The 8051 Microcontroller and Embedded Systems

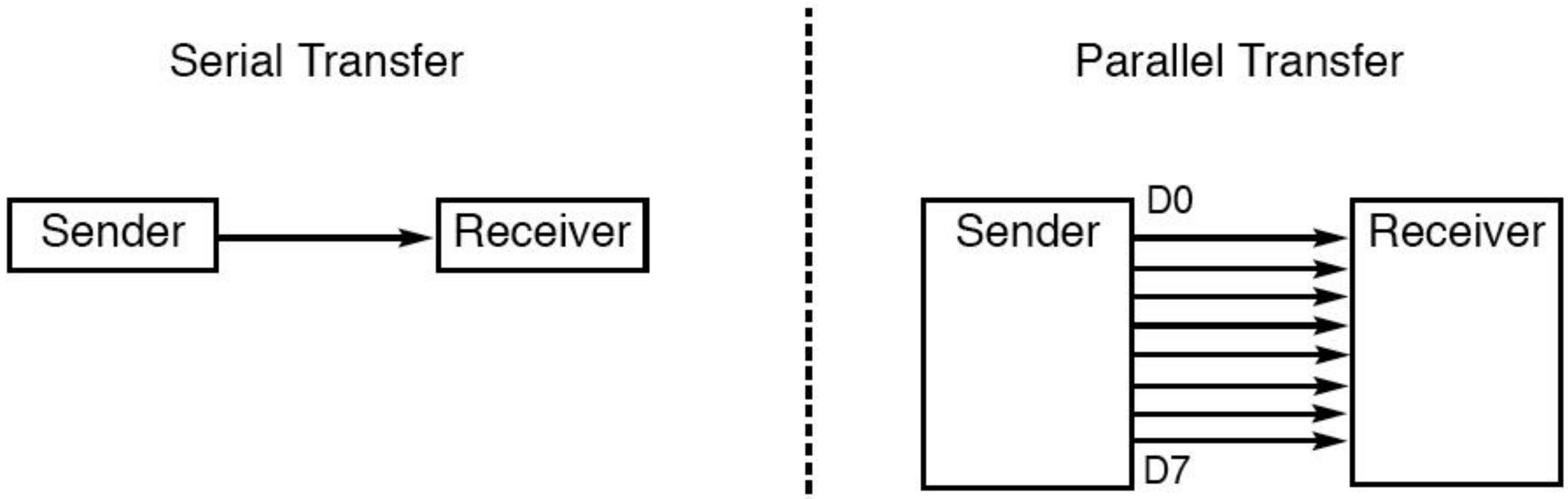
8051 SERIAL PORT PROGRAMMING IN ASSEMBLY

OBJECTIVES

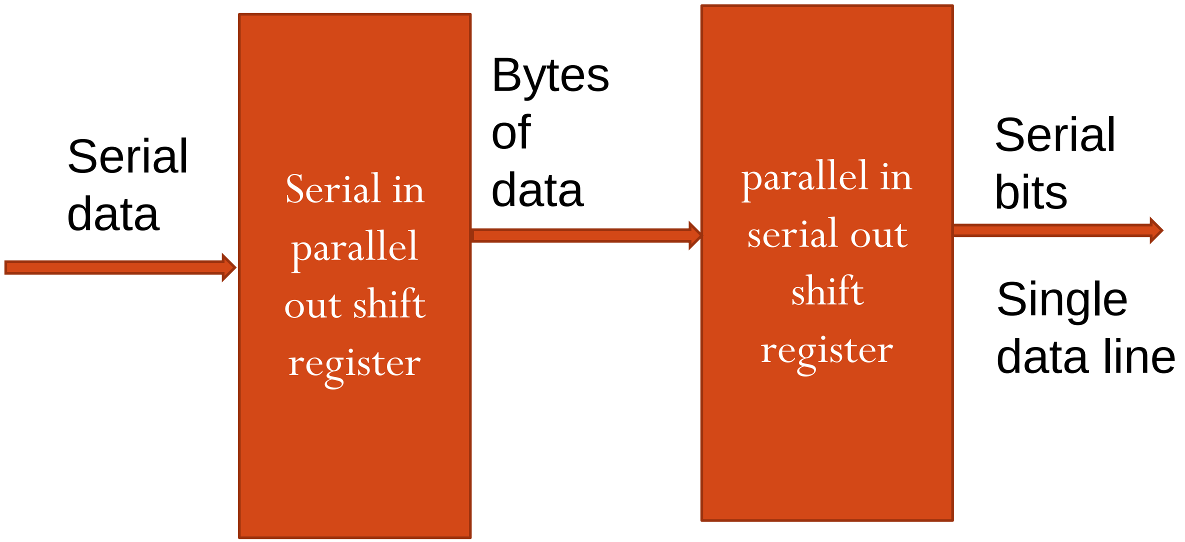
- Contrast and compare serial versus parallel communication
- List the advantages of serial communication over parallel
- Explain serial communication protocol
- Contrast synchronous versus asynchronous communication
- Contrast half-versus full-duplex transmission
- Explain the process of data framing
- Describe data transfer rate and bps rate
- Define the RS232 standard
- Explain the use of the MAX232 and MAX233 chips

Basics of Serial Communication

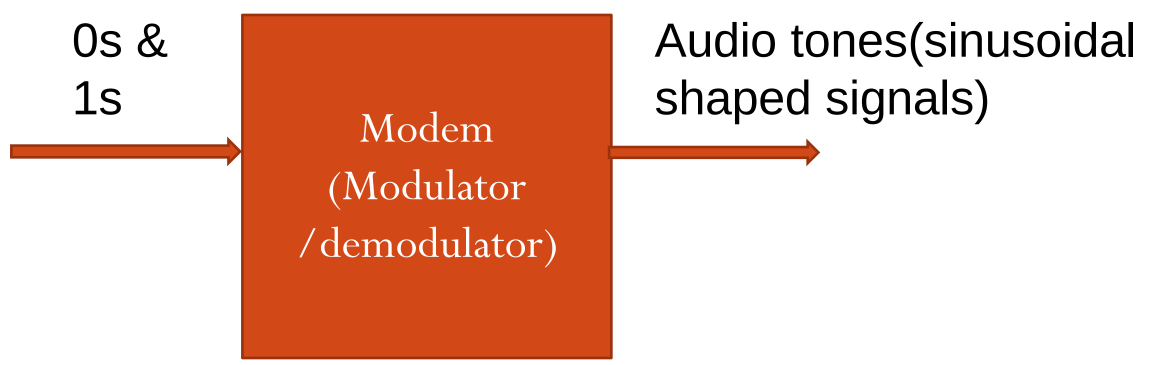
- Serial communication is used for transferring data between 2 systems located at distances of hundreds of feet to millions of miles apart.



