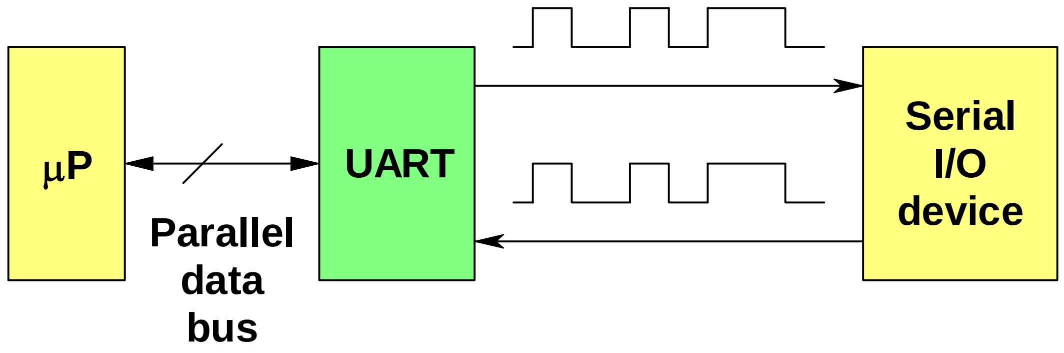
- Baud rate is modem terminology
  - Defined as signaling speed.
- Rate of data transfer (bps) is the number of bits of information that are transmitted/received in a unit time (per second).
- In modems, a single signal change could correspond to transfer of several bits.
- The two are the same as far as the conductor wire is concerned.

Character Encoding

- The most common form of character encoding is the American Standard Code for Information Interchange (ASCII). This is a seven-bit code that allows for 128 characters.
- The codes that are less than 20H are not alpha-numeric characters, but control codes such as carriage return or backspace.
- The numbers 0 - 9 are encoded as 30H - 39H. The capital letters A - Z have codes 41H - 5AH, and the lower case letters a - z are 61H - 7FH.

Universal Asynchronous Receiver/Transmitter (UART)

- Asynchronous serial data communication uses a special IC (UART) operating in asynchronous mode to serialize and de-serialize data.
- The UART also controls the bit rate, generates the start and stop bits, and provides various control functions.



Functions of the UART

- During transmission the UART:
  - Converts data from parallel to serial.
  - Adds the proper parity bit.
  - Adds start and stop bits.
  - Clocks the data out at a proper bit rate.
  - Notifies the microprocessor each time a character has been sent so the microprocessor can make more data available.

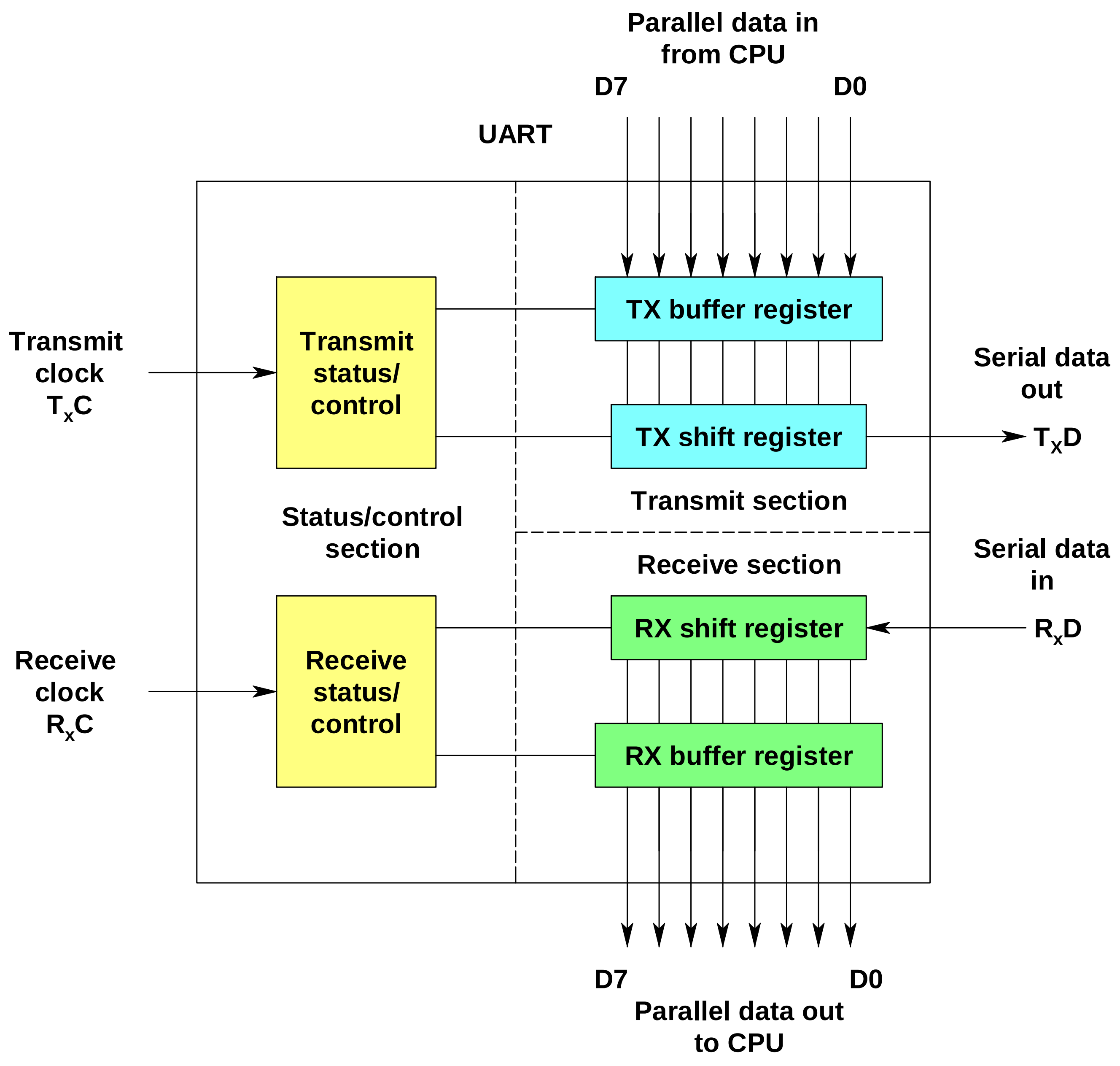
Functions of the UART

- During reception the UART:
  - Looks for the start bit to anticipate the next incoming data word.
  - Converts the incoming data from serial to parallel.
  - Checks parity.
  - Checks framing by looking for the stop bit.
  - Notifies the microprocessor each time a complete character (one byte) has been received.
- The start and stop bits are collectively referred to as framing bits. The absence of an expected stop bit is defined as a framing error.

Three Modules of a UART

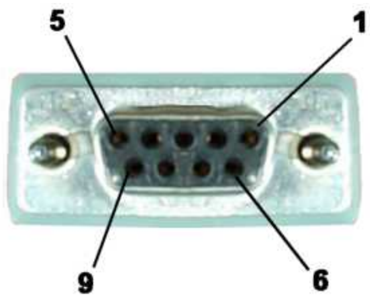
- A UART consists of three main modules:
- a transmit module, which converts data from the computer’s parallel data format to the serial format used by the serial communication channel.
- a receive module, which converts the serial data received from the communication channel to the parallel format needed by the computer’s data bus.
- a status/control module, which regulates and monitors the operation of the other two modules.

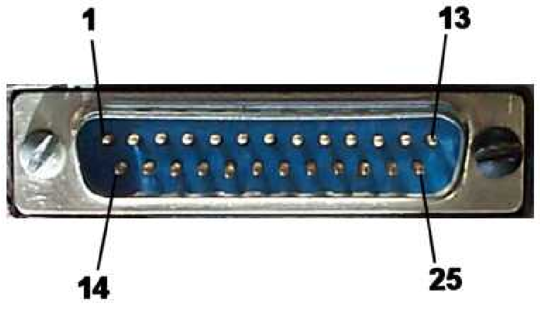
Block Diagram of the UART



RS-232C

- TTL logic signals are not what is transmitted between equipment such as modems, computers, and terminals.
- One standard that specifies the signals for asynchronous communication is the RS-232C standard.
- The RS-232C specifies 25 signal pins with a male DTE (Data Terminal Equipment) and a female DCE (Data Communications Equipment) connector. Most of these signals are not used in most applications, so a 9 pin subset is used (popularized by the IBM PC or compatible).
- Actually in our 8051 interfacing case, only three wires are used. They are the Transmit (TxD), Receive (RxD) and GND signals.





Important Asynchronous Signals in RS-232C

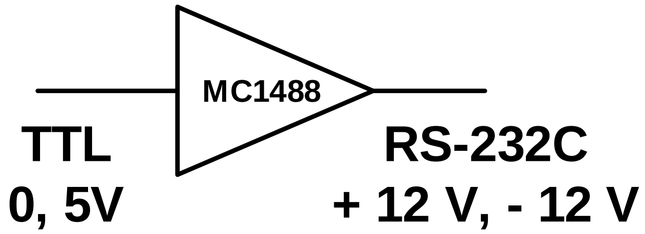
Description	PIN # (9-PIN)	From	Abbreviation
Data Leads			
Transmit Data	3	DTE	TD
Receive Data	2	DCE	RD
Power On Indicator Leads			
Data set ready	6	DCE	DSR
Data terminal ready	4	DTE	DTR
Leads that announce that an outside event has taken place			
Data carrier detect	1	DCE	CD
Ring Indicator	9	DCE	RI
Ready to send/receive handshake leads			
Request to send	7	DTE	RTS
Clear to send	8	DCE	CTS
Ground leads			
Signal ground	5		SG

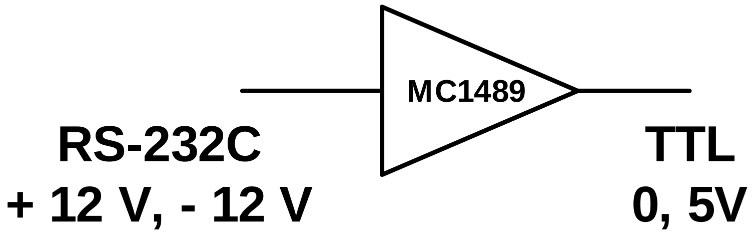
RS-232C Voltage Specifications

- Consider voltages from point of view of TX and RX.
- Need to recall difference between levels on DATA and CONTROL lines.
  - For data
    - logic 1 is positive, logic 0 is negative.
  - For controls
    - Asserted is positive, de-asserted is negative.
- A logic high or mark is between - 3V and - 15V under load.
- A logic low or space is between + 3V and + 15 V under load.
- Typically, +12V and -12 volts are used. The voltage swing is > the TTL 5 volts for noise immunity.

RS232 Voltage Level Convertors

- UARTs are generally used with RS-232 hardware that converts the 5V to -12V and the 0V to +12V and vice-versa.
- If -12 V and +12 V are available in the system, then a 1488 and 1489 are generally used as driver and receiver.

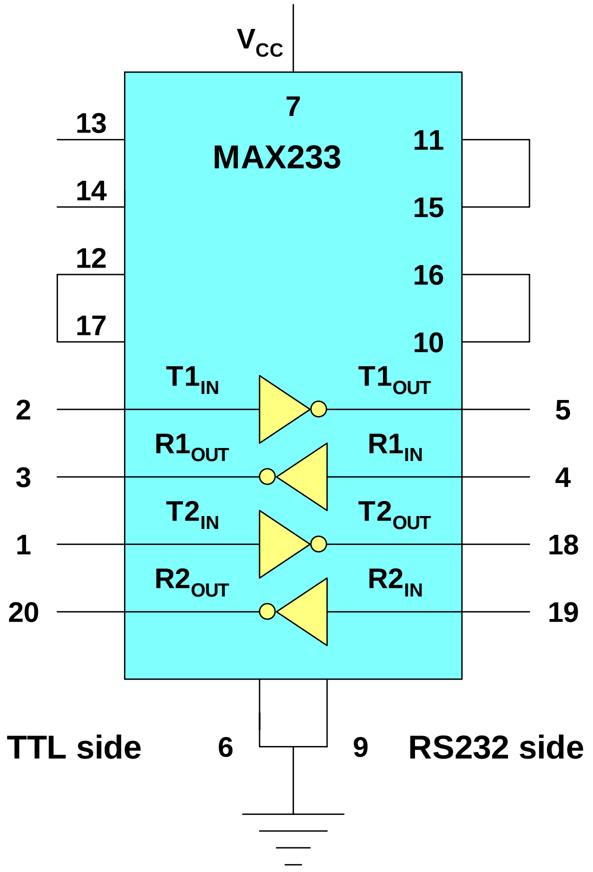


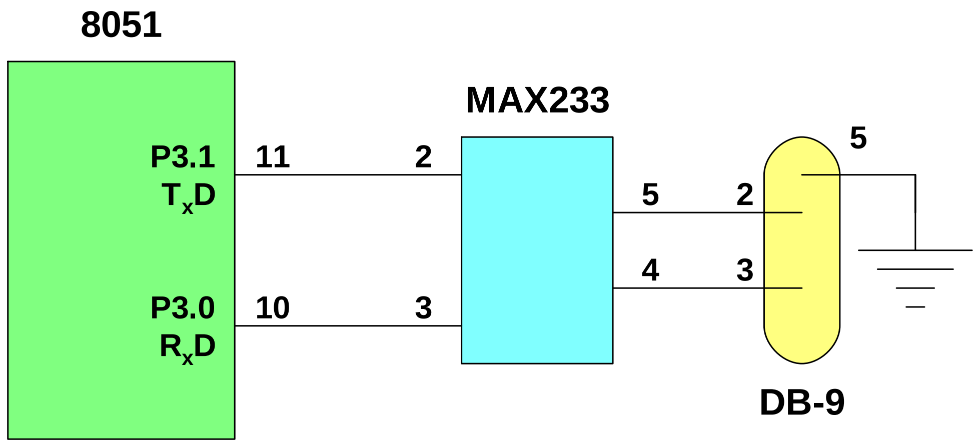


RS232 to TTL Converter with MAX-233

- Nowadays the MAX233 is a good solution because it generates its own -12 V and +12 V from 0 and 5 V, respectively, and because it integrates both driver and receiver.
- The MAX233 basically performs the same job as the MAX232 but eliminates the need for the capacitors.
- However, the MAX233 chip is much more expensive than the MAX232 and they are not pin compatible. You cannot take a MAX232 out of a board and directly replace it with a MAX233.

Interfacing MAX233 with the 8051





Serial Communication on the 8051

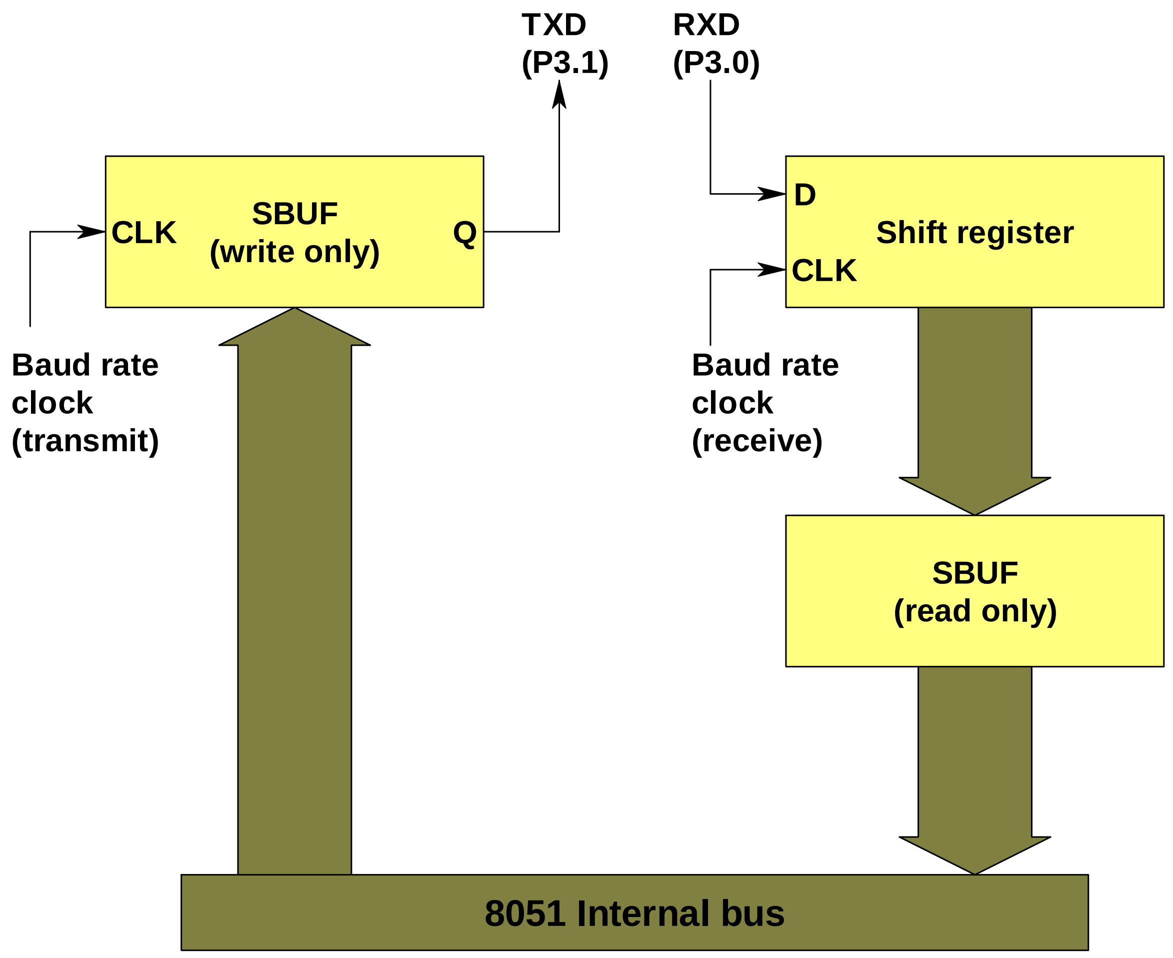
8051's Built-in UART

- One of the 8051's many powerful features is its built-in UART, otherwise known as a serial port.
- The serial port is full-duplex, so it can receive and transmit data simultaneously.
- The port is also buffered in receive mode, so it can receive a second data byte before the first data byte has been read from the register.
- SBUF: SFR to hold data at 99H.
  - This is physically two registers.
  - One is write-only and is used to hold data to be transmitted.
  - The other is read-only and is used to hold received data.

Control Registers

- The 8051 has two SFR registers that control the serial port
  - SCON: Serial Port Control Register at 98H, bit-addressable SFR. It is the serial port control and status register.
  - PCON: Power Control Register at 87H. It is the SFR used to control data rates. Not bit-addressable.

8051 Serial Port Block Diagram



SCON: Serial Control Register

SCON Address = 98H Reset Value = 0000 0000B
Bit Addressable

Bit:	7	6	5	4	3	2	1	0
	SMO/FE	SM1	SM2	REN	TB8	RB8	T1	R1

Symbol	Function
FE	Framing Error bit. This bit is set by the receiver when an invalid stop bit is detected. The FE bit is not cleared by valid frames but should be cleared by software. The SMOD0 bit must be set to enable access to the FE bit. SMOD0 is located at PCON6
SM0	Serial Port Mode Bit 0 SMOD0 must = 0 to access bit SM0. SMOD0 is located at PCON6.
SM1	Serial Port Mode Bit 1.

SM0	SM1	Mode	Description	Baud Rate**
0	0	0	shift register	FOSC/12
0	1	1	8-bit UART	variable
1	0	2	9-bit UART	FOSC/64 or FOSC/32
1	1	3	9-bit UART	variable

**FOSC = oscillator frequency

SCON: Serial Control Register (Cont.)

Symbol	Function
SM2	Enables the Automatic Address Recognition feature in Modes 2 or 3. If SM2 = 1 then RI will not be set unless the received 9th data bit (RB8) is 1, indicating an address, and the received byte is a Given or Broadcast Address. In Mode 1, if SM2 = 1 then RI will not be activated unless a valid stop bit was received, and the received byte is a Given or Broadcast Address. In Mode 0, SM2 should be 0.
REN	Enables serial reception. Set by software to enable reception. Clear by software to disable reception.
TB8	The 9th data bit that will be transmitted in Modes 2 and 3. Set or clear by software as desired.
RB8	In modes 2 and 3, the 9th data bit that was received. In Mode 1, if SM2 = 0, RB8 is the stop bit that was received. In Mode 0, RB8 is not used.
T1	Transmit interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or at the beginning of the stop bit in the other modes, in any serial transmission. Must be cleared by software.
R1	Receive interrupt flag. Set by hardware at the end of the 8th bit time in Mode 0, or halfway through the stop bit time in the other modes, in any serial reception (except see SM2). Must be cleared by software.

PCON: Power Control Register

PCON **Address = 87H** **Reset Value = 00XX 0000B**
Not Bit Addressable

Bit:	7	6	5	4	3	2	1	0
	SMOD1	SMOD0	--	N/A	N/A	N/A	N/A	N/A

Symbol	Function
SMOD1	Double Baud rate bit. When set to a 1 and Timer 1 is used to generate baud rates, and the Serial Port is used in modes 1, 2, or 3.
SMOD0	When set, Read/Write accesses to SCON.7 are to the FE bit. When clear, Read/Write accesses to SCON.7 are to the SM0 bit.
--	Not implemented, reserved for future use.
N/A	Not Applicable. These bits are used for power control capability and have nothing to do with the serial port.

8051 Serial Interface Modes

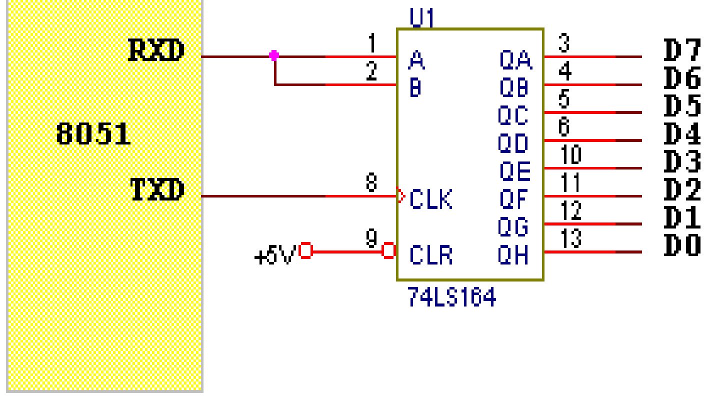
SM0	SM1	Mode	Description	Baud rate
0	0	0	Shift register	fosc/12
0	1	1	8-bit UART	Variable
1	0	2	9-bit UART	fosc/64 or fosc/32
1	1	3	9-bit UART	Variable

- For the above modes, the 8051 uses port 3, bits 1 and 0 as the serial TxD (output) and serial RxD (input) pins, respectively.

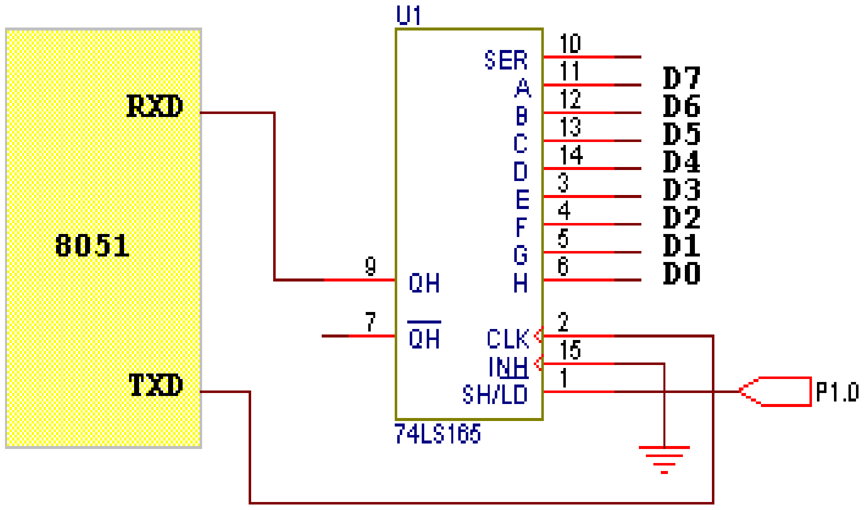
Serial Interface Mode 0 Shift Register

- Synchronous mode (half-duplex)
- Fixed baud rate (f_{osc}/12)
  - (Max. 1 MHz for 12MHz osc.)
- 8 data bits (LSB first)
- TxD pin used as shift clock
- RxD pin used for data
- This mode is not intended for data communication. It is mainly used for serial I/O expansion.
- Typical external shift register for output is 74LS164, and for input is 74LS165.

Mode 0 Interfacing



Serial I/O Expansion OUTPUT Port



Serial I/O Expansion INPUT Port

Mode 0 Program Example 1

SCON bit 7 … bit 0

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	0	0	0	0	0	0	0

- The following program segment shows how to make use of mode 0 to shift the content of address 20H out to an external shift register (74LS164).

```
START:  MOV   SCON,#00H ;Set mode 0.
        MOV   R0,#20H   ;Set data address.
        MOV   A,@R0     ;Get the parallel output data.
        MOV   SBUF,A    ;Load SBUF for transmission.
WAIT:   JNB   TI,WAIT   ;If TI = 0, wait.
                        ;Data byte not yet fully transmitted
       CLR    TI        ;TI must clear by software.
```

Cautions of Shift Register Data Transmission

- Data transmission begins when a byte is written to SBUF by the program, and lasts for eight clock pulses. Data is shifted out LSB first. TI is set after the last data bit has been clocked out by TxD.
- Software must check TI to ensure that the last byte has been transmitted before writing another byte to SBUF.
- The program can easily move bytes to SBUF faster than they can be transmitted.
- Shift register parallel data output is normally enabled after all eight serial bits have been shifted.
- Parallel output data should be latched into an external latch to stabilize the data.

Mode 0 Program Example 2

SCON bit 7 bit 0

SM0	SM1	SM2	REN	TB8	RB8	TI	RI
0	0	0	1	0	0	0	0

- The following program fragment shows how to make use of mode 0 to shift in the content of an external shift register (74LS165) and store the result in address 20H.

```
        CLR     P1.0        ;Set 74LS165 in load mode, to load data
                            ;from its parallel port first.
        SETB    P1.0        ;Now set 74LS165 to shift mode.
        MOV     SCON,#10H  ;Set up mode 0 for data reception.
WAIT:   JNB     RI,WAIT     ;Wait for RI to go high.
        CLR     RI          ;RI must be cleared by software
        MOV     20H,SBUF     ;Save the shift reg. to address 20H
```

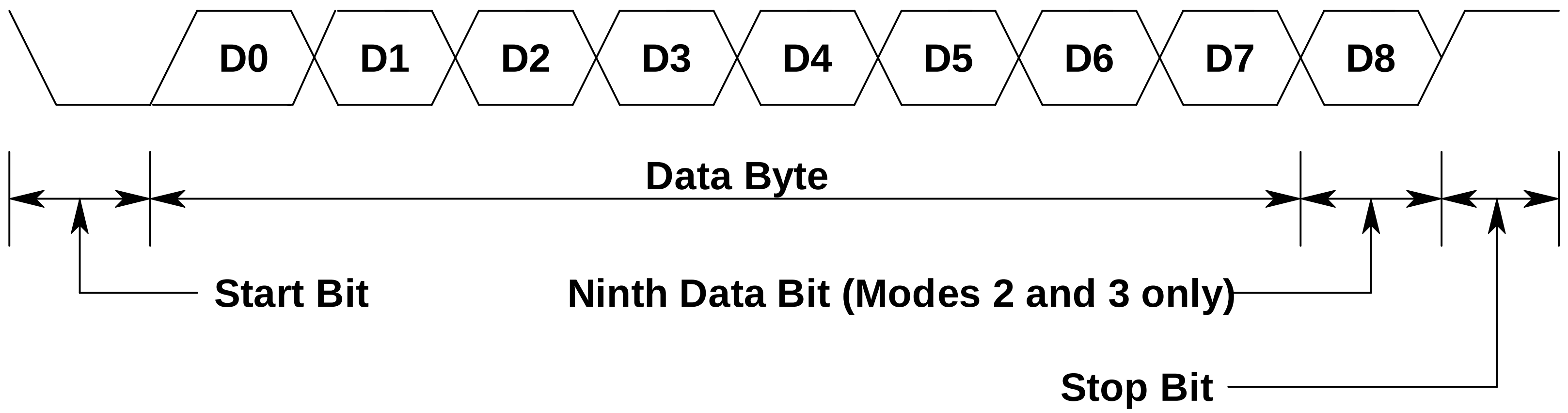
Cautions of Shift Register Data Reception

- Data reception is enabled by setting REN and clearing RI.
- RI is set when the MSB of the received data byte has been shifted into SBUF.
- The program must check to ensure RI is set before reading data from SBUF and then reset RI.
- Having to reset RI to enable further data reception is unique to mode 0. All other modes may receive data irrespective of the state of RI.

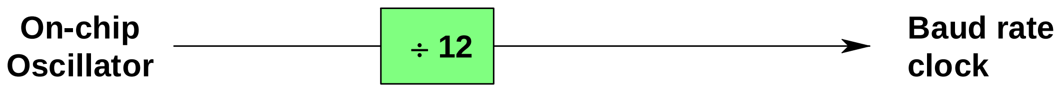
8051 Serial Port Operation Mode 1

- Asynchronous Mode (Full Duplex)
- Variable Baud Rate - can go up to 104.2 KHz (20 MHz osc.).
- 8 data bits (LSB first)
- 1 Start Bit (0); 1 Stop Bit (1)
- The data frame format is compatible with the COM port of IBM/compatible PCs.
  - Makes it the most useful mode.
- On receive, the stop bit goes into RB8 in special function register SCON.

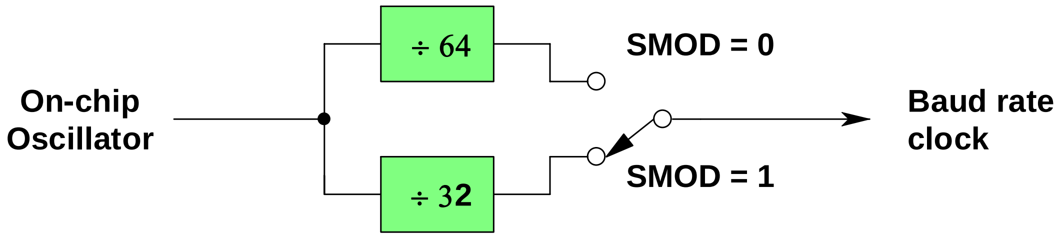
Data Frame Format For Modes 1, 2 and 3



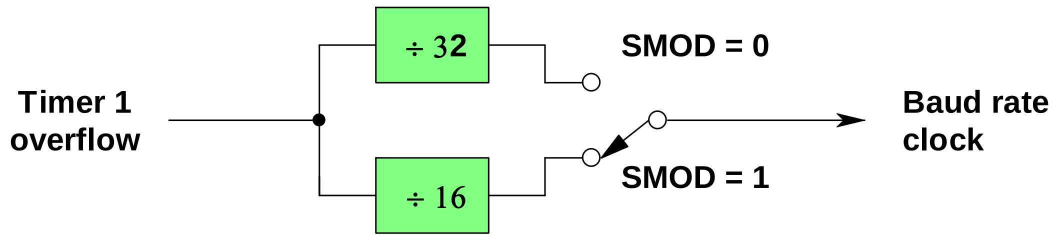
Serial Port Clocking Sources



(a) Mode 0



(b) Mode 2



(c) Modes 1 and 3

SMOD is the control bit in PCON and can be 0 or 1.
When SMOD = 1, it doubles the baud rate.

Mode 1 Baud Rates

- Timer 1 is used to generate the baud rate for mode 1 by using the overflow flag of the timer to determine the baud frequency.
- When the timer overflows, it signals the processor to send/receive a data bit out/in to the serial port.
- Typically, timer 1 is configured in auto-reload mode (mode 2), and the baud rate is given as:

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times \frac{\text{oscillator frequency}}{12 \times [256 - \text{TH1}]}$$

- Very often, the baud rate is given, what we need to determine is the timer reload value.

$$\text{TH1} = 256 - \frac{2^{\text{SMOD}}}{32} \times \frac{\text{oscillator frequency}}{12 \times \text{Baud rate}}$$

Mode 1 Baud Rates (Cont.)

- The oscillator frequency is chosen to help generate both standard and nonstandard baud rates.
- That is why an oddball crystal frequency of 11.0592 MHz (instead of 12 MHz) is used so often for 8051 designs. This is to achieve exact clock rates for most standard baud rates, i.e., 9600, 19200.

Mode 1 Baud Rates (Cont.)

- If timer 1 is not run in timer mode 2, then the baud rate is:

$$\text{Baud rate} = \frac{2^{\text{SMOD}}}{32} \times (\text{Timer 1 overflow rate})$$

- The baud rate is determined by how frequently timer 1 overflows. The more frequently timer 1 overflows, the higher the baud rate.
- The timer 1 interrupt should be disabled in this application.
- The timer itself can be configured for either “timer” or “counter” operation, in any of its 3 running modes (mode 0, 1 and 2).

Initialize UART

```
TMOD |= 0x20; //Timer 1 mode 2
TH1 = -24;   //1200 baud @ 11.059 Mhz
TR1 = 1;     //start timer
SCON = 0x50; //SM1+REN
```

Output a character

```
char c;

SBUF= c; //write char to SBUF
while(!TI); //wait until done
```

Receive a character

```
char c;

while(!RI); //wait for done
RI=0;
c= SBUF;
```

Happy Thanksgiving!