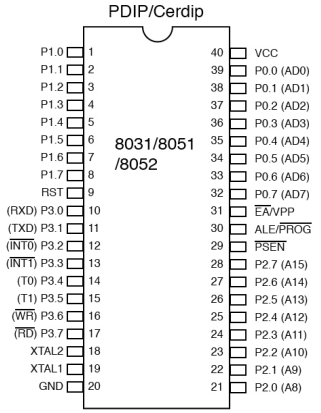
8051 Microcontroller:

Serial Communication



Virendra Singh

Computer Architecture and Dependable Systems Lab

Department of Electrical Engineering

Indian Institute of Technology Bombay

http://www.ee.iitb.ac.in/~viren/

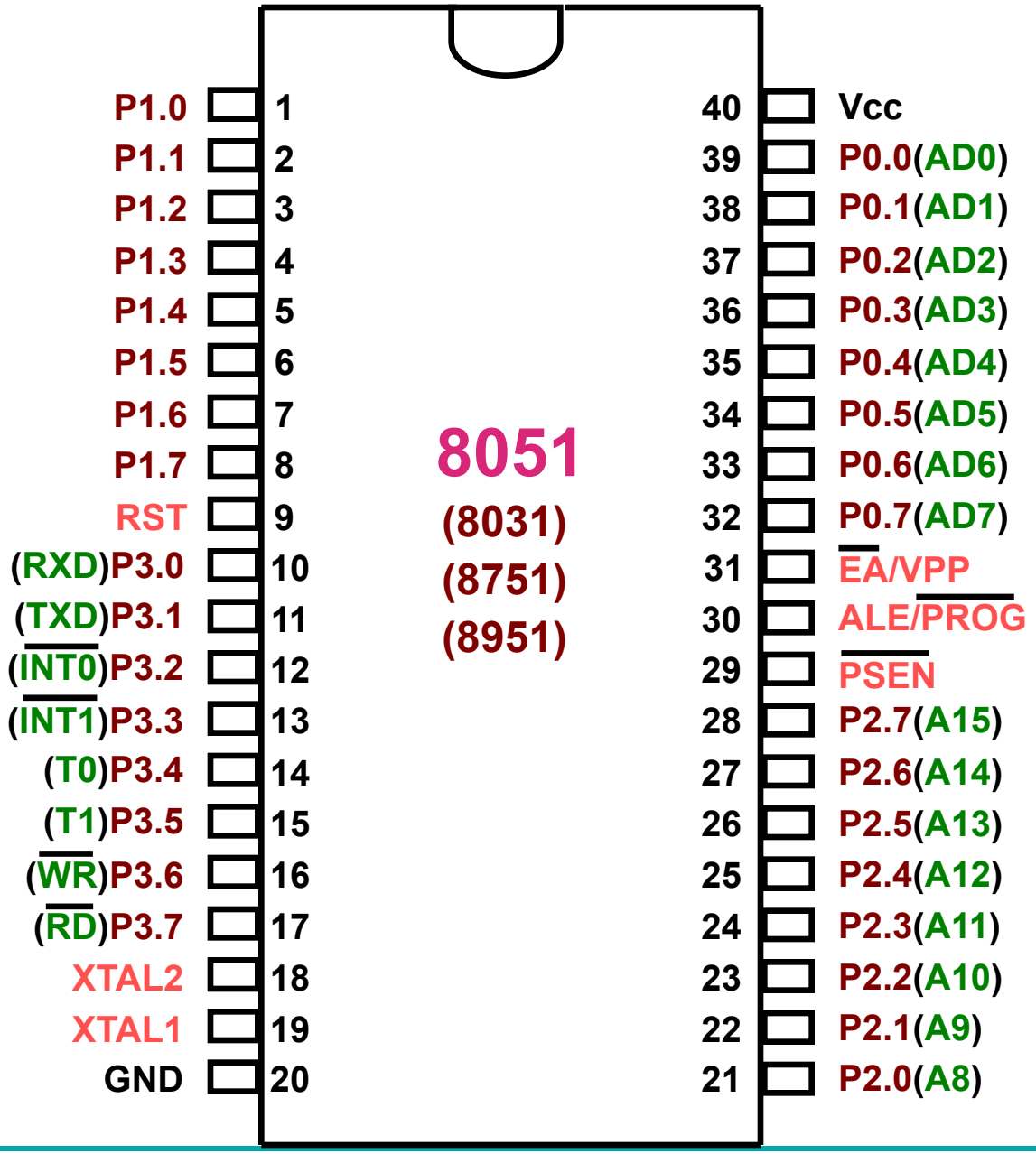
E-mail: viren@ee.iitb.ac.in

EE-309: Microprocessors

Lecture 16 (24 Aug 2015)

CADSL

8051 Pin Diagram



SCON Serial Port Control Register (Bit Addressable)

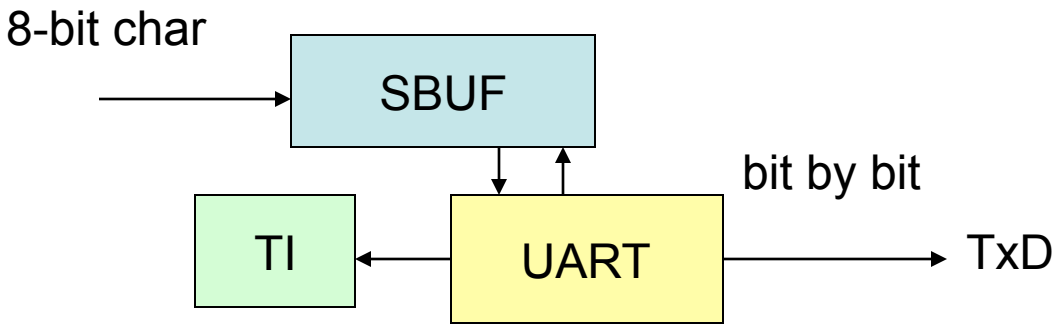
SM0	SM1	SM2	REN	TB8	RB8	TI	RI

SM0	SCON.7	Serial port mode specifier
SM1	SCON.6	Serial port mode specifier
SM2	SCON.5	Used for multiprocessor communication. (Make it 0)
REN	SCON.4	Set/cleared by software to enable/disable reception.
TB8	SCON.3	Not widely used.
RB8	SCON.2	Not widely used.
TI	SCON.1	Transmit interrupt flag. Set by hardware at the beginning of the stop bit in mode 1. Must be cleared by software.
RI	SCON.0	Receive interrupt flag. Set by hardware halfway through the stop bit time in mode 1. Must be cleared by software.

Note: Make SM2, TB8, and RB8 = 0.

Transfer Data with TI Flag

- The following sequence is the steps that the 8051 goes through in transmitting a character via TxD:
  1. The byte character to be transmitted is written into the SBUF register.
  2. It transfers the **start bit**.
  3. The **8-bit character** is transferred one bit at a time.
  4. The **stop bit** is transferred.



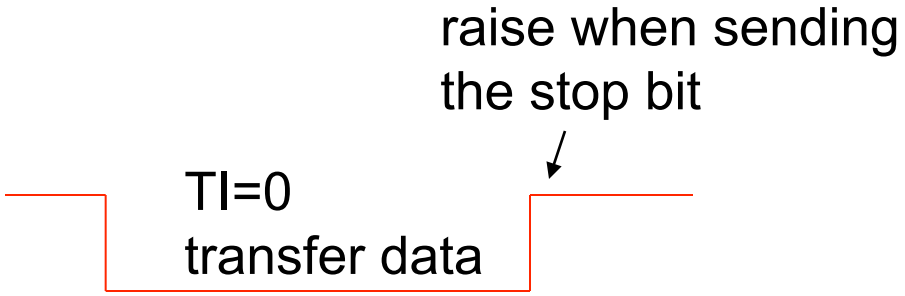
Transfer Data with the TI flag

5. During the transfer of the stop bit, the 8051 raises the TI flag, indicating that the last character was transmitted and it is ready to transfer the next character.
6. By monitoring the **TI** flag, we know whether or not the 8051 is ready to transfer another byte.
   - We will not overloading the SBUF register.
   - If we write another byte into the SBUF before TI is raised, the untransmitted portion of the previous byte will be lost.
   - We can use interrupt to transfer data
7. After SBUF is loaded with a new byte, the TI flag bit must be cleared by the programmer.

Transferring Data Serially in 8051

1. Use the timer 1 in mode 2
   - `MOV TMOD,#20H`
2. Set the value TH1 to chose baud rate.
   - `MOV TH1,#FDH` **;Baud rate = 9600bps**
3. Set SCON register in mode 1.
   - `MOV SCON,#50H`
4. Start the timer.
   - `SETB TR1`

Transferring Data Serially in 8051



5. Clear TI flag.
   - `CLR TI`
6. The character byte to be transferred serially is written into the SBUF register.
   - `MOV SBUF,#'A'`
7. Keep monitoring the Transmit Interrupt (TI) to see if it is raised.
   - `HERE: JNB TI, HERE`
8. To transfer the next character, go to Step 5.

Example: Serial Transmission

Write a program for the 8051 to transfer letter "A" serially at 4800 baud, continuously.

```
        MOV   TMOD,#20H     ;timer 1, mode 2
        MOV   TH1,#-6       ;4800 baud rate
        MOV   SCON,#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
```

Example: Serial Transmission

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

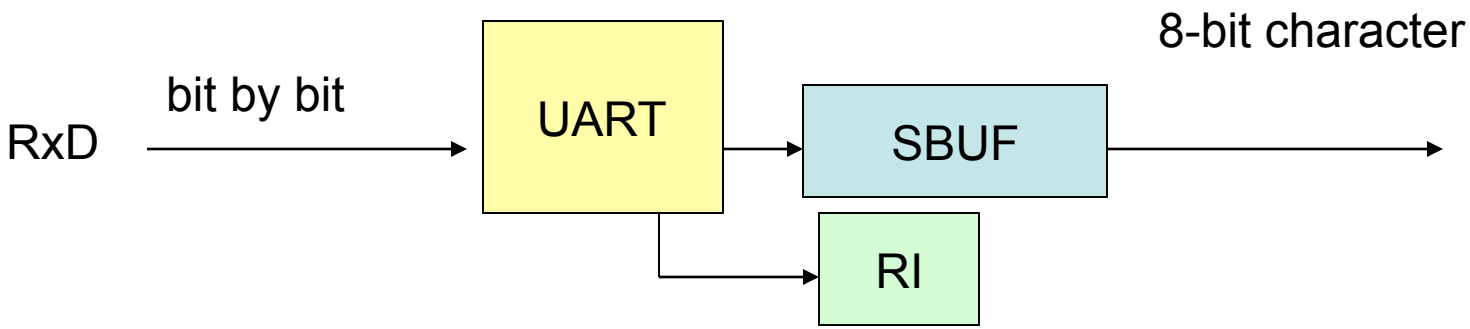
```
        MOV   TMOD,#20H  ;timer 1, mode 2
        MOV   TH1,#-3    ;9600 baud
        MOV   SCON,#50H  ;8-bit,1 stop,REN enabled
        SETB  TR1
AGAIN:  MOV   A,#"Y"     ;transfer "Y"
        ACALL TRANS
        ---------------------------------
        MOV   A,#"E"     ;transfer "E"
        ACALL TRANS
        ---------------------------------
        MOV   A,#"S"     ;transfer "S"
        ACALL TRANS
        SJMP  AGAIN      ;keep doing it
```

Example: Serial Transmission

```
;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
```

Receive Data with RI Flag

- The following sequence is the steps that the 8051 goes through in receiving a character via RxD:
  1. 8051 receives the **start bit** indicating that the next bit is the first bit of the character to be received.
  2. The **8-bit character** is received one bit at a time. When the last bit is received, a byte is **formed** and placed in SBUF.



Receive Data with TI Flag

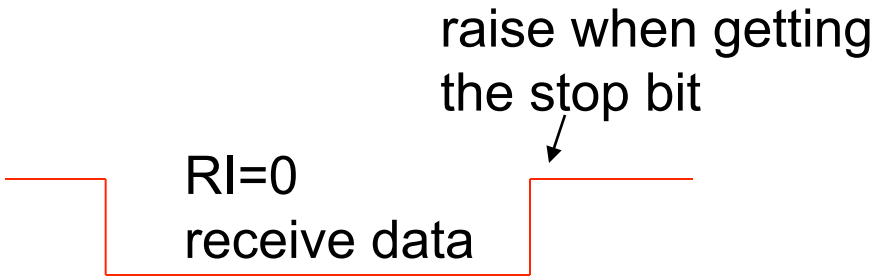
3. The **stop bit** is received. During receiving the stop bit, the 8051 make **RI=1**, indicating that an entire character was been received and **must be picked up** before it gets **overwritten** by an incoming character.
4. By monitoring the **RI** flag, we know whether or not the 8051 has received a character byte.
   - If we fail to copy SBUF into a safe place, we risk the loss of the received byte.
   - We can use interrupt to transfer data
5. After SBUF is copied into a safe place, the RI flag bit must be cleared by the programmer.

Receiving Data Serially in 8051

1. Use the timer 1 in mode 2
   - `MOV TMOD,#20H`
2. Set the value TH1 to chose baud rate.
   - `MOV TH1,#FDH` **;Baud rate = 9600bps**
3. Set SCON register in mode 1.
   - `MOV SCON,#50H`
4. Start the timer.
   - `SETB TR1`

Receiving Data Serially in 8051

5. Clear RI flag.
   - **`CLR RI`**



6. Keep monitoring the Receive Interrupt (RI) to see if it is raised.
   - **`HERE: JNB RI, HERE`**
7. When RI is raised, SBUF has the whole byte. Move the content of SBUF to a safe place.
   - **`MOV A,SBUF`**
8. To receive the next character, go to Step 5.

Example: Serial Data Receive

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  ;timer1, mode 2 (auto reload)
        MOV   TH1,#-6    ;4800 baud
        MOV   SCON,#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
```

Example: Serial Data Receive

Assume that the 8051 serial port is connected to the COM port of the IBM PC, and on the PC we are using the terminal.exe program to send and receive data serially. P1 and P2 of the 8051 are connected to LEDs and switches, respectively.

Write an 8051 program to

(a) send to the PC the message ”We Are Ready”,

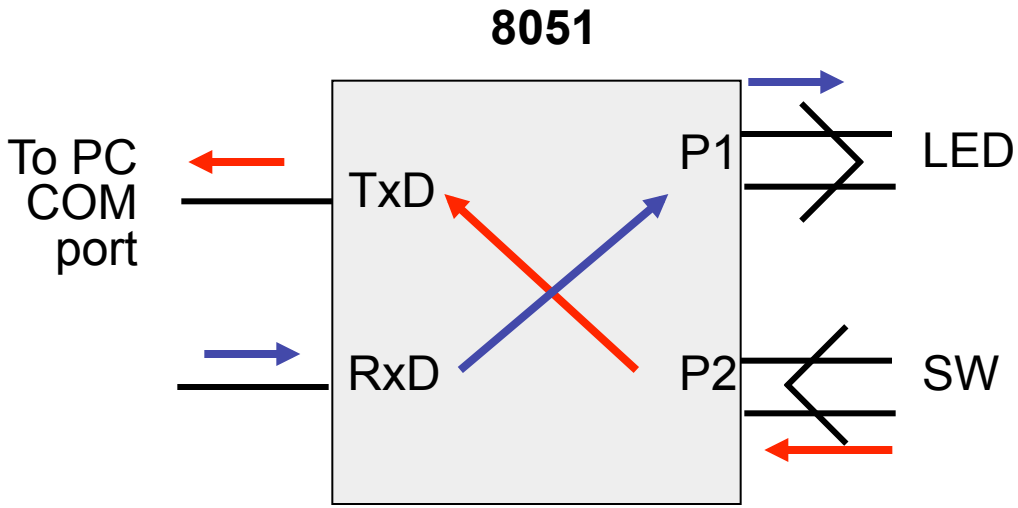
(b) → receive any data sent by the PC and put it on LEDs connected to P1, and

(c) ← get data on switches connected to P2 and send it to the PC serially.

The program should perform part (a) once, but parts (b) and (c) continuously.

Use the 4800 baud rate.

Example: Serial Data Receive




```
ORG   0
MOV   P2,#0FFH        ;make P2 an input port
MOV   TMOD,#20H
MOV   TH1,#0FAH       ;4800 baud rate
MOV   SCON,#50H       ;8-bit, 1 stop, REN enabled
SETB  TR1             ;start timer 1
MOV   DPTR,#MYDATA    ;load pointer for message
```

Example: Serial Data Receive

```
        MOV   DPTR,#MYDATA    ;load pointer for message
H1:     CLR   A
        MOVC  A,@A+DPTR       ;get the character
        JZ    B1              ;if last character get out
(a)     ACALL SEND
        INC   DPTR
        SJMP  H1              ;next character
B1:     MOV   A,P2            ;read data on P2
(c)     ACALL SEND            ;transfer it serially
        ACALL RECV            ;get the serial data
(b)     MOV   P1,A            ;display it on LEDs
        SJMP  B1              ;stay in loop indefinitely
```

Example: Serial Data Receive

```
;----------serial data transfer. ACC has the data
SEND:   MOV   SBUF,A      ;load the data
H2:     JNB   TI,H2       ;stay here until last bit gone
        CLR   TI          ;get ready for next char
        RET
;------------- Receive data serially in ACC
RECV:   JNB   RI,RECV     ;wait here for char
        MOV   A,SBUF      ;save it in ACC
        CLR   RI          ;get ready for next char
        RET
MYDATA:DB     "We Are Ready",0  ;--The message to send
        END
```

Doubling the Baud Rate in the 8051

- There are two ways to increase the baud rate of data transfer in the 8051:
  1. To use a higher frequency crystal.
     - It is not feasible in many situations since the system crystal is fixed.
     - Many new crystal may not be compatible with the IBM PC serial COM ports baud rate.
  2. To change a bit in the PCON register.
     - This is a software way by setting SMOD=1.

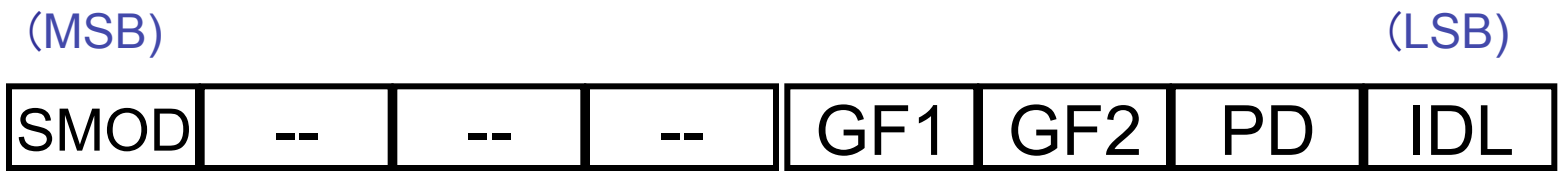
PCON Register

SMOD Double baud rate. If Timer 1 is used to generate baud and SMOD=1, the baud rate is doubled when the Serial Port is used in modes 1,2,3

GF1,GF0 General purpose flag bit.

PD Power down bit. Setting this bit activates "Power Down" operation in the 80C51BH. (precedence)

IDL Idle Mode bit. Setting this bit activates "Idle Mode" operation in the 80C51BH.



* PCON is not bit-addressable.

SMOD Flag of the PCON Register

- Power control register: **PCON**

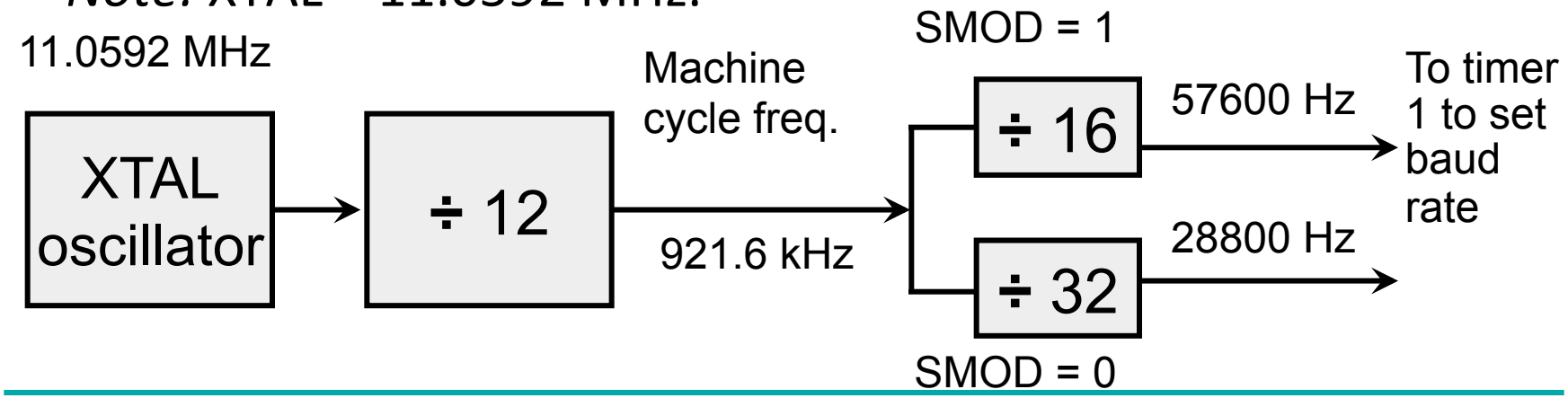
```
MOV A, PCON
SETB ACC.7
MOV PCON,A          ;Not to modify other bits
```

  - An 8-bit register
  - Not bit-addressable
  - SCOM=0: default
  - SCOM=1: double the baud rate

Baud Rate Comparison for SMOD = 0 and SMOD =1

TH1 (Decimal)	(Hex)	SMOD = 0	SMOD = 1
-3	FD	9,600	19,200
-6	FA	4,800	9,600
-12	F4	2,400	4,800
-24	E8	1,200	2,400

Note: XTAL = 11.0592 MHz.



Baud Rates for SMOD=0

- When SMOD=0, the 8051 divides 1/12 of the crystal frequency by 32, and uses that frequency for timer 1 to set the baud rate.
  - XTAL = 11.0592 MHz
  - The system frequency = 11.0592 MHz / 12 = 921.6 kHz
  - Timer 1 has 921.6 kHz/ 32 = 28,800 Hz as source.
  - TH1=256 - Crystal frequency/(12*32*Baud rate)
- Default on reset

Baud Rates for SMOD=1

- When SMOD=0, the 8051 divides 1/12 of the crystal frequency by 16, and uses that frequency for timer 1 to set the baud rate.
  - XTAL = 11.0592 MHz
  - The system frequency = 11.0592 MHz / 12 = 921.6 kHz
  - Timer 1 has 921.6 kHz/ 16 = 57,600 Hz as source.
  - TH1=256 - Crystal frequency/(12*16*Baud rate)

Example

Assuming that XTAL = 11.0592 MHz for the following program, state (a) what this program does, (b) compute the frequency used by timer 1 to set the baud rate, and (c) find the baud rate of the data transfer.

(a) This program transfers ASCII letter B (01000010 binary) continuously.

(b) and (c) With XTAL = 11.0592 MHz and SMOD = 1

11.0592 / 12 = 921.6 kHz machine cycle frequency.

921.6 /16 = 57,600 Hz frequency used by timer 1 to set the baud rate.

57,600 / 3 = 19,200, the baud rate.

Example

```
     MOV   A,PCON
     SETB  ACC.7
     MOV   PCON,A      ;SMOD=1, double baud rate
     MOV   TMOD,#20H   ;Timer 1, mode 2,auto reload
     MOV   TH1,#-3     ;19200 baud rate
     MOV   SCON,#50H   ;8-bit data,1 stop bit, RI enabled
     SETB  TR1         ;start Timer 1
     MOV   A,#"B"      ;transfer letter B
A1:  CLR   TI          ;make sure TI=0
     MOV   SBUF,A      ;transfer it
H1:  JNB   TI H1       ;check TI
     SJMP  A1          ;do again
```

Thank You