	Serial transmission	Parallel Transmission
Data transfer	1.Data transfer done by bit by bit.	1.Data transfer done by more than 1 bit at a time(8 bit data)
Data Transfer rate	Slower	faster
Wiring requirement	For long distance can be used as single wire required	For long distance can't be used as more wires required
Cost	Low	High



If data is to be transferred on telephone line



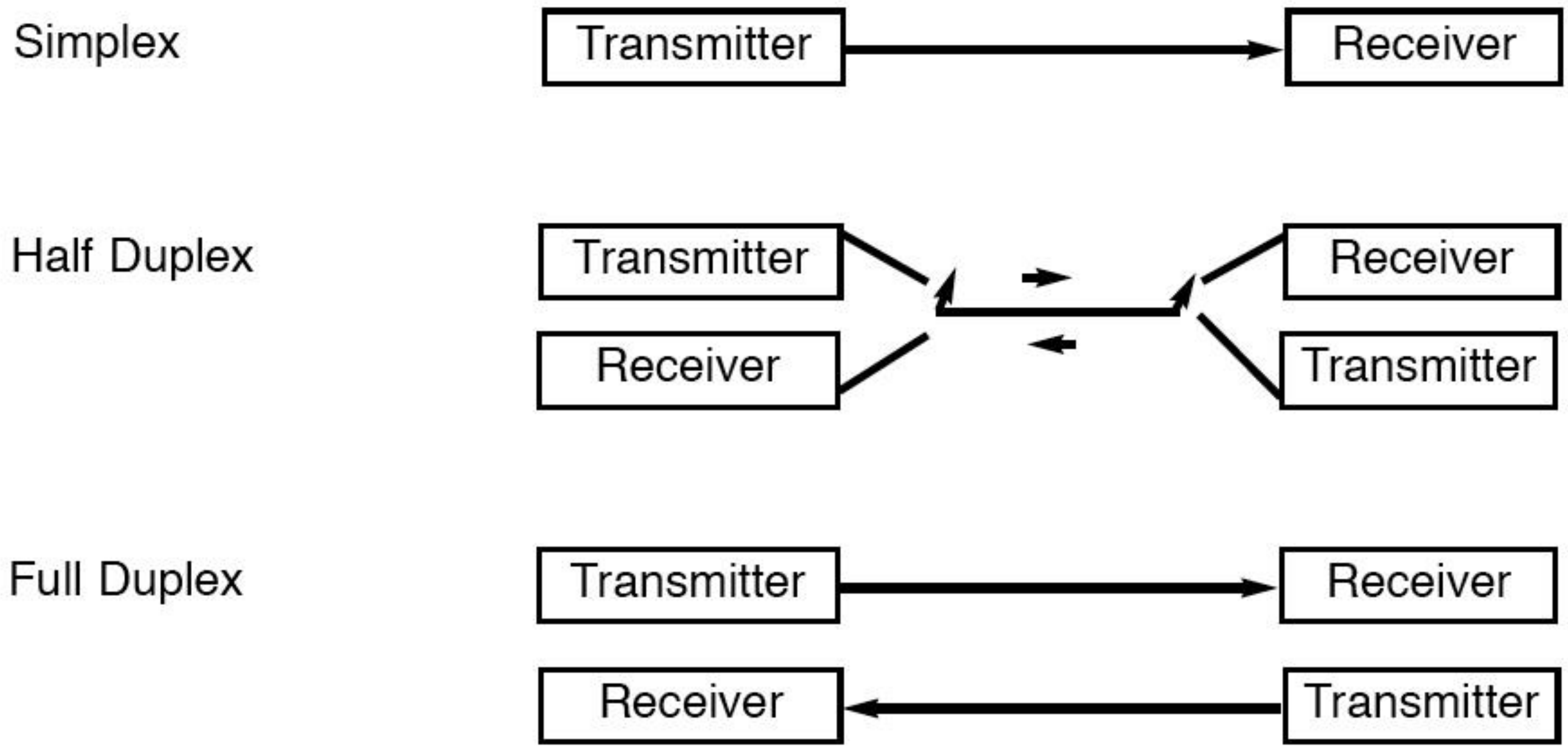
- When distance is short digital signal can transferred as it is on single wire & requires no modulation. e.g.IBM PC keyboard transfer data to motherboard.
- For long distance data transfer using communication lines such as telephone, serial data communication requires a modem to modulate(convert from 0s & 1s to audio tones) & demodulate(converting from audio tones to 0s & 1s)

Serial data communication methods

- Synchronous :-transfer block of data (characters) at a time
- Asynchronous:- transfer a single byte at a time
- By Software ,this method programs are tedious & long.

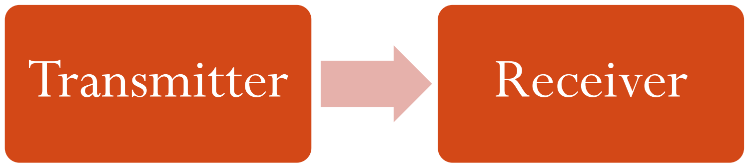
Serial data communication chips

- UART(universal asynchronous receiver transmitter)
- USART(universal synchronous -asynchronous receiver transmitter)
- 8051 has in built UART chip


Simplex
Transmitter
Receiver
Half Duplex
Transmitter
Receiver
Receiver
Transmitter
Full Duplex
Transmitter
Receiver
Receiver
Transmitter

Half duplex & full duplex transmission

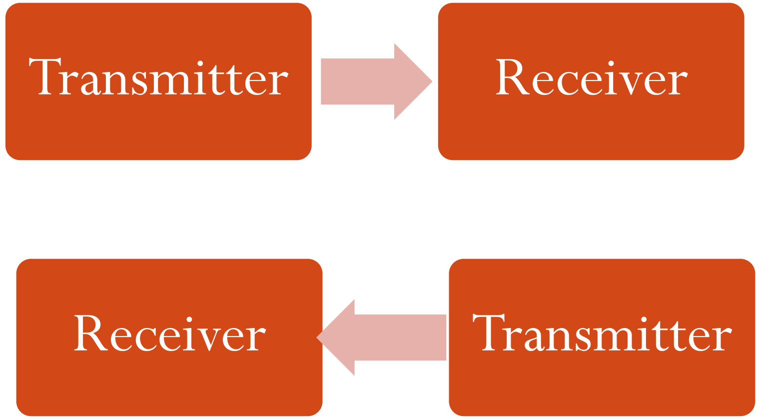
- Simplex transmission:-e.g. printers (computer only sends data)



- In data transmission if data can be transmitted & received it is duplex transmission.
- Half duplex :-data is transmitted 1 way at a time.

Half duplex & full duplex transmission

- Full duplex :-data can go both ways at same time.



- It requires 2 wire conductors for data lines (in addition to signal ground),1 for transmission & 1 for reception, in order to transfer & receive data simultaneously.

Port Pin	Alternate Function
P3.0	RXD (serial input port)
P3.1	TXD (serial output port)
P3.2	$\overline{INT0}$ (external interrupt)
P3.3	$\overline{INT1}$ (external interrupt)
P3.4	T0 (Timer/Counter 0 external input)
P3.5	T1 (Timer/Counter 1 external input)
P3.6	$\overline{WR}$ (external data memory write strobe)
P3.7	$\overline{RD}$ (external data memory read strobe)

SECTION 10.1: BASICS OF SERIAL COMMUNICATION

- **Asynchronous serial communication and data framing**
  - **data coming in 0s and 1s**
  - **to make sense of the data sender and receiver agree on a set of rules**
  - **Protocol**
    - **how the data is packed**
    - **how many bits/character**
    - **when the data begins and ends**

SECTION 10.1: BASICS OF SERIAL COMMUNICATION

- **Start and stop bits**
  - **asynchronous method, each character is placed between start and stop bits**
  - **called *framing***
  - **start bit is always one bit**
  - **stop bit can be one or two bits**
  - **start bit is always a 0 (low)**
  - **stop bit(s) is 1 (high)**
  - **LSB is sent out first**

SECTION 10.1: BASICS OF SERIAL COMMUNICATION

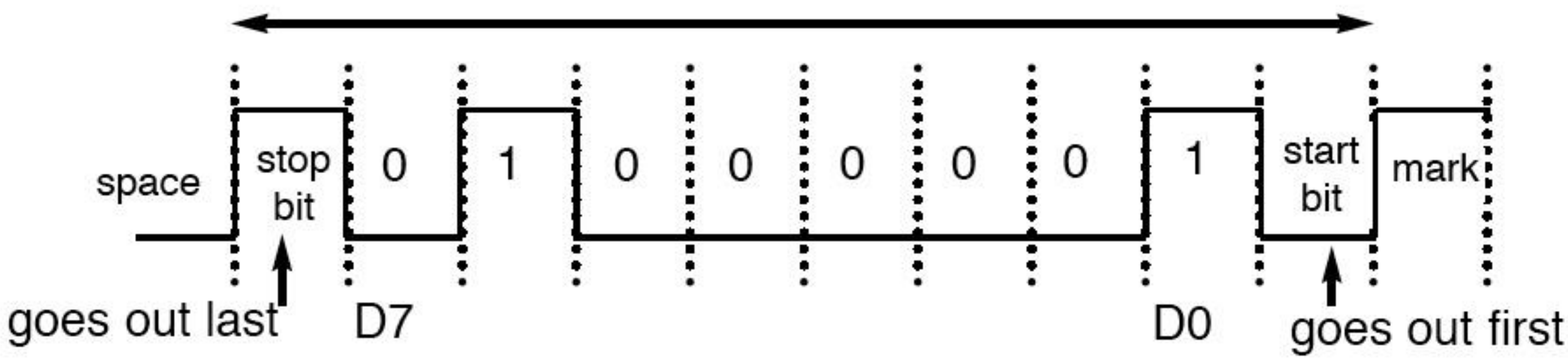


Figure 10–3 Framing ASCII “A” (41H)

SECTION 10.1: BASICS OF SERIAL COMMUNICATION

- **in modern PCs one stop bit is standard**
- **when transferring a text file of ASCII characters using 1 stop bit there is total of 10 bits for each character**
- **8 bits for the ASCII code (1 parity bit), 1 bit each for the start and stop bits**
- **for each 8-bit character there are an extra 2 bits, which gives 20% overhead**

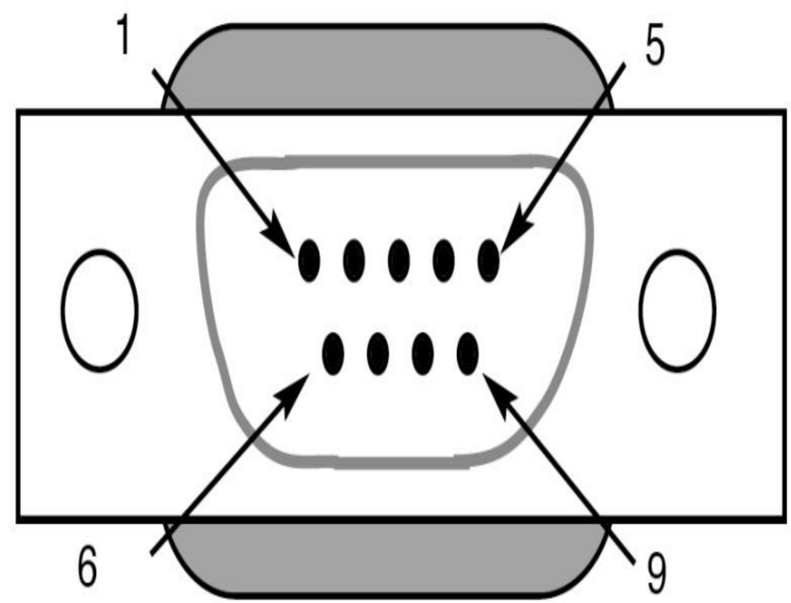
SECTION 10.1: BASICS OF SERIAL COMMUNICATION

- **Data transfer rate**
  - **rate of data transfer *bps* (bits per second)**
  - **widely used terminology for bps is *baud rate***
  - **baud and bps rates are not necessarily equal**
  - **baud rate is defined as the number of signal changes per second**

9-Pin Connector

Pin	Description
1	Data carrier detect ($\overline{DCD}$)
2	Received data (RxD)
3	Transmitted data (TxD)
4	Data terminal ready (DTR)
5	Signal ground (GND)
6	Data set ready ($\overline{DSR}$)
7	Request to send ($\overline{RTS}$)
8	Clear to send ($\overline{CTS}$)
9	Ring indicator (RI)

IBM PC DB-9 Signals



DB-9 9-Pin Connector

Data communication classification

- **Data communication classification**
  - **DTE (data terminal equipment)**
  - **DCE (data communication equipment)**
  - **DTE - terminals and computers that send and receive data**
  - **DCE - communication equipment responsible for transferring the data**
  - **simplest connection between a PC and microcontroller requires a minimum of three pins, TxD, RxD, and ground**

DB-9 9-Pin Connector

- 1.DTR(data terminal ready):-When a terminal (PC com port)is turned on, it sends signal DTR to indicate that it is ready for communication.
- 2.DSR(Data set ready) :-When DCE(data communication equipment)i.e. modem is on,DSR indicate that it is ready to communicate. Active low signal.
- 3.RTS(Request to send):-When DTE(data terminal equipment)(PC)has byte to transmit, RTS send the signal to the modem that it has a byte of data to trasmit.Active low .
- 4. CTS(clear to send):-modem sends CTS signal to DTE(PC)to indicate that it can receive data now.

DB-9 9-Pin Connector

- 5.DCD(Carrier detect or data carrier detect):-Modem sends DCD signal to inform DTE(PC)that valid carrier is detected & that contact between it & other modem is established.
- 6.RI(ring indicator):-An o/p from modem(DCE)& i/p to PC(DTE) indicates that telephone is ringing.

Null Modem Connection

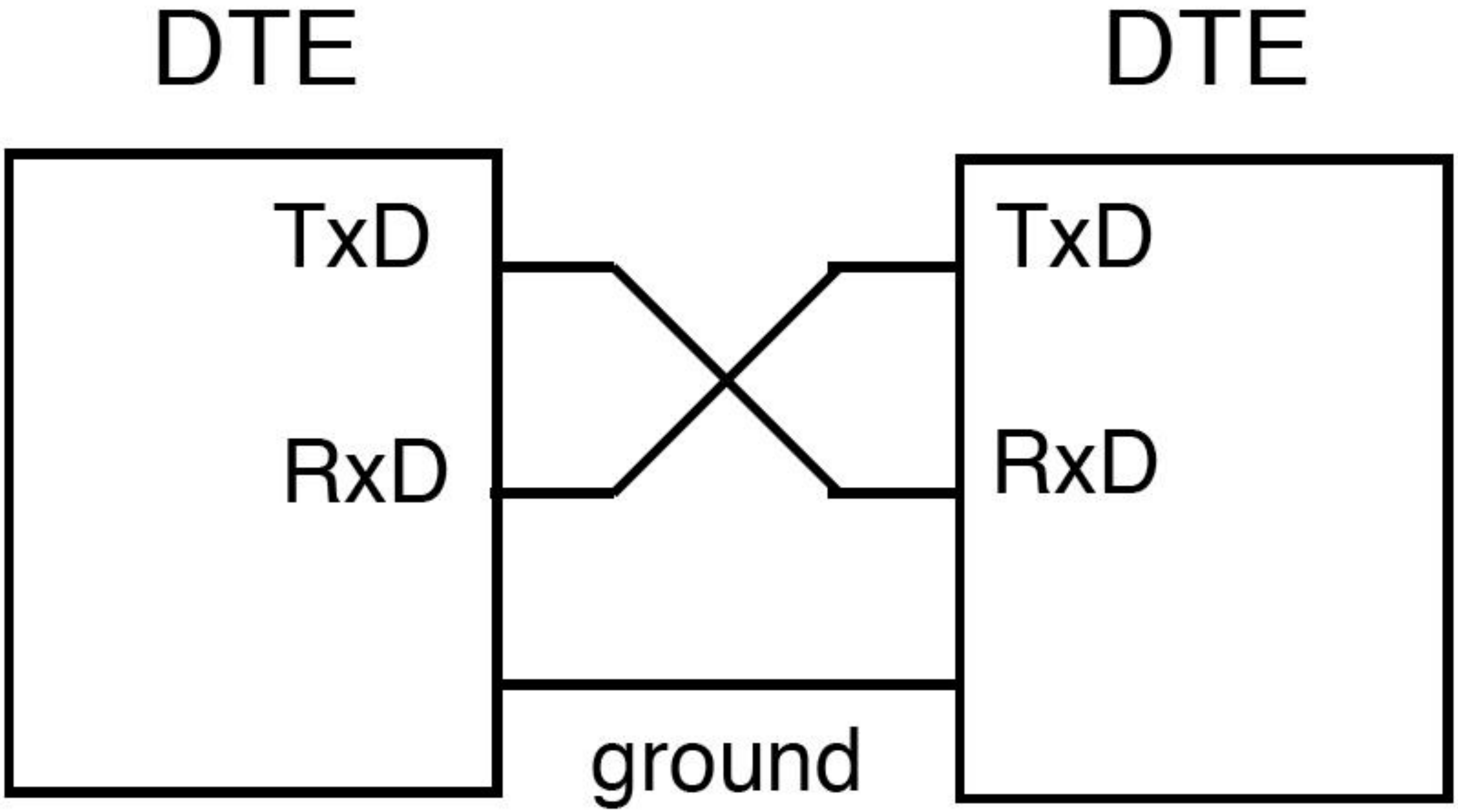
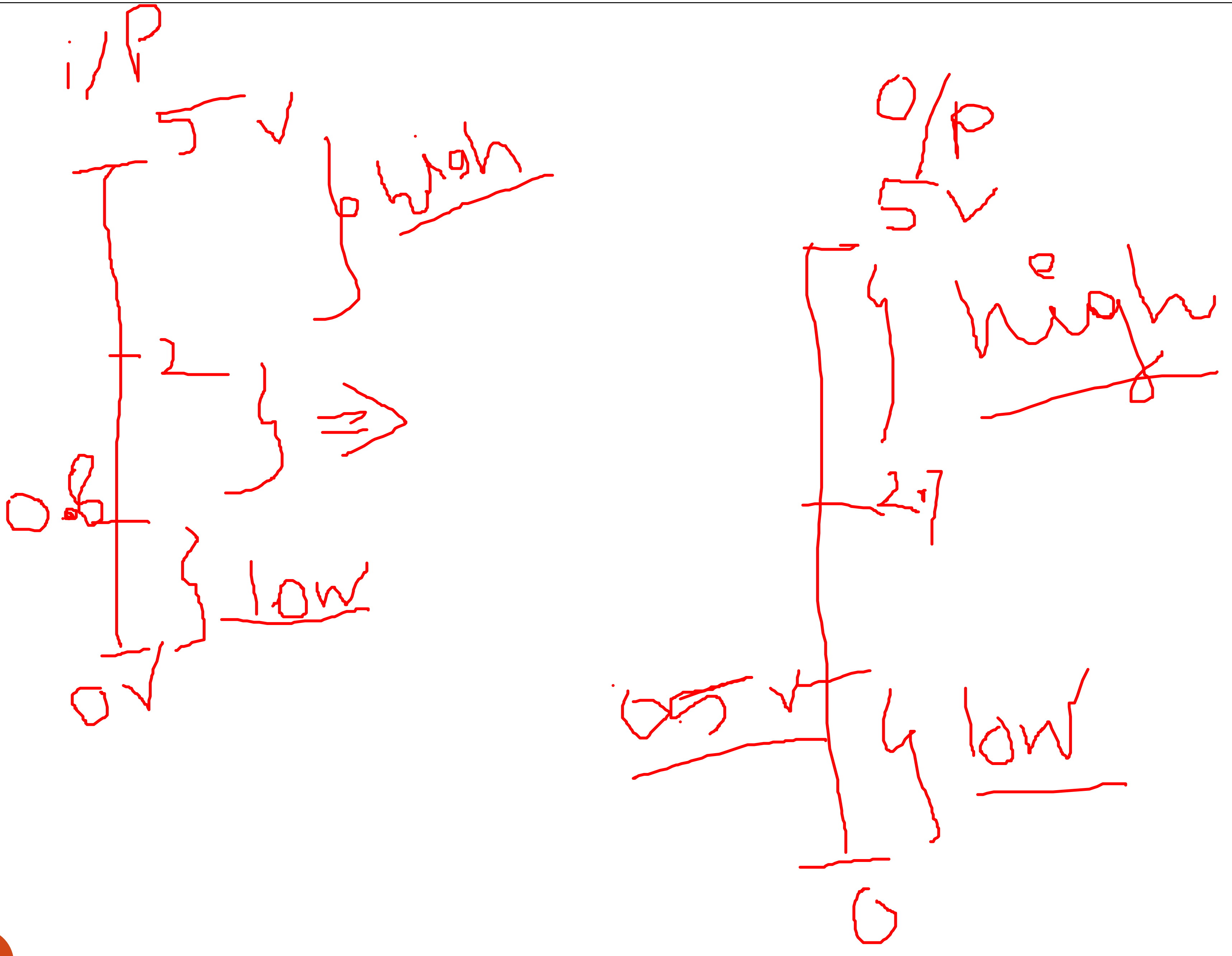


Figure 10–6 Null Modem Connection

SECTION 10.2: 8051 CONNECTION TO RS232

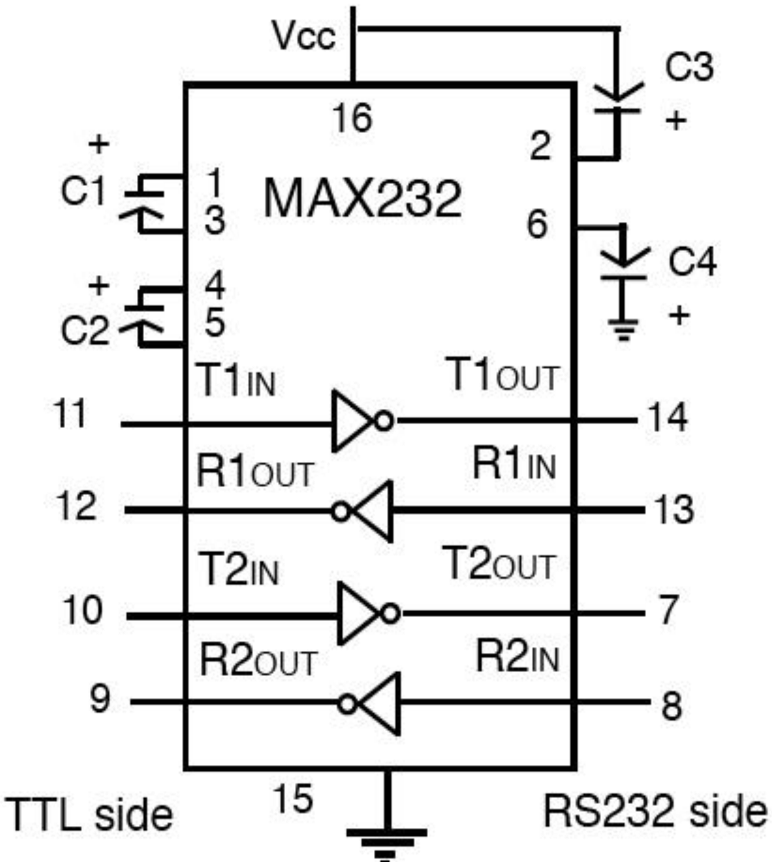
- **RxD and TxD pins in the 8051**
  - **8051 has two pins used for transferring and receiving data serially**
  - **TxD and RxD are part of the port 3 group**
  - **pin 11 (P3.1) is assigned to TxD**
  - **pin 10 (P3.0) is designated as RxD**
  - **these pins are TTL compatible**
  - **require a line driver to make them RS232 compatible**
  - **driver is the MAX232 chip**


i/p
5V
high
2
0.8
low
0V
o/p
5V
high
2.7
0.5V
low
0

SECTION 10.2: 8051 CONNECTION TO RS232

- **MAX232 (Line Driver) (voltage converter)**
  - **converts from RS232 voltage levels to TTL voltage levels**
  - **uses a +5 V power source**
  - **MAX232 has two sets of line drivers for transferring and receiving data**
  - **line drivers used for TxD are called T1 and T2**
  - **line drivers for RxD are designated as R1 and R2**
  - **T1 and R1 are used together for TxD and RxD of the 8051**
  - **4 capacitors used(1 to 22 UF)**

SECTION 10.2: 8051 CONNECTION TO RS232



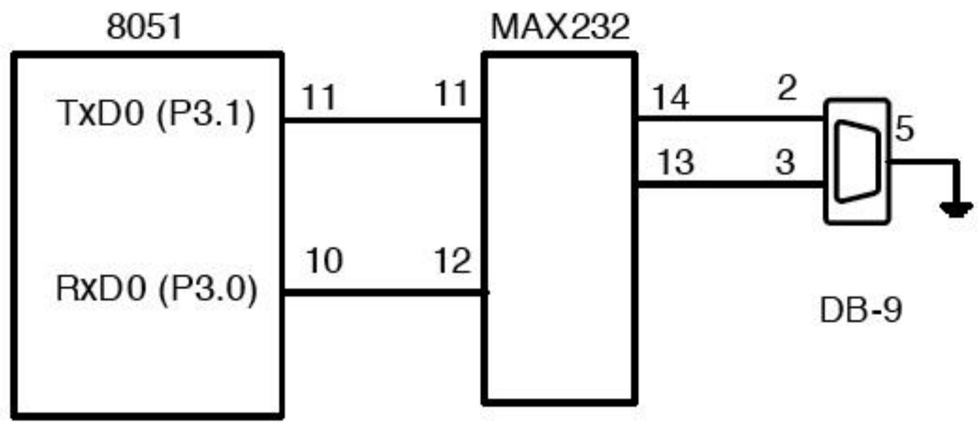


Figure 10–7 (a) Inside MAX232
(b) its Connection to the 8051 (Null Modem)

SECTION 10.2: 8051 CONNECTION TO RS232

- **MAX233**
  - **MAX233 performs the same job as the MAX232**
  - **eliminates the need for capacitors**
  - **much more expensive than the MAX232**

SECTION 10.2: 8051 CONNECTION TO RS232

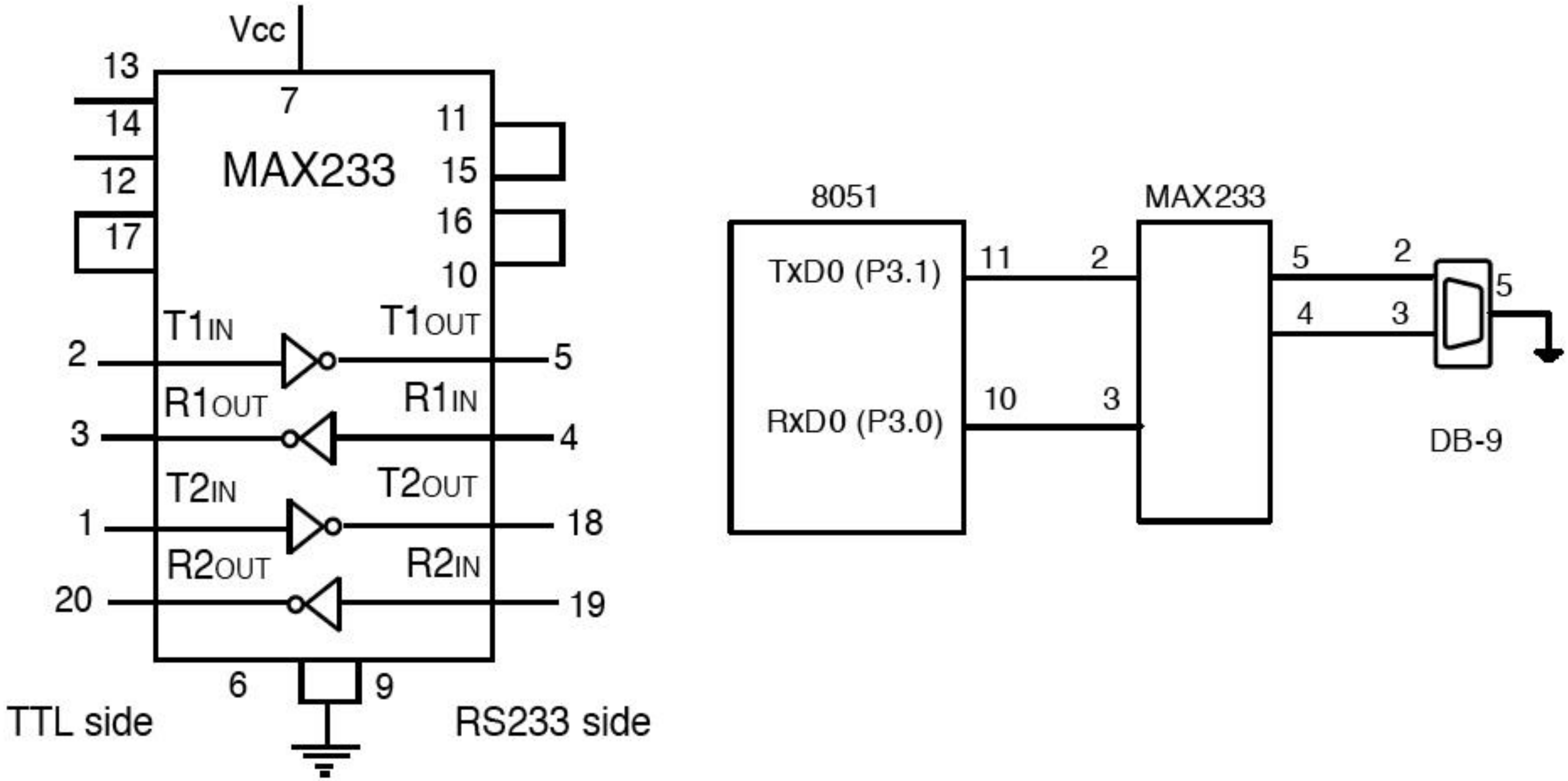


Figure 10–8 (a) Inside MAX233
(b) Its Connection to the 8051 (Null Modem)

Baud rate in the 8051

- **Baud rate in the 8051**
  - **serial communications of the 8051 with the COM port of the PC**
  - **To allow data transfer between PC & 8051 w/o error, we must make sure that the baud rate of the 8051 system matches the baud rate of the PC's COM port.**
  - **can use Windows HyperTerminal program**

PC Baud Rates

110
150
300
600
1200
2400
4800
9600
19200

Note: Some of the Baud rates supported by 486/Pentium IBM PC BIOS.

Table 10–3 PC Baud Rates

Baud rate in the 8051

- **baud rate in the 8051 is programmable**
- **done with the help of Timer 1**
- **relationship between the crystal frequency and the baud rate in the 8051**
- **8051 divides the crystal frequency by 12 to get the machine cycle frequency**
- **XTAL = 11.0592 MHz, the machine cycle frequency is 921.6 kHz**
- **8051's UART divides the machine cycle frequency of 921.6 kHz by 32 once more before it is used by Timer 1 to set the baud rate**
- **921.6 kHz divided by 32 gives 28,800 Hz**
- **Timer 1 must be programmed in mode 2, that is 8-bit, auto-reload**

Timer 1 TH1 Register Values for Various Baud Rates

Baud Rate	TH1 (Decimal)	TH1 (Hex)
9600	–3	FD
4800	–6	FA
2400	–12	F4
1200	–24	E8

Note: XTAL = 11.0592 MHz.

Table 10–4 Timer 1 TH1 Register Values for Various Baud Rates

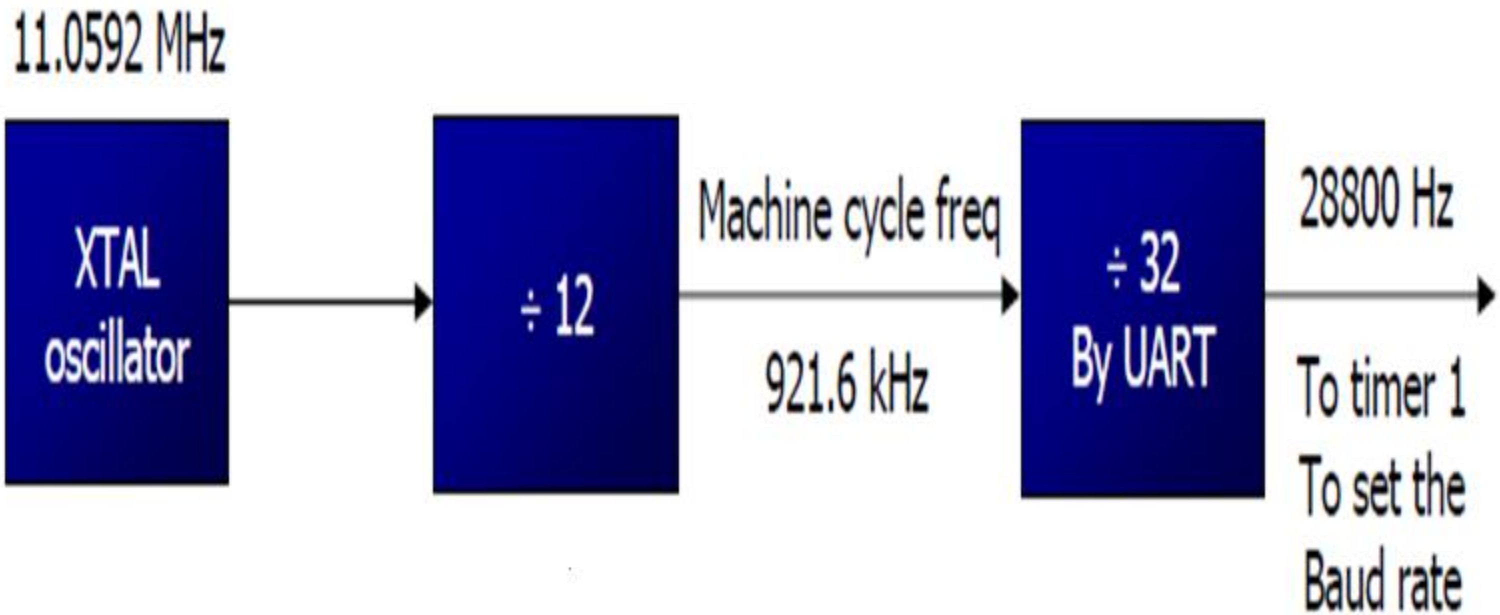
Example 10-1

With XTAL = 11.0592 MHz, find the TH1 value needed to have the following baud rates. (a) 9600 (b) 2400 (c) 1200

- **machine cycle frequency**
  = 11.0592 MHz / 12 = 921.6 kHz
- **Timer 1 frequency provided by 8051 UART**
  = 921.6 kHz / 32 = 28,800 Hz

(a) 28,800 / 3 = 9600	**where -3**	**= FD (hex)**
(b) 28,800 / 12 = 2400	**where -12**	**= F4 (hex)**
(c) 28,800 / 24 = 1200	**where -24**	**= E8 (hex)**

SECTION 10.3: 8051 SERIAL PORT PROGRAMMING IN ASSEMBLY



SBUF (serial buffer) register (8 bit)

- a byte of data to be transferred via the TxD line must be placed in the SBUF register
- SBUF holds the byte of data when it is received by the RxD line
- can be accessed like any other register

```
MOV SBUF,#'D'      ;load SBUF=44H, ASCII for 'D'
MOV SBUF,A         ;copy accumulator into SBUF
MOV A,SBUF         ;copy SBUF into accumulator
```

- when a byte is written, it is framed with the start and stop bits and transferred serially via the TxD pin
- when the bits are received serially via RxD, it is deframe by eliminating the stop and start bits, making a byte out of the data received, and then placing it in the SBUF .

Serial Control Register (SCON)

Bit 7							Bit 0
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

Bit	Operation
SM0 SM1	Serial communication mode selection bits SM0 SM1 Mode Description Baud rate 0 0 Mode 0 8 bit shift register mode fosc/12 0 1 Mode 1 8 bit UART variable (set by timer 1) 1 0 Mode 2 9 bit UART fosc/32 or fosc/64 1 1 Mode 3 9 bit UART variable (set by timer 1)
SM2	In mode 2 and 3, if set, this will enable multiprocessor communication (Always 0)
REN	Set/cleared by s/w to enable/disable serial reception(Receive enable)
TB8	Transfer bit 8,Used in serial mode 2 and 3 (Not used ,Always 0)
RB8	Receive bit 8,Used in serial mode 2 and 3 (Not used ,Always 0) In mode 1, if SM2=0, then RB8 is the stop bit that is received
TI	Transmit interrupt flag, set by hardware must be cleared by software
RI	Receive interrupt flag, set by hardware must be cleared by software

Serial Control Register (SCON)

- SM0 and SM1 determine the mode
- only mode 1 is important
- when mode 1 is chosen, the data framing is 8 bits, 1 stop bit, and 1 start bit
- compatible with the COM port of PCs
- mode 1 allows the baud rate to be variable and is set by Timer 1 of the 8051
- for each character a total of 10 bits are transferred, where the first bit is the start bit, followed by 8 bits of data, and finally 1 stop bit.

Serial Control Register (SCON)

- REN (receive enable)
- REN=1, allows 8051 to receive data on the RxD
- if 8051 is to both transfer and receive data, REN must be set to 1
- REN=0, the receiver is disabled
- SETB SCON.4 and CLR SCON.4,

Serial Control Register (SCON)

- TI (transmit interrupt)
  - when 8051 finishes the transfer of the 8-bit character, it raises the TI flag to indicate that it is ready to transfer another byte
- RI (receive interrupt)
  - when the 8051 receives data serially via RxD, it places the byte in the SBUF register
  - then raises the RI flag bit to indicate that a byte has been received and should be picked up before it is lost

Programming 8051 to transfer data serially

1. **TMOD register is loaded with the value 20H i.e. Timer 1 in mode 2**
2. **TH1 is loaded with value to set the baud rate**
3. **SCON register is loaded with the value 50H i.e. serial mode 1 & 8 bit data framed with start & stop bits.**
4. **TR1 is set to 1 to start Timer1**
5. **transmit character byte is written into the SBUF register**
6. **TI flag bit is monitored with use of instruction "JNB TI,XX" to see if the character has been transferred completely**
7. **TI is cleared by the "CLR TI" instruction**
8. **to transfer the next character, go to Step 5.**

Example 10-2

Write a program to transfer letter "A" serially at 4800 baud, continuously.

```
MOV TMOD,#20H                ;Timer 1, mode 2(auto-reload)
MOV TH1,#-6                  ;4800 baud rate
MOV TCON,#50H                ;8-bit, 1 stop, REN enabled
SETB TR1                     ;start Timer 1
AGAIN: MOV SBUF,#"A"         ;letter "A" to be transferred
HERE: JNB TI,HERE            ;wait for the last bit
CLR TI                       ;clear TI for next char
SJMP AGAIN                   ;keep sending A
END
```

Example 10-3

Write a program to transfer the message "YES" serially at 9600 baud, 8-bit data, 1 stop bit. Do this continuously.

```
MOV TMOD,#20H                 ;Timer 1 Mode 2
MOV TH1,#-3                   ;9600 baud
MOV TCON,#50H                 ;8-bit, 1 stop bit, REN enabled
SETB TR1                      ;start Timer 1
AGAIN: MOV A,#"Y"             ;transfer "Y"
ACALL TRANS
MOV A,#"E"                    ;transfer "E"
ACALL TRANS
MOV A, #"S"                   ;transfer "S"
ACALL TRANS
SJMP AGAIN                    ;keep doing it
;-------------------------------
;serial data transfer subroutine
;-------------------------------
TRANS: MOV SBUF,A             ;load SBUF
HERE: JNB TI,HERE             ;wait for last bit to transfer
CLR TI                        ;get ready for next byte
RET

END
```

SECTION 10.3: 8051 SERIAL PORT PROGRAMMING IN ASSEMBLY

- **Importance of the TI flag**
  - **check the TI flag bit, we know whether can transfer another byte**
  - **TI flag bit is raised by the 8051**
  - **TI flag cleared by the programmer**
  - **writing a byte into SBUF before the TI flag bit is raised, may lead to loss of a portion of the byte being transferred**

Programming 8051 to receive data serially

1. **TMOD register is loaded with the value 20H i.e. Timer 1 in mode 2**
2. **TH1 is loaded with value to set the baud rate**
3. **SCON register is loaded with the value 50H i.e. serial mode 1 & 8 bit data framed with start & stop bits & RI= ON**
4. **TR1 is set to 1 to start Timer1**
5. **RI is cleared by the "CLR RI" instruction**
6. **RI flag bit is monitored with use of instruction "JNB RI,XX" to see if the character has been received yet.**
7. **When RI is raised,SBUF has byte. Its content moved to safe place.**
8. **to receive the next character, go to Step 5.**

Example 10-4

Program the 8051 to receive bytes of data serially, and put them in P1. Set the baud rate at 4800, 8-bit data, and 1 stop bit.

```
MOV TMOD,#20H        ;Timer 1, mode 2 (auto reload)
MOV TH1,#-6          ;4800 baud
MOV TCON,#50H        ;8-bit, 1 stop, REN enabled
SETB TR1             ;start Timer 1
HERE: JNB RI,HERE    ;wait for char to come in
MOV A,SBUF           ;save incoming byte in A
MOV P1,A             ;send to port 1
CLR RI               ;get ready to receive next byte
SJMP HERE            ;keep getting data

END
```

SECTION 10.3: 8051 SERIAL PORT PROGRAMMING IN ASSEMBLY

- **Importance of the RI flag bit**
  1. **it receives the start bit, next bit is the first bit of the character**
  2. **when the last bit is received, a byte is formed and placed in SBUF**
  3. **when stop bit is received, makes RI = 1**
  4. **when RI=1, received byte is in the SBUF register, copy SBUF contents to a safe place**
  5. **after the SBUF contents are copied the RI flag bit must be cleared to 0**

PCON Register: Power control register

- PCON (Power control) register is used to force the 8051 microcontroller into power saving mode.
- Power control register of 8051 contains two power saving mode bits and one serial baud rate control bit.

SERIAL COMMUNICATION PROGRAMMING

Doubling Baud Rate

- There are two ways to increase the baud rate of data transfer
  - To use a higher frequency crystal
  - To change a bit in the PCON register

The system crystal is fixed

- PCON register is an 8-bit register
  - When 8051 is powered up, SMOD is zero
  - We can set it to high by software and thereby double the baud rate

SMOD	--	--	--	GF1	GF0	PD	IDL

It is not a bit-addressable register

```
MOV   A,PCON      ;place a copy of PCON in ACC
SETB  ACC.7       ;make D7=1
MOV   PCON,A      ;changing any other bits
```

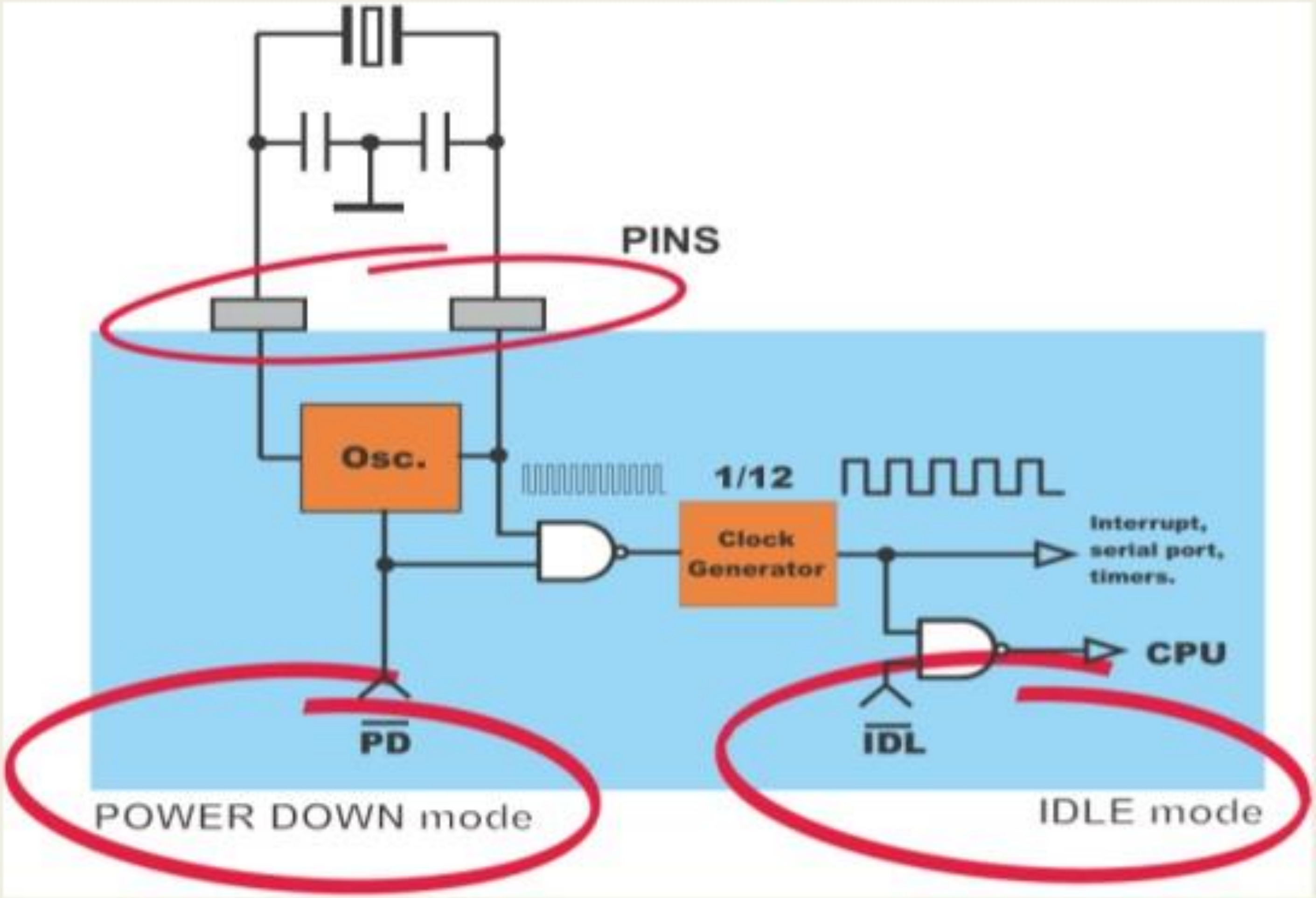
7	6	5	4	3	2	1	0	
SMOD	---	---	---	GF1	GF0	PD	IDL	PCON

- **Bit 7 – SMOD**
  - **1 =** Baud rate is doubled in UART mode 1, 2 and 3.
  - **0 =** No effect on Baud rate.
- **Bit 3:2 – GF1 & GF0:**
  - These are general purpose bit for user.
- **Bit 1 – PD:** Power Down
  - **1** = Enable Power Down mode. In this mode, Oscillator clock turned OFF and both CPU and peripherals clock stopped. Hardware reset can cancel this mode.
  - **0** = Disable Power down mode.
- **Bit 0 – IDL:** Idle
  - **1** = Enable Idle mode. CPU clock turned off whereas internal peripheral module such as timer, serial port, interrupts works normally. Interrupt and H/W reset can cancel this mode.
  - **0** = Disable Idle mode.

Why Power saving required?

- Generally speaking, the microcontroller is inactive for the most part and just waits for some external signal in order to takes its role in a show.
- This can cause some problems in case batteries are used for power supply.
- In extreme cases, the only solution is to set the whole electronics in sleep mode in order to minimize consumption.
- A typical example is a TV remote controller: it can be out of use for months but when used again it takes less than a second to send a command to TV receiver.
- The AT89S53 uses approximately 25mA for regular operation, which doesn't make it a pover-saving microcontroller.
- Anyway, it doesn't have to be always like that, it can easily switch the operating mode in order to reduce its total consumption to approximately 40uA.
- Actually, there are two power-saving modes of operation: *Idle* and *Power Down*.

Power Saving Modes



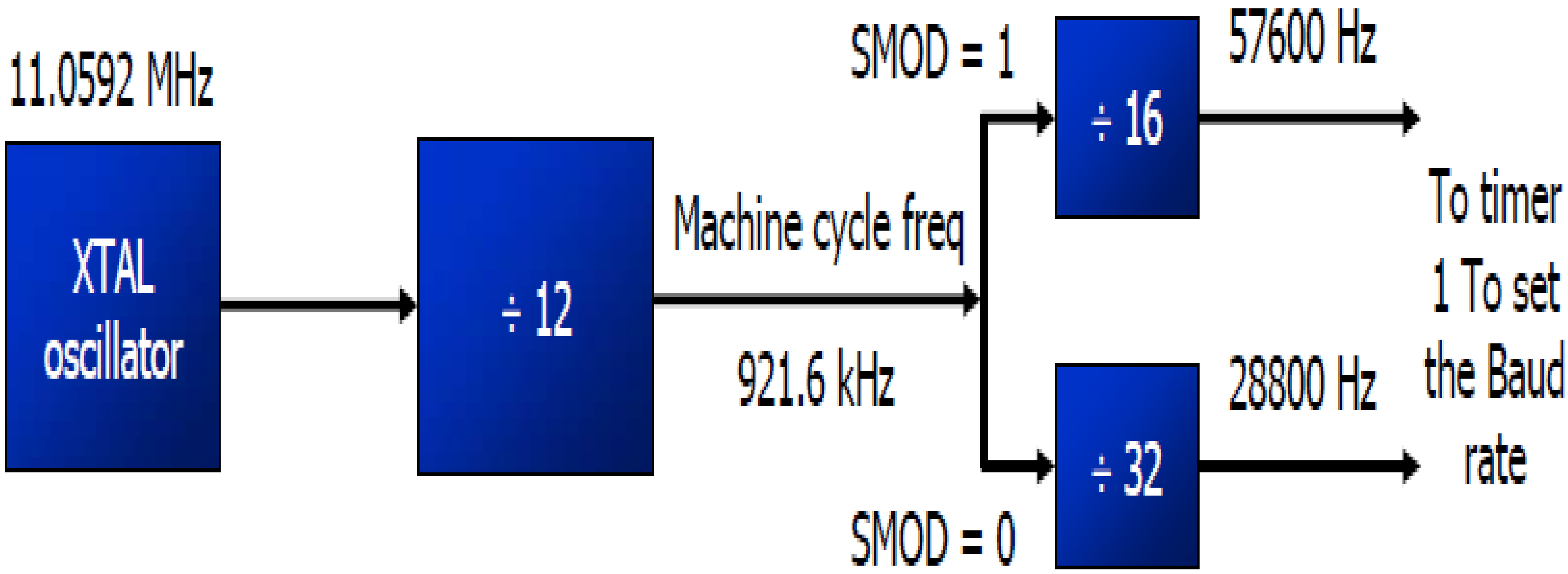
Power Down mode

- By setting the PD bit of the PCON register from within the program, the microcontroller is set to Power down mode, thus turning off its internal oscillator and reduces power consumption enormously.
- The microcontroller can operate using only 2V power supply in power-down mode, while a total power consumption is less than 40uA.
- The only way to get the microcontroller back to normal mode is by reset.
- While the microcontroller is in Power Down mode, the state of all SFR registers and I/O ports remains unchanged.
- By setting it back into the normal mode, the contents of the SFR register is lost, but the content of internal RAM is saved.
- Reset signal must be long enough, approximately 10mS, to enable stable operation of the quartz oscillator.

Idle mode

- Upon the IDL bit of the PCON register is set, the microcontroller turns off the greatest power consumer- CPU unit while peripheral units such as serial port, timers and interrupt system continue operating normally consuming 6.5mA.
- In Idle mode, the state of all registers and I/O ports remains unchanged. In order to exit the Idle mode and make the microcontroller operate normally, it is necessary to enable and execute any interrupt or reset.
- It will cause the IDL bit to be automatically cleared and the program resumes operation from instruction having set the IDL bit.
- It is recommended that first three instructions to execute now are NOP instructions.
- They don't perform any operation but provide some time for the microcontroller to stabilize and prevents undesired changes on the I/O ports.

Doubling Baud Rate



Baud Rate comparison for SMOD=0 and SMOD=1

TH1 (Decimal)	(Hex)	SMOD=0	SMOD=1
-3	FD	9600	19200
-6	FA	4800	9600
-12	F4	2400	4800
-24	E8	1200	2400

- If the crystal frequency is 22 MHz, what will be baud rate if : (a) TH1=-3; (b) TH1=-12 with SMOD =0 & SMOD =1

Solution:-

With SMOD =0,Machine Cycle freq.= 22/12 =1833 KHz

& 1833/32 =57.281 KHz =57,281

- (a) With TH1 =-3,baud rate =57,281/3 =19,093
- (b) With TH1 =-12,baud rate =57,281/12 = 4773

With SMOD =1, baud rate are doubled

(a) With TH1 =-3,baud rate = 38,186

- (b) With TH1 =-12,baud rate =9546

- Find the baud rate if TH1 = -2, SMOD = 1, and XTAL = 11.0592 MHz. Is this baud rate supported by IBM compatible PCs?

Solution:

With XTAL = 11.0592 and SMOD = 1, we have timer frequency =57,600 Hz. The baud rate is 57,600/2 = 28,800. This baud rate is not supported by the BIOS of the PCs; however, the PC can be programmed to do data transfer at such a speed. Also, HyperTerminal in Windows supports this and other baud rates